Truly Concurrent Process Algebra with Timing

Yong Wang
College of Computer Science and Technology,
Faculty of Information Technology,
Beijing University of Technology, Beijing, China

Abstract. We extend truly concurrent process algebra APTC with timing related properties. Just like ACP with timing, APTC with timing also has four parts: discrete relative timing, discrete absolute timing, continuous relative timing and continuous absolute timing.

Keywords: True Concurrency; Process Algebra; Timing

1. Introduction

In true concurrency, there are various structures, such as Petri net, event structure and domain et al \cite{5, 6, 7}, to model true concurrency. There are also some kinds of bisimulations to capture the behavioral equivalence between these structures, including pomset bisimulation, step bisimulation, history-preserving (hp-) bisimulation, and the finest hereditary history-preserving (hhp-) bisimulation \cite{8, 9}. Based on these truly concurrent semantics models, several logics to relate the logic syntaxes and the semantics models, such as the reversible logic \cite{11, 12}, the truly concurrent logic SFL \cite{13} based on the interleaving mu-calculi \cite{10}, and a uniform logic \cite{14, 15} to cover the above truly concurrent bisimulations. We also discussed the weakly truly concurrent bisimulations and their logics \cite{16}, that is, which are related to true concurrency with silent step $\tau$.

Process algebras CCS \cite{3, 2} and ACP \cite{1, 4} are based on the interleaving bisimulation. For the lack of process algebras based on truly concurrent bisimulations, we developed a calculus for true concurrency CTC \cite{18}, an axiomatization for true concurrency APTC \cite{17} and a calculus of truly concurrent mobile processes $\pi_{tc}$ \cite{19}, which are corresponding to CCS, ACP and $\pi$ based on interleaving bisimulation. There are correspondence between APTC and process algebra ACP \cite{4}, in this paper, we extend APTC with timing related properties. Just like ACP with timing \cite{23, 24, 25}, APTC with timing also has four parts: discrete relative timing, discrete absolute timing, continuous relative timing and continuous absolute timing.

This paper is organized as follows. In section 2, we introduce some preliminaries on APTC and timing.

---

Correspondence and offprint requests to: Yong Wang, Pingleyuan 100, Chaoyang District, Beijing, China. e-mail: wangy@bjut.edu.cn
In section 3, section 4, section 5 and section 6 we introduce APTC with discrete relative timing, APTC with discrete absolute timing, APTC with continuous relative timing and APTC with continuous absolute timing, respectively. We introduce recursion and abstraction in section 7 and section 8. We take an example to illustrate the usage of APTC with timing in section 9. The extension mechanism is discussed in section 10. Finally, in section 11 we conclude our work.

2. Backgrounds

For the convenience of the readers, we introduce some backgrounds about our previous work on truly concurrent process algebra [17] [18] [19], and also timing [23] [24] [25] in traditional process algebra ACP [4].

2.1. Truly Concurrent Process Algebra

In this subsection, we introduce the preliminaries on truly concurrent process algebra [17] [18] [19], which is based on truly concurrent operational semantics.

For this paper is an extension to APTC with timing, in the following, we introduce APTC briefly, for details, please refer to APTC [17].

APTC eliminates the differences of structures of transition system, event structure, etc, and discusses their behavioral equivalences. It considers that there are two kinds of causality relations: the chronological order modeled by the sequential composition and the causal order between different parallel branches modeled by the communication merge. It also considers that there exist two kinds of confliction relations: the structural confliction modeled by the alternative composition and the conflictions in different parallel branches which should be eliminated. Based on conservative extension, there are four modules in APTC: BATC (Basic Algebra for True Concurrency), APTC (Algebra for Parallelism in True Concurrency), recursion and abstraction.

2.1.1. Basic Algebra for True Concurrency

BATC has sequential composition · and alternative composition + to capture the chronological ordered causality and the structural confliction. The constants are ranged over A, the set of atomic actions. The algebraic laws on · and + are sound and complete modulo truly concurrent bisimulation equivalences (including pomset bisimulation, step bisimulation, hp-bisimulation and hhp-bisimulation).

Definition 2.1 (Prime event structure with silent event). Let \( \Lambda \) be a fixed set of labels, ranged over \( a, b, c, \cdots \) and \( \tau \). A (\( \Lambda \)-labelled) prime event structure with silent event \( \tau \) is a tuple \( \mathcal{E} = (\mathbb{E}, \leq, \#, \lambda) \), where \( \mathbb{E} \) is a denumerable set of events, including the silent event \( \tau \), being the empty event. Let \( \mathbb{E} = \mathbb{E}\{\tau\} \), exactly excluding \( \tau \), it is obvious that \( \tau^* = \epsilon \), where \( \epsilon \) is the empty event. Let \( \lambda : \mathbb{E} \rightarrow \Lambda \) be a labelling function and let \( \lambda(\tau) = \tau \). And \( \leq, \# \) are binary relations on \( \mathbb{E} \), called causality and conflict respectively, such that:

1. \( \leq \) is a partial order and \( [e] = \{e' \in \mathbb{E}| e' \leq e\} \) is finite for all \( e \in \mathbb{E} \). It is easy to see that \( e \leq \tau^* \leq e' = e \leq \tau \leq \cdots \leq \tau \leq e' \), then \( e \leq e' \).
2. \( \# \) is irreflexive, symmetric and hereditary with respect to \( \leq \), that is, for all \( e, e', e'' \in \mathbb{E} \), if \( e \# e' \leq e'' \), then \( e \# e'' \).

Then, the concepts of consistency and concurrency can be drawn from the above definition:

1. \( e, e' \in \mathbb{E} \) are consistent, denoted as \( e \sim e' \), if \( \neg(e \# e') \). A subset \( X \subseteq \mathbb{E} \) is called consistent, if \( e \sim e' \) for all \( e, e' \in X \).
2. \( e, e' \in \mathbb{E} \) are concurrent, denoted as \( e \parallel e' \), if \( \neg(e \leq e') \), \( \neg(e' \leq e) \), and \( \neg(e \# e') \).

The prime event structure without considering silent event \( \tau \) is the original one in [5] [6] [7].

Definition 2.2 (Configuration). Let \( \mathcal{E} \) be a PES. A (finite) configuration in \( \mathcal{E} \) is a (finite) consistent subset of events \( C \subseteq \mathcal{E} \), closed with respect to causality (i.e., \( \hat{C} = C \)). The set of finite configurations of \( \mathcal{E} \) is denoted by \( C(\mathcal{E}) \). We let \( \hat{C} = C\{\tau\} \).
Table 1. Axioms of BATC

| No. | Axiom                                      |
|-----|--------------------------------------------|
| A1  | \(x + y = y + x\)                          |
| A2  | \((x + y) + z = x + (y + z)\)              |
| A3  | \(x + x = x\)                              |
| A4  | \((x + y) \cdot z = x \cdot z + y \cdot z\) |
| A5  | \((x \cdot y) \cdot z = x \cdot (y \cdot z)\) |

A consistent subset of \(X \subseteq \mathbb{E}\) of events can be seen as a pomset. Given \(X, Y \subseteq \mathbb{E}, \hat{X} \sim \hat{Y}\) if \(\hat{X}\) and \(\hat{Y}\) are isomorphic as pomsets. In the following of the paper, we say \(C_1 \sim C_2\), we mean \(\hat{C}_1 \sim \hat{C}_2\).

**Definition 2.3** (Pomset transitions and step). Let \(E\) be a PES and let \(C \in C(E)\), and \(\emptyset \neq X \subseteq E\), if \(C \cap X = \emptyset\) and \(C' = C \cup X \in C(E)\), then \(C \xrightarrow{X} C'\) is called a pomset transition from \(C\) to \(C'\). When the events in \(X\) are pairwise concurrent, we say that \(C \xrightarrow{X} C'\) is a step.

**Definition 2.4** (Pomset, step bisimulation). Let \(\mathcal{E}_1, \mathcal{E}_2\) be PESs. A pomset bisimulation is a relation \(R \subseteq C(\mathcal{E}_1) \times C(\mathcal{E}_2)\), such that if \((C_1, C_2) \in R\), and \(C_1 \xrightarrow{X_1} C_1'\) then \(C_2 \xrightarrow{X_2} C_2'\), with \(X_1 \subseteq \mathcal{E}_1\), \(X_2 \subseteq \mathcal{E}_2\), \(X_1 \sim X_2\), and \((C_1', C_2') \in R\), and vice-versa. We say that \(\mathcal{E}_1, \mathcal{E}_2\) are pomset bisimilar, written \(\mathcal{E}_1 \sim_g \mathcal{E}_2\), if there exists a pomset bisimulation \(R\), such that \((\emptyset, \emptyset) \in R\). By replacing pomset transitions with steps, we can get the definition of step bisimulation. When PESs \(\mathcal{E}_1\) and \(\mathcal{E}_2\) are step bisimilar, we write \(\mathcal{E}_1 \sim_s \mathcal{E}_2\).

**Definition 2.5** (Posetal product). Given two PESs \(\mathcal{E}_1, \mathcal{E}_2\), the posetal product of their configurations, denoted \(C(\mathcal{E}_1) \times C(\mathcal{E}_2)\), is defined as

\[
\{(C_1, f, C_2) \mid C_1 \in C(\mathcal{E}_1), C_2 \in C(\mathcal{E}_2), f : C_1 \rightarrow C_2 \text{ isomorphism}\}.
\]

A subset \(R \subseteq C(\mathcal{E}_1) \times C(\mathcal{E}_2)\) is called a posetal relation. We say that \(R\) is downward closed when for any \((C_1, f, C_2), (C_1', f', C_2') \in C(\mathcal{E}_1) \times C(\mathcal{E}_2)\), if \((C_1, f, C_2) \subseteq (C_1', f', C_2')\) pointwise and \((C_1', f', C_2') \in R\), then \((C_1, f, C_2) \in R\).

For \(f : X_1 \rightarrow X_2\), we define \(f[x_1 \mapsto x_2] : X_1 \cup \{x_1\} \rightarrow X_2 \cup \{x_2\}\), \(z \in X_1 \cup \{x_1\}\), \((1)f[x_1 \mapsto x_2](z) = x_2\) if \(z = x_1\); \((2)f[x_1 \mapsto x_2](z) = f(z)\), otherwise. Where \(X_1 \subseteq \mathcal{E}_1\), \(X_2 \subseteq \mathcal{E}_2\), \(x_1 \subseteq \mathcal{E}_1\), \(x_2 \subseteq \mathcal{E}_2\).

**Definition 2.6** ((Hereditary) history-preserving bisimulation). A history-preserving (hp-) bisimulation is a posetal relation \(R \subseteq C(\mathcal{E}_1) \times C(\mathcal{E}_2)\) such that if \((C_1, f, C_2) \in R\), and \(C_1 \xrightarrow{e_1} C_1'\), then \(C_2 \xrightarrow{e_2} C_2'\), with \((C_1', f[e_1 \mapsto e_2], C_2') \in R\), and vice-versa. \(\mathcal{E}_1, \mathcal{E}_2\) are history-preserving (hp)bisimilar and are written \(\mathcal{E}_1 \sim_{hp} \mathcal{E}_2\) if there exists a hp-bisimulation \(R\) such that \((\emptyset, \emptyset, \emptyset) \in R\).

A hereditary history-preserving (hhp)bisimulation is a downward closed hp-bisimulation. \(\mathcal{E}_1, \mathcal{E}_2\) are hereditary history-preserving (hhp)bisimilar and are written \(\mathcal{E}_1 \sim_{hhp} \mathcal{E}_2\).

In the following, let \(e_1, e_2, e_1', e_2' \in \mathcal{E}\), and let variables \(x, y, z\) range over the set of terms for true concurrency, \(p, q, s\) range over the set of closed terms. The set of axioms of BATC consists of the laws given in Table 1.

We give the operational transition rules of operators \(\cdot\) and \(+\) as Table 2 shows. And the predicate \(\xrightarrow{\sqrt{\cdot}}\) represents successful termination after execution of the event \(e\).

**Theorem 2.7** (Soundness of BATC modulo truly concurrent bisimulation equivalences). The axiomatization of BATC is sound modulo truly concurrent bisimulation equivalences \(\sim_p, \sim_s, \sim_{hp}\) and \(\sim_{hhp}\). That is,

1. let \(x\) and \(y\) be BATC terms. If \(\text{BATC} \vdash x = y\), then \(x \sim_p y\);
2. let \(x\) and \(y\) be BATC terms. If \(\text{BATC} \vdash x = y\), then \(x \sim_s y\);
3. let \(x\) and \(y\) be BATC terms. If \(\text{BATC} \vdash x = y\), then \(x \sim_{hp} y\);
4. let \(x\) and \(y\) be BATC terms. If \(\text{BATC} \vdash x = y\), then \(x \sim_{hhp} y\).
Theorem 2.10 (Completeness of APTC modulo truly concurrent bisimulation equivalences). The axiomatization of APTC is complete modulo truly concurrent bisimulation equivalences.

\[
\begin{align*}
& x \xrightarrow{c} \checkmark \\
& x + y \xrightarrow{c} \checkmark \\
& x \xrightarrow{c} x' \\
& x + y \xrightarrow{c} x' \\
& y \xrightarrow{c} y' \\
& y \xrightarrow{c} x \\
& x \xrightarrow{c} y \\
& x \xrightarrow{c} y \\
& x \cdot y \xrightarrow{} y \\
& x \cdot y \xrightarrow{} y \\
& x \cdot y \xrightarrow{} x' \cdot y
\end{align*}
\]

Table 2. Transition rules of BATC

Theorem 2.8 (Completeness of BATC modulo truly concurrent bisimulation equivalences). The axiomatization of BATC is complete modulo truly concurrent bisimulation equivalences \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhp} \). That is,

1. let \( p \) and \( q \) be closed BATC terms, if \( p \sim_p q \) then \( p = q \);
2. let \( p \) and \( q \) be closed BATC terms, if \( p \sim_s q \) then \( p = q \);
3. let \( p \) and \( q \) be closed BATC terms, if \( p \sim_{hp} q \) then \( p = q \);
4. let \( p \) and \( q \) be closed BATC terms, if \( p \sim_{hhp} q \) then \( p = q \).

2.1.2. Algebra for Parallelism in True Concurrency

APTC uses the whole parallel operator \( \parallel \), the auxiliary binary parallel \( || \) to model parallelism, and the communication merge \( | \) to model communications among different parallel branches, and also the unary conflict elimination operator \( \Theta \) and the binary unless operator \( \triangleleft \) to eliminate conflicts among different parallel branches. Since a communication may be blocked, a new constant called deadlock \( \delta \) is extended to \( A \), and also a new unary encapsulation operator \( \partial_H \) is introduced to eliminate \( \delta \), which may exist in the processes. The algebraic laws on these operators are also sound and complete modulo truly concurrent bisimulation equivalences (including pomset bisimulation, step bisimulation, hp-bisimulation, but not hhp-bisimulation). Note that, the parallel operator \( || \) in a process cannot be eliminated by deductions on the process using axioms of APTC, but other operators can eventually be steadied by \( \sim, + \) and \( \parallel \). This is also why truly concurrent bisimulations are called a truly concurrent semantics.

We design the axioms of APTC in Table 3, including algebraic laws of parallel operator \( \parallel \), communication operator \( | \), conflict elimination operator \( \Theta \) and unless operator \( \triangleleft \), encapsulation operator \( \partial_H \), the deadlock constant \( \delta \), and also the whole parallel operator \( \parallel \).

we give the transition rules of APTC in Table 4; it is suitable for all truly concurrent behavioral equivalence, including pomset bisimulation, step bisimulation, hp-bisimulation and hhp-bisimulation.

Theorem 2.9 (Soundness of APTC modulo truly concurrent bisimulation equivalences). The axiomatization of APTC is sound modulo truly concurrent bisimulation equivalences \( \sim_p, \sim_s, \sim_{hp} \). That is,

1. let \( x \) and \( y \) be APTC terms. If \( APTC \vdash x = y \), then \( x \sim_p y \);
2. let \( x \) and \( y \) be APTC terms. If \( APTC \vdash x = y \), then \( x \sim_s y \);
3. let \( x \) and \( y \) be APTC terms. If \( APTC \vdash x = y \), then \( x \sim_{hp} y \).

Theorem 2.10 (Completeness of APTC modulo truly concurrent bisimulation equivalences). The axiomatization of APTC is complete modulo truly concurrent bisimulation equivalences \( \sim_p, \sim_s, \sim_{hp} \). That is,

1. let \( p \) and \( q \) be closed APTC terms, if \( p \sim_p q \) then \( p = q \);
2. let \( p \) and \( q \) be closed APTC terms, if \( p \sim_s q \) then \( p = q \);
3. let \( p \) and \( q \) be closed APTC terms, if \( p \sim_{hp} q \) then \( p = q \).
In the following, concurrent bisimulation equivalences (including pomset bisimulation, step bisimulation and hp-bisimulation). For a guarded recursive specifications, they are sound with respect to APTC with guarded recursion modulo Specification Principle and RDP (Recursive Definition Principle), RDP says the solutions of a recursive specification can represent the behaviors of the specification, while RSP says that a guarded recursive specification has only one solution, they are sound with respect to APTC with guarded recursion modulo several truly concurrent bisimulation equivalences (including pomset bisimulation, step bisimulation and hp-bisimulation), and they are complete with respect to APTC with linear recursion modulo several truly concurrent bisimulation equivalences (including pomset bisimulation, step bisimulation and hp-bisimulation).

In the following, $E, F, G$ are recursion specifications, $X, Y, Z$ are recursive variables.

For a guarded recursive specifications $E$ with the form

$$X_1 = t_1(X_1, \cdots, X_n)$$
Table 5. Transition rules of guarded recursion

Table 4. Transition rules of APTC

The behavior of the solution $\langle X_i | E \rangle$ for the recursion variable $X_i$ in $E$, where $i \in \{1, \ldots, n\}$, is exactly the behavior of their right-hand sides $t_i(X_1, \ldots, X_n)$, which is captured by the two transition rules in Table 5.

The $RDP$ (Recursive Definition Principle) and the $RSP$ (Recursive Specification Principle) are shown in Table 6.

**Theorem 2.11** (Soundness of APTC with guarded recursion). Let $x$ and $y$ be APTC with guarded recursion terms. If APTC with guarded recursion $\vdash x = y$, then

1. $x \sim_s y$;
bisimulation resulted APTC with silent step and abstraction operator is called APTC

Definition 2.13
To abstract away internal implementations from the external behaviors, a new constant $C$ is added to $A$ and also a new unary abstraction operator $\tau$ is used to rename actions in $I$ into $\tau$ (the resulted APTC with silent step and abstraction operator is called APTC-$\tau$). The recursive specification is adapted to guarded linear recursion to prevent infinite $\tau$-loops specifically. The axioms of $\tau$ and $\tau_1$ are sound modulo rooted branching truly concurrent bisimulation equivalences (several kinds of weakly truly concurrent bisimulation equivalences, including rooted branching pomset bisimulation, rooted branching step bisimulation and rooted branching hp-bisimulation). To eliminate infinite $\tau$-loops caused by $\tau_1$ and obtain the completeness, CFAR (Cluster Fair Abstraction Rule) is used to prevent infinite $\tau$-loops in a constructible way.

Definition 2.14 (Weak pomset transitions and weak step). Let $E$ be a PES and let $C \in C(E)$, and $\varnothing \neq X \in \hat{E}$,

if $C \cap X = \varnothing$ and $\hat{C}' = \hat{C} \cup X \in C(E)$, then $C \overset{X}{\Rightarrow} C'$ is called a weak pomset transition from $C$ to $C'$, where we define $\overset{X}{\Rightarrow} \overset{\tau}{\Rightarrow} \overset{e}{\Rightarrow} \overset{\tau_1}{\Rightarrow}$ for every $e \in X$. When the events in $X$ are pairwise concurrent, we say that $C \overset{X}{\Rightarrow} C'$ is a weak step.

Definition 2.15 (Branching pomset, step bisimulation). Assume a special termination predicate $\downarrow$, and let $\sqrt{\cdot}$ represent a state with $\sqrt{\downarrow}$. Let $E_1, E_2$ be PESs. A branching pomset bisimulation is a relation $R \subseteq C(E_1) \times C(E_2)$, such that:

1. if $(C_1, C_2) \in R$, and $C_1 \overset{X}{\Rightarrow} C_1'$ then
   - either $X \equiv \tau^*$, and $(C_1', C_2) \in R$;
   - or there is a sequence of (zero or more) $\tau$-transitions $C_2 \overset{\tau^*}{\Rightarrow} C_2'$, such that $(C_1, C_2^0) \in R$ and $C_2^0 \overset{X}{\Rightarrow} C_2'$ with $(C_1', C_2') \in R$;

2. if $(C_1, C_2) \in R$, and $C_2 \overset{X}{\Rightarrow} C_2'$ then
   - either $X \equiv \tau^*$, and $(C_1', C_2) \in R$;
   - or there is a sequence of (zero or more) $\tau$-transitions $C_1 \overset{\tau^*}{\Rightarrow} C_1'$, such that $(C_1^0, C_2) \in R$ and $C_1^0 \overset{X}{\Rightarrow} C_1'$ with $(C_1', C_2') \in R$;

3. if $(C_1, C_2) \in R$ and $C_1 \downarrow$, then there is a sequence of (zero or more) $\tau$-transitions $C_2 \overset{\tau^*}{\Rightarrow} C_2^0$ such that $(C_1, C_2^0) \in R$ and $C_2^0 \downarrow$;

4. if $(C_1, C_2) \in R$ and $C_2 \downarrow$, then there is a sequence of (zero or more) $\tau$-transitions $C_1 \overset{\tau^*}{\Rightarrow} C_1^0$ such that $(C_1^0, C_2) \in R$ and $C_1^0 \downarrow$.

We say that $E_1, E_2$ are branching pomset bisimilar, written $E_1 \equiv_{hp} E_2$, if there exists a branching pomset bisimulation $R$, such that $(\varnothing, \varnothing) \in R$. 

Table 6. Recursive definition and specification principle

| No. | Axiom |
|-----|-------|
| RDP | $(X_i|E) = t_i((X_1|E, \ldots, X_n|E))$ $i \in \{1, \ldots, n\}$ |
| RSP | if $y_i = t_i(y_1, \ldots, y_n)$ for $i \in \{1, \ldots, n\}$, then $y_i = (X_i|E)$ $i \in \{1, \ldots, n\}$ |

Theorem 2.12 (Completeness of APTC with linear recursion). Let $p$ and $q$ be closed APTC with linear recursion terms, then,

1. if $p \sim q$ then $p \equiv q$;
2. if $p \sim q$ then $p \equiv q$;
3. if $p \sim_{hp} q$ then $p \equiv q$.

2.1.4. Abstraction

Draft of Truly Concurrent Process Algebra with Timing
By replacing pomset transitions with steps, we can get the definition of branching step bisimulation. When PESs \( E_1 \) and \( E_2 \) are branching step bisimilar, we write \( E_1 \approx_{bhp} E_2 \).

**Definition 2.15** (Rooted branching pomset, step bisimulation). Assume a special termination predicate \( \downarrow \), and let \( \sqrt{\cdot} \) represent a state with \( \sqrt{\cdot} \downarrow \). Let \( E_1, E_2 \) be PESs. A branching pomset bisimulation is a relation \( R \subseteq \mathcal{C}(E_1) \times \mathcal{C}(E_2) \), such that:

1. if \( (C_1, f, C_2) \in R \), and \( C_1 \overset{\tau}{\rightarrow} C_1' \), then \( C_2 \overset{\tau}{\rightarrow} C_2' \) with \( C_1' \approx_{bhp} C_2' \);
2. if \( (C_1, f, C_2) \in R \), and \( C_2 \overset{\tau}{\rightarrow} C_2' \), then \( C_1 \overset{\tau}{\rightarrow} C_1' \) with \( C_1' \approx_{bhp} C_2' \);
3. if \( (C_1, C_2) \in R \) and \( C_1 \downarrow \), then \( C_2 \downarrow \);
4. if \( (C_1, C_2) \in R \) and \( C_2 \downarrow \), then \( C_1 \downarrow \).

We say that \( E_1, E_2 \) are rooted branching pomset bisimilar, written \( E_1 \approx_{rhp} E_2 \), if there exists a rooted branching pomset bisimulation \( R \), such that \( (\varnothing, \varnothing) \in R \).

By replacing pomset transitions with steps, we can get the definition of rooted branching step bisimulation. When PESs \( E_1 \) and \( E_2 \) are rooted branching step bisimilar, we write \( E_1 \approx_{r_{bhp}} E_2 \).

**Definition 2.16** (Branching (hereditary) history-preserving bisimulation). Assume a special termination predicate \( \downarrow \), and let \( \sqrt{\cdot} \) represent a state with \( \sqrt{\cdot} \downarrow \). A branching history-preserving (hp-)bisimulation is a weakly posetal relation \( R \subseteq \mathcal{C}(E_1) \times \mathcal{C}(E_2) \) such that:

1. if \( (C_1, C_2) \in R \), and \( C_1 \overset{\tau}{\rightarrow} C_1' \), then
   - either \( e_1 \equiv \tau \), and \( (C_1', f[e_1 \mapsto \tau], C_2) \in R \);
   - or there is a sequence of (zero or more) \( \tau \)-transitions \( C_2 \overset{\tau^*}{\rightarrow} C_0 \), such that \( (C_1, f, C_2) \in R \) and \( C_2 \equiv C_0 \); with \( (C_1', f[e_1 \mapsto e_2], C_2') \in R \);
2. if \( (C_1, C_2) \in R \), and \( C_2 \overset{\tau}{\rightarrow} C_2' \), then
   - either \( X \equiv \tau \), and \( (C_1, f[e_2 \mapsto \tau], C_2') \in R \);
   - or there is a sequence of (zero or more) \( \tau \)-transitions \( C_1 \overset{\tau^*}{\rightarrow} C_1' \), such that \( (C_1', f, C_2) \in R \) and \( C_1 \equiv C_1' \); with \( (C_1', f[e_2 \mapsto e_1], C_2') \in R \);
3. if \( (C_1, C_2) \in R \) and \( C_1 \downarrow \), then there is a sequence of (zero or more) \( \tau \)-transitions \( C_2 \overset{\tau^*}{\rightarrow} C_0 \) such that \( (C_1, f, C_2) \in R \) and \( C_2 \downarrow \);
4. if \( (C_1, C_2) \in R \) and \( C_2 \downarrow \), then there is a sequence of (zero or more) \( \tau \)-transitions \( C_1 \overset{\tau^*}{\rightarrow} C_0 \) such that \( (C_1, f, C_2) \in R \) and \( C_1 \downarrow \).

\( E_1, E_2 \) are branching history-preserving (hp-)bisimilar and are written \( E_1 \approx_{bhp} E_2 \) if there exists a branching hp-bisimulation \( R \) such that \( (\varnothing, \varnothing) \in R \).

A branching hereditary history-preserving (hhp-)bisimulation is a downward closed branching hhp-bisimulation. \( E_1, E_2 \) are branching hereditary history-preserving (hhp-)bisimilar and are written \( E_1 \approx_{b_{hhp}} E_2 \).

**Definition 2.17** (Rooted branching (hereditary) history-preserving bisimulation). Assume a special termination predicate \( \downarrow \), and let \( \sqrt{\cdot} \) represent a state with \( \sqrt{\cdot} \downarrow \). A rooted branching history-preserving (hp-)bisimulation is a weakly posetal relation \( R \subseteq \mathcal{C}(E_1) \times \mathcal{C}(E_2) \) such that:

1. if \( (C_1, f, C_2) \in R \), and \( C_1 \overset{\tau}{\rightarrow} C_1' \), then \( C_2 \overset{\tau}{\rightarrow} C_2' \) with \( C_1' \approx_{bhp} C_2' \);
2. if \( (C_1, f, C_2) \in R \), and \( C_2 \overset{\tau}{\rightarrow} C_2' \), then \( C_1 \overset{\tau}{\rightarrow} C_1' \) with \( C_1' \approx_{bhp} C_2' \);
3. if \( (C_1, f, C_2) \in R \) and \( C_1 \downarrow \), then \( C_2 \downarrow \);
4. if \( (C_1, f, C_2) \in R \) and \( C_2 \downarrow \), then \( C_1 \downarrow \).

\( E_1, E_2 \) are rooted branching history-preserving (hp-)bisimilar and are written \( E_1 \approx_{r_{bhp}} E_2 \) if there exists rooted a branching hp-bisimulation \( R \) such that \( (\varnothing, \varnothing, \varnothing) \in R \).

A rooted branching hereditary history-preserving (hhp-)bisimulation is a downward closed rooted branching
Table 7. Axioms of APTC

| No. | Axiom |
|-----|-------|
| B1  | $e \cdot \tau = e$ |
| B2  | $e \cdot (\tau \cdot (x + y) + x) = e \cdot (x + y)$ |
| B3  | $x \parallel \tau = x$ |
| T11 | $e \notin I \quad \tau_I(e) = e$ |
| T12 | $e \notin I \quad \tau_I(e) = \tau$ |
| T13 | $\tau_I(\delta) = \delta$ |
| T14 | $\tau_I(x + y) = \tau_I(x) + \tau_I(y)$ |
| T15 | $\tau_I(x \cdot y) = \tau_I(x) \cdot \tau_I(y)$ |
| T16 | $\tau_I(x \parallel y) = \tau_I(x) \parallel \tau_I(y)$ |

**APTC** if X is in a cluster for I with exits

$(a_{11} \parallel \cdots \parallel a_{1i})Y_1, \cdots, (a_{m1} \parallel \cdots \parallel a_{ml})Y_m, b_{i1} \parallel \cdots \parallel b_{i} \parallel \cdots \parallel b_{nj})$

then $\tau \cdot \tau_I((X|E)) = \tau \cdot \tau_I((a_{11} \parallel \cdots \parallel a_{1i})(Y_1|E) + \cdots + (a_{m1} \parallel \cdots \parallel a_{ml})(Y_m|E) + b_{i1} \parallel \cdots \parallel b_{i} \parallel \cdots \parallel b_{nj})$

Table 7. Axioms of APTC$_\tau$

- $\tau \rightarrow \sqrt{\cdot}$
- $\tau_I(x) \rightarrow \sqrt{\cdot}$
- $\tau_I(x) \rightarrow \sqrt{\cdot}$
- $\tau_I(x) \rightarrow \sqrt{\cdot}$
- $\tau_I(x) \rightarrow \sqrt{\cdot}$
- $\tau_I(x) \rightarrow \sqrt{\cdot}$

Table 8. Transition rule of APTC$_\tau$

$hhp$-bisimulation. $E_1, E_2$ are rooted branching hereditary history-preserving (hhp-)bisimilar and are written $E_1 \equiv_{rhhp} E_2$.

The axioms and transition rules of APTC$_\tau$ are shown in Table 7 and Table 8

**Theorem 2.18** (Soundness of APTC$_\tau$ with guarded linear recursion). Let $x$ and $y$ be APTC$_\tau$ with guarded linear recursion terms. If APTC$_\tau$ with guarded linear recursion $\vdash x = y$, then

1. $x \equiv_{rbs} y$;
2. $x \equiv_{rbp} y$;
3. $x \equiv_{rhhp} y$.

**Theorem 2.19** (Soundness of CFAR). CFAR is sound modulo rooted branching truly concurrent bisimulation equivalences $\equiv_{rbs}$, $\equiv_{rbp}$ and $\equiv_{rhhp}$.

**Theorem 2.20** (Completeness of APTC$_\tau$ with guarded linear recursion and CFAR). Let $p$ and $q$ be closed APTC$_\tau$ with guarded linear recursion and CFAR terms, then,

1. if $p \equiv_{rbs} q$ then $p = q$;
2. if $p \equiv_{rbp} q$ then $p = q$;
3. if $p \equiv_{rhhp} q$ then $p = q$.

2.2. Timing

Process algebra with timing [23 24 25] can be used to describe or analyze systems with time-dependent behaviors, which is an extension of process algebra ACP [4]. The timing of actions is either relative or absolute, and the time scale on which the time is measured is either discrete or continuous. The four resulted theories are generalizations of ACP without timing.

This work (truly concurrent process algebra with timing) is a generalization of APTC without timing (see
Similarly to process algebra with timing \cite{23} \cite{24} \cite{25}, there are also four resulted theories. The four theories with timing (four truly concurrent process algebras with timing and four process algebras with timing) will be explained in details in the following sections, and we do not repeat again in this subsection.

3. Discrete Relative Timing

In this section, we will introduce a version of APTC with relative timing and time measured on a discrete time scale. Measuring time on a discrete time scale means that time is divided into time slices and timing of actions is done with respect to the time slices in which they are performed. With respect to relative timing, timing is relative to the execution time of the previous action, and if the previous action does not exist, the start-up time of the whole process.

Like APTC without timing, let us start with a basic algebra for true concurrency called BATC^{drt} (BATC with discrete relative timing). Then we continue with APTC^{drt} (APTC with discrete relative timing).

3.1. Basic Definitions

In this subsection, we will introduce some basic definitions about timing. These basic concepts come from \cite{25}, we introduce them into the backgrounds of true concurrency.

**Definition 3.1 (Undelayable actions).** Undelayable actions are defined as atomic processes that perform an action in the current time slice and then terminate successfully. We use a constant \(a\) to represent the undelayable action, that is, the atomic process that performs the action \(a\) in the current time slice and then terminates successfully.

**Definition 3.2 (Undelayable deadlock).** Undelayable deadlock \(\delta\) is an additional process that is neither capable of performing any action nor capable of idling till the next time slice.

**Definition 3.3 (Relative delay).** The relative delay of the process \(p\) for \(n (n \in \mathbb{N})\) time slices is the process that idles till the \(n\)th-next time slice and then behaves like \(p\). The operator \(\sigma_{rel}\) is used to represent the relative delay, and let \(\sigma_{rel}^n(t) = n\sigma_{rel}\).

**Definition 3.4 (Deadlocked process).** Deadlocked process \(\dot{\delta}\) is an additional process that has deadlocked before the current time slice. After a delay of one time slice, the undelayable deadlock \(\delta\) and the deadlocked process \(\dot{\delta}\) are indistinguishable from each other.

**Definition 3.5 (Truly concurrent bisimulation equivalences with time-related capabilities).** The following requirement with time-related capabilities is added to truly concurrent bisimulation equivalences \(\sim_p, \sim_s, \sim_{hp}\) and \(\sim_{hhp}\):

- if a process is capable of first idling till a certain time slice and next going on as another process, then any equivalent process must be capable of first idling till the same time slice and next going on as a process equivalent to the other process;
- if a process has deadlocked before the current time slice, then any equivalent process must have deadlocked before the current time slice.

**Definition 3.6 (Relative time-out).** The relative time-out \(\upsilon_{rel}\) of a process \(p\) after \(n (n \in \mathbb{N})\) time slices behaves either like the part of \(p\) that does not idle till the \(n\)th-next time slice, or like the deadlocked process after a delay of \(n\) time slices if \(p\) is capable of idling till the \(n\)th-next time slice; otherwise, like \(p\). And let \(\upsilon_{rel}^n(t) = n\upsilon_{rel}\).

**Definition 3.7 (Relative initialization).** The relative initialization \(\uparrow_{rel}\) of a process \(p\) after \(n (n \in \mathbb{N})\) time slices behaves like the part of \(p\) that idles till the \(n\)th-next time slice if \(p\) is capable of idling till that time slice; otherwise, like the deadlocked process after a delay of \(n\) time slices. And we let \(\uparrow_{rel}^n(t) = n\uparrow_{rel}\).
In this subsection, we will introduce the theory BATC\textsuperscript{drt}.

3.2.1. The Theory BATC\textsuperscript{drt}

**Definition 3.8** (Signature of BATC\textsuperscript{drt}). The signature of BATC\textsuperscript{drt} consists of the sort \( P_{rel} \) of processes with discrete relative timing, the undelayable action constants \( a \in A_\delta \), the undelayable deadlock constant \( \delta \), the alternative composition operator \( + : P_{rel} \times P_{rel} \rightarrow P_{rel} \), the sequential composition operator \( ; : P_{rel} \times P_{rel} \rightarrow P_{rel} \), the relative delay operator \( \sigma_{rel} : N \times P_{rel} \rightarrow P_{rel} \), the deadlocked process \( \tau_{rel} : N \times P_{rel} \rightarrow P_{rel} \), the relative time-out operator \( \nu_{rel} : N \times P_{rel} \rightarrow P_{rel} \), and the relative initialization operator \( \iota_{rel} : N \times P_{rel} \rightarrow P_{rel} \).

The set of axioms of BATC\textsuperscript{drt} consists of the laws given in Table 9.

| No. | Axiom |
|-----|-------|
| A1  | \( x + y = y + x \) |
| A2  | \( (x + y) + z = x + (y + z) \) |
| A3  | \( x + x = x \) |
| A4  | \( (x + y) \cdot z = x \cdot z + y \cdot z \) |
| A5  | \( (x \cdot y) \cdot z = x \cdot (y \cdot z) \) |
| A6ID| \( x + \delta = x \) |
| A7ID| \( \delta \cdot x = \delta \) |
| DRT1| \( \sigma_{rel}^0(x) = x \) |
| DRT2| \( \sigma_{rel}^m(\sigma_{rel}^n(x)) = \sigma_{rel}^{m+n}(x) \) |
| DRT3| \( \sigma_{rel}^0(\sigma_{rel}^n(x)) = \sigma_{rel}^n(\nu_{rel}(x)) \) |
| DRT4| \( \sigma_{rel}^0(x \cdot y) = \sigma_{rel}^n(x \cdot y) \) |
| DRT7| \( \sigma_{rel}^0(\delta) = \delta \) |
| A6DRA| \( a + \delta = \delta \) |
| DRT0| \( \nu_{rel}(\delta) = \delta \) |
| DRT01| \( \nu_{rel}(\delta) = (\delta) \) |
| DRT02| \( \sigma_{rel}^n(\nu_{rel}(\delta)) = \sigma_{rel}^n(\delta) \) |
| DRT03| \( \nu_{rel}(\sigma_{rel}(x)) = \sigma_{rel}(\nu_{rel}(x)) \) |
| DRT04| \( \nu_{rel}(\nu_{rel}(x + y)) = \nu_{rel}(x) + \nu_{rel}(y) \) |
| DRT05| \( \nu_{rel}(\nu_{rel}(x \cdot y)) = \nu_{rel}(x) \cdot \nu_{rel}(y) \) |
| DRT0| \( \nu_{rel}(\delta) = \sigma_{rel}(\delta) \) |
| DRT1| \( \nu_{rel}(x) = x \) |
| DRT2| \( \nu_{rel}(\delta) = \sigma_{rel}(\delta) \) |
| DRT3| \( \nu_{rel}(\sigma_{rel}(x)) = \sigma_{rel}(\nu_{rel}(x)) \) |
| DRT4| \( \nu_{rel}(x + y) = \nu_{rel}(x) + \nu_{rel}(y) \) |
| DRT5| \( \nu_{rel}(x \cdot y) = \nu_{rel}(x) \cdot \nu_{rel}(y) \) |

Table 9. Axioms of BATC\textsuperscript{drt} (\( a \in A_\delta, m, n \geq 0 \))

The operational semantics of BATC\textsuperscript{drt} are defined by the transition rules in Table 10. Where \( \uparrow \) is a unary deadlocked predicate, and \( t \uparrow \equiv -(t \uparrow) \); \( t \uparrow \equiv t \downarrow \) means that process \( t \) is capable of first idling till the \( m \)-th next time slice, and then proceeding as process \( t' \).

3.2.2. Elimination

**Definition 3.9** (Basic terms of BATC\textsuperscript{drt}). The set of basic terms of BATC\textsuperscript{drt}, \( B(BATC\textsuperscript{drt}) \), is inductively defined as follows by two auxiliary sets \( B_0(BATC\textsuperscript{drt}) \) and \( B_1(BATC\textsuperscript{drt}) \):

1. If \( a \in A_\delta \), then \( a \in B_1(BATC\textsuperscript{drt}) \);
2. If \( a \in A \) and \( t \in B(BATC\textsuperscript{drt}) \), then \( a \cdot t \in B_1(BATC\textsuperscript{drt}) \);
Theorem 3.11 (Generalization of BATC<sub>drt</sub>). By the definitions of \( a = a \) for each \( a \in A \) and \( \delta = \delta \), BATC<sub>drt</sub> is a generalization of BATC.

Proof. It follows from the following two facts.

1. The transition rules of BATC in section 2.1 are all source-dependent;
2. The sources of the transition rules of BATC<sub>drt</sub> contain an occurrence of \( \delta, a, a, \sigma_{rel}^m, v_{rel}^m \) and \( \overline{v}_{rel}^m \).

So, BATC is an embedding of BATC<sub>drt</sub>, as desired.\[\square\]
3.2.4. Congruence

Theorem 3.12 (Congruence of BATC\textsuperscript{drt}). Truly concurrent bisimulation equivalences are all congruences with respect to BATC\textsuperscript{drt}. That is,

- pomset bisimulation equivalence $\sim_p$ is a congruence with respect to BATC\textsuperscript{drt};
- step bisimulation equivalence $\sim_s$ is a congruence with respect to BATC\textsuperscript{drt};
- hp-bisimulation equivalence $\sim_{hp}$ is a congruence with respect to BATC\textsuperscript{drt};
- hhp-bisimulation equivalence $\sim_{hhp}$ is a congruence with respect to BATC\textsuperscript{drt}.

Proof. It is easy to see that $\sim_p$, $\sim_s$, $\sim_{hp}$ and $\sim_{hhp}$ are all equivalent relations on BATC\textsuperscript{drt} terms, it is only sufficient to prove that $\sim_p$, $\sim_s$, $\sim_{hp}$ and $\sim_{hhp}$ are all preserved by the operators $\sigma_{\text{rel}}^n$, $v_{\text{rel}}^m$ and $\uparrow_{\text{rel}}$. It is trivial and we omit it.

3.2.5. Soundness

Theorem 3.13 (Soundness of BATC\textsuperscript{drt}). The axiomatization of BATC\textsuperscript{drt} is sound modulo truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$, $\sim_{hp}$ and $\sim_{hhp}$. That is,

1. let $x$ and $y$ be BATC\textsuperscript{drt} terms. If BATC\textsuperscript{drt}$\vdash x = y$, then $x \sim_p y$;
2. let $x$ and $y$ be BATC\textsuperscript{drt} terms. If BATC\textsuperscript{drt}$\vdash x = y$, then $x \sim_s y$;
3. let $x$ and $y$ be BATC\textsuperscript{drt} terms. If BATC\textsuperscript{drt}$\vdash x = y$, then $x \sim_{hp} y$;
4. let $x$ and $y$ be BATC\textsuperscript{drt} terms. If BATC\textsuperscript{drt}$\vdash x = y$, then $x \sim_{hhp} y$.

Proof. Since $\sim_p$, $\sim_s$, $\sim_{hp}$ and $\sim_{hhp}$ are both equivalent and congruent relations, we only need to check if each axiom in Table 9 is sound modulo $\sim_p$, $\sim_s$, $\sim_{hp}$ and $\sim_{hhp}$ respectively.

1. We only check the soundness of the non-trivial axiom $DRT03$ modulo $\sim_s$. Let $p$ be BATC\textsuperscript{drt} processes, and $v_{\text{rel}}^m(\sigma_{\text{rel}}^n(p)) = \sigma_{\text{rel}}^n(v_{\text{rel}}^m(p))$, it is sufficient to prove that $v_{\text{rel}}^m(\sigma_{\text{rel}}^n(p)) \sim_s \sigma_{\text{rel}}^n(v_{\text{rel}}^m(p))$. By the transition rules of operator $\sigma_{\text{rel}}^n$ and $v_{\text{rel}}^m$ in Table 10, we get

\[
\begin{align*}
&v_{\text{rel}}^m(\sigma_{\text{rel}}^n(p)) \mapsto^n v_{\text{rel}}^m(\sigma_{\text{rel}}(p)) \\
&\sigma_{\text{rel}}^n(v_{\text{rel}}^m(p)) \mapsto^n \sigma_{\text{rel}}^0(v_{\text{rel}}^m(p))
\end{align*}
\]

There are several cases:

\[
\begin{align*}
&\quad p \xrightarrow{a} \sqrt{} \\
&v_{\text{rel}}^m(\sigma_{\text{rel}}^0(p)) \xrightarrow{a} \sqrt{} \\
&\quad \sigma_{\text{rel}}^n(v_{\text{rel}}^m(p)) \xrightarrow{a} \sqrt{} \\
&\quad p \xrightarrow{a} p' \\
&v_{\text{rel}}^m(\sigma_{\text{rel}}(p)) \xrightarrow{a} p' \\
&\quad p \xrightarrow{a} p' \\
&\sigma_{\text{rel}}^0(v_{\text{rel}}^m(p)) \xrightarrow{a} p' \\
&\quad p \uparrow \\
&v_{\text{rel}}^m(\sigma_{\text{rel}}^0(p)) \uparrow
\end{align*}
\]
\[
\frac{p \vdash \sigma_n^\text{rel}(\nu^\text{rel}_n(p)) \uparrow}{\sigma_n^\text{rel}(\nu^\text{rel}_n(p)) \uparrow}
\]
So, we see that each case leads to \( \nu^\text{rel}_n(\sigma_n^\text{rel}(p)) \sim \sigma_n^\text{rel}(\nu^\text{rel}_n(p)) \), as desired.

2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \( \cdot \)) or in concurrency (implicitly defined by \( \cdot \) and \(+\), and explicitly defined by \( \uparrow \)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \( P = \{a, b : a \cdot b\} \). Then the pomset transition labeled by the above \( P \) is just composed of one single event transition labeled by \( a \) succeeded by another single event transition labeled by \( b \), that is, \( \frac{a \uparrow}{\vdash} \).

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 9 is sound modulo pomset bisimulation equivalence, we omit them.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product (\( C_1, f, C_2 \)), \( f : C_1 \to C_2 \) isomorphism. Two process terms \( s \) related to \( C_1 \) and \( t \) related to \( C_2 \), and \( f : C_1 \to C_2 \) isomorphism. Initially, \( (C_1, f, C_2) = (\emptyset, \emptyset, \emptyset) \), and \((\emptyset, \emptyset, \emptyset) \sim_{hp} \). When \( s \sim s' (C_1 \sim C_1') \), there will be \( t \sim t' (C_2 \sim C_2') \), and we define \( f' = f[a \sim a] \). Then, if \((C_1, f, C_2) \sim_{hp} \), then \((C_1', f', C_2') \sim_{hp} \).

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 9 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

4. We just need to add downward-closed condition to the soundness modulo hp-bisimulation equivalence, we omit them.

\[\Box\]

3.2.6. Completeness

**Theorem 3.14** (Completeness of \( \text{BATC}^{\text{dirt}} \)). The axiomatization of \( \text{BATC}^{\text{dirt}} \) is complete modulo truly concurrent bisimulation equivalences \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhp} \). That is,

1. let \( p \) and \( q \) be closed \( \text{BATC}^{\text{dirt}} \) terms, if \( p \sim_p q \) then \( p = q \);
2. let \( p \) and \( q \) be closed \( \text{BATC}^{\text{dirt}} \) terms, if \( p \sim_h q \) then \( p = q \);
3. let \( p \) and \( q \) be closed \( \text{BATC}^{\text{dirt}} \) terms, if \( p \sim_{hp} q \) then \( p = q \);
4. let \( p \) and \( q \) be closed \( \text{BATC}^{\text{dirt}} \) terms, if \( p \sim_{hhp} q \) then \( p = q \).

**Proof.** 1. Firstly, by the elimination theorem of \( \text{BATC}^{\text{dirt}} \), we know that for each closed \( \text{BATC}^{\text{dirt}} \) term \( p \), there exists a closed basic \( \text{BATC}^{\text{dirt}} \) term \( p' \), such that \( \text{BATC}^{\text{dirt}} \vdash p \equiv p' \), so, we only need to consider closed basic \( \text{BATC}^{\text{dirt}} \) terms.

The basic terms modulo associativity and commutativity (AC) of conflict + (defined by axioms A1 and A2 in Table 6), and this equivalence is denoted by \( =_{AC} \). Then, each equivalence class \( s \) modulo AC of + has the following normal form

\[ s_1 + \cdots + s_k \]

with each \( s_i \) either an atomic event or of the form \( t_1 \cdot t_2 \), and each \( s_i \) is called the summand of \( s \).

Now, we prove that for normal forms \( n \) and \( n' \), if \( n \sim_s n' \) then \( n =_{AC} n' \). It is sufficient to induct on the sizes of \( n \) and \( n' \). We can get \( n =_{AC} n' \).

Finally, let \( s \) and \( t \) be basic terms, and \( s \sim_s t \), there are normal forms \( n \) and \( n' \), such that \( s = n \) and \( t = n' \). The soundness theorem of \( \text{BATC}^{\text{dirt}} \) modulo step bisimulation equivalence yields \( s \sim_s n \) and \( t \sim_s n' \), so \( n \sim_s s \sim_s t \sim_s n' \). Since if \( n \sim_s n' \) then \( n =_{AC} n' \), \( s = n =_{AC} n' = t \), as desired.

2. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_p \).
3. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_{hp} \).
4. This case can be proven similarly, just by replacement of $\sim_s$ by $\sim_{hbp}$.

3.3. Algebra for Parallelism in True Concurrency with Discrete Relative Timing

In this subsection, we will introduce APTC\textsuperscript{drt}.

3.3.1. The Theory APTC\textsuperscript{drt}

**Definition 3.15** (Signature of APTC\textsuperscript{drt}). The signature of APTC\textsuperscript{drt} consists of the signature of BATC\textsuperscript{drt}, and the whole parallel composition operator $\parallel : \mathcal{P}_{rel} \times \mathcal{P}_{rel} \rightarrow \mathcal{P}_{rel}$, the parallel operator $| : \mathcal{P}_{rel} \times \mathcal{P}_{rel} \rightarrow \mathcal{P}_{rel}$, the communication merger operator $\triangleright : \mathcal{P}_{rel} \times \mathcal{P}_{rel} \rightarrow \mathcal{P}_{rel}$, and the encapsulation operator $\partial_H : \mathcal{P}_{rel} \rightarrow \mathcal{P}_{rel}$ for all $H \subseteq A$.

The set of axioms of APTC\textsuperscript{drt} consists of the laws given in Table 11.

The operational semantics of APTC\textsuperscript{drt} are defined by the transition rules in Table 12.

3.3.2. Elimination

**Definition 3.16** (Basic terms of APTC\textsuperscript{drt}). The set of basic terms of APTC\textsuperscript{drt}, $\mathcal{B}(\text{APTC}\textsuperscript{drt})$, is inductively defined as follows by two auxiliary sets $\mathcal{B}_0(\text{APTC}\textsuperscript{drt})$ and $\mathcal{B}_1(\text{APTC}\textsuperscript{drt})$:

1. if $a \in A$, then $a \in \mathcal{B}_1(\text{APTC}\textsuperscript{drt})$;
2. if $a \in A$ and $t \in \mathcal{B}(\text{APTC}\textsuperscript{drt})$, then $a \cdot t \in \mathcal{B}_1(\text{APTC}\textsuperscript{drt})$;
3. if $t, t' \in \mathcal{B}_1(\text{APTC}\textsuperscript{drt})$, then $t + t' \in \mathcal{B}_1(\text{APTC}\textsuperscript{drt})$;
4. if $t, t' \in \mathcal{B}_1(\text{APTC}\textsuperscript{drt})$, then $t \parallel t' \in \mathcal{B}_1(\text{APTC}\textsuperscript{drt})$;
5. if $\delta \in \mathcal{B}(\text{APTC}\textsuperscript{drt})$;
6. if $\sigma^n_{rel}(t) \in \mathcal{B}_0(\text{APTC}\textsuperscript{drt})$;
7. if $n > 0$, $t \in \mathcal{B}_1(\text{APTC}\textsuperscript{drt})$ and $t' \in \mathcal{B}_0(\text{APTC}\textsuperscript{drt})$, then $t + \sigma^n_{rel}(t') \in \mathcal{B}_0(\text{APTC}\textsuperscript{drt})$;

**Theorem 3.17** (Elimination theorem). Let $p$ be a closed APTC\textsuperscript{drt} term. Then there is a basic APTC\textsuperscript{drt} term $q$ such that $\text{APTC}\textsuperscript{drt} \vdash p = q$.

**Proof.** It is sufficient to induct on the structure of the closed APTC\textsuperscript{drt} term $p$. It can be proven that $p$ combined by the constants and operators of APTC\textsuperscript{drt} exists an equal basic term $q$, and the other operators not included in the basic terms, such as $v_{rel}$, $\overline{\varepsilon}_{rel}$, $\subseteq$, $\parallel$, $\triangleright$, $\partial_H$, $\Theta$, $\sigma^n_{rel}$, and $\sigma^n_{rel}$ can be eliminated.

3.3.3. Connections

**Theorem 3.18** (Generalization of APTC\textsuperscript{drt}). 1. By the definitions of $a = a$ for each $a \in A$ and $\delta = \delta$, APTC\textsuperscript{drt} is a generalization of APTC.
2. APTC\textsuperscript{drt} is a generalization of BATC\textsuperscript{drt}

**Proof.** 1. It follows from the following two facts.

   (a) The transition rules of APTC in section 2.4 are all source-dependent;
No.  Axiom
P1  \( x \vdash y = x \parallel y + x \mid y \)
P2  \( x \parallel y = y \parallel x \)
P3  \( (x \parallel y) \parallel z = x \parallel (y \parallel z) \)
P4DR  \( \parallel (x \parallel y) = (\parallel x) \parallel y \)
P5DR  \( \parallel (x \parallel y) = (\parallel y) \parallel x \)
P6DR  \( \parallel (x \parallel y) = (\parallel x) \parallel (y \parallel x) \)
P7  \( (x + y) \parallel z = (x \parallel z) + (y \parallel z) \)
P8  \( x \parallel (y + z) = (x \parallel y) + (x \parallel z) \)
DRP9ID  \( \sigma^-\sigma^+_{\text{rel}}(x) = \delta \)
DRP10ID  \( \sigma^-\sigma^+_{\text{rel}}(y) = \delta \)
DRP11  \( \sigma^-\sigma^+_{\text{rel}}(x) \parallel \sigma^-\sigma^+_{\text{rel}}(y) = \delta \)
PID12  \( \delta \parallel x = \delta \)
PID13  \( x \mid \delta = \delta \)
C14DR  \( \delta = \gamma(a, b) \)
C15DR  \( \delta = \gamma(a, b) - y \)
C16DR  \( \delta = a - \gamma(a, b) \parallel x \)
C17DR  \( \delta = \gamma(a, b) - y \parallel (x \mid y) \)
C18  \( (x + y) \parallel z = (x \parallel z) + (y \parallel z) \)
C19  \( x \mid (y + z) = (x \parallel y) + (x \parallel z) \)
DRC20ID  \( \sigma^-\sigma^+_{\text{rel}}(y) \parallel \sigma^-\sigma^+_{\text{rel}}(y) = \delta \)
DRC21ID  \( \sigma^-\sigma^+_{\text{rel}}(x) \parallel \sigma^-\sigma^+_{\text{rel}}(y) = \delta \)
DRC22  \( \sigma^-\sigma^+_{\text{rel}}(x) \parallel \sigma^-\sigma^+_{\text{rel}}(y) = \sigma^-\sigma^+_{\text{rel}}(x \parallel y) \)
C1D23  \( \delta \mid x = \delta \)
C1D24  \( x \mid \delta = \delta \)
CE26DRID  \( \Theta(y) = a \parallel \delta \)
CE27  \( \Theta(x + y) = \Theta(x) \ll y + \Theta(y) \ll x \)
CE28  \( \Theta(x \parallel y) = \Theta(x) \cdot \Theta(y) \)
CE29  \( \Theta(x \parallel y) = (\Theta(x) \ll y) \parallel (\Theta(y) \ll x) \parallel x \)
CE30  \( \Theta(x \parallel y) = (\Theta(x) \ll y) \parallel (\Theta(y) \ll x) \parallel x \)
U31DRID  \( (\parallel_{a} b) \mid c = a \parallel_{a} (b \parallel c) \)
U32DRID  \( (\parallel_{a} b) \mid c = a \parallel_{a} (b \parallel c) \)
U33DRID  \( (\parallel_{a} b) \mid c = a \parallel_{a} (b \parallel c) \)
U34DRID  \( a \parallel_{a} (b \parallel c) = a \parallel_{a} (b \parallel c) \)
U35DRID  \( a \parallel_{a} (b \parallel c) = a \parallel_{a} (b \parallel c) \)
U36  \( (x + y) \ll z = (x \ll z) + (y \ll z) \)
U37  \( (x \parallel y) \ll z = (x \ll z) \cdot (y \ll z) \)
U38  \( (x \parallel y) \ll z = (x \ll z) \parallel (y \ll z) \)
U39  \( (x \parallel y) \ll z = (x \ll z) \parallel (y \ll z) \)
U40  \( x \ll (y + z) = (x \parallel y) \ll z \)
U41  \( x \ll (y + z) = (x \parallel y) \ll z \)
U42  \( x \ll (y \parallel z) = (x \parallel y) \ll z \)
U43  \( x \ll (y \parallel z) = (x \parallel y) \ll z \)
D1DRID  \( \delta \parallel_{H} H \parallel_{H} \delta \parallel_{H} \)
D2DRID  \( \delta \parallel_{H} H \parallel_{H} \delta \parallel_{H} \)
D3DRID  \( \delta \parallel_{H} H \parallel_{H} \delta \parallel_{H} \)
D4  \( \delta \parallel_{H} H \parallel_{H} \delta \parallel_{H} \)
D5  \( \delta \parallel_{H} H \parallel_{H} \delta \parallel_{H} \)
D6  \( \delta \parallel_{H} H \parallel_{H} \delta \parallel_{H} \)
D7  \( \delta \parallel_{H} H \parallel_{H} \delta \parallel_{H} \)

Table 11. Axioms of APTCdr\( (a, b, c \in A_{\delta}, n \geq 0) \)

(b) The sources of the transition rules of APTCdr\( contain an occurrence of \( \delta, a, \sigma_{\text{rel}}^{n}, \nu_{\text{rel}}^{n} \) and \( \tau_{\text{rel}}^{n} \).

So, APTC\( is an embedding of APTCdr\(, as desired.

2. It follows from the following two facts.

(a) The transition rules of BATCdr\( are all source-dependent;

(b) The sources of the transition rules of APTCdr\( contain an occurrence of \( \delta, a, \parallel, \mid, \Theta, \ll, \partial_{H} \).
\[
\begin{align*}
\frac{x \xrightarrow{a} \sqrt{y \xrightarrow{b}}} {x \parallel y \xrightarrow{(a,b)} \sqrt{y}} & \quad \frac{x \xrightarrow{a} x' \xrightarrow{b}} {x \parallel y \xrightarrow{(a,b)} x'} \\
\frac{x \xrightarrow{a} \sqrt{y \xrightarrow{b}}'} {x \parallel y \xrightarrow{(a,b)} y'} & \quad \frac{x \xrightarrow{a} x' \xrightarrow{b}'} {x \parallel y \xrightarrow{(a,b)} y'} \\
\frac{x \xrightarrow{m} x' \xrightarrow{m} y'} {x \parallel y \xrightarrow{(a,b)} x' \parallel y'} & \quad \frac{x \xrightarrow{\Uparrow} y \xrightarrow{\Uparrow}} {x \parallel y \xrightarrow{\Uparrow}} \\
\end{align*}
\]

\[
\begin{align*}
\frac{x \xrightarrow{a} \sqrt{y \xrightarrow{b}}'} {x \parallel y \xrightarrow{(a,b)} y'} & \quad \frac{x \xrightarrow{a} x' \xrightarrow{b}'} {x \parallel y \xrightarrow{(a,b)} x'} \\
\frac{x \xrightarrow{a} \sqrt{y \xrightarrow{b}}'} {x \parallel y \xrightarrow{(a,b)} y'} & \quad \frac{x \xrightarrow{a} x' \xrightarrow{b}'} {x \parallel y \xrightarrow{(a,b)} x'} \\
\frac{x \xrightarrow{m} x' \xrightarrow{m} y'} {x \parallel y \xrightarrow{(a,b)} x' \parallel y'} & \quad \frac{x \xrightarrow{\Uparrow} y \xrightarrow{\Uparrow}} {x \parallel y \xrightarrow{\Uparrow}} \\
\end{align*}
\]

Table 12. Transition rules of \(\text{APTC}^{\text{drt}}(a,b,c \in A, m > 0)\)

So, \(\text{BATC}^{\text{drt}}\) is an embedding of \(\text{APTC}^{\text{drt}}\), as desired.

\[\square\]

3.3.4. Congruence

**Theorem 3.19** (Congruence of \(\text{APTC}^{\text{drt}}\)). Truly concurrent bisimulation equivalences \(\sim_p, \sim_s\) and \(\sim_{hp}\) are all congruences with respect to \(\text{APTC}^{\text{drt}}\). That is,

- pomset bisimulation equivalence \(\sim_p\) is a congruence with respect to \(\text{APTC}^{\text{drt}}\);
- step bisimulation equivalence \(\sim_s\) is a congruence with respect to \(\text{APTC}^{\text{drt}}\);
• hp-bisimulation equivalence ∼hp is a congruence with respect to APTC\textsuperscript{drt}.

Proof. It is easy to see that ∼p, ∼s, and ∼hp are all equivalent relations on APTC\textsuperscript{drt} terms, it is only sufficient to prove that ∼p, ∼s, and ∼hp are all preserved by the operators σ\textsuperscript{rel}n, ν\textsuperscript{rel}n and τ\textsuperscript{rel}. It is trivial and we omit it.

3.3.5. Soundness

Theorem 3.20 (Soundness of APTC\textsuperscript{drt}). The axiomatization of APTC\textsuperscript{drt} is sound modulo truly concurrent bisimulation equivalences ∼p, ∼s, and ∼hp. That is,

1. let x and y be APTC\textsuperscript{drt} terms. If APTC\textsuperscript{drt} ⊨ x = y, then x ∼ s y;
2. let x and y be APTC\textsuperscript{drt} terms. If APTC\textsuperscript{drt} ⊨ x = y, then x ∼ p y;
3. let x and y be APTC\textsuperscript{drt} terms. If APTC\textsuperscript{drt} ⊨ x = y, then x ∼ hp y.

Proof. Since ∼p, ∼s, and ∼hp are both equivalent and congruent relations, we only need to check if each axiom in Table 11 is sound modulo ∼p, ∼s, and ∼hp respectively.

1. We only check the soundness of the non-trivial axiom DRP11 modulo ∼s. Let p, q be APTC\textsuperscript{drt} processes, and σ\textsuperscript{rel}n(p) ∥ σ\textsuperscript{rel}n(q) = σ\textsuperscript{rel}n(p ∥ q), it is sufficient to prove that σ\textsuperscript{rel}n(p) ∥ σ\textsuperscript{rel}n(q) ∼ s σ\textsuperscript{rel}n(p ∥ q). By the transition rules of operator σ\textsuperscript{rel} and ∥ in Table 11 we get

| σ\textsuperscript{rel}n(p) ∥ σ\textsuperscript{rel}n(q) | →n σ\textsuperscript{rel}n(p ∥ q) |
|---------------------------------|------------------|
| σ\textsuperscript{rel}n(p) ∥ σ\textsuperscript{rel}n(q) | →n σ\textsuperscript{rel}n(p ∥ q) |

There are several cases:

\[ \frac{p \xrightleftharpoons{a} q \xrightarrow{b}}{\sigma_{\text{rel}}^n(p) \parallel \sigma_{\text{rel}}^n(q) \xrightarrow{\{a,b\}}} \sqrt{v} \]

\[ \frac{p \xrightleftharpoons{a} q \xrightarrow{b}}{\sigma_{\text{rel}}^n(p) \parallel \sigma_{\text{rel}}^n(q) \xrightarrow{\{a,b\}} p'} \]

\[ \frac{p \xrightarrow{a} q \xrightarrow{b}}{\sigma_{\text{rel}}^n(p) \parallel \sigma_{\text{rel}}^n(q) \xrightarrow{\{a,b\}} p'} \]

\[ \frac{p \xrightarrow{a} q \xrightarrow{b}}{\sigma_{\text{rel}}^n(p) \parallel \sigma_{\text{rel}}^n(q) \xrightarrow{\{a,b\}} q'} \]

\[ \frac{p \xleftleftharpoons{a} q \xrightarrow{b}}{\sigma_{\text{rel}}^n(p) \parallel \sigma_{\text{rel}}^n(q) \xrightarrow{\{a,b\}} q'} \]

\[ \frac{p \xrightarrow{a} q \xrightarrow{b}}{\sigma_{\text{rel}}^n(p) \parallel \sigma_{\text{rel}}^n(q) \xrightarrow{\{a,b\}} q'} \]

\[ \frac{p \xleftleftharpoons{a} q \xrightarrow{b}}{\sigma_{\text{rel}}^n(p) \parallel \sigma_{\text{rel}}^n(q) \xrightarrow{\{a,b\}} p' \parallel q'} \]
\[
\begin{align*}
\frac{p \xrightarrow{a} p' \quad q \xrightarrow{b} q'}{
\sigma^0_{rel}(p \parallel q) \xrightarrow{(a,b)} p' \parallel q'}
\end{align*}
\]

1. Firstly, by the elimination theorem of \(\text{APT}\text{C}^{\text{drt}}\), we know that \(\text{APT}\text{C}^{\text{drt}}\text{C}^{\text{drt}}\) is complete modulo truly concurrent bisimulation equivalences \(\sim_p\), \(\sim_s\), and \(\sim_{hp}\). That is,

2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \(\cdot\)) or in concurrency (implicitly defined by \(+\)), and explicitly defined by \(\parallel\). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \(P = \{a, b : a \cdot b\}\).

Then the pomset transition labeled by the above \(P\) is just composed of one single event transition labeled by \(a\) succeeded by another single event transition labeled by \(b\), that is, \(\xrightarrow{a} \parallel a \rightarrow b\).

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table [1] is sound modulo pomset bisimulation equivalence, we omit them.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product \((C_1, f, C_2), f : C_1 \rightarrow C_2\) isomorphism. Two process terms \(s\) related to \(C_1\) and \(t\) related to \(C_2\), and \(f : C_1 \rightarrow C_2\) isomorphism. Initially, \((C_1, f, C_2) = (\varnothing, \varnothing, \varnothing)\), and \((\varnothing, \varnothing, \varnothing) \in \sim_{hp}\). When \(s \xrightarrow{a} s' (C_1 \xrightarrow{a} C'_1)\), there will be \(t \xrightarrow{a} t' (C_2 \xrightarrow{a} C'_2)\), and we define \(f' = f[a \rightarrow a]\). Then, if \((C_1, f, C_2) \in \sim_{hp}\), then \((C'_1, f', C'_2) \in \sim_{hp}\).

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table [1] is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

\[\square\]

3.3.6. Completeness

**Theorem 3.21** (Completeness of APTC\(^{\text{drt}}\)). The axiomatization of APTC\(^{\text{drt}}\) is complete modulo truly concurrent bisimulation equivalences \(\sim_p\), \(\sim_s\), and \(\sim_{hp}\). That is,

1. let \(p\) and \(q\) be closed APTC\(^{\text{drt}}\) terms, if \(p \sim_s q\) then \(p = q\);
2. let \(p\) and \(q\) be closed APTC\(^{\text{drt}}\) terms, if \(p \sim_p q\) then \(p = q\);
3. let \(p\) and \(q\) be closed APTC\(^{\text{drt}}\) terms, if \(p \sim_{hp} q\) then \(p = q\).

**Proof.** 1. Firstly, by the elimination theorem of APTC\(^{\text{drt}}\), we know that for each closed APTC\(^{\text{drt}}\) term \(p\), there exists a closed basic APTC\(^{\text{drt}}\) term \(p'\), such that APTC\(^{\text{drt}}\) \(\vdash p = p'\), so, we only need to consider closed basic APTC\(^{\text{drt}}\) terms.

The basic terms modulo associativity and commutativity (AC) of conflict \(+\) (defined by axioms A1 and A2 in Table [9]) and associativity and commutativity (AC) of parallel \(\parallel\) (defined by axioms P2 and P3 in Table [1]), and these equivalences is denoted by \(=_{AC}\). Then, each equivalence class \(s\) modulo AC of \(+\) and \(\parallel\) has the following normal form
with each $s_i$ either an atomic event or of the form
\[ t_1 \cdot \cdots \cdot t_m \]
with each $t_j$ either an atomic event or of the form
\[ u_1 \parallel \cdots \parallel u_n \]
with each $u_l$ an atomic event, and each $s_i$ is called the summand of $s$.

Now, we prove that for normal forms $n$ and $n'$, if $n \sim s n'$ then $n =_{AC} n'$. It is sufficient to induct on the sizes of $n$ and $n'$. We can get $n =_{AC} n'$.

Finally, let $s$ and $t$ be basic APTC\textsuperscript{drt} terms, and $s \sim_s t$, there are normal forms $n$ and $n'$, such that $s = n$ and $t = n'$. The soundness theorem modulo step bisimulation equivalence yields $s \sim_s n$ and $t \sim_s n'$, so $n \sim_s s \sim_s t \sim_s n'$. Since if $n \sim_s n'$ then $n =_{AC} n'$, $s = n =_{AC} n' = t$, as desired.

2. This case can be proven similarly, just by replacement of $\sim_s$ by $\sim_p$.

3. This case can be proven similarly, just by replacement of $\sim_s$ by $\sim_{hp}$.

\[ \square \]

4. Discrete Absolute Timing

In this section, we will introduce a version of APTC with absolute timing and time measured on a discrete time scale. Measuring time on a discrete time scale means that time is divided into time slices and timing of actions is done with respect to the time slices in which they are performed. While in absolute timing, all timing is counted from the start of the whole process.

Like APTC without timing, let us start with a basic algebra for true concurrency called BATC\textsuperscript{dat} (BATC with discrete absolute timing). Then we continue with APTC\textsuperscript{dat} (APTC with discrete absolute timing).

4.1. Basic Definitions

In this subsection, we will introduce some basic definitions about timing. These basic concepts come from [25], we introduce them into the backgrounds of true concurrency.

**Definition 4.1** (Undelayable actions). Undelayable actions are defined as atomic processes that perform an action in the current time slice and then terminate successfully. We use a constant $a$ to represent the undelayable action, that is, the atomic process that performs the action $a$ in the current time slice and then terminates successfully.

**Definition 4.2** (Undelayable deadlock). Undelayable deadlock $\delta$ is an additional process that is neither capable of performing any action nor capable of idling till after time slice 1.

**Definition 4.3** (Absolute delay). The absolute delay of the process $p$ for $n$ ($n \in \mathbb{N}$) time slices is the process that idles $n$ time slices longer than $p$ and otherwise behaves like $p$. The operator $\sigma_{abs}$ is used to represent the absolute delay, and let $\sigma_{abs}^n(t) = n \sigma_{abs} t$.

**Definition 4.4** (Deadlocked process). Deadlocked process $\mathring{\delta}$ is an additional process that has deadlocked before time slice 1. After a delay of one time slice, the undelayable deadlock $\delta$ and the deadlocked process $\mathring{\delta}$ are indistinguishable from each other.

**Definition 4.5** (Truly concurrent bisimulation equivalences with time-related capabilities). The following requirement with time-related capabilities is added to truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$, $\sim_{hp}$ and $\sim_{hhp}$ and Definition 3.5:

- In case of absolute timing, the requirements in Definition 3.5 apply to the capabilities in a certain time slice.
Definition 4.6 (Absolute time-out). The absolute time-out \( \nu_{\text{abs}} \) of a process \( p \) at time \( n \) \( (n \in \mathbb{N}) \) behaves either like the part of \( p \) that does not idle till time slice \( n+1 \), or like the deadlocked process after a delay of \( n \) time slices if \( p \) is capable of idling till time slice \( n+1 \); otherwise, like \( p \). And let \( \nu_{\text{abs}}^n(t) = n \nu_{\text{abs}}.t \).

Definition 4.7 (Absolute initialization). The absolute initialization \( \nu_{\text{abs}} \) of a process \( p \) at time \( n \) \( (n \in \mathbb{N}) \) behaves like the part of \( p \) that idles till time slice \( n+1 \) if \( p \) is capable of idling till that time slice; otherwise, like the deadlocked process after a delay of \( n \) time slices. And we let \( \nu_{\text{abs}}^n(t) = n \nu_{\text{abs}}.t \).

4.2. Basic Algebra for True Concurrency with Discrete Absolute Timing

In this subsection, we will introduce the theory \( \text{BATC}^{\text{dat}} \).

4.2.1. The Theory \( \text{BATC}^{\text{dat}} \)

Definition 4.8 (Signature of \( \text{BATC}^{\text{dat}} \)). The signature of \( \text{BATC}^{\text{dat}} \) consists of the sort \( \mathcal{P}_{\text{abs}} \) of processes with discrete absolute timing, the undelayable action constants \( a :\rightarrow \mathcal{P}_{\text{abs}} \) for each \( a \in A \), the undelayable deadlock constant \( \delta :\rightarrow \mathcal{P}_{\text{abs}} \), the alternative composition operator \( + : \mathcal{P}_{\text{abs}} \times \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}} \), the sequential composition operator \( \circ : \mathcal{P}_{\text{abs}} \times \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}} \), the absolute delay operator \( \sigma_{\text{abs}} : \mathbb{N} \times \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}} \), the deadlock operator \( \delta : \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}} \), the absolute time-out operator \( \nu_{\text{abs}} : \mathbb{N} \times \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}} \) and the absolute initialization operator \( \nu_{\text{abs}}^n : \mathbb{N} \rightarrow \mathcal{P}_{\text{abs}} \).

The set of axioms of \( \text{BATC}^{\text{dat}} \) consists of the laws given in Table 13.

| No. | Axiom |
|-----|-------|
| A1  | \( x + y = y + x \) |
| A2  | \( (x + y) + z = x + (y + z) \) |
| A3  | \( x + x = x \) |
| A4  | \( (x + y) \cdot z = x \cdot (z + y) \) |
| A5  | \( (x \cdot y) \cdot z = x \cdot (y \cdot z) \) |
| A6ID| \( x + \delta = x \) |
| A7ID| \( \delta \cdot x = \delta \) |
| DAT1| \( \sigma_{\text{abs}}^0(x) = \nu_{\text{abs}}^0(x) \) |
| DAT2| \( \sigma_{\text{abs}}^n_x(x) = \sigma_{\text{abs}}^{n+1}_x(x) \) |
| DAT3| \( \sigma_{\text{abs}}^n_x(x) + \sigma_{\text{abs}}^m_y(y) = \sigma_{\text{abs}}^{n+m}_x + y \) |
| DAT4| \( \sigma_{\text{abs}}^n_x(x) \cdot \nu_{\text{abs}}^m_y(y) = \sigma_{\text{abs}}^{n+m}_x \cdot y \) |
| DAT5| \( \sigma_{\text{abs}}^n_x(x) \cdot (\nu_{\text{abs}}^m_y(y) + \sigma_{\text{abs}}^n_z(z)) = \sigma_{\text{abs}}^{n+m}_x \cdot (y \cdot \nu_{\text{abs}}^n(z)) \) |
| DAT6| \( \sigma_{\text{abs}}^n_x(\delta) \cdot x = \sigma_{\text{abs}}^{n+1}_x(\delta) \) |
| DAT7| \( \sigma_{\text{abs}}^n(\delta) = \hat{\delta} \) |
| DAT00| \( \nu_{\text{abs}}^n(\delta) = \delta \) |
| DAT01| \( \nu_{\text{abs}}^n(x) = x \) |
| DAT02| \( \nu_{\text{abs}}^n(\hat{x}) = \hat{x} \) |
| DAT03| \( \nu_{\text{abs}}^n(x) = \sigma_{\text{abs}}^n(x) \) |
| DAT04| \( \nu_{\text{abs}}^n(x + y) = \nu_{\text{abs}}^n(x) + \nu_{\text{abs}}^n(y) \) |
| DAT05| \( \nu_{\text{abs}}^n(x \cdot y) = \nu_{\text{abs}}^n(x) \cdot \nu_{\text{abs}}^n(y) \) |
| DAT06| \( \nu_{\text{abs}}^n(\delta) = \delta \) |
| DAT07| \( \nu_{\text{abs}}^n(x) = \sigma_{\text{abs}}^n(x) \) |
| DAT08| \( \nu_{\text{abs}}^n(\hat{x}) = \sigma_{\text{abs}}^{n+1}(\delta) \) |
| DAT09| \( \nu_{\text{abs}}^n(x) = \sigma_{\text{abs}}^n(x) \cdot y \) |
| DAT10| \( \nu_{\text{abs}}^n(x + y) = \nu_{\text{abs}}^n(x) + \nu_{\text{abs}}^n(y) \) |
| DAT11| \( \nu_{\text{abs}}^n(x \cdot y) = \nu_{\text{abs}}^n(x) \cdot \nu_{\text{abs}}^n(y) \) |

Table 13. Axioms of \( \text{BATC}^{\text{dat}} \) (\( a \in A, m, n \geq 0 \))

The operational semantics of \( \text{BATC}^{\text{dat}} \) are defined by the transition rules in Table 14. The transition rules are defined on \( (t, n) \), where \( t \) is a term and \( n \in \mathbb{N} \). Where \( \uparrow \) is a unary deadlocked predicate, and
\( \langle t, n \rangle \uparrow \equiv \neg((t, n) \uparrow); \{ t, n \} \rightarrow^m \{ t', n' \} \) means that process \( t \) is capable of first idling till the \( m \)th-next time slice, and then proceeding as process \( t' \) and \( m + n = n' \).

4.2.2. Elimination

**Definition 4.9 (Basic terms of BATC\(_{\text{dat}}\)).** The set of basic terms of BATC\(_{\text{dat}}\), \( \mathcal{B}(\text{BATC}\_{\text{dat}}) \), is inductively defined as follows by two auxiliary sets \( \mathcal{B}_0(\text{BATC}\_{\text{dat}}) \) and \( \mathcal{B}_1(\text{BATC}\_{\text{dat}}) \):

1. if \( a \in A_{\delta} \), then \( a \in \mathcal{B}_1(\text{BATC}\_{\text{dat}}) \);
2. if \( a \in A \) and \( t \in \mathcal{B}(\text{BATC}\_{\text{dat}}) \), then \( a \cdot t \in \mathcal{B}_1(\text{BATC}\_{\text{dat}}) \);
3. if \( t, t' \in \mathcal{B}_1(\text{BATC}\_{\text{dat}}) \), then \( t + t' \in \mathcal{B}_1(\text{BATC}\_{\text{dat}}) \);
4. if \( t \in \mathcal{B}_1(\text{BATC}\_{\text{dat}}) \), then \( t \in \mathcal{B}_0(\text{BATC}\_{\text{dat}}) \);
5. if \( n > 0 \) and \( t \in \mathcal{B}_0(\text{BATC}\_{\text{dat}}) \), then \( \sigma^n_{\text{abs}}(t) \in \mathcal{B}_0(\text{BATC}\_{\text{dat}}) \);
6. if \( n > 0 \), \( t \in \mathcal{B}_1(\text{BATC}\_{\text{dat}}) \) and \( t' \in \mathcal{B}_0(\text{BATC}\_{\text{dat}}) \), then \( t + \sigma^n_{\text{abs}}(t') \in \mathcal{B}_0(\text{BATC}\_{\text{dat}}) \);
7. \( \delta \in \mathcal{B}(\text{BATC}\_{\text{dat}}) \);
8. if \( t \in \mathcal{B}_0(\text{BATC}\_{\text{dat}}) \), then \( t \in \mathcal{B}(\text{BATC}\_{\text{dat}}) \).

**Theorem 4.10 (Elimination theorem).** Let \( p \) be a closed BATC\(_{\text{dat}}\) term. Then there is a basic BATC\(_{\text{dat}}\) term \( q \) such that \( \text{BATC}_{\text{dat}} \vdash p = q \).

**Proof.** It is sufficient to induct on the structure of the closed BATC\(_{\text{dat}}\) term \( p \). It can be proven that \( p \) combined by the constants and operators of BATC\(_{\text{dat}}\) exists an equal basic term \( q \), and the other operators not included in the basic terms, such as \( v_{\text{abs}} \) and \( \overline{v}_{\text{abs}} \) can be eliminated.

4.2.3. Connections

**Theorem 4.11 (Generalization of BATC\(_{\text{dat}}\)).** By the definitions of \( a = a \) for each \( a \in A \) and \( \delta = \delta \). BATC\(_{\text{dat}}\) is a generalization of BATC.

**Proof.** It follows from the following two facts.

1. The transition rules of BATC in section 2.1 are all source-dependent;
2. The sources of the transition rules of BATC\(_{\text{dat}}\) contain an occurrence of \( \delta, a, \sigma^n_{\text{abs}}, v^n_{\text{abs}} \) and \( \overline{v}^n_{\text{abs}} \).

So, BATC is an embedding of BATC\(_{\text{dat}}\), as desired.

4.2.4. Congruence

**Theorem 4.12 (Congruence of BATC\(_{\text{dat}}\)).** Truly concurrent bisimulation equivalences are all congruences with respect to BATC\(_{\text{dat}}\). That is,

- pomset bisimulation equivalence \( \sim_p \) is a congruence with respect to BATC\(_{\text{dat}}\);
- step bisimulation equivalence \( \sim_s \) is a congruence with respect to BATC\(_{\text{dat}}\);
- hp-bisimulation equivalence \( \sim_{hp} \) is a congruence with respect to BATC\(_{\text{dat}}\);
- hhp-bisimulation equivalence \( \sim_{hhp} \) is a congruence with respect to BATC\(_{\text{dat}}\).

**Proof.** It is easy to see that \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhp} \) are all equivalent relations on BATC\(_{\text{dat}}\) terms, it is only sufficient to prove that \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhp} \) are all preserved by the operators \( \sigma^n_{\text{abs}}, v^n_{\text{abs}} \) and \( \overline{v}^n_{\text{abs}} \). It is trivial and we omit it.
| Transition Rule | Description |
|-----------------|-------------|
| $(x, n) \xrightarrow{a} (x', n)$ | $\langle \sigma_{\text{abs}}^n(x), n \rangle \xrightarrow{a} \langle \sigma_{\text{abs}}^n(x'), n + m \rangle$ |
| $(x, n) \xrightarrow{\nu} (\sqrt{n})$ | $\langle \sigma_{\text{abs}}^n(x), n + n' \rangle \xrightarrow{\nu} \langle \sigma_{\text{abs}}^n(x'), n + m \rangle$ |

Table 14. Transition rules of $\text{BATC}^\text{lat}(a \in A, m > 0, n, n' \geq 0)$
4.2.5. Soundness

Theorem 4.13 (Soundness of BATC\textsuperscript{dat}). The axiomatization of BATC\textsuperscript{dat} is sound modulo truly concurrent bisimulation equivalences \(\sim_p, \sim_s, \sim_{hp}\) and \(\sim_{hp}\). That is,

1. let x and y be BATC\textsuperscript{dat} terms. If BATC\textsuperscript{dat} \(\vdash x = y\), then \(x \sim s y\);
2. let x and y be BATC\textsuperscript{dat} terms. If BATC\textsuperscript{dat} \(\vdash x = y\), then \(x \sim p y\);
3. let x and y be BATC\textsuperscript{dat} terms. If BATC\textsuperscript{dat} \(\vdash x = y\), then \(x \sim hp y\);
4. let x and y be BATC\textsuperscript{dat} terms. If BATC\textsuperscript{dat} \(\vdash x = y\), then \(x \sim hhp y\).

Proof. Since \(\sim_p, \sim_s, \sim_{hp}\) and \(\sim_{hp}\) are both equivalent and congruent relations, we only need to check if each axiom in Table 13 is sound modulo \(\sim_p, \sim_s, \sim_{hp}\) and \(\sim_{hp}\) respectively.

1. We only check the soundness of the non-trivial axiom DAT\textsubscript{O3} modulo \(\sim_s\). Let \(p\) be BATC\textsuperscript{dat} processes, and \(\nu_{abs}(\sigma^n_{abs}(p)) = \sigma^n_{abs}(\nu_{abs}(p))\), it is sufficient to prove that \(\nu_{abs}(\sigma^n_{abs}(p)) \sim_s \sigma^n_{abs}(\nu_{abs}(p))\). By the transition rules of operator \(\sigma^n_{abs}\) and \(\nu_{abs}\) in Table 14 we get

\[
\begin{align*}
(p,0) & \vdash (p,0) \\
\nu^m_{abs}(\sigma^n_{abs}(p)), n') & \Rightarrow (\nu^m_{abs}(\sigma^n_{abs}(p)), n'+n) \\
\sigma^n_{abs}(\nu^m_{abs}(p)), n') & \Rightarrow (\sigma^n_{abs}(\nu^m_{abs}(p)), n'+n)
\end{align*}
\]

There are several cases:

\[
\begin{align*}
(p,n') & \Rightarrow (\sqrt{n'},n') \\
\nu^m_{abs}(\sigma^n_{abs}(p)), n'+n & \Rightarrow (\sqrt{n'}+n) \\
\sigma^n_{abs}(\nu^m_{abs}(p)), n'+n & \Rightarrow (\sqrt{n'}+n) \\
\sigma^n_{abs}(\nu^m_{abs}(p)), n'+n & \Rightarrow (\sigma^n_{abs}(p'), n'+n) \\
(p,n') & \Rightarrow (p',n') \\
(p,n') & \Rightarrow (p',n') \\
\sigma^n_{abs}(\nu^m_{abs}(p)), n'+n & \Rightarrow (\sigma^n_{abs}(p'), n'+n) \\
\sigma^n_{abs}(\nu^m_{abs}(p)), n'+n & \Rightarrow (\sigma^n_{abs}(p'), n'+n) \\
(p,n') & \Rightarrow (p',n') \\
\sigma^n_{abs}(\nu^m_{abs}(p)), n'+n & \Rightarrow (\sigma^n_{abs}(p'), n'+n) \\
(p,n') & \Rightarrow (p',n') \\
\sigma^n_{abs}(\nu^m_{abs}(p)), n'+n & \Rightarrow (\sigma^n_{abs}(p'), n'+n)
\end{align*}
\]

So, we see that each case leads to \(\nu^m_{abs}(\sigma^n_{abs}(p)) \sim_s \sigma^n_{abs}(\nu^m_{abs}(p))\), as desired.

2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \(\cdot\)) or in concurrency (implicitly defined by \(\cdot\) and \(+\), and explicitly defined by \(\uparrow\)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \(P = \{a, b : a \cdot b\}\). Then the pomset transition labeled by the above \(P\) is just composed of one single event transition labeled by \(a\) succeeded by another single event transition labeled by \(b\), that is, \(\Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow\).

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 13 is sound modulo pomset bisimulation equivalence, we omit them.
3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product \((C_1, f, C_2, f) : C_1 \rightarrow C_2\) isomorphism. Two process terms \(s\) related to \(C_1\) and \(t\) related to \(C_2\), and \(f : C_1 \rightarrow C_2\) isomorphism. Initially, \((C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)\), and \((\emptyset, \emptyset, \emptyset) \notin \s\). When \(s \sim t' (C_1, \emptyset, C_2')\), there will be \(t \rightarrow t' (C_2, \emptyset, C_2')\), and we define \(f' = f[a \rightarrow a]\). Then, if \((C_1, f, C_2) \in \sim\), then \((C_1', f', C_2') \in \sim\).

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 13 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

4. We just need to add downward-closed condition to the soundness modulo hp-bisimulation equivalence, we omit them.

\[\square\]

4.2.6. Completeness

**Theorem 4.14** (Completeness of BATC\(^{\text{dat}}\)). The axiomatization of BATC\(^{\text{dat}}\) is complete modulo truly concurrent bisimulation equivalences \(\sim, \sim\) and \(\sim_h\). That is,

1. let \(p\) and \(q\) be closed BATC\(^{\text{dat}}\) terms, if \(p \sim q\) then \(p = q\);
2. let \(p\) and \(q\) be closed BATC\(^{\text{dat}}\) terms, if \(p \sim_h q\) then \(p = q\);
3. let \(p\) and \(q\) be closed BATC\(^{\text{dat}}\) terms, if \(p \sim_h q\) then \(p = q\);
4. let \(p\) and \(q\) be closed BATC\(^{\text{dat}}\) terms, if \(p \sim_h q\) then \(p = q\).

**Proof.** 1. Firstly, by the elimination theorem of BATC\(^{\text{dat}}\), we know that for each closed BATC\(^{\text{dat}}\) term \(p\), there exists a closed basic BATC\(^{\text{dat}}\) term \(p'\), such that BATC\(^{\text{dat}}\) \(\vdash p = p'\), so, we only need to consider closed basic BATC\(^{\text{dat}}\) terms.

The basic terms modulo associativity and commutativity (AC) of conflict + (defined by axioms A1 and A2 in Table 13), and this equivalence is denoted by \(\sim_{AC}\). Then, each equivalence class \(s\) modulo AC of + has the following normal form

\[s_1 + \cdots + s_k\]

with each \(s_i\) either an atomic event or of the form \(t_1 \cdot t_2\), and each \(s_i\) is called the summand of \(s\).

Now, we prove that for normal forms \(n\) and \(n'\), if \(n \sim n'\) then \(n = AC n'\). It is sufficient to induct on the sizes of \(n\) and \(n'\). We can get \(n = AC n'\).

Finally, let \(s\) and \(t\) be basic terms, and \(s \sim t\), there are normal forms \(n\) and \(n'\), such that \(s = n\) and \(t = n'\). The soundness theorem of BATC\(^{\text{dat}}\) modulo step bisimulation equivalence yields \(s \sim_n t \sim_n n'\). Since if \(n \sim n'\) then \(n = AC n'\), \(s = n = AC n' = t\), as desired.

2. This case can be proven similarly, just by replacement of \(\sim_s\) by \(\sim_p\).
3. This case can be proven similarly, just by replacement of \(\sim_s\) by \(\sim_{hp}\).
4. This case can be proven similarly, just by replacement of \(\sim_s\) by \(\sim_{hp}\).

\[\square\]

4.3. Algebra for Parallelism in True Concurrency with Discrete Absolute Timing

In this subsection, we will introduce APTC\(^{\text{dat}}\).

**4.3.1. The Theory APTC\(^{\text{dat}}\)**

**Definition 4.15** (Signature of APTC\(^{\text{dat}}\)). The signature of APTC\(^{\text{dat}}\) consists of the signature of BATC\(^{\text{dat}}\), and the whole parallel composition operator \(\parallel : P_{\text{abs}} \times P_{\text{abs}} \rightarrow P_{\text{abs}}\), the parallel operator \(| : P_{\text{abs}} \times P_{\text{abs}} \rightarrow P_{\text{abs}}\), the communication merger operator \(\mid : P_{\text{abs}} \times P_{\text{abs}} \rightarrow P_{\text{abs}}\), and the encapsulation operator \(\partial_H : P_{\text{abs}} \rightarrow P_{\text{abs}}\) for all \(H \subseteq A\).

The set of axioms of APTC\(^{\text{dat}}\) consists of the laws given in Table 13.
The operational semantics of APTC\textsuperscript{dat} are defined by the transition rules in Table 15.

### Table 15. Axioms of APTC\textsuperscript{dat} (a, b, c ∈ A₅, n ≥ 0)

| No. | Axiom |
|-----|-------|
| P1  | x \parallel y = x \parallel y + x \parallel y |
| P2  | x \parallel y = y \parallel x |
| P3  | (x \parallel y) \parallel z = x \parallel (y \parallel z) |
| P4DA | a \parallel (b \parallel y) = (a \parallel b) \parallel y |
| P5DA | (a \parallel x) \parallel z = (a \parallel z) \parallel x |
| P6DA | (a \parallel x) \parallel (b \parallel y) = (a \parallel b) \parallel (x \parallel y) |
| P7  | (x \parallel y) \parallel z = (x \parallel z) \parallel (y \parallel z) |
| P8  | x \parallel (y \parallel z) = (x \parallel y) \parallel (x \parallel z) |
| DAP9ID | (v\textsubscript{1}ID(x) \parallel\textbf{Δ}) \parallel σ\textsubscript{abs}(y) = \textbf{Δ} |
| DAP10ID | σ\textsubscript{abs}(x) \parallel (v\textsubscript{1}ID(y) \parallel\textbf{Δ}) = \textbf{Δ} |
| DAP11 | σ\textsubscript{abs}(x) \parallel σ\textsubscript{abs}(y) = σ\textsubscript{abs}(x \parallel y) |
| P1D12 | \delta \parallel x = \delta |
| P1D13 | x \parallel \delta = \delta |
| C14DA | a \parallel \delta = \gamma(a, b) |
| C15DA | (a \parallel b) \parallel \delta = \gamma(a, b) \parallel \delta |
| C16DA | (a \parallel x) \parallel \delta = \gamma(a, b) \parallel x |
| C17DA | (a \parallel x) \parallel (b \parallel y) = \gamma(a, b) \parallel (x \parallel y) |
| C18 | (x \parallel y) \parallel z = (x \parallel z) \parallel (y \parallel z) |
| C19 | x \parallel (y \parallel z) = (x \parallel y) \parallel (x \parallel z) |
| DAC20ID | (v\textsubscript{1}ID(x) \parallel\textbf{Δ}) \parallel σ\textsubscript{abs}(y) = \textbf{Δ} |
| DAC21ID | σ\textsubscript{abs}(x) \parallel (v\textsubscript{1}ID(y) \parallel\textbf{Δ}) = \textbf{Δ} |
| DAC22 | σ\textsubscript{abs}(x) \parallel σ\textsubscript{abs}(y) = σ\textsubscript{abs}(x \parallel y) |
| C1D23 | \delta \parallel x = \delta |
| CE25DA | \Theta(y) = a |
| CE26DAID | \Theta(\delta) = \delta |
| CE27 | \Theta(x \parallel y) = \Theta(x) \parallel \delta + \Theta(y) |
| CE28 | \Theta(x \parallel y) = \Theta(x \parallel \delta + \Theta(y)) |
| CE29 | \Theta(x \parallel y) = \Theta((x \parallel y) \parallel \delta) + \Theta((\delta \parallel x) \parallel y) |
| CE30 | \Theta(x \parallel y) = \Theta((x \parallel y) \parallel \delta) + \Theta((\delta \parallel x) \parallel y) |
| U3IDAID | (\{a, b\} \parallel \delta) = a \parallel \delta |
| U32IDAID | (\{a, b\} \parallel \delta) = a \parallel \delta |
| U33IDAID | (\{b, a\} \parallel \delta) = a \parallel \delta |
| U34IDAID | (\{b, a\} \parallel \delta) = a \parallel \delta |
| U35IDAID | (\{b, a\} \parallel \delta) = a \parallel \delta |
| U36 | (x \parallel y) \parallel z = (x \parallel z) \parallel (y \parallel z) |
| U37 | (x \parallel y) \parallel z = (x \parallel z) \parallel (y \parallel z) |
| U38 | (x \parallel y) \parallel z = (x \parallel z) \parallel (y \parallel z) |
| U39 | (x \parallel y) \parallel z = (x \parallel z) \parallel (y \parallel z) |
| U40 | x \parallel (y \parallel z) = (x \parallel y) \parallel z |
| U41 | x \parallel (y \parallel z) = (x \parallel y) \parallel z |
| U42 | x \parallel (y \parallel z) = (x \parallel y) \parallel z |
| U43 | x \parallel (y \parallel z) = (x \parallel y) \parallel z |
| D1DAID | a \in H \parallel \Theta_H(y) = a |
| D2DAID | a \in H \parallel \Theta_H(x) = \delta |
| D3DAID | \Theta_H(\delta) = \delta |
| D4 | \Theta_H(x \parallel y) = \Theta_H(x) \parallel \Theta_H(y) |
| D5 | \Theta_H(x \parallel y) = \Theta_H(x) \parallel \Theta_H(y) |
| D6 | \Theta_H(x \parallel y) = \Theta_H(x) \parallel \Theta_H(y) |
| DADT | \Theta_H(\sigma_{\text{abs}}(x)) = \sigma_{\text{abs}}(\Theta_H(x)) |

4.3.2. Elimination

**Definition 4.16 (Basic terms of APTC\textsuperscript{dat}).** The set of basic terms of APTC\textsuperscript{dat}, \(B(\text{APTC}\textsuperscript{dat})\), is inductively defined as follows by two auxiliary sets \(B_0(\text{APTC}\textsuperscript{dat})\) and \(B_1(\text{APTC}\textsuperscript{dat})\):
Table 16. Transition rules of APTC\textsuperscript{dat} (a, b, c \in A, m > 0, n \geq 0)

1. if \( a \in A_b \), then \( a \in B_1(\text{APTC}_{\text{dat}}) \);
2. if \( a \in A \) and \( t \in B(\text{APTC}_{\text{dat}}) \), then \( a \cdot t \in B_1(\text{APTC}_{\text{dat}}) \);
3. if \( t, t' \in B_1(\text{APTC}_{\text{dat}}) \), then \( t + t' \in B_1(\text{APTC}_{\text{dat}}) \);
4. if \( t, t' \in B_1(\text{APTC}_{\text{dat}}) \), then \( t \parallel t' \in B_1(\text{APTC}_{\text{dat}}) \);
5. if \( t \in B_1(\text{APTC}_{\text{dat}}) \), then \( t \in B_0(\text{APTC}_{\text{dat}}) \);
6. if \( n > 0 \) and \( t \in B_0(\text{APTC}_{\text{dat}}) \), then \( \sigma^n_{\text{abs}}(t) \in B_0(\text{APTC}_{\text{dat}}) \);
7. if \( n > 0 \), \( t \in B_1(\text{APTC}_{\text{dat}}) \) and \( t' \in B_0(\text{APTC}_{\text{dat}}) \), then \( t + \sigma^n_{\text{abs}}(t') \in B_0(\text{APTC}_{\text{dat}}) \);
8. \( \delta \in B(\text{APTC}_{\text{dat}}) \);
9. if \( t \in B_0(\text{APTC}_{\text{dat}}) \), then \( t \in B(\text{APTC}_{\text{dat}}) \).

**Theorem 4.17** (Elimination theorem). Let \( p \) be a closed \( \text{APTC}_{\text{dat}} \) term. Then there is a basic \( \text{APTC}_{\text{dat}} \) term \( q \) such that \( \text{APTC}_{\text{dat}} \vdash p = q \).

**Proof.** It is sufficient to induct on the structure of the closed \( \text{APTC}_{\text{dat}} \) term \( p \). It can be proven that \( p \) combined by the constants and operators of \( \text{APTC}_{\text{dat}} \) exists an equal basic term \( q \), and the other operators not included in the basic terms, such as \( v_{\text{abs}}, \tau_{\text{abs}}, \|, |, \partial_H, \Theta \) and \( \triangleright \) can be eliminated. \( \square \)

### 4.3.3. Connections

**Theorem 4.18** (Generalization of \( \text{APTC}_{\text{dat}} \)).
1. By the definitions of \( a = a \) for each \( a \in A \) and \( \delta = \delta \), \( \text{APTC}_{\text{dat}} \) is a generalization of \( \text{APTC} \).
2. \( \text{APTC}_{\text{dat}} \) is a generalization of \( \text{BATC}_{\text{dat}} \)

**Proof.**
1. It follows from the following two facts.
   - The transition rules of \( \text{APTC} \) in section 2.1 are all source-dependent;
   - The sources of the transition rules of \( \text{APTC}_{\text{dat}} \) contain an occurrence of \( \delta, a, \sigma_{\text{abs}}^n, v_{\text{abs}}^n \) and \( \tau_{\text{abs}}^n \).

   So, \( \text{APTC} \) is an embedding of \( \text{APTC}_{\text{dat}} \), as desired.
2. It follows from the following two facts.
   - The transition rules of \( \text{BATC}_{\text{dat}} \) are all source-dependent;
   - The sources of the transition rules of \( \text{APTC}_{\text{dat}} \) contain an occurrence of \( \|, |, \Theta, \triangleright, \partial_H \).

   So, \( \text{BATC}_{\text{dat}} \) is an embedding of \( \text{APTC}_{\text{dat}} \), as desired. \( \square \)

### 4.3.4. Congruence

**Theorem 4.19** (Congruence of \( \text{APTC}_{\text{dat}} \)). Truly concurrent bisimulation equivalences \( \sim_p \), \( \sim_s \) and \( \sim_{hp} \) are all congruences with respect to \( \text{APTC}_{\text{dat}} \). That is,
- pomset bisimulation equivalence \( \sim_p \) is a congruence with respect to \( \text{APTC}_{\text{dat}} \);
- step bisimulation equivalence \( \sim_s \) is a congruence with respect to \( \text{APTC}_{\text{dat}} \);
- hp-bisimulation equivalence \( \sim_{hp} \) is a congruence with respect to \( \text{APTC}_{\text{dat}} \).

**Proof.** It is easy to see that \( \sim_p \), \( \sim_s \), and \( \sim_{hp} \) are all equivalent relations on \( \text{APTC}_{\text{dat}} \) terms, it is only sufficient to prove that \( \sim_p \), \( \sim_s \), and \( \sim_{hp} \) are all preserved by the operators \( \sigma_{\text{abs}}^n, v_{\text{abs}}^n \) and \( \tau_{\text{abs}}^n \). It is trivial and we omit it. \( \square \)

### 4.3.5. Soundness

**Theorem 4.20** (Soundness of \( \text{APTC}_{\text{dat}} \)). The axiomatization of \( \text{APTC}_{\text{dat}} \) is sound modulo truly concurrent bisimulation equivalences \( \sim_p \), \( \sim_s \), and \( \sim_{hp} \). That is,
1. let \( x \) and \( y \) be \( \text{APTC}_{\text{dat}} \) terms. If \( \text{APTC}_{\text{dat}} \vdash x = y \), then \( x \sim y \);
2. let \( x \) and \( y \) be \( \text{APTC}_{\text{dat}} \) terms. If \( \text{APTC}_{\text{dat}} \vdash x = y \), then \( x \sim_p y \);
3. let \( x \) and \( y \) be \( \text{APTC}_{\text{dat}} \) terms. If \( \text{APTC}_{\text{dat}} \vdash x = y \), then \( x \sim_{hp} y \).

**Proof.** Since \( \sim_p \), \( \sim_s \), and \( \sim_{hp} \) are both equivalent and congruent relations, we only need to check if each axiom in Table 15 is sound modulo \( \sim_p \), \( \sim_s \), and \( \sim_{hp} \) respectively.
1. We only check the soundness of the non-trivial axiom \(\text{DAP11 modulo } \sim_s\). Let \(p, q\) be APTC\textsuperscript{dat} processes, and \(\sigma^n_{\text{abs}}(p) \parallel \sigma^n_{\text{abs}}(q) = \sigma^n_{\text{abs}}(p \parallel q)\), it is sufficient to prove that \(\sigma^n_{\text{abs}}(p) \parallel \sigma^n_{\text{abs}}(q) \sim_s \sigma^n_{\text{abs}}(p \parallel q)\). By the transition rules of operator \(\sigma^n_{\text{abs}}\) and \(\parallel\) in Table \(\ref{tab:transition-rules}\) we get

\[
\begin{align*}
\langle p, 0 \rangle \uparrow & \quad \langle \sigma^n_{\text{abs}}(p) \parallel \sigma^n_{\text{abs}}(q), n' \rangle \rightarrow^n \langle \sigma^n_{\text{abs}}(p) \parallel \sigma^n_{\text{abs}}(q), n' + n \rangle \\
\langle p, 0 \rangle \uparrow & \quad \langle \sigma^n_{\text{abs}}(p \parallel q), n' \rangle \rightarrow^n \langle \sigma^n_{\text{abs}}(p \parallel q), n' + n \rangle
\end{align*}
\]

There are several cases:

\[
\begin{align*}
\langle p, n' \rangle \overset{a \in A}{\rightarrow} \langle \sqrt{,}, n' \rangle & \quad \langle q, n' \rangle \overset{b \in B}{\rightarrow} \langle \sqrt{,}, n' \rangle \\
\sigma^n_{\text{abs}}(p) \parallel \sigma^n_{\text{abs}}(q), n' + n \overset{\{a,b\}}{\rightarrow} \langle \sqrt{,}, n' + n \rangle \\
\langle p, n' \rangle \overset{a \in A}{\rightarrow} \langle \sqrt{,}, n' \rangle & \quad \langle q, n' \rangle \overset{b \in B}{\rightarrow} \langle \sqrt{,}, n' \rangle \\
\sigma^n_{\text{abs}}(p \parallel q), n' + n \overset{\{a,b\}}{\rightarrow} \langle \sqrt{,}, n' + n \rangle \\
\langle p, n' \rangle \overset{a \in A}{\rightarrow} \langle p', n' \rangle & \quad \langle q, n' \rangle \overset{b \in B}{\rightarrow} \langle \sqrt{,}, n' \rangle \\
\sigma^n_{\text{abs}}(p) \parallel \sigma^n_{\text{abs}}(p'), n' + n \overset{\{a,b\}}{\rightarrow} \langle \sigma^n_{\text{abs}}(p'), n' + n \rangle \\
\langle p, n' \rangle \overset{a \in A}{\rightarrow} \langle p', n' \rangle & \quad \langle q, n' \rangle \overset{b \in B}{\rightarrow} \langle q', n' \rangle \\
\sigma^n_{\text{abs}}(p \parallel q), n' + n \overset{\{a,b\}}{\rightarrow} \langle \sigma^n_{\text{abs}}(q'), n' + n \rangle \\
\langle p, n' \rangle \overset{a \in A}{\rightarrow} \langle \sqrt{,}, n' \rangle & \quad \langle q, n' \rangle \overset{b \in B}{\rightarrow} \langle q', n' \rangle \\
\sigma^n_{\text{abs}}(p \parallel q), n' + n \overset{\{a,b\}}{\rightarrow} \langle \sigma^n_{\text{abs}}(q'), n' + n \rangle \\
\langle p, n' \rangle \overset{a \in A}{\rightarrow} \langle p', n' \rangle & \quad \langle q, n' \rangle \overset{b \in B}{\rightarrow} \langle q', n' \rangle \\
\sigma^n_{\text{abs}}(p \parallel q), n' + n \overset{\{a,b\}}{\rightarrow} \langle \sigma^n_{\text{abs}}(q'), n' + n \rangle \\
\langle p, n' \rangle \overset{a \in A}{\rightarrow} \langle p', n' \rangle & \quad \langle q, n' \rangle \overset{b \in B}{\rightarrow} \langle q', n' \rangle \\
\sigma^n_{\text{abs}}(p \parallel q), n' + n \overset{\{a,b\}}{\rightarrow} \langle \sigma^n_{\text{abs}}(q'), n' + n \rangle \\
\langle p, n' \rangle \uparrow & \quad \langle \sigma^n_{\text{abs}}(p) \parallel \sigma^n_{\text{abs}}(q), n' + n \rangle \uparrow \\
\langle q, n' \rangle \uparrow & \quad \langle \sigma^n_{\text{abs}}(p \parallel q), n' + n \rangle \uparrow
\end{align*}
\]

So, we see that each case leads to \(\sigma^n_{\text{abs}}(p) \parallel \sigma^n_{\text{abs}}(q) \sim_s \sigma^n_{\text{abs}}(p \parallel q)\), as desired.
2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by ·) or in concurrency (implicitly defined by + and explicitly defined by ⊤), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of $P = \{a, b : a \cdot b\}$. Then the pomset transition labeled by the above $P$ is just composed of one single event transition labeled by $a$ succeeded by another single event transition labeled by $b$, that is, $\frac{P}{\Rightarrow} = \frac{a}{\Rightarrow} \cdot \frac{b}{\Rightarrow}$.

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 15 is sound modulo pomset bisimulation equivalence, we omit them.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product $(C_1, f, C_2), f : C_1 \rightarrow C_2$ isomorphism. Two process terms $s$ related to $C_1$ and $t$ related to $C_2$, and $f : C_1 \rightarrow C_2$ isomorphism. Initially, $(C_1, f, C_2) = (\varnothing, \varnothing, \varnothing)$, and $(\varnothing, \varnothing, \varnothing) \in \sim_{hp}$. When $s \xrightarrow{a} s'$ $(C_1 \xrightarrow{a} C_1')$, there will be $t \xrightarrow{a'} (C_2 \xrightarrow{a'} C_2')$, and we define $f' = f[a \rightarrow a]$. Then, if $(C_1, f, C_2) \in \sim_{hp}$, then $(C_1', f', C_2') \in \sim_{hp}$.

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 15 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

\[\square\]

4.3.6. Completeness

**Theorem 4.21** (Completeness of APTC$^{\text{dat}}$). The axiomatization of APTC$^{\text{dat}}$ is complete modulo truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$, and $\sim_{hp}$. That is,

1. let $p$ and $q$ be closed APTC$^{\text{dat}}$ terms, if $p \sim_s q$ then $p = q$;
2. let $p$ and $q$ be closed APTC$^{\text{dat}}$ terms, if $p \sim_p q$ then $p = q$;
3. let $p$ and $q$ be closed APTC$^{\text{dat}}$ terms, if $p \sim_{hp} q$ then $p = q$.

**Proof.** 1. Firstly, by the elimination theorem of APTC$^{\text{dat}}$, we know that for each closed APTC$^{\text{dat}}$ term $p$, there exists a closed basic APTC$^{\text{dat}}$ term $p'$, such that APTC$^{\text{dat}}$ ⊢ $p = p'$, so, we only need to consider closed basic APTC$^{\text{dat}}$ terms. The basic terms modulo associativity and commutativity (AC) of conflict + (defined by axioms A1 and A2 in Table 13) and associativity and commutativity (AC) of parallel || (defined by axioms P2 and P3 in Table 15), and these equivalences is denoted by $=_{AC}$. Then, each equivalence class $s$ modulo AC of $+$ and $\parallel$ has the following normal form

$$s_1 + \cdots + s_k$$

with each $s_i$ either an atomic event or of the form

$$t_1 \cdots t_m$$

with each $t_j$ either an atomic event or of the form

$$u_1 \parallel \cdots \parallel u_n$$

with each $u_i$ an atomic event, and each $s_i$ is called the summand of $s$.

Now, we prove that for normal forms $n$ and $n'$, if $n \sim_s n'$ then $n =_{AC} n'$. It is sufficient to induct on the sizes of $n$ and $n'$. We can get $n =_{AC} n'$.

Finally, let $s$ and $t$ be basic APTC$^{\text{dat}}$ terms, and $s \sim_s t$, there are normal forms $n$ and $n'$, such that $s = n$ and $t = n'$. The soundness theorem modulo step bisimulation equivalence yields $s \sim_s n$ and $t \sim_s n'$, so $n \sim_s n \sim_s n'$. Since if $n \sim_s n'$ then $n =_{AC} n'$, $s = n =_{AC} n' = t$, as desired.

2. This case can be proven similarly, just by replacement of $\sim_s$ by $\sim_p$.
3. This case can be proven similarly, just by replacement of $\sim_s$ by $\sim_{hp}$.

\[\square\]
4.4. Discrete Initial Abstraction

In this subsection, we will introduce APTC$^{\text{dat}}$ with discrete initial abstraction called APTC$^{\text{dat}/\sim}$.

4.4.1. Basic Definition

Definition 4.22 (Discrete initial abstraction). Discrete initial abstraction $\sqrt{d}$ is an abstraction mechanism to form functions from natural numbers to processes with absolute timing, that map each natural number $n$ to a process initialized at time $n$.

4.4.2. The Theory APTC$^{\text{dat}/\sim}$

Definition 4.23 (Signature of APTC$^{\text{dat}/\sim}$). The signature of APTC$^{\text{dat}/\sim}$ consists of the signature of APTC$^{\text{dat}}$ and the discrete initial abstraction operator $\sqrt{d} : \mathbb{N}.P^*_{abs} \to P^*_{abs}$. Where $P^*_{abs}$ is the sorts with discrete initial abstraction.

The set of axioms of APTC$^{\text{dat}/\sim}$ consists of the laws given in Table 17. Where $i, j, \ldots$ are variables of sort $\mathbb{N}$, $F, G, \ldots$ are variables of sort $\mathbb{N}.P^*_{abs}$, $K, L, \ldots$ are variables of sort $\mathbb{N}, \mathbb{N}.P^*_{abs}$, and we write $\sqrt{d}.i.t$ for $\sqrt{d}(i.t)$.

It is sufficient to extend bisimulations $CI/\sim$ of APTC$^{\text{dat}}$ to

\[(CI/\sim)^* = \{f : \mathbb{N} \to CI/\sim \mid \forall i \in \mathbb{N}. f(i) = \text{abs}(f(i))\}\]

and define the constants and operators of APTC$^{\text{dat}/\sim}$ on $(CI/\sim)^*$ as in Table 18 and the $* : CI/\sim \times (CI/\sim)^* \to CI/\sim$ is defined in Table 19.

4.4.3. Connections
It follows from the following two facts. By the definitions of constants and operators of ACT\textsuperscript{drt} in APTC\textsuperscript{dat}ψ in Table 20, a relatively timed process with discrete initialization abstraction of the time spectrum tail operator \( \mu : \mathcal{P}_{\text{abs}}^* \to \mathcal{P}_{\text{abs}}^* \) in Table 21, Table 22 and Table 23. APTC\textsuperscript{dat}ψ is a generalization of APTC\textsuperscript{drt}.

2. APTC\textsuperscript{dat}ψ is a generalization of BATC\textsuperscript{dat}.

Proof. 1. It follows from the following two facts. By the definitions of constants and operators of ACT\textsuperscript{drt} in APTC\textsuperscript{dat}ψ in Table 20, a relatively timed process with discrete initialization abstraction of the time spectrum tail operator \( \mu : \mathcal{P}_{\text{abs}}^* \to \mathcal{P}_{\text{abs}}^* \) in Table 21, Table 22 and Table 23.

(a) the transition rules of ACT\textsuperscript{drt} are all source-dependent;
(b) the sources of the transition rules of APTC\textsuperscript{dat}ψ contain an occurrence of \( \sqrt{d} \).

So, ACT\textsuperscript{drt} is an embedding of APTC\textsuperscript{dat}ψ, as desired.

2. It follows from the following two facts.

(a) The transition rules of APTC\textsuperscript{dat} are all source-dependent;
(b) The sources of the transition rules of APTC\textsuperscript{dat}ψ contain an occurrence of \( \sqrt{d} \).

So, APTC\textsuperscript{dat} is an embedding of APTC\textsuperscript{dat}ψ, as desired.

\[ 0.4.4. \text{ Congruence} \]

\textbf{Theorem 4.25} (Congruence of APTC\textsuperscript{dat}ψ). Truly concurrent bisimulation equivalences \( \sim_p, \sim_s \) and \( \sim_{hp} \) are all congruences with respect to APTC\textsuperscript{dat}ψ. That is,
Draft of Truly Concurrent Process Algebra with Timing

\[ DPTST0 \quad \mu(\delta) = \delta \]
\[ DPTST1 \quad \mu(\emptyset) = \delta \]
\[ DPTST2 \quad \mu(\sigma_{abs}^{n+1}(x)) = \sigma_{abs}^{n}(x) \]
\[ DPTST3 \quad \mu(x + y) = \mu(x)+\mu(y) \]
\[ DPTST4 \quad \mu(x \cdot y) = \mu(x) \cdot \mu(y) \]
\[ DPTST5 \quad \mu(x \parallel y) = \mu(x) \parallel \mu(y) \]
\[ DPTST6 \quad \mu(x) = \sqrt{\lambda k.\mu(\tau_{abs}(x))} \]

Table 21. Axioms of time spectrum tail \((a \in A_{\text{sm}} n \geq 0)\)

\[
\begin{align*}
(x,n+1) & \xrightarrow{a} (x',n+1) & (x,n+1) & \xrightarrow{\vee} (\sqrt{\cdot},n+1) \\
(\mu(x),n) & \xrightarrow{\Delta} (\mu(x'),n) & (\mu(x),n) & \xrightarrow{\vee} (\sqrt{\cdot},n) \\
(x,n+1) & \xrightarrow{m} (x,n+m+1) & (x,n+1) & \xrightarrow{\downarrow} (x,0) \xrightarrow{m} \\\n(\mu(x),n) & \xrightarrow{m} (\mu(x),n+m) & (\mu(x),n) & \xrightarrow{\downarrow} (\mu(x),n) \xrightarrow{\downarrow}
\end{align*}
\]

Table 22. Transition rules of time spectrum tail \((a \in A, m > 0, n \geq 0)\)

- pomset bisimulation equivalence \(\sim_p\) is a congruence with respect to \(\text{APTC}_{\text{dat}}^{\vee}\);
- step bisimulation equivalence \(\sim_s\) is a congruence with respect to \(\text{APTC}_{\text{dat}}^{\vee}\);
- \(hp\)-bisimulation equivalence \(\sim_{hp}\) is a congruence with respect to \(\text{APTC}_{\text{dat}}^{\vee}\).

**Proof.** It is easy to see that \(\sim_p\), \(\sim_s\), and \(\sim_{hp}\) are all equivalent relations on \(\text{APTC}_{\text{dat}}^{\vee}\) terms, it is only sufficient to prove that \(\sim_p\), \(\sim_s\), and \(\sim_{hp}\) are all preserved by the operators \(\sigma_{abs}^n\), \(\nu_{abs}^n\), and \(\nu_{abs}^k\). It is trivial and we omit it. \(\square\)

4.4.5. Soundness

**Theorem 4.26** (Soundness of \(\text{APTC}_{\text{dat}}^{\vee}\)). The axiomatization of \(\text{APTC}_{\text{dat}}^{\vee}\) is sound modulo truly concurrent bisimulation equivalences \(\sim_p\), \(\sim_s\), and \(\sim_{hp}\). That is,

1. let \(x\) and \(y\) be \(\text{APTC}_{\text{dat}}^{\vee}\) terms. If \(\text{APTC}_{\text{dat}}^{\vee} \vdash x = y\), then \(x \sim_s y\);
2. let \(x\) and \(y\) be \(\text{APTC}_{\text{dat}}^{\vee}\) terms. If \(\text{APTC}_{\text{dat}}^{\vee} \vdash x = y\), then \(x \sim_p y\);
3. let \(x\) and \(y\) be \(\text{APTC}_{\text{dat}}^{\vee}\) terms. If \(\text{APTC}_{\text{dat}}^{\vee} \vdash x = y\), then \(x \sim_{hp} y\).

**Proof.** Since \(\sim_p\), \(\sim_s\), and \(\sim_{hp}\) are both equivalent and congruent relations, we only need to check if each axiom in Table 17 is sound modulo \(\sim_p\), \(\sim_s\), and \(\sim_{hp}\) respectively.

1. Each axiom in Table 17 can be checked that it is sound modulo step bisimulation equivalence, by \(\lambda\)-definitions in Table 18, Table 19. We omit them.
2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \(\cdot\)) or in concurrency (implicitly defined by \(\cdot\) and \(+\), and explicitly defined by \(\parallel\)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \(P = \{a,b:a\times b\}\). Then the pomset transition labeled by the above \(P\) is just composed of one single event transition labeled by \(a\) succeeded by another single event transition labeled by \(b\) that is, \(P = \xrightarrow{a} b\).

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 17 is sound modulo pomset bisimulation equivalence, we omit them.
3. From the definition of \(hp\)-bisimulation, we know that \(hp\)-bisimulation is defined on the posetal product

\[ \mu(f) = \lambda k.\mu(f(k+1)) \]

Table 23. Definition of time spectrum tail on \((\mathcal{CI}/\sim)^*\)
| No. | Axiom |
|-----|-------|
| BOOL1 | ¬t = f |
| BOOL2 | ¬f = t |
| BOOL3 | ¬¬b = b |
| BOOL4 | t ∨ b = t |
| BOOL5 | f ∨ b = b |
| BOOL6 | b ∨ b' = b' ∨ b |
| BOOL7 | b ∧ b' = ¬(¬b ∨ ¬b') |

Table 24. Axioms of logical operators

(C₁, f, C₂), f : C₁ → C₂ isomorphism. Two process terms s related to C₁ and t related to C₂, and f : C₁ → C₂ isomorphism. Initially, (C₁, f, C₂) = (∅, ∅, ∅), and (∅, ∅, ∅) ∈∼hp. When s ↦ t (C₁ ↦ C₁'), there will be t ↦ t' (C₂ ↦ C₂'), and we define f' = f[a ↦ a]. Then, if (C₁, f, C₂) ∈∼hp, then (C₁', f', C₂') ∈∼hp. Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 17 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

4.5. Time-Dependent Conditions

In this subsection, we will introduce APTC\textsuperscript{dat}\sqrt{C} with time-dependent conditions called APTC\textsuperscript{dat}\sqrt{C}.

4.5.1. Basic Definition

**Definition 4.27** (Time-dependent conditions). The basic kinds of time-dependent conditions are in-time-slice and in-time-slice-greater-than. In-time-slice n (n ∈ \mathbb{N}) is the condition that holds only in time slice n and in-time-slice-greater-than n (n ∈ \mathbb{N}) is the condition that holds in all time slices greater than n. t is as the truth and f is as falsity.

4.5.2. The Theory APTC\textsuperscript{dat}\sqrt{C}

**Definition 4.28** (Signature of APTC\textsuperscript{dat}\sqrt{C}). The signature of APTC\textsuperscript{dat}\sqrt{C} consists of the signature of APTC\textsuperscript{dat}\sqrt{C}, and the in-time-slice operator sl : N → B\textsuperscript{*}, the in-time-slice-greater-than operator sl
\textsubscript{>} : N → B\textsuperscript{*}, the logical constants and operators t :→ B\textsuperscript{*}, f :→ B\textsuperscript{*}, ¬ : B\textsuperscript{*} → B\textsuperscript{*}, ∨ : B\textsuperscript{*} × B\textsuperscript{*} → B\textsuperscript{*}, ∧ : B\textsuperscript{*} × B\textsuperscript{*} → B\textsuperscript{*}, the absolute initialization operator υ\textsubscript{abs} : N × B\textsuperscript{*} → B\textsuperscript{*}, the discrete initial abstraction operator √\textsubscript{d} : N × B\textsuperscript{*} → B\textsuperscript{*}, and the conditional operator ::= : B\textsuperscript{*} × P\textsuperscript{*} \textsubscript{abs} → P\textsuperscript{*} \textsubscript{abs}. Where B\textsuperscript{*} is the sort of time-dependent conditions.

The set of axioms of APTC\textsuperscript{dat}\sqrt{C} consists of the laws given in Table 24, Table 25, and Table 26. Where b is a condition.

The operational semantics of APTC\textsuperscript{dat}\sqrt{C} are defined by the transition rules in Table 27 and Table 28.

4.5.3. Elimination

**Definition 4.29** (Basic terms of APTC\textsuperscript{dat}\sqrt{C}). The set of basic terms of APTC\textsuperscript{dat}\sqrt{C}, B(APTC\textsuperscript{dat}\sqrt{C}), is inductively defined as follows by two auxiliary sets B\textsubscript{0}(APTC\textsuperscript{dat}\sqrt{C}) and B\textsubscript{1}(APTC\textsuperscript{dat}\sqrt{C}):
Table 25. Axioms of conditions (m, n ≥ 0)

{\begin{array}{|c|c|}
\hline
No. & Axiom \\
\hline
SGC1 & t \triangleright x \\
SGC21D & f \triangleright x = \delta \\
DASGC1 & t_\text{abs}(b \triangleright x) = t_\text{abs}(b) \triangleright x + t_\text{abs}(x) + t_\text{abs}(\delta) \\
DASGC2 & x = \sum_{d \in \{0, 1\}} (s_i(k + 1) \triangleright t_\text{abs}(x)) + s_i(n + 1) \triangleright x \\
SGC31D & b \triangleright x = \delta \\
DASGC3 & b \triangleright x \triangleright y = (b \triangleright x) \triangleright y \\
SGC4 & b \triangleright (x \triangleright y) = (b \triangleright x) \triangleright (b \triangleright y) \\
DASGC4 & b \triangleright (x \triangleright y) = (b \triangleright x) \triangleright (b \triangleright y) \\
DASGC5 & (b \triangleright x) \triangleright (b \triangleright y) = (b \triangleright x) \triangleright (b \triangleright y) \\
DASGC6 & (b \triangleright x) \triangleright (b \triangleright y) = (b \triangleright x) \triangleright (b \triangleright y) \\
DASGC7 & b \triangleright \Theta(x) = \Theta(b \triangleright x) \\
DASGC8 & b \triangleright (x \triangleright y) = (b \triangleright x) \triangleright (b \triangleright y) \\
DASGC9 & b \triangleright 0_\text{H}(x) = 0_\text{H}(b \triangleright x) \\
DASGC10 & (\sqrt[\text{abs}]{F(i)}) = \sqrt[\text{abs}]{F(i)} \\
DASGC11 & (\sqrt[\text{abs}]{C(i)}) \triangleright x = \sqrt[\text{abs}]{C(i)} \triangleright t_\text{abs}(x) \\
\hline
\end{array}}

Table 26. Axioms of conditionals (n ≥ 0)

{\begin{array}{|c|c|}
\hline
\hline
\hline
\end{array}}

Table 27. Transition rules of APTC\text{d\text{at}}\sqrt{C(a \in A, m > 0, n ≥ 0)}

\begin{align*}
\begin{array}{|c|c|}
\hline
\hline
\hline
\end{array}}

Table 28. Definitions of conditional operator on (CJ/\sim)^*
1. if $a \in A$, then $a \in B_1(\text{APTC}^\text{dat} \sqrt{C})$
2. if $a \in A$ and $t \in B(\text{APTC}^\text{dat} \sqrt{C})$, then $a \cdot t \in B_1(\text{APTC}^\text{dat} \sqrt{C})$
3. if $t, t' \in B_1(\text{APTC}^\text{dat} \sqrt{C})$, then $t + t' \in B_1(\text{APTC}^\text{dat} \sqrt{C})$
4. if $t, t' \in B_1(\text{APTC}^\text{dat} \sqrt{C})$, then $t \parallel t' \in B_1(\text{APTC}^\text{dat} \sqrt{C})$
5. if $t \in B_1(\text{APTC}^\text{dat} \sqrt{C})$, then $t \in B_0(\text{APTC}^\text{dat} \sqrt{C})$
6. if $n > 0$ and $t \in B_0(\text{APTC}^\text{dat} \sqrt{C})$, then $\sigma^n_{\text{abs}}(t) \in B_0(\text{APTC}^\text{dat} \sqrt{C})$
7. if $n > 0$, $t \in B_1(\text{APTC}^\text{dat} \sqrt{C})$ and $t' \in B_0(\text{APTC}^\text{dat} \sqrt{C})$, then $t + \sigma^n_{\text{abs}}(t') \in B_0(\text{APTC}^\text{dat} \sqrt{C})$
8. if $n > 0$ and $t \in B_0(\text{APTC}^\text{dat} \sqrt{C})$, then $\sqrt{\Delta} t(n) \in B_0(\text{APTC}^\text{dat} \sqrt{C})$
9. $\delta \in B(\text{APTC}^\text{dat} \sqrt{C})$
10. if $t \in B_0(\text{APTC}^\text{dat} \sqrt{C})$, then $t \in B(\text{APTC}^\text{dat} \sqrt{C})$

**Theorem 4.30** (Elimination theorem). Let $p$ be a closed $\text{APTC}^\text{dat} \sqrt{C}$ term. Then there is a basic $\text{APTC}^\text{dat} \sqrt{C}$ term $q$ such that $\text{APTC}^\text{dat} \sqrt{C} \vdash p = q$.

**Proof.** It is sufficient to induct on the structure of the closed $\text{APTC}^\text{dat} \sqrt{C}$ term $p$. It can be proven that $p$ combined by the constants and operators of $\text{APTC}^\text{dat} \sqrt{C}$ exists an equal basic term $q$, and the other operators not included in the basic terms, such as $\nu_{\text{abs}}, \nu_{\text{abs}}, \star$, $\partial_H$, $\Theta$, $<$, and the constants and operators related to conditions can be eliminated.

4.5.4. Congruence

**Theorem 4.31** (Congruence of $\text{APTC}^\text{dat} \sqrt{C}$). Truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$ and $\sim_{hp}$ are all congruences with respect to $\text{APTC}^\text{dat} \sqrt{C}$. That is,

- pomset bisimulation equivalence $\sim_p$ is a congruence with respect to $\text{APTC}^\text{dat} \sqrt{C}$
- step bisimulation equivalence $\sim_s$ is a congruence with respect to $\text{APTC}^\text{dat} \sqrt{C}$
- $hp$-bisimulation equivalence $\sim_{hp}$ is a congruence with respect to $\text{APTC}^\text{dat} \sqrt{C}$

**Proof.** It is easy to see that $\sim_p$, $\sim_s$, and $\sim_{hp}$ are all equivalent relations on $\text{APTC}^\text{dat} \sqrt{C}$ terms, it is only sufficient to prove that $\sim_p$, $\sim_s$, and $\sim_{hp}$ are all preserved by the operators $\sigma^n_{\text{abs}}, \nu^n_{\text{abs}}$ and $\nu^n_{\text{abs}}$. It is trivial and we omit it.

4.5.5. Soundness

**Theorem 4.32** (Soundness of $\text{APTC}^\text{dat} \sqrt{C}$). The axiomatization of $\text{APTC}^\text{dat} \sqrt{C}$ is sound modulo truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$, and $\sim_{hp}$. That is,

1. let $x$ and $y$ be $\text{APTC}^\text{dat} \sqrt{C}$ terms. If $\text{APTC}^\text{dat} \sqrt{C} \vdash x = y$, then $x \sim_s y$
2. let $x$ and $y$ be $\text{APTC}^\text{dat} \sqrt{C}$ terms. If $\text{APTC}^\text{dat} \sqrt{C} \vdash x = y$, then $x \sim_p y$
3. let $x$ and $y$ be $\text{APTC}^\text{dat} \sqrt{C}$ terms. If $\text{APTC}^\text{dat} \sqrt{C} \vdash x = y$, then $x \sim_{hp} y$

**Proof.** Since $\sim_p$, $\sim_s$, and $\sim_{hp}$ are both equivalent and congruent relations, we only need to check if each axiom in Table 24, Table 25 and Table 26 is sound modulo $\sim_p$, $\sim_s$, and $\sim_{hp}$ respectively.

1. Each axiom in Table 24, Table 25 and Table 26 can be checked that it is sound modulo step bisimulation equivalence, by transition rules of conditionals in Table 27. We omit them.
2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by $\cdot$) or in concurrency (implicitly defined by $\star$, and explicitly defined by $\parallel$), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of $P = \{a, b : a \cdot b\}$. 

Then the pomset transition labeled by the above $P$ is just composed of one single event transition labeled by $a$ succeeded by another single event transition labeled by $b$, that is, $P \xrightarrow{a} P \xrightarrow{b}$.

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 17 is sound modulo pomset bisimulation equivalence, we omit them.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product $(C_1, f, C_2)\mapsto C_1 \rightarrow C_2$ isomorphism. Two process terms $s$ related to $C_1$ and $t$ related to $C_2$, and if $f : C_1 \rightarrow C_2$ isomorphism. Initially, $(C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)$, and $(\emptyset, \emptyset, \emptyset) \in \sim_{hp}$. When $s \xrightarrow{a} s' (C_1 \xrightarrow{a} C_1')$, there will be $t \xrightarrow{a} t'$ ($C_2 \xrightarrow{a} C_2'$), and we define $f' = f[a \mapsto a]$. Then, if $(C_1, f, C_2) \in \sim_{hp}$, then $(C_1', f', C_2') \in \sim_{hp}$.

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 17 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

\[ \square \]

4.5.6. Completeness

**Theorem 4.33 (Completeness of APTC\textsuperscript{dat} √C).** The axiomatization of APTC\textsuperscript{dat} √C is complete modulo truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$, and $\sim_{hp}$. That is,

1. let $p$ and $q$ be closed APTC\textsuperscript{dat} √C terms, if $p \sim_s q$ then $p = q$;
2. let $p$ and $q$ be closed APTC\textsuperscript{dat} √C terms, if $p \sim_p q$ then $p = q$;
3. let $p$ and $q$ be closed APTC\textsuperscript{dat} √C terms, if $p \sim_{hp} q$ then $p = q$.

**Proof.** 1. Firstly, by the elimination theorem of APTC\textsuperscript{dat} √C, we know that for each closed APTC\textsuperscript{dat} √C term $p$, there exists a closed basic APTC\textsuperscript{dat} √C term $p'$, such that APTC\textsuperscript{dat} √C $\vdash p = p'$, so, we only need to consider closed basic APTC\textsuperscript{dat} √C terms.

The basic terms modulo associativity and commutativity (AC) of conflict $+$ (defined by axioms A1 and A2 in Table 13) and associativity and commutativity (AC) of parallel $∥$ (defined by axioms P2 and P3 in Table 15), and these equivalences is denoted by $=_{AC}$. Then, each equivalence class $s$ modulo AC of $+$ and $∥$ has the following normal form

\[ s_1 + \cdots + s_k \]

with each $s_i$ either an atomic event or of the form

\[ t_1 \cdots t_m \]

with each $t_j$ either an atomic event or of the form

\[ u_1 \parallel \cdots \parallel u_n \]

with each $u_i$ an atomic event, and each $s_i$ is called the summand of $s$.

Now, we prove that for normal forms $n$ and $n'$, if $n \sim_s n'$ then $n =_{AC} n'$. It is sufficient to induct on the sizes of $n$ and $n'$. We can get $n =_{AC} n'$.

Finally, let $s$ and $t$ be basic APTC\textsuperscript{dat} √C terms, and $s \sim_s t$, there are normal forms $n$ and $n'$, such that $s = n$ and $t = n'$. The soundness theorem modulo step bisimulation equivalence yields $s \sim_s n$ and $t \sim_s n'$, so $n \sim_s s \sim_s t \sim_s n'$. Since if $n \sim_s n'$ then $n =_{AC} n'$, $s = n =_{AC} n' = t$, as desired.

2. This case can be proven similarly, just by replacement of $\sim_s$ by $\sim_p$.
3. This case can be proven similarly, just by replacement of $\sim_s$ by $\sim_{hp}$.

\[ \square \]

5. Continuous Relative Timing

In this section, we will introduce a version of APTC with relative timing and time measured on a continuous time scale. Measuring time on a continuous time scale means that timing is now done with respect to time
points on a continuous time scale. With respect to relative timing, timing is relative to the execution time of the previous action, and if the previous action does not exist, the start-up time of the whole process.

Like APTC without timing, let us start with a basic algebra for true concurrency called BATC$^\text{cert}$ (BATC with continuous relative timing). Then we continue with APTC$^\text{cert}$ (APTC with continuous relative timing).

5.1. Basic Definitions

In this subsection, we will introduce some basic definitions about timing. These basic concepts come from [25], we introduce them into the backgrounds of true concurrency.

**Definition 5.1** (Undelayable actions). Undelayable actions are defined as atomic processes that perform an action and then terminate successfully. We use a constant $\tilde{a}$ to represent the undelayable action, that is, the atomic process that performs the action $a$ and then terminates successfully.

**Definition 5.2** (Undelayable deadlock). Undelayable deadlock $\tilde{\delta}$ is an additional process that is neither capable of performing any action nor capable of idling beyond the current point of time.

**Definition 5.3** (Relative delay). The relative delay of the process $\tilde{p}$ for a period of time $\tilde{r}$ ($\tilde{r} \in \mathbb{R}^\delta$) is the process that idles for a period of time $\tilde{r}$ and then behaves like $\tilde{p}$. The operator $\sigma_{\text{rel}}$ is used to represent the relative delay, and let $\sigma_{\text{rel}}(t) = \sigma_{\text{rel}}$.

**Definition 5.4** (Deadlocked process). Deadlocked process $\tilde{\delta}$ is an additional process that has deadlocked before the current point of time. After a delay of a period of time, the undelayable deadlock $\tilde{\delta}$ and the deadlocked process $\tilde{\delta}$ are indistinguishable from each other.

**Definition 5.5** (Truly concurrent bisimulation equivalences with time-related capabilities). The following requirement with time-related capabilities is added to truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$, $\sim_{\text{hp}}$ and $\sim_{\text{hdp}}$:

- if a process is capable of first idling till a period of time and next going on as another process, then any equivalent process must be capable of first idling till the same period of time and next going on as a process equivalent to the other process;
- if a process has deadlocked before the current point of time, then any equivalent process must have deadlocked before the current point of time.

**Definition 5.6** (Integration). Let $f$ be a function from $\mathbb{R}^\delta$ to processes with continuous relative timing and $V \subseteq \mathbb{R}^\delta$. The integration $\int f$ of $f$ over $V$ is the process that behaves like one of the process in $\{f(r)\mid r \in V\}$.

**Definition 5.7** (Relative time-out). The relative time-out $\upsilon_{\text{rel}}$ of a process $\tilde{p}$ after a period of time $\tilde{r}$ ($\tilde{r} \in \mathbb{R}^\delta$) behaves either like the part of $\tilde{p}$ that does not idle till the $\text{pth}$-next time slice, or like the deadlocked process after a delay of $\tilde{r}$ time units if $\tilde{p}$ is capable of idling for the period of time $\tilde{r}$; otherwise, like $\tilde{p}$. And let $\upsilon_{\text{rel}}(t) = r\upsilon_{\text{rel}}$.

**Definition 5.8** (Relative initialization). The relative initialization $\overline{\upsilon}_{\text{rel}}$ of a process $\tilde{p}$ after a period of time $\tilde{r}$ ($\tilde{r} \in \mathbb{R}^\delta$) behaves like the part of $\tilde{p}$ that idles for a period of time $\tilde{r}$ if $\tilde{p}$ is capable of idling for $\tilde{r}$; otherwise, like the deadlocked process after a delay of $\tilde{r}$. And we let $\overline{\upsilon}_{\text{rel}}(t) = r\overline{\upsilon}_{\text{rel}}$.

5.2. Basic Algebra for True Concurrency with Continuous Relative Timing

In this subsection, we will introduce the theory BATC$^\text{cert}$.

5.2.1. The Theory BATC$^\text{cert}$

**Definition 5.9** (Signature of BATC$^\text{cert}$). The signature of BATC$^\text{cert}$ consists of the sort $\mathcal{P}_{\text{rel}}$ of processes with continuous relative timing, the undelayable action constants $\tilde{a} : \mathcal{P}_{\text{rel}}$ for each $a \in A$, the undelayable deadlock constant $\tilde{\delta} : \mathcal{P}_{\text{rel}}$, the alternative composition operator $+ : \mathcal{P}_{\text{rel}} \times \mathcal{P}_{\text{rel}} \rightarrow \mathcal{P}_{\text{rel}}$, the sequential composition operator $\cdot : \mathcal{P}_{\text{rel}} \rightarrow \mathcal{P}_{\text{rel}}$.
The set of axioms of BATC\textsuperscript{crt} consists of the laws given in Table 29.

The operational semantics of BATC\textsuperscript{crt} are defined by the transition rules in Table 30. Where \(\uparrow\) is a unary deadlocked predicate, and \(t \uparrow \overline{\uparrow} \neg(t \uparrow)\); \(t \mapsto t'\) means that process \(t\) is capable of first idling for \(q\), and then proceeding as process \(t'\).

### 5.2.2. Elimination

**Definition 5.10** (Basic terms of BATC\textsuperscript{crt}). The set of basic terms of BATC\textsuperscript{crt}, \(\mathcal{B}(\text{BATC}\textsuperscript{crt})\), is inductively defined as follows by two auxiliary sets \(\mathcal{B}_0(\text{BATC}\textsuperscript{crt})\) and \(\mathcal{B}_1(\text{BATC}\textsuperscript{crt})\):

1. if \(a \in A\), then \(\tilde{a} \in \mathcal{B}_1(\text{BATC}\textsuperscript{crt})\);
2. if \(a \in A\) and \(t \in \mathcal{B}(\text{BATC}\textsuperscript{crt})\), then \(\tilde{a} \cdot t \in \mathcal{B}_1(\text{BATC}\textsuperscript{crt})\);
3. if \(t, t' \in \mathcal{B}_1(\text{BATC}\textsuperscript{crt})\), then \(t + t' \in \mathcal{B}_1(\text{BATC}\textsuperscript{crt})\);
4. if \(t \in \mathcal{B}_1(\text{BATC}\textsuperscript{crt})\), then \(t \in \mathcal{B}_0(\text{BATC}\textsuperscript{crt})\);
5. if \(p > 0\) and \(t \in \mathcal{B}_0(\text{BATC}\textsuperscript{crt})\), then \(\sigma_{\text{rel}}^p(t) \in \mathcal{B}_0(\text{BATC}\textsuperscript{crt})\);
6. if \(p > 0\), \(t \in \mathcal{B}_1(\text{BATC}\textsuperscript{crt})\) and \(t' \in \mathcal{B}_0(\text{BATC}\textsuperscript{crt})\), then \(t + \sigma_{\text{rel}}^p(t') \in \mathcal{B}_0(\text{BATC}\textsuperscript{crt})\);
7. \(\tilde{\delta} \in \mathcal{B}(\text{BATC}\textsuperscript{crt})\);
8. if \(t \in \mathcal{B}_0(\text{BATC}\textsuperscript{crt})\), then \(t \in \mathcal{B}(\text{BATC}\textsuperscript{crt})\).
Proof. Combined by the constants and operators of BATC srt with respect to BATC srt, Theorem 5.12 is a generalization of BATC srt.

5.2.4. Congruence

Table 30. Transition rules of BATC srt (a ∈ A, p ≥ 0, r, s > 0)

Theorem 5.11 (Elimination theorem). Let p be a closed BATC srt term. Then there is a basic BATC srt term q such that BATC srt ⊢ p = q.

Proof. It is sufficient to induct on the structure of the closed BATC srt term p. It can be proven that p combined by the constants and operators of BATC srt exists an equal basic term q, and the other operators not included in the basic terms, such as \( v_{\text{rel}} \) and \( \overline{v}_{\text{rel}} \) can be eliminated. □

5.2.3. Connections

Theorem 5.12 (Generalization of BATC srt). By the definitions of \( a = \tilde{a} \) for each \( a \in A \) and \( \delta = \tilde{\delta} \), BATC srt is a generalization of BATC.

Proof. It follows from the following two facts.

1. The transition rules of BATC in section 2.1 are all source-dependent;
2. The sources of the transition rules of BATC srt contain an occurrence of \( \tilde{\delta}, \tilde{a}, \sigma_{\text{rel}}^p, v_{\text{rel}}^p \) and \( \overline{v}_{\text{rel}}^p \).

So, BATC is an embedding of BATC srt, as desired. □

5.2.4. Congruence

Theorem 5.13 (Congruence of BATC srt). Truly concurrent bisimulation equivalences are all congruences with respect to BATC srt. That is,

- pomset bisimulation equivalence \( \sim_p \) is a congruence with respect to BATC srt;
- step bisimulation equivalence \( \sim_s \) is a congruence with respect to BATC srt;
- hp-bisimulation equivalence \( \sim_{hp} \) is a congruence with respect to BATC srt;
- hhp-bisimulation equivalence \( \sim_{hhp} \) is a congruence with respect to BATC srt.
Proof. It is easy to see that \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhp} \) are all equivalent relations on \( \text{BATC}_{\text{arr}} \) terms, it is only sufficient to prove that \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhp} \) are all preserved by the operators \( \sigma_{\text{rel}}^p, v_{\text{rel}}^p \) and \( \overline{p}_{\text{rel}} \). It is trivial and we omit it.

5.2.5. Soundness

Theorem 5.14 (Soundness of \( \text{BATC}_{\text{arr}} \)). The axiomatization of \( \text{BATC}_{\text{arr}} \) is sound modulo truly concurrent bisimulation equivalences \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhp} \). That is,

1. let \( x \) and \( y \) be \( \text{BATC}_{\text{arr}} \) terms. If \( \text{BATC}_{\text{arr}} \vdash x = y \), then \( x \sim_s y \);
2. let \( x \) and \( y \) be \( \text{BATC}_{\text{arr}} \) terms. If \( \text{BATC}_{\text{arr}} \vdash x = y \), then \( x \sim_p y \);
3. let \( x \) and \( y \) be \( \text{BATC}_{\text{arr}} \) terms. If \( \text{BATC}_{\text{arr}} \vdash x = y \), then \( x \sim_{hp} y \);
4. let \( x \) and \( y \) be \( \text{BATC}_{\text{arr}} \) terms. If \( \text{BATC}_{\text{arr}} \vdash x = y \), then \( x \sim_{hhp} y \).

Proof. Since \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhp} \) are both equivalent and congruent relations, we only need to check if each axiom in Table 29 is sound modulo \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhp} \) respectively.

1. We only check the soundness of the non-trivial axiom \( \text{SRTO3} \) modulo \( \sim_s \). Let \( p \) be \( \text{BATC}_{\text{arr}} \) processes, and 
\[
\nu_{\text{rel}}^s(\sigma_{\text{rel}}^s(p)) = \sigma_{\text{rel}}^s(\nu_{\text{rel}}^s(p)),
\]
it is sufficient to prove that 
\[
\nu_{\text{rel}}^s(\sigma_{\text{rel}}^s(p)) \sim_s \sigma_{\text{rel}}^s(\nu_{\text{rel}}^s(p)).
\]
By the transition rules of operator \( \sigma_{\text{rel}}^s \) and \( v_{\text{rel}}^s \) in Table 30, we get

\[
\frac{\sigma_{\text{rel}}^s(\nu_{\text{rel}}^s(p)) \rightarrow^* \sigma_{\text{rel}}^s(\nu_{\text{rel}}^s(p))}{\sigma_{\text{rel}}^s(\nu_{\text{rel}}^s(p)) \rightarrow^* \sigma_{\text{rel}}^s(\nu_{\text{rel}}^s(p))}
\]

There are several cases:

\[
\frac{p \rightarrow \sqrt{ } \quad \nu_{\text{rel}}^s(\sigma_{\text{rel}}^s(p)) \rightarrow \sqrt{ }}{p \rightarrow \sqrt{ } \quad \sigma_{\text{rel}}^s(\nu_{\text{rel}}^s(p)) \rightarrow \sqrt{ } \quad \nu_{\text{rel}}^s(\sigma_{\text{rel}}^s(p)) \rightarrow p' \quad \sigma_{\text{rel}}^s(\nu_{\text{rel}}^s(p)) \rightarrow p' \quad p \rightarrow p'}
\]

So, we see that each case leads to 
\[
\nu_{\text{rel}}^s(\sigma_{\text{rel}}^s(p)) \sim_s \sigma_{\text{rel}}^s(\nu_{\text{rel}}^s(p)),
\]
as desired.

2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \( \cdot \)) or in concurrency (implicitly defined by \( \cdot \) and \( \ast \), and explicitly defined by \( \perp \)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \( P = \{\tilde{a}, \tilde{b} : \tilde{a} \cdot \tilde{b}\} \).
Then the pomset transition labeled by the above \( P \) is just composed of one single event transition labeled by \( \tilde{a} \) succeeded by another single event transition labeled by \( \tilde{b} \), that is, \( \frac{P}{\tilde{a} \to b} \).

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 29 is sound modulo pomset bisimulation equivalence, we omit them.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product \((C_1, f, C_2), f : C_1 \to C_2 \) isomorphism. Two process terms \( s \) related to \( C_1 \) and \( t \) related to \( C_2 \), and \( f : C_1 \to C_2 \) isomorphism. Initially, \((C_1, f, C_2) = (\emptyset, \emptyset, \emptyset) \) and \((\emptyset, \emptyset, \emptyset) \) are \( \sim_{hp} \). When \( s \sim s' (C_1 \sim C_1') \), there will be \( t \sim t' (C_2 \sim C_2') \), and we define \( f = f[a \to a] \). Then, if \((C_1, f, C_2) \sim_{hp}, (C_1', f', C_2') \sim_{hp} \).

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 29 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

4. We just need to add downward-closed condition to the soundness modulo hp-bisimulation equivalence, we omit them.

\[ \square \]

5.2.6. Completeness

**Theorem 5.15** (Completeness of BATC\(^srt\)). The axiomatization of BATC\(^srt\) is complete modulo truly concurrent bisimulation equivalences \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhp} \). That is,

1. let \( p \) and \( q \) be closed BATC\(^srt\) terms, if \( p \sim_s q \) then \( p = q \);
2. let \( p \) and \( q \) be closed BATC\(^srt\) terms, if \( p \sim_p q \) then \( p = q \);
3. let \( p \) and \( q \) be closed BATC\(^srt\) terms, if \( p \sim_{hp} q \) then \( p = q \);
4. let \( p \) and \( q \) be closed BATC\(^srt\) terms, if \( p \sim_{hhp} q \) then \( p = q \).

**Proof.** 1. Firstly, by the elimination theorem of BATC\(^srt\), we know that for each closed BATC\(^srt\) term \( p \), there exists a closed basic BATC\(^srt\) term \( p' \), such that BATC\(^srt\) \( \vdash p = p' \), so, we only need to consider closed basic BATC\(^srt\) terms.

The basic terms modulo associativity and commutativity (AC) of conflict + (defined by axioms A1 and A2 in Table 29), and this equivalence is denoted by \( =_{AC} \). Then, each equivalence class \( s \) modulo AC of + has the following normal form

\[ s_1 + \cdots + s_k \]

with each \( s_i \) either an atomic event or of the form \( t_1 \cdot t_2 \), and each \( s_i \) is called the summand of \( s \).

Now, we prove that for normal forms \( n \) and \( n' \), if \( n \sim_s n' \) then \( n =_{AC} n' \). It is sufficient to induct on the sizes of \( n \) and \( n' \). We can get \( n =_{AC} n' \).

Finally, let \( s \) and \( t \) be basic terms, and \( s \sim_t t \), there are normal forms \( n \) and \( n' \), such that \( s = n \) and \( t = n' \). The soundness theorem of BATC\(^srt\) modulo step bisimulation equivalence yields \( s \sim_s n \) and \( t \sim_s n' \), so \( n \sim_s n' \) since \( n \sim_s n' \) then \( n =_{AC} n' \), \( s = n =_{AC} n' = t \), as desired.

2. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_p \).
3. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_{hp} \).
4. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_{hhp} \).

\[ \square \]

5.3. BATC\(^srt\) with Integration

In this subsection, we will introduce the theory BATC\(^srt\) with integration called BATC\(^srt\)I.

5.3.1. The Theory BATC\(^srt\)I

**Definition 5.16** (Signature of BATC\(^srt\)I). The signature of BATC\(^srt\)I consists of the signature of BATC\(^srt\) and the integration operator \( f : \mathcal{P}(\mathbb{R}^2) \times \mathbb{R}^2 \rightarrow \mathcal{P}_{rel} \).
Theorem 5.18

The set of axioms of $\text{BATC}^{\text{cert}I}$ consists of the laws given in Table 31.

The operational semantics of $\text{BATC}^{\text{cert}I}$ are defined by the transition rules in Table 32.

5.3.2. Elimination

Definition 5.17 (Basic terms of $\text{BATC}^{\text{cert}I}$). The set of basic terms of $\text{BATC}^{\text{cert}I}$, $B(\text{BATC}^{\text{cert}I})$, is inductively defined as follows by two auxiliary sets $B_0(\text{BATC}^{\text{cert}I})$ and $B_1(\text{BATC}^{\text{cert}I})$:

1. if $a \in A_{\delta}$, then $\tilde{a} \in B_1(\text{BATC}^{\text{cert}I})$;
2. if $a \in A$ and $t \in B(\text{BATC}^{\text{cert}I})$, then $\tilde{a} \cdot t \in B_1(\text{BATC}^{\text{cert}I})$;
3. if $t, t' \in B(\text{BATC}^{\text{cert}I})$, then $t + t' \in B_1(\text{BATC}^{\text{cert}I})$;
4. if $t \in B_1(\text{BATC}^{\text{cert}I})$, then $t \in B_0(\text{BATC}^{\text{cert}I})$;
5. if $p > 0$ and $t \in B_0(\text{BATC}^{\text{cert}I})$, then $\sigma_{\text{rel}}^p(t) \in B_0(\text{BATC}^{\text{cert}I})$;
6. if $p > 0$, $t \in B_1(\text{BATC}^{\text{cert}I})$ and $t' \in B_0(\text{BATC}^{\text{cert}I})$, then $t + \sigma_{\text{rel}}^p(t') \in B_0(\text{BATC}^{\text{cert}I})$;
7. if $t \in B_0(\text{BATC}^{\text{cert}I})$, then $\int_{v \in V}(t) \in B_0(\text{BATC}^{\text{cert}I})$;
8. if $\delta \in B(\text{BATC}^{\text{cert}I})$;
9. if $t \in B_0(\text{BATC}^{\text{cert}I})$, then $t \in B(\text{BATC}^{\text{cert}I})$.

Theorem 5.18 (Elimination theorem). Let $p$ be a closed $\text{BATC}^{\text{cert}I}$ term. Then there is a basic $\text{BATC}^{\text{cert}I}$ term $q$ such that $\text{BATC}^{\text{cert}I} \vdash p = q$. 

---

Table 31. Axioms of $\text{BATC}^{\text{cert}I}(p \geq 0)$

| No. | Axiom |
|-----|-------|
| INT1 | $\int_{v \in V} F(v) = \int_{v \in V} F(v)$ |
| INT2 | $\int_{v \in V} F(v) = \delta$ |
| INT3 | $\int_{v \in V} F(v) = F(p)$ |
| INT4 | $\int_{v \in V \cup W} F(v) = \int_{v \in V} F(v) + \int_{v \in W} F(v)$ |
| INT5 | $\forall v \in V. F(v) = G(v) \Rightarrow \int_{v \in V} F(v) = \int_{v \in V} G(v)$ |
| INT6 | $\int_{v \in V} F(v) = G(v) \Rightarrow \int_{v \in V} F(v) = \int_{v \in V} G(v)$ |

Table 32. Transition rules of $\text{BATC}^{\text{cert}I}(a \in A, p, q \geq 0, r > 0)$

\[
\begin{align*}
\frac{F(q) \xrightarrow{a} x'}{\int_{v \in V} F(v) \xrightarrow{a} x'} (q \in V) \\
\frac{F(q) \xrightarrow{a} }{\int_{v \in V} F(v) \xrightarrow{\sqrt{}}} (q \in V) \\
\frac{F(q) \xrightarrow{\tau} F_{1}(q) \mid q \in V_1, \ldots, F(q) \xrightarrow{\tau} F_{n}(q) \mid q \in V_n, \{F(q) \xrightarrow{\tau} \mid q \in V_{n+1}\}}{\int_{v \in V} \xrightarrow{\tau} \int_{v \in V_1} F_{1}(v) + \cdots + \int_{v \in V_n} F_{n}(v)} (\{V_1, \ldots, V_n\} \text{ partition of } V \setminus V_{n+1}, V_{n+1} \subset V) \\
\frac{F(q) \xrightarrow{\tau} \top \mid q \in V}{\int_{v \in V} F(v) \xrightarrow{\top}} \end{align*}
\]
Proof. It is sufficient to induct on the structure of the closed BATC$^\text{cert}$I term $p$. It can be proven that $p$ combined by the constants and operators of BATC$^\text{cert}$I exists an equal basic term $q$, and the other operators not included in the basic terms, such as $\nu_{\text{rel}}$ and $\tau_{\text{rel}}$ can be eliminated.

5.3.3. Connections

Theorem 5.19 (Generalization of BATC$^\text{cert}$I). 1. By the definitions of $a = \int_{v \in [0, \infty)} \sigma_v(\tilde{a})$ for each $a \in A$ and $\delta = \int_{v \in [0, \infty)} \sigma_v(\tilde{\delta})$, BATC$^\text{cert}$I is a generalization of BATC.

2. BATC$^\text{cert}$I is a generalization of BATC$^\text{cert}$.

Proof. 1. It follows from the following two facts.

(a) The transition rules of BATC in section 2.1 are all source-dependent;
(b) The sources of the transition rules of BATC$^\text{cert}$I contain an occurrence of $\tilde{\delta}$, $\tilde{a}$, $\sigma^p$, $\nu^p$, $\tau^p$ and $\int$.

So, BATC is an embedding of BATC$^\text{cert}$I, as desired.

2. It follows from the following two facts.

(a) The transition rules of BATC$^\text{cert}$ are all source-dependent;
(b) The sources of the transition rules of BATC$^\text{cert}$I contain an occurrence of $\int$.

So, BATC$^\text{cert}$ is an embedding of BATC$^\text{cert}$I, as desired.

5.3.4. Congruence

Theorem 5.20 (Congruence of BATC$^\text{cert}$I). Truly concurrent bisimulation equivalences are all congruences with respect to BATC$^\text{cert}$I. That is,

- pomset bisimulation equivalence $\sim_p$ is a congruence with respect to BATC$^\text{cert}$I;
- step bisimulation equivalence $\sim_s$ is a congruence with respect to BATC$^\text{cert}$I;
- $\nu$-bisimulation equivalence $\sim_{\nu}$ is a congruence with respect to BATC$^\text{cert}$I;
- $\nu$-bisimulation equivalence $\sim_{\nu}$ is a congruence with respect to BATC$^\text{cert}$I.

Proof. It is easy to see that $\sim_p$, $\sim_s$, $\sim_{\nu}$ and $\sim_{\nu}$ are all equivalent relations on BATC$^\text{cert}$I terms, it is only sufficient to prove that $\sim_p$, $\sim_s$, $\sim_{\nu}$ and $\sim_{\nu}$ are all preserved by the operators $\int$. It is trivial and we omit it.

5.3.5. Soundness

Theorem 5.21 (Soundness of BATC$^\text{cert}$I). The axiomatization of BATC$^\text{cert}$I is sound modulo truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$, $\sim_{\nu}$ and $\sim_{\nu}$. That is,

1. let $x$ and $y$ be BATC$^\text{cert}$I terms. If BATC$^\text{cert}$I $\vdash x = y$, then $x \sim_p y$;
2. let $x$ and $y$ be BATC$^\text{cert}$I terms. If BATC$^\text{cert}$I $\vdash x = y$, then $x \sim_s y$;
3. let $x$ and $y$ be BATC$^\text{cert}$I terms. If BATC$^\text{cert}$I $\vdash x = y$, then $x \sim_{\nu} y$;
4. let $x$ and $y$ be BATC$^\text{cert}$I terms. If BATC$^\text{cert}$I $\vdash x = y$, then $x \sim_{\nu} y$.

Proof. Since $\sim_p$, $\sim_s$, $\sim_{\nu}$ and $\sim_{\nu}$ are both equivalent and congruent relations, we only need to check if each axiom in Table 31 is sound modulo $\sim_p$, $\sim_s$, $\sim_{\nu}$ and $\sim_{\nu}$ respectively.

1. We can check the soundness of each axiom in Table 31 by the transition rules in Table 32 it is trivial and we omit them.
2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by $\cdot$) or in concurrency (implicitly defined by $\cdot$ and $+$, and
Proof. 1. Firstly, by the elimination theorem of BATC
\sim concurrent bisimulation equivalences. Then the pomset transition labeled by the above \( P \) is just composed of one single event transition labeled by \( \tilde{a} \) succeeded by another single event transition labeled by \( \tilde{b} \), that is, \( \tilde{a} \rightarrow \tilde{b} \rightarrow \).

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 31 is sound modulo step bisimulation equivalence, we omit them.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product \((C_1, f, C_2), f : C_1 \rightarrow C_2\) isomorphism. Two process terms \( s \) related to \( C_1 \) and \( t \) related to \( C_2 \), and \( f : C_1 \rightarrow C_2\) isomorphism. Initially, \((C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)\), and \((\emptyset, \emptyset, \emptyset) \in \sim_{hp}\). When \( s \xrightarrow{a} s' (C_1 \xrightarrow{a} C_1') \), there will be \( t \xrightarrow{a} t' (C_2 \xrightarrow{a} C_2') \), and we define \( f' = f[a \rightarrow a] \). Then, if \((C_1, f, C_2) \in \sim_{hp}\), then \((C_1', f', C_2') \in \sim_{hp}\).

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 31 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

4. We just need to add downward-closed condition to the soundness modulo hp-bisimulation equivalence, we omit them.

\[\square\]

5.3.6. Completeness

**Theorem 5.22** (Completeness of BATC\textsubscript{cert}). The axiomatization of BATC\textsubscript{cert} is complete modulo truly concurrent bisimulation equivalences \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hp} \). That is,

1. let \( p \) and \( q \) be closed BATC\textsubscript{cert} terms, if \( p \sim_s q \) then \( p = q \);
2. let \( p \) and \( q \) be closed BATC\textsubscript{cert} terms, if \( p \sim_p q \) then \( p = q \);
3. let \( p \) and \( q \) be closed BATC\textsubscript{cert} terms, if \( p \sim_{hp} q \) then \( p = q \);
4. let \( p \) and \( q \) be closed BATC\textsubscript{cert} terms, if \( p \sim_{hp} q \) then \( p = q \).

**Proof.** 1. Firstly, by the elimination theorem of BATC\textsubscript{cert}, we know that for each closed BATC\textsubscript{cert} term \( p \), there exists a closed basic BATC\textsubscript{cert} term \( p' \), such that BATC\textsubscript{cert} \( \vdash p \sim p' \), so, we only need to consider closed basic BATC\textsubscript{cert} terms.

The basic terms modulo associativity and commutativity (AC) of conflict + (defined by axioms A1 and A2 in Table 29), and this equivalence is denoted by \( =_{AC} \). Then, each equivalence class \( s \) modulo AC of + has the following normal form

\[ s_1 + \cdots + s_k \]

with each \( s_i \) either an atomic event or of the form \( t_1 \cdot t_2 \), and each \( s_i \) is called the summand of \( s \).

Now, we prove that for normal forms \( n \) and \( n' \), if \( n \sim_s n' \) then \( n =_{AC} n' \). It is sufficient to induct on the sizes of \( n \) and \( n' \). We can get \( n =_{AC} n' \).

Finally, let \( s \) and \( t \) be basic terms, and \( s \sim_s t \), there are normal forms \( n \) and \( n' \), such that \( s = n \) and \( t = n' \).

The soundness theorem of BATC\textsubscript{cert} modulo step bisimulation equivalence yields \( s \sim_s n \) and \( t \sim_s n' \), so \( n \sim_s t \sim_s n' \). Since if \( n \sim_s n' \) then \( n =_{AC} n' \), \( s = n =_{AC} n' = t \), as desired.

2. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_p \).
3. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_{hp} \).
4. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_{hp} \).

\[\square\]

5.4. Algebra for Parallelism in True Concurrency with Continuous Relative Timing

In this subsection, we will introduce APTC\textsubscript{cert}. 

5.4.1. Basic Definition

**Definition 5.23** (Relative undelayable time-out). The relative undelayable time-out $\nu_{rel}$ of a process $p$ behaves like the part of $p$ that starts to perform actions at the current point of time if $p$ is capable of performing actions at the current point of time; otherwise, like undelayable deadlock. And let $\nu^r_{rel}(t) = \nu_{rel}$.

5.4.2. The Theory APTC$^\text{cert}$

**Definition 5.24** (Signature of APTC$^\text{cert}$). The signature of APTC$^\text{cert}$ consists of the signature of BATC$^\text{cert}$, and the whole parallel composition operator $\parallel: P_{rel} \times P_{rel} \to P_{rel}$, the parallel operator $\|: P_{rel} \times P_{rel} \to P_{rel}$, the communication merger operator $\vdash: P_{rel} \times P_{rel} \to P_{rel}$, the encapsulation operator $\partial_H: P_{rel} \to P_{rel}$ for all $H \subseteq A$, and the relative undelayable time-out operator $\nu_{rel}: P_{rel} \to P_{rel}$.

The set of axioms of APTC$^\text{cert}$ consists of the laws given in Table 33.

The operational semantics of APTC$^\text{cert}$ are defined by the transition rules in Table 34.

5.4.3. Elimination

**Definition 5.25** (Basic terms of APTC$^\text{cert}$). The set of basic terms of APTC$^\text{cert}$, $B(\text{APTC}^\text{cert})$, is inductively defined as follows by two auxiliary sets $B_0(\text{APTC}^\text{cert})$ and $B_1(\text{APTC}^\text{cert})$:

1. if $a \in A_\delta$, then $\tilde{a} \in B_1(\text{APTC}^\text{cert})$;
2. if $a \in A$ and $t \in B(\text{APTC}^\text{cert})$, then $\tilde{a} \cdot t \in B_1(\text{APTC}^\text{cert})$;
3. if $t, t' \in B_1(\text{APTC}^\text{cert})$, then $t + t' \in B_1(\text{APTC}^\text{cert})$;
4. if $t, t' \in B_1(\text{APTC}^\text{cert})$, then $t \parallel t' \in B_1(\text{APTC}^\text{cert})$;
5. if $t \in B_1(\text{APTC}^\text{cert})$, then $t \in B_0(\text{APTC}^\text{cert})$;
6. if $p > 0$ and $t \in B_0(\text{APTC}^\text{cert})$, then $\sigma_{rel}^p(t) \in B_0(\text{APTC}^\text{cert})$;
7. if $p > 0$, $t \in B_1(\text{APTC}^\text{cert})$ and $t' \in B_0(\text{APTC}^\text{cert})$, then $t + \sigma_{rel}^p(t') \in B_0(\text{APTC}^\text{cert})$;
8. if $t \in B_0(\text{APTC}^\text{cert})$, then $\nu_{rel}(t) \in B_0(\text{APTC}^\text{cert})$;
9. $\tilde{\delta} \in B(\text{APTC}^\text{cert})$;
10. if $t \in B_0(\text{APTC}^\text{cert})$, then $t \in B(\text{APTC}^\text{cert})$.

**Theorem 5.26** (Elimination theorem). Let $p$ be a closed APTC$^\text{cert}$ term. Then there is a basic APTC$^\text{cert}$ term $q$ such that $\text{APTC}^\text{cert} \vdash p = q$.

**Proof.** It is sufficient to induct on the structure of the closed APTC$^\text{dat}$ term $p$. It can be proven that $p$ combined by the constants and operators of APTC$^\text{dat}$ exists an equal basic term $q$, and the other operators not included in the basic terms, such as $\nu_{rel}$, $\Theta$, $\nu_{rel}$, $\partial_H$, $\Theta$ and $\prec$ can be eliminated.

5.4.4. Connections

**Theorem 5.27** (Generalization of APTC$^\text{cert}$). 1. By the definitions of $a = \tilde{a}$ for each $a \in A$ and $\delta = \tilde{\delta}$, APTC$^\text{cert}$ is a generalization of APTC.

2. APTC$^\text{cert}$ is a generalization of BATC$^\text{cert}$

**Proof.** 1. It follows from the following two facts.

(a) The transition rules of APTC in section 2.1 are all source-dependent;

(b) The sources of the transition rules of APTC$^\text{cert}$ contain an occurrence of $\tilde{\delta}, \tilde{a}, \sigma_{rel}^p, \nu_{rel}^p, \nu_{rel}^p$, and $\nu_{rel}$.

So, APTC is an embedding of APTC$^\text{cert}$, as desired.
Table 33. Axioms of APTC^ext

| No. | Axiom |
|-----|-------|
| P1  | $x \cdot y = x \parallel y + x \mid y$ |
| P2  | $x \parallel y = y \parallel x$ |
| P3  | $(x \parallel y) \parallel z = x \parallel (y \parallel z)$ |
| P4SR| $\tilde{a} \parallel (\tilde{b} \cdot y) = (\tilde{a} \parallel \tilde{b}) \cdot y$ |
| P5SR| $(\tilde{a} \cdot x) \parallel \tilde{b} = (\tilde{a} \parallel \tilde{b}) \cdot x$ |
| P6SR| $(\tilde{a} \cdot x) \parallel (\tilde{b} \cdot y) = (\tilde{a} \parallel \tilde{b}) \cdot (x \parallel y)$ |
| P7  | $(x \parallel y) \parallel z = (x \parallel z) + (y \parallel z)$ |
| P8  | $x \parallel (y + z) = (x \parallel y) + (x \parallel z)$ |
| SRP9ID| $(\nu_{rel}(x) + \tilde{\delta}) \parallel \sigma_{rel}^p(y) = \tilde{\delta}$ |
| SRP10ID| $\sigma_{rel}^p(x) \parallel (\nu_{rel}(y) + \tilde{\delta}) = \tilde{\delta}$ |
| SRP11| $\sigma_{rel}^p(x) \parallel \sigma_{rel}^p(y) = \sigma_{rel}^p(x \parallel y)$ |
| PID12| $\tilde{\delta} \parallel x = \tilde{\delta}$ |
| PID13| $x \parallel \tilde{\delta} = \tilde{\delta}$ |
| C14SR| $\tilde{a} \cdot \tilde{b} = \gamma(\tilde{a}, \tilde{b})$ |
| C15SR| $\tilde{a} \parallel (\tilde{b} \cdot y) = \gamma(\tilde{a}, \tilde{b}) \cdot y$ |
| C16SR| $(\tilde{a} \cdot x) \parallel \tilde{b} = \gamma(\tilde{a}, \tilde{b}) \cdot x$ |
| C17SR| $(\tilde{a} \cdot x) \parallel (\tilde{b} \cdot y) = \gamma(\tilde{a}, \tilde{b}) \cdot (x \parallel y)$ |
| C18  | $(x + y) \parallel z = (x \parallel z) + (y \parallel z)$ |
| C19  | $x \parallel (y + z) = (x \parallel y) + (x \parallel z)$ |
| DRC20ID| $(\nu_{rel}(x) + \tilde{\delta}) \parallel \sigma_{rel}^p(y) = \tilde{\delta}$ |
| DRC21ID| $\sigma_{rel}^p(x) \parallel (\nu_{rel}(y) + \tilde{\delta}) = \tilde{\delta}$ |
| DRC22| $\sigma_{rel}^p(x) \parallel \sigma_{rel}^p(y) = \sigma_{rel}^p(x \parallel y)$ |
| C1D23| $\tilde{\delta} \parallel x = \tilde{\delta}$ |
| C1D24| $x \parallel \tilde{\delta} = \tilde{\delta}$ |
| CE25DR| $\Theta(\tilde{a}) = \tilde{a}$ |
| CE26DRID| $\Theta(\tilde{a}) = \tilde{\delta}$ |
| CE27| $\Theta(x + y) = \Theta(x) \triangleleft y + \Theta(y) \triangleleft x$ |
| CE28| $\Theta(x \cdot y) = \Theta(x) \cdot \Theta(y)$ |
| CE29| $\Theta(x \parallel y) = ((\Theta(x) \parallel y) \parallel y) + ((\Theta(y) \parallel x) \parallel x)$ |
| CE30| $\Theta(x \parallel y) = ((\Theta(x) \parallel y) \parallel y) + ((\Theta(y) \parallel x) \parallel x)$ |
| U3ISRID| $(\tilde{f}(\tilde{a}, \tilde{b})) \parallel \tilde{a} \triangleleft \tilde{b} = \tilde{\delta}$ |
| U32SRID| $(\tilde{f}(\tilde{a}, \tilde{b}), \tilde{b} \triangleleft \tilde{c}) \parallel \tilde{a} \triangleleft \tilde{c} = \tilde{a}$ |
| U33SRID| $(\tilde{f}(\tilde{a}, \tilde{b}), \tilde{b} \triangleleft \tilde{c}) \parallel \tilde{c} \triangleleft \tilde{a} = \tilde{\delta}$ |
| U34SRID| $\tilde{a} \triangleleft \tilde{b} = \tilde{a}$ |
| U35SRID| $\tilde{a} \parallel \tilde{a} = \tilde{\delta}$ |
| U36| $(x + y) \triangleleft z = (x \triangleleft z) + (y \triangleleft z)$ |
| U37| $(x \parallel y) \triangleleft z = (x \parallel z) - (y \triangleleft z)$ |
| U38| $(x \parallel y) \parallel z = (x \parallel z) \parallel (y \parallel z)$ |
| U39| $(x \parallel y) \parallel z = (x \parallel z) + (y \parallel z)$ |
| U40| $x \triangleleft (y + z) = (x \triangleleft y) + (z \triangleleft y)$ |
| U41| $x \triangleleft (y \cdot z) = (x \triangleleft y) \cdot (z \triangleleft y)$ |
| U42| $x \triangleleft (y \parallel z) = (x \triangleleft y) \parallel (z \triangleleft y)$ |
| U43| $x \triangleleft (y \parallel z) = (x \triangleleft y) \parallel (z \triangleleft y)$ |
| D1SRID| $\tilde{a} \in H \cdot \partial_H(\tilde{a}) = \tilde{a}$ |
| D2SRID| $\tilde{a} \in H \cdot \partial_H(\tilde{a}) = \tilde{\delta}$ |
| D3SRID| $\partial_H(\tilde{\delta}) = \tilde{\delta}$ |
| D4| $\partial_H(x + y) = \partial_H(x) + \partial_H(y)$ |
| D5| $\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)$ |
| D6| $\partial_H(x \parallel y) = \partial_H(x) \parallel \partial_H(y)$ |
| SRD7| $\partial_H(\sigma_{rel}^p(x)) = \sigma_{rel}^p(\partial_H(x))$ |
| SRU0| $\nu_{rel}(\tilde{\delta}) = \tilde{\delta}$ |
| SRU1| $\nu_{rel}(\tilde{a}) = \tilde{a}$ |
| SRU2| $\nu_{rel}(\sigma_{rel}^p(x)) = \tilde{\delta}$ |
| SRU3| $\nu_{rel}(x + y) = \nu_{rel}(x) + \nu_{rel}(y)$ |
| SRU4| $\nu_{rel}(x \cdot y) = \nu_{rel}(x) \cdot y$ |
| SRU5| $\nu_{rel}(x \parallel y) = \nu_{rel}(x) \parallel \nu_{rel}(y)$ |

Table 33. Axioms of APTC^ext ($a, b, c \in A_\delta, p \geq 0, r > 0$)
Table 34. Transition rules of APTC$^a_{\mathfrak{a}_t}$ (a, b, c ∈ A, r > 0)
2. It follows from the following two facts.
   (a) The transition rules of BATC
       srt
      are all source-dependent;
   (b) The sources of the transition rules of APTC
       srt
      contain an occurrence of \( \downarrow, \|, \vert, \Theta, \triangleleft, \partial_H \) and \( \nu_{\text{rel}} \).

So, BATC
       srt
      is an embedding of APTC
       srt
      , as desired.

5.4.5. Congruence

**Theorem 5.28 (Congruence of APTC
       srt
     ).** Truly concurrent bisimulation equivalences \( \sim_p \), \( \sim_s \) and \( \sim_{hp} \) are all congruences with respect to APTC
       srt
      . That is,
   - pomset bisimulation equivalence \( \sim_p \) is a congruence with respect to APTC
       srt
      ;
   - step bisimulation equivalence \( \sim_s \) is a congruence with respect to APTC
       srt
      ;
   - \( hp \)-bisimulation equivalence \( \sim_{hp} \) is a congruence with respect to APTC
       srt
      .

**Proof.** It is easy to see that \( \sim_p \), \( \sim_s \), and \( \sim_{hp} \) are all equivalent relations on APTC
       srt
      terms, it is only sufficient to prove that \( \sim_p \), \( \sim_s \), and \( \sim_{hp} \) are all preserved by the operators \( \sigma_{\text{rel}}^p \), \( \nu_{\text{rel}}^p \), \( \nu_{\text{rel}}^p \), and \( \nu_{\text{rel}} \). It is trivial and we omit it.

5.4.6. Soundness

**Theorem 5.29 (Soundness of APTC
       srt
     ).** The axiomatization of APTC
       srt
      is sound modulo truly concurrent bisimulation equivalences \( \sim_p \), \( \sim_s \), and \( \sim_{hp} \). That is,
   1. let \( x \) and \( y \) be APTC
       srt
      terms. If APTC
       srt
      \( \vdash x = y \), then \( x \sim_s y \);
   2. let \( x \) and \( y \) be APTC
       srt
      terms. If APTC
       srt
      \( \vdash x = y \), then \( x \sim_p y \);
   3. let \( x \) and \( y \) be APTC
       srt
      terms. If APTC
       srt
      \( \vdash x = y \), then \( x \sim_{hp} y \).

**Proof.** Since \( \sim_p \), \( \sim_s \), and \( \sim_{hp} \) are both equivalent and congruent relations, we only need to check if each axiom in Table 33 is sound modulo \( \sim_p \), \( \sim_s \), and \( \sim_{hp} \) respectively.

1. We only check the soundness of the non-trivial axiom \( \text{SRP11} \) modulo \( \sim_s \). Let \( p, q \) be APTC
       srt
      processes, and \( \sigma_{\text{rel}}^s(p) \parallel \sigma_{\text{rel}}^s(q) = \sigma_{\text{rel}}^s(p \parallel q) \), it is sufficient to prove that \( \sigma_{\text{rel}}^s(p) \parallel \sigma_{\text{rel}}^s(q) \sim_s \sigma_{\text{rel}}^s(p \parallel q) \). By the transition rules of operator \( \sigma_{\text{rel}}^s \) and \( \parallel \) in Table 34, we get

\[
\sigma_{\text{rel}}^s(p) \parallel \sigma_{\text{rel}}^s(q) \rightarrow^* \sigma_{\text{rel}}^0(p) \parallel \sigma_{\text{rel}}^0(q)
\]

\[
\sigma_{\text{rel}}^0(p \parallel q) \rightarrow^* \sigma_{\text{rel}}^0(p \parallel q)
\]

There are several cases:

\[
p \overset{a}{\rightarrow} q \overset{b}{\rightarrow} \sqrt{ }
\]

\[
\sigma_{\text{rel}}^0(p \parallel q) \rightarrow^* \sigma_{\text{rel}}^0(p \parallel q) \quad \{a,b\}
\]

\[
p \overset{a}{\rightarrow} q \overset{b}{\rightarrow} \sqrt{ }
\]

\[
\sigma_{\text{rel}}^0(p \parallel q) \rightarrow^* \sigma_{\text{rel}}^0(p \parallel q) \quad \{a,b\}
\]

\[
p \overset{a}{\rightarrow} p' q \overset{b}{\rightarrow} \sqrt{ }
\]

\[
\sigma_{\text{rel}}^0(p \parallel q) \rightarrow^* \sigma_{\text{rel}}^0(p \parallel q) \quad \{a,b\}
\]

\[
\sigma_{\text{rel}}^0(p) \parallel \sigma_{\text{rel}}^0(q) \rightarrow^* p'
\]
From the definition of \( \text{hp-bisimulation} \), we know that \( \text{hp-bisimulation} \) is defined on the posetal product. Similarly to the proof of soundness modulo \( \text{step-bisimulation equivalence} \), we can prove that each axiom in Table 3.3 is sound modulo \( \text{hp-bisimulation equivalence} \), we just need additionally to check the above conditions on \( \text{hp-bisimulation} \), we omit them.

---

\[ p \xrightarrow{a} p', q \xrightarrow{b} q' \]

\[ \sigma^0_{\text{rel}}(p \parallel q) \xrightarrow{\{a,b\}} p' \]

\[ p \xrightarrow{a} q \xrightarrow{b} q' \]

\[ \sigma^0_{\text{rel}}(p) \parallel \sigma^0_{\text{rel}}(q) \xrightarrow{\{a,b\}} q' \]

\[ p \xrightarrow{a} q \xrightarrow{b} q' \]

\[ \sigma^0_{\text{rel}}(p) \parallel \sigma^0_{\text{rel}}(q) \xrightarrow{\{a,b\}} q' \]

\[ p \xrightarrow{a} p', q \xrightarrow{b} q' \]

\[ \sigma^0_{\text{rel}}(p) \parallel \sigma^0_{\text{rel}}(q) \xrightarrow{\{a,b\}} q' \]

\[ p \xrightarrow{a} \sigma^0_{\text{rel}}(p) \parallel \sigma^0_{\text{rel}}(q) \xrightarrow{\{a,b\}} q' \]

\[ \sigma^0_{\text{rel}}(p) \parallel \sigma^0_{\text{rel}}(q) \xrightarrow{\{a,b\}} q' \]

\[ \sigma^0_{\text{rel}}(p) \parallel \sigma^0_{\text{rel}}(q) \xrightarrow{\{a,b\}} q' \]

So, we see that each case leads to \( \sigma^*_{\text{rel}}(p) \parallel \sigma^*_{\text{rel}}(q) \sim_{\sigma^*_{\text{rel}}(p \parallel q)} \), as desired.

2. From the definition of \( \text{pomset-bisimulation} \), we know that \( \text{pomset-bisimulation} \) is defined by \( \text{pomset transitions} \), which are labeled by \( \text{pomsets} \). In a \( \text{pomset transition} \), the \( \text{events} \) (actions) in the \( \text{pomset} \) are either within \( \text{causality relations} \) (defined by \( \cdot \)) or in \( \text{concurrency} \) (implicitly defined by \( + \) and \( \cdot \)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo \( \text{step-bisimulation equivalence} \)), so, we only need to prove the case of events in causality. Without loss of generality, we take a \( \text{pomset of} \ P = \{ \tilde{a}, \tilde{b} : \tilde{a} \cdot \tilde{b} \} \). Then the \( \text{pomset transition labeled by the above} \ P \) is just composed of one single \( \text{event transition labeled} \) by \( \tilde{a} \) succeeded by another single \( \text{event transition labeled by} \ \tilde{b} \), that is, \( P = \tilde{a} \rightarrow \tilde{b} \).

Similarly to the proof of soundness modulo \( \text{step-bisimulation equivalence} \), we can prove that each axiom in Table 3.3 is sound modulo \( \text{pomset-bisimulation equivalence} \), we omit them.

3. From the definition of \( \text{hp-bisimulation} \), we know that \( \text{hp-bisimulation} \) is defined on the posetal product \( (C_1, f, C_2), f : C_1 \rightarrow C_2 \) isomorphism. Two \( \text{process terms} \ s \) related to \( C_1 \) and \( t \) related to \( C_2 \), and \( f : C_1 \rightarrow C_2 \) isomorphism. Initially, \( (C_1, f, C_2) = (\emptyset, \emptyset, \emptyset) \), and \( (\emptyset, \emptyset, \emptyset) \in \epsilon_{\text{hp}} \). When \( s = \tilde{a} \rightarrow s' (C_1 = C_1') \), there will be \( t = \tilde{a} \rightarrow t' (C_2 = C_2') \), and we define \( f' = f[a \rightarrow a] \). Then, if \( (C_1, f, C_2) \in \epsilon_{\text{hp}} \), then \( (C_1', f', C_2') \in \epsilon_{\text{hp}} \).

Similarly to the proof of soundness modulo \( \text{hp-bisimulation equivalence} \), we can prove that each axiom in Table 3.3 is sound modulo \( \text{hp-bisimulation equivalence} \), we just need additionally to check the above conditions on \( \text{hp-bisimulation} \), we omit them. 

\[ \square \]
5.4.7. Completeness

**Theorem 5.30** (Completeness of APTC\textsuperscript{srt}). The axiomatization of APTC\textsuperscript{srt} is complete modulo truly concurrent bisimulation equivalences \(\sim_p\), \(\sim_s\), and \(\sim_{hp}\). That is,

1. let \(p\) and \(q\) be closed APTC\textsuperscript{srt} terms, if \(p \sim_s q\) then \(p = q\);
2. let \(p\) and \(q\) be closed APTC\textsuperscript{srt} terms, if \(p \sim_p q\) then \(p = q\);
3. let \(p\) and \(q\) be closed APTC\textsuperscript{srt} terms, if \(p \sim_{hp} q\) then \(p = q\).

**Proof.** 1. Firstly, by the elimination theorem of APTC\textsuperscript{srt}, we know that for each closed APTC\textsuperscript{srt} term \(p\), there exists a closed basic APTC\textsuperscript{srt} term \(p\)' , such that APTC\textsuperscript{srt} \(\vdash p = p\)' , so, we only need to consider closed basic APTC\textsuperscript{srt} terms.

The basic terms modulo associativity and commutativity (AC) of conflict \(+\) (defined by axioms A1 and A2 in Table 29) and associativity and commutativity (AC) of parallel \(\parallel\) (defined by axioms P1 and P3 in Table 33), and these equivalences is denoted by \(=_{AC}\). Then, each equivalence class \(s\) modulo AC of \(+\) and \(\parallel\) has the following normal form

\[s_1 + \cdots + s_k\]

with each \(s_i\) either an atomic event or of the form

\[t_1 \cdot \cdots \cdot t_m\]

with each \(t_j\) either an atomic event or of the form

\[u_1 \parallel \cdots \parallel u_n\]

with each \(u_1\) an atomic event, and each \(s_i\) is called the summand of \(s\).

Now, we prove that for normal forms \(n\) and \(n'\), if \(n \sim_s n'\) then \(n =_{AC} n'\). It is sufficient to induct on the sizes of \(n\) and \(n'\). We can get \(n =_{AC} n'\).

Finally, let \(s\) and \(t\) be basic APTC\textsuperscript{srt} terms, and \(s \sim_s t\), there are normal forms \(n\) and \(n'\), such that \(s = n\) and \(t = n'\). The soundness theorem modulo step bisimulation equivalence yields \(s \sim_s n\) and \(t \sim_s n'\), so \(n \sim_s n'\). Since if \(n \sim_s n'\) then \(n =_{AC} n'\), \(s = n =_{AC} n' = t\), as desired.

2. This case can be proven similarly, just by replacement of \(\sim_s\) by \(\sim_p\).
3. This case can be proven similarly, just by replacement of \(\sim_s\) by \(\sim_{hp}\).

5.5. APTC\textsuperscript{srt} with Integration

In this subsection, we will introduce the theory APTC\textsuperscript{srt} with integration called APTC\textsuperscript{srt\textsubscript{I}}.

5.5.1. The Theory APTC\textsuperscript{srt\textsubscript{I}}

**Definition 5.31** (Signature of APTC\textsuperscript{srt\textsubscript{I}}). The signature of APTC\textsuperscript{srt\textsubscript{I}} consists of the signature of APTC\textsuperscript{srt} and the integration operator \(\int : \mathcal{P}(\mathbb{R}^2) \times \mathbb{R}^2 \cdot \mathcal{P}_{rel} \to \mathcal{P}_{rel}\).

The set of axioms of APTC\textsuperscript{srt\textsubscript{I}} consists of the laws given in Table 35.

The operational semantics of APTC\textsuperscript{srt\textsubscript{I}} are defined by the transition rules in Table 32.

5.5.2. Elimination

**Definition 5.32** (Basic terms of APTC\textsuperscript{srt\textsubscript{I}}). The set of basic terms of APTC\textsuperscript{srt\textsubscript{I}}, \(\mathcal{B}(\text{APTC}^{\text{srt\textsubscript{I}}})\), is inductively defined as follows by two auxiliary sets \(\mathcal{B}_0(\text{APTC}^{\text{srt\textsubscript{I}}})\) and \(\mathcal{B}_1(\text{APTC}^{\text{srt\textsubscript{I}}})\):

1. if \(a \in A\), then \(\hat{a} \in \mathcal{B}_1(\text{APTC}^{\text{srt\textsubscript{I}}})\);
Table 35. Axioms of APTC

| No. | Axiom |
|-----|-------|
| INT13 | \( \int_{x \in V} (F(x) \parallel x) = (\int_{x \in V} F(x)) \parallel x \) |
| INT14 | \( \int_{x \in V} x \parallel F(x) = x \parallel (\int_{x \in V} F(x)) \) |
| INT15 | \( \int_{x \in V} F(x) \parallel x = (\int_{x \in V} F(x)) \parallel x \) |
| INT16 | \( \int_{x \in V} (x \parallel F(x)) = x \parallel (\int_{x \in V} F(x)) \) |
| INT17 | \( \int_{x \in V} \partial_u F(x) = \partial_u (\int_{x \in V} F(x)) \) |
| INT18 | \( \int_{x \in V} \Theta(F(x)) = \Theta(\int_{x \in V} F(x)) \) |
| INT19 | \( \int_{x \in V} F(x) \parallel x = (\int_{x \in V} F(x)) \parallel x \) |
| SRU5  | \( \nu_{\text{rel}}(\int_{x \in V} P) = \int_{x \in V} \nu_{\text{rel}}(P) \) |

2. If \( a \in A \) and \( t \in B(\text{APTC}^{\text{cert}}) \), then \( \tilde{a} \cdot t \in B_1(\text{APTC}^{\text{cert}}) \);
3. If \( t, t' \in B_1(\text{APTC}^{\text{cert}}) \), then \( t + t' \in B_1(\text{APTC}^{\text{cert}}) \);
4. If \( t, t' \in B_1(\text{APTC}^{\text{cert}}) \), then \( t \parallel t' \in B_1(\text{APTC}^{\text{cert}}) \);
5. If \( t \in B_1(\text{APTC}^{\text{cert}}) \), then \( t \in B_0(\text{APTC}^{\text{cert}}) \);
6. If \( p > 0 \) and \( t \in B_0(\text{APTC}^{\text{cert}}) \), then \( \sigma_{\text{rel}}(t) \in B_0(\text{APTC}^{\text{cert}}) \);
7. If \( p > 0 \), \( t \in B_1(\text{APTC}^{\text{cert}}) \) and \( t' \in B_0(\text{APTC}^{\text{cert}}) \), then \( t + \sigma_{\text{rel}}(t') \in B_0(\text{APTC}^{\text{cert}}) \);
8. If \( t \in B_0(\text{APTC}^{\text{cert}}) \), then \( \nu_{\text{rel}}(t) \in B_0(\text{APTC}^{\text{cert}}) \);
9. If \( t \in B_0(\text{APTC}^{\text{cert}}) \), then \( \int_{x \in V} t \in B_0(\text{APTC}^{\text{cert}}) \);
10. \( \delta \in B(\text{APTC}^{\text{cert}}) \);
11. If \( t \in B_0(\text{APTC}^{\text{cert}}) \), then \( t \in B(\text{APTC}^{\text{cert}}) \).

**Theorem 5.33** (Elimination theorem). Let \( p \) be a closed APTC^{\text{cert}} term. Then there is a basic APTC^{\text{cert}} term \( q \) such that \( APTC^{\text{cert}} \vdash p = q \).

**Proof.** It is sufficient to induct on the structure of the closed APTC^{\text{dat}} term \( p \). It can be proven that \( p \) combined by the constants and operators of APTC^{\text{dat}} exists an equal basic term \( q \), and the other operators not included in the basic terms, such as \( \nu_{\text{rel}}, \pi_{\text{rel}}, \parallel, \partial_H, \Theta \) and \( \triangleleft \) can be eliminated.

5.5.3. Connections

**Theorem 5.34** (Generalization of APTC^{\text{cert}}). 1. By the definitions of \( a = \int_{x \in [0, \infty)} \sigma_{\text{rel}}(\tilde{a}) \) for each \( a \in A \) and \( \delta = \int_{x \in [0, \infty)} \sigma_{\text{rel}}(\tilde{\delta}) \), APTC^{\text{cert}} is a generalization of APTC.

2. APTC^{\text{cert}} is a generalization of APTC^{\text{cert}}.

**Proof.** 1. It follows from the following two facts.
   (a) The transition rules of APTC in section 2.1 are all source-dependent;
   (b) The sources of the transition rules of APTC^{\text{cert}} contain an occurrence of \( \tilde{\delta}, \tilde{a}, \sigma_{\text{rel}}, \nu_{\text{rel}}, \pi_{\text{rel}}, \nu_{\text{rel}}, \) and \( f \).

So, APTC is an embedding of APTC^{\text{cert}}, as desired.

2. It follows from the following two facts.
   (a) The transition rules of APTC^{\text{cert}} are all source-dependent;
   (b) The sources of the transition rules of APTC^{\text{cert}} contain an occurrence of \( f \).

So, APTC^{\text{cert}} is an embedding of APTC^{\text{cert}}, as desired.

5.5.4. Congruence

**Theorem 5.35** (Congruence of APTC^{\text{cert}}). Truly concurrent bisimulation equivalences are all congruences with respect to APTC^{\text{cert}}. That is,
• pomset bisimulation equivalence $\sim_p$ is a congruence with respect to APTC$^\text{ert}$I;
• step bisimulation equivalence $\sim_s$ is a congruence with respect to APTC$^\text{ert}$I;
• hp-bisimulation equivalence $\sim_{hp}$ is a congruence with respect to APTC$^\text{ert}$I.

Proof. It is easy to see that $\sim_p$, $\sim_s$, and $\sim_{hp}$ are all preserved by the operators $f$. It is trivial and we omit it.

5.5.5. Soundness

Theorem 5.36 (Soundness of APTC$^\text{ert}$I). The axiomatization of APTC$^\text{ert}$I is sound modulo truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$, and $\sim_{hp}$. That is,

1. let $x$ and $y$ be APTC$^\text{ert}$I terms. If $\text{APTC}^\text{ert}I \vdash x = y$, then $x \sim_s y$;
2. let $x$ and $y$ be APTC$^\text{ert}$I terms. If $\text{APTC}^\text{ert}I \vdash x = y$, then $x \sim_p y$;
3. let $x$ and $y$ be APTC$^\text{ert}$I terms. If $\text{APTC}^\text{ert}I \vdash x = y$, then $x \sim_{hp} y$.

Proof. Since $\sim_p$, $\sim_s$, and $\sim_{hp}$ are both equivalent and congruent relations, we only need to check the soundness of each axiom in Table 35 is sound modulo $\sim_p$, $\sim_s$, and $\sim_{hp}$ respectively.

1. We can check the soundness of each axiom in Table 35 by the transition rules in Table 32, it is trivial and we omit them.

2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by $\cdot$) or in concurrency (implicitly defined by $\cdot$ and $+$, and explicitly defined by $\parallel$), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of $P = \{\tilde{a}, \tilde{b} : \tilde{a} \cdot \tilde{b}\}$. Then the pomset transition labeled by the above $P$ is just composed of one single event transition labeled by $\tilde{a}$ succeeded by another single event transition labeled by $\tilde{b}$, that is, $\frac{P}{\tilde{a}/\tilde{b}} \rightarrow a \cdot b$. Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 35 is sound modulo pomset bisimulation equivalence, we omit them.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product $(C_1, f, C_2), f : C_1 \to C_2$ isomorphism. Two process terms $s$ related to $C_1$ and $t$ related to $C_2$, and $f : C_1 \to C_2$ isomorphism. Initially, $(C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)$, and $(\emptyset, \emptyset, \emptyset \sim_{hp})$. Then, $s \xrightarrow{a} s' (C_1 \xrightarrow{a} C_1')$, there will be $t \xrightarrow{a} t' (C_2 \xrightarrow{a} C_2')$, and we define $f' = f[a \mapsto a]$. Then, $(C_1, f, C_2) \sim_{hp}$, then $(C_1', f', C_2') \sim_{hp}$. Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 35 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

4. We just need to add downward-closed condition to the soundness modulo hp-bisimulation equivalence, we omit them.

5.5.6. Completeness

Theorem 5.37 (Completeness of APTC$^\text{ert}$I). The axiomatization of APTC$^\text{ert}$I is complete modulo truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$, $\sim_{hp}$ and $\sim_{hp}$. That is,

1. let $p$ and $q$ be closed APTC$^\text{ert}$I terms, if $p \sim_s q$ then $p = q$;
2. let $p$ and $q$ be closed APTC$^\text{ert}$I terms, if $p \sim_p q$ then $p = q$;
3. let $p$ and $q$ be closed APTC$^\text{ert}$I terms, if $p \sim_{hp} q$ then $p = q$.

Proof. 1. Firstly, by the elimination theorem of APTC$^\text{ert}$I, we know that for each closed APTC$^\text{ert}$I term $p$, there exists a closed basic APTC$^\text{ert}$I term $p'$, such that APTC$^\text{ert}I \vdash p = p'$, so, we only need to consider closed basic APTC$^\text{ert}$I terms.
The basic terms modulo associativity and commutativity (AC) of conflict + (defined by axioms A1 and A2 in Table 29) and associativity and commutativity (AC) of parallel \parallel (defined by axioms P2 and P3 in Table 33), and these equivalences is denoted by $=_{\text{AC}}$. Then, each equivalence class $s$ modulo AC of $+$ and $\parallel$ has the following normal form

$$s_1 + \cdots + s_k$$

with each $s_i$ either an atomic event or of the form

$$t_1 \cdots \cdot t_m$$

with each $t_j$ either an atomic event or of the form

$$u_1 \parallel \cdots \parallel u_n$$

with each $u_l$ an atomic event, and each $s_i$ is called the summand of $s$.

Now, we prove that for normal forms $n$ and $n'$, if $n \sim_s n'$ then $n =_{\text{AC}} n'$. It is sufficient to induct on the sizes of $n$ and $n'$. We can get $n =_{\text{AC}} n'$.

Finally, let $s$ and $t$ be basic APTC terms, and $s \sim_s t$, there are normal forms $n$ and $n'$, such that $s = n$ and $t = n'$. The soundness theorem modulo step bisimulation equivalence yields $s \sim_s n$ and $t \sim_s n'$, so $n \sim_s s \sim_s t \sim_s n'$. Since if $n \sim_s n'$ then $n =_{\text{AC}} n'$, $s = n =_{\text{AC}} n' = t$, as desired.

2. This case can be proven similarly, just by replacement of $\sim_s$ by $\sim_p$.

3. This case can be proven similarly, just by replacement of $\sim_s$ by $\sim_{hp}$.

\section{Continuous Absolute Timing}

In this section, we will introduce a version of APTC with absolute timing and time measured on a continuous time scale. Measuring time on a continuous time scale means that timing is now done with respect to time points on a continuous time scale. While in absolute timing, all timing is counted from the start of the whole process.

Like APTC without timing, let us start with a basic algebra for true concurrency called BATC sat (BATC with continuous absolute timing). Then we continue with APTC sat (APTC with continuous absolute timing).

\subsection{Basic Definitions}

In this subsection, we will introduce some basic definitions about timing. These basic concepts come from [25], we introduce them into the backgrounds of true concurrency.

\begin{definition}[Undelayable actions]
Undelayable actions are defined as atomic processes that perform an action and then terminate successfully. We use a constant $\tilde{a}$ to represent the undelayable action, that is, the atomic process that performs the action $a$ and then terminates successfully.
\end{definition}

\begin{definition}[Undelayable deadlock]
Undelayable deadlock $\tilde{\delta}$ is an additional process that is neither capable of performing any action nor capable of idling beyond the current point of time.
\end{definition}

\begin{definition}[Absolute delay]
The absolute delay of the process $p$ for a period of time $r$ ($r \in \mathbb{R}$) is the process that idles a period of time $r$ longer than $p$ and otherwise behaves like $p$. The operator $\sigma_{\text{abs}}$ is used to represent the absolute delay, and let $\sigma_{\text{abs}}(t) = r\sigma_{\text{abs}}$.
\end{definition}

\begin{definition}[Deadlocked process]
Deadlocked process $\tilde{\delta}$ is an additional process that has deadlocked before point of time 0. After a delay of a period of time, the undelayable deadlock $\tilde{\delta}$ and the deadlocked process $\tilde{\delta}$ are indistinguishable from each other.
\end{definition}

\begin{definition}[Truly concurrent bisimulation equivalences with time-related capabilities]
The following requirement with time-related capabilities is added to truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$, $\sim_{hp}$ and $\sim_{hdp}$ and Definition 5.5:
\end{definition}
• in case of absolute timing, the requirements in Definition 6.3 apply to the capabilities in a certain period of time.

Definition 6.6 (Integration). Let \( f \) be a function from \( \mathbb{R}^2 \) to processes with continuous absolute timing and \( V \subseteq \mathbb{R}^2 \). The integration \( f \) of \( f \) over \( V \) is the process that behaves like one of the process in \( \{ f(r) | r \in V \} \).

Definition 6.7 (Absolute time-out). The absolute time-out \( \upsilon_{\text{abs}} \) of a process \( p \) at point of time \( r \) \((r \in \mathbb{R}^2)\) behaves either like the part of \( p \) that does not idle till point of time \( r \), or like the deadlocked process after a delay of a period of time \( r \) if \( p \) is capable of idling till point of time \( r \); otherwise, like \( p \). And let \( \upsilon'_{\text{abs}}(t) = r \upsilon_{\text{abs}}t \).

Definition 6.8 (Absolute initialization). The absolute initialization \( \overline{\upsilon}_{\text{abs}} \) of a process \( p \) at point of time \( r \) \((r \in \mathbb{R}^2)\) behaves like the part of \( p \) that idles till point of time \( r \) if \( p \) is capable of idling till that point of time; otherwise, like the deadlocked process after a delay of a period of time \( r \). And we let \( \overline{\upsilon}_{\text{abs}}(t) = r \overline{\upsilon}_{\text{abs}}t \).

6.2. Basic Algebra for True Concurrency with Continuous Absolute Timing

In this subsection, we will introduce the theory \( \text{BATC}^{\text{sat}} \).

6.2.1. The Theory \( \text{BATC}^{\text{sat}} \)

Definition 6.9 (Signature of \( \text{BATC}^{\text{sat}} \)). The signature of \( \text{BATC}^{\text{sat}} \) consists of the sort \( \mathcal{P}_{\text{abs}} \) of processes with continuous absolute timing, the undelayable action constants \( \alpha : \mathcal{P}_{\text{abs}} \) for each \( \alpha \in \mathcal{A} \), the undelayable deadlock constant \( \delta : \mathcal{P}_{\text{abs}} \), the alternative composition operator \( + : \mathcal{P}_{\text{abs}} \times \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}} \), the sequential composition operator \( \cdot : \mathcal{P}_{\text{abs}} \times \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}} \), the absolute delay operator \( \sigma_{\text{abs}} : \mathbb{R}^2 \times \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}} \), the deadlocked process constant \( \delta : \mathcal{P}_{\text{abs}} \), the absolute time-out operator \( \upsilon_{\text{abs}} : \mathbb{R}^2 \times \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}} \) and the absolute initialization operator \( \overline{\upsilon}_{\text{abs}} : \mathbb{R}^2 \times \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}} \).

The set of axioms of \( \text{BATC}^{\text{sat}} \) consists of the laws given in Table 36.

The operational semantics of \( \text{BATC}^{\text{sat}} \) are defined by the transition rules in Table 37. The transition rules are defined on \( (t,p) \), where \( t \) is a term and \( p \in \mathbb{R}^2 \). Where \( \uparrow \) is a unary deadlocked predicate, and \( (t,p) \uparrow = -((t,p) \uparrow) \); \( (t,p) \rightarrow ^q (t',p') \) means that process \( t \) is capable of first idling till the \( q \)-th next time slice, and then proceeding as process \( t' \) and \( q + p = p' \).

6.2.2. Elimination

Definition 6.10 (Basic terms of \( \text{BATC}^{\text{sat}} \)). The set of basic terms of \( \text{BATC}^{\text{sat}} \), \( \mathcal{B}(\text{BATC}^{\text{sat}}) \), is inductively defined as follows by two auxiliary sets \( \mathcal{B}_0(\text{BATC}^{\text{sat}}) \) and \( \mathcal{B}_1(\text{BATC}^{\text{sat}}) \):

1. if \( a \in \mathcal{A} \), then \( \alpha \in \mathcal{B}_1(\text{BATC}^{\text{sat}}) \);
2. if \( a \in \mathcal{A} \) and \( t \in \mathcal{B}(\text{BATC}^{\text{sat}}) \), then \( \alpha \cdot t \in \mathcal{B}_1(\text{BATC}^{\text{sat}}) \);
3. if \( t,t' \in \mathcal{B}_1(\text{BATC}^{\text{sat}}) \), then \( t + t' \in \mathcal{B}_1(\text{BATC}^{\text{sat}}) \);
4. if \( t \in \mathcal{B}_1(\text{BATC}^{\text{sat}}) \), then \( t \in \mathcal{B}_0(\text{BATC}^{\text{sat}}) \);
5. if \( p > 0 \) and \( t \in \mathcal{B}_0(\text{BATC}^{\text{sat}}) \), then \( \sigma_{\text{abs}}(t) \in \mathcal{B}_0(\text{BATC}^{\text{sat}}) \);
6. if \( p > 0 \), \( t \in \mathcal{B}_1(\text{BATC}^{\text{sat}}) \) and \( t' \in \mathcal{B}_0(\text{BATC}^{\text{sat}}) \), then \( t + \sigma_{\text{abs}}(t') \in \mathcal{B}_0(\text{BATC}^{\text{sat}}) \);
7. \( \delta \in \mathcal{B}(\text{BATC}^{\text{sat}}) \);
8. if \( t \in \mathcal{B}_0(\text{BATC}^{\text{sat}}) \), then \( t \in \mathcal{B}(\text{BATC}^{\text{sat}}) \).

Theorem 6.11 (Elimination theorem). Let \( p \) be a closed \( \text{BATC}^{\text{sat}} \) term. Then there is a basic \( \text{BATC}^{\text{sat}} \) term \( q \) such that \( \text{BATC}^{\text{sat}} \vdash p = q \).

Proof. It is sufficient to induct on the structure of the closed \( \text{BATC}^{\text{sat}} \) term \( p \). It can be proven that \( p \) combined by the constants and operators of \( \text{BATC}^{\text{sat}} \) exists an equal basic term \( q \), and the other operators not included in the basic terms, such as \( \upsilon_{\text{abs}} \) and \( \overline{\upsilon}_{\text{abs}} \) can be eliminated. \( \square \)
Proof. It is easy to see that \( \sim \) is sufficient to prove that \( \text{SAT} \) with respect to \( \text{BATC} \).

Theorem 6.13

\[ 6.2.4. \text{Connections} \]

Table 36. Axioms of \( \text{BATC}^{\text{sat}} \) (\( a \in A, p, q \geq 0, r > 0 \))

6.2.3. Connections

Theorem 6.12 (Generalization of \( \text{BATC}^{\text{sat}} \)). By the definitions of \( a = \tilde{a} \) for each \( a \in A \) and \( \delta = \tilde{\delta} \), \( \text{BATC}^{\text{sat}} \) is a generalization of \( \text{BATC} \).

Proof. It follows from the following two facts.

1. The transition rules of \( \text{BATC} \) in section 2.1 are all source-dependent;
2. The sources of the transition rules of \( \text{BATC}^{\text{sat}} \) contain an occurrence of \( \tilde{\delta}, \tilde{a}, \sigma_p^{\text{rel}}, v_p^{\text{abs}} \) and \( \overline{v}_p^{\text{abs}} \).

So, \( \text{BATC} \) is an embedding of \( \text{BATC}^{\text{sat}} \), as desired.

6.2.4. Congruence

Theorem 6.13 (Congruence of \( \text{BATC}^{\text{sat}} \)). Truly concurrent bisimulation equivalences are all congruences with respect to \( \text{BATC}^{\text{sat}} \). That is,

- pomset bisimulation equivalence \( \sim_p \) is a congruence with respect to \( \text{BATC}^{\text{sat}} \);
- step bisimulation equivalence \( \sim_s \) is a congruence with respect to \( \text{BATC}^{\text{sat}} \);
- hp-bisimulation equivalence \( \sim_{hp} \) is a congruence with respect to \( \text{BATC}^{\text{sat}} \);
- hhp-bisimulation equivalence \( \sim_{hhp} \) is a congruence with respect to \( \text{BATC}^{\text{sat}} \).

Proof. It is easy to see that \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhp} \) are all equivalent relations on \( \text{BATC}^{\text{sat}} \) terms, it is only sufficient to prove that \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhp} \) are all preserved by the operators \( \sigma_p^{\text{abs}}, v_p^{\text{abs}} \) and \( \overline{v}_p^{\text{abs}} \). It is trivial and we omit it.
Table 37. Transition rules of BATC\textsuperscript{sat} ($a \in A, p, q \geq 0, r > 0$)

| Rule                                      | Meaning                                                                 |
|-------------------------------------------|-------------------------------------------------------------------------|
| $(\delta, p) \uparrow$                    | $(\delta, r) \uparrow$                                                 |
| $(\delta, 0) \rightarrow (\emptyset, 0)$ | $(\delta, r) \uparrow$                                                 |
| $(x, p) \xrightarrow{a} (x', p)$          | $(x, p) \xrightarrow{a} (x', p)$                                      |
| $(\sigma_{abs}^0(x), p) \xrightarrow{+} (x', p)$ | $(\sigma_{abs}^0(x), p + r) \xrightarrow{+} (\sigma_{abs}^0(x'), p + r)$ |
| $(x, p) \xrightarrow{a} (\emptyset, p)$  | $(x, p) \xrightarrow{a} (\emptyset, p)$                              |
| $(\sigma_{abs}^q(x), p + q) \xrightarrow{+} (\sigma_{abs}^q(x), p + q)$ | $(\sigma_{abs}^q(x), p + q + r) \xrightarrow{+} (\sigma_{abs}^q(x), p + q + r)$ |
| $(x, p) \xrightarrow{a} (x', p)$          | $(x, p) \xrightarrow{a} (x', p)$                                      |
| $(x + y, p) \xrightarrow{+} (x', p)$      | $(x + y, p) \xrightarrow{+} (x', p)$                                  |
| $(x, p) \xrightarrow{a} (\emptyset, p)$  | $(x, p) \xrightarrow{a} (\emptyset, p)$                              |
| $(x, p) \xrightarrow{(\emptyset, p)}$    | $(x, p) \xrightarrow{(\emptyset, p)}$                                |
| $(x, p) \xrightarrow{a} (x', p)$          | $(x, p) \xrightarrow{a} (x', p)$                                      |
| $(v_{abs}^0(x), p) \xrightarrow{a} (x', p)$ | $(v_{abs}^0(x), p + r) \xrightarrow{a} (v_{abs}^0(x'), p + r)$         |
| $(x, p) \xrightarrow{a} (v_{abs}^0(x), p)$ | $(x, p) \xrightarrow{a} (v_{abs}^0(x), p + r)$                        |
| $(v_{abs}^q(x), p) \xrightarrow{+} (v_{abs}^q(x), p + r)$ | $(v_{abs}^q(x), p) \xrightarrow{+} (v_{abs}^q(x), p + r)$ |
| $(x, p) \xrightarrow{a} (v_{abs}^0(x), p)$ | $(x, p) \xrightarrow{a} (v_{abs}^0(x), p + r)$                        |
| $(x, p) \xrightarrow{(\emptyset, p)}$    | $(x, p) \xrightarrow{(\emptyset, p)}$                                |
| $(x, p) \xrightarrow{a} (v_{abs}^q(x), p)$ | $(x, p) \xrightarrow{a} (v_{abs}^q(x), p + r)$                        |
| $(v_{abs}^{\tau q}(x), p) \xrightarrow{+} (v_{abs}^{\tau q}(x), p + r)$ | $(v_{abs}^{\tau q}(x), p) \xrightarrow{+} (v_{abs}^{\tau q}(x), p + r)$ |
| $(x, p) \xrightarrow{a} (v_{abs}^{\tau q}(x), p)$ | $(x, p) \xrightarrow{a} (v_{abs}^{\tau q}(x), p + r)$ |
| $(x, r + q) \uparrow$                      | $(x, r + q) \uparrow$                                                 |
| $(\tau_{abs}^{\tau q}(x), p) \xrightarrow{+} (\tau_{abs}^{\tau q}(x), p + r)$ | $(\tau_{abs}^{\tau q}(x), p) \xrightarrow{+} (\tau_{abs}^{\tau q}(x), r + q)$ |
| $(x, p) \xrightarrow{a} (\tau_{abs}^{\tau q}(x), p)$ | $(x, p) \xrightarrow{a} (\tau_{abs}^{\tau q}(x), p + r)$ |

Table 37. Transition rules of BATC\textsuperscript{sat} ($a \in A, p, q \geq 0, r > 0$)
6.2.5. Soundness

**Theorem 6.14** (Soundness of BATC\textsuperscript{sat}). The axiomatization of BATC\textsuperscript{sat} is sound modulo truly concurrent bisimulation equivalences \(\sim_p, \sim_s, \sim_h\) and \(\sim_{hp}\). That is,

1. let \(x\) and \(y\) be BATC\textsuperscript{sat} terms. If \(\text{BATC} \vdash x = y\), then \(x \sim_s y\);
2. let \(x\) and \(y\) be BATC\textsuperscript{sat} terms. If \(\text{BATC} \vdash x = y\), then \(x \sim_p y\);
3. let \(x\) and \(y\) be BATC\textsuperscript{sat} terms. If \(\text{BATC} \vdash x = y\), then \(x \sim_h y\);
4. let \(x\) and \(y\) be BATC\textsuperscript{sat} terms. If \(\text{BATC} \vdash x = y\), then \(x \sim_{hp} y\).

**Proof.** Since \(\sim_p, \sim_s, \sim_h\) and \(\sim_{hp}\) are both equivalent and congruent relations, we only need to check if each axiom in Table 36 is sound modulo \(\sim_p, \sim_s, \sim_h\) and \(\sim_{hp}\) respectively.

1. We only check the soundness of the non-trivial axiom \(SAT03\) modulo \(\sim_s\). Let \(p\) be BATC\textsuperscript{sat} processes, and \(v_{\text{abs}}^r(\sigma_{\text{abs}}^s(p)) = \sigma_{\text{abs}}^s(v_{\text{abs}}^r(p))\), it is sufficient to prove that \(v_{\text{abs}}^r(\sigma_{\text{abs}}^s(p)) \sim_s \sigma_{\text{abs}}^s(v_{\text{abs}}^r(p))\). By the transition rules of operator \(\sigma_{\text{abs}}^s\) and \(v_{\text{abs}}^r\) in Table 37, we get

\[
\begin{align*}
\langle p, 0 \rangle & \cdot \\
\langle v_{\text{abs}}^r(\sigma_{\text{abs}}^s(p)), s' \rangle \mapsto^* & \langle v_{\text{abs}}^r(\sigma_{\text{abs}}^s(p)), s' + s \rangle \\
\langle p, 0 \rangle & \cdot \\
\langle \sigma_{\text{abs}}^s(v_{\text{abs}}^r(p)), s' \rangle \mapsto^* & \langle \sigma_{\text{abs}}^s(v_{\text{abs}}^r(p)), s' + s \rangle
\end{align*}
\]

There are several cases:

\[
\begin{align*}
\langle p, s' \rangle & \overset{a}{\rightarrow} \langle \sqrt{\cdot}, s' \rangle \\
\langle v_{\text{abs}}^r(\sigma_{\text{abs}}^s(p)), s' + s \rangle & \overset{\cdot}{\rightarrow} \langle \sqrt{\cdot}, s' + s \rangle \\
\langle p, s' \rangle & \overset{a}{\rightarrow} \langle \sqrt{\cdot}, s' \rangle \\
\langle \sigma_{\text{abs}}^s(v_{\text{abs}}^r(p)), s' + s \rangle & \overset{\cdot}{\rightarrow} \langle \sqrt{\cdot}, s' + s \rangle \\
\langle p, s' \rangle & \overset{a}{\rightarrow} \langle p', s' \rangle \\
\langle v_{\text{abs}}^r(\sigma_{\text{abs}}^s(p)), s' + s \rangle & \overset{\cdot}{\rightarrow} \langle \sigma_{\text{abs}}^s(p'), s' + s \rangle \\
\langle p, s' \rangle & \overset{a}{\rightarrow} \langle p', s' \rangle \\
\langle \sigma_{\text{abs}}^s(v_{\text{abs}}^r(p)), s' + s \rangle & \overset{\cdot}{\rightarrow} \langle \sigma_{\text{abs}}^s(p'), s' + s \rangle \\
\langle p, s' \rangle & \overset{\cdot}{\rightarrow} \\
\langle v_{\text{abs}}^r(\sigma_{\text{abs}}^s(p)), s' + s \rangle & \overset{\cdot}{\rightarrow} \\
\langle p, s' \rangle & \overset{\cdot}{\rightarrow}
\end{align*}
\]

So, we see that each case leads to \(v_{\text{abs}}^r(\sigma_{\text{abs}}^s(p)) \sim_s \sigma_{\text{abs}}^s(v_{\text{abs}}^r(p))\), as desired.

2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \(\cdot\)) or in concurrency (implicitly defined by \(\cdot\) and \(+\), and explicitly defined by \(\uparrow\)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \(P = \{a, b : \tilde{a} \sim \tilde{b}\}\). Then the pomset transition labeled by the above \(P\) is just composed of one single event transition labeled by \(\tilde{a}\) succeeded by another single event transition labeled by \(\tilde{b}\), that is, \(\tilde{a} \rightarrow \tilde{b}\).

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 36 is sound modulo pomset bisimulation equivalence, we omit them.
3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product \((C_1, f, C_2), f : C_1 \to C_2\) isomorphism. Two process terms \(s\) related to \(C_1\) and \(t\) related to \(C_2\), and \(f : C_1 \to C_2\) isomorphism. Initially, \((C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)\), and \((\emptyset, \emptyset, \emptyset) \sim_{hp}\). When \(s \xrightarrow{a} s' (C_1 \xrightarrow{a} C_1')\), there will be \(t \xrightarrow{a} t' (C_2 \xrightarrow{a} C_2')\), and we define \(f' = f[a \mapsto a]\). Then, if \((C_1, f, C_2) \sim_{hp}\), then \((C_1', f', C_2') \sim_{hp}\).

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 36 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

4. We just need to add downward-closed condition to the soundness modulo hp-bisimulation equivalence, we omit them.

\[ \square \]

6.2.6. Completeness

**Theorem 6.15** (Completeness of \(\text{BATC}^{\text{sat}}\)). The axiomatization of \(\text{BATC}^{\text{sat}}\) is complete modulo truly concurrent bisimulation equivalences \(\sim_p, \sim_s, \sim_{hp}\) and \(\sim_{hhp}\). That is,\\

1. let \(p\) and \(q\) be closed \(\text{BATC}^{\text{sat}}\) terms, if \(p \sim_p q\) then \(p = q\);
2. let \(p\) and \(q\) be closed \(\text{BATC}^{\text{sat}}\) terms, if \(p \sim_s q\) then \(p = q\);
3. let \(p\) and \(q\) be closed \(\text{BATC}^{\text{sat}}\) terms, if \(p \sim_{hp} q\) then \(p = q\);
4. let \(p\) and \(q\) be closed \(\text{BATC}^{\text{sat}}\) terms, if \(p \sim_{hhp} q\) then \(p = q\).

**Proof.** 1. Firstly, by the elimination theorem of \(\text{BATC}^{\text{sat}}\), we know that for each closed \(\text{BATC}^{\text{sat}}\) term \(p\), there exists a closed basic \(\text{BATC}^{\text{sat}}\) term \(p'\), such that \(\text{BATC}^{\text{sat}} \vdash p = p'\), so, we only need to consider closed basic \(\text{BATC}^{\text{sat}}\) terms.

The basic terms modulo associativity and commutativity (AC) of conflict + (defined by axioms A1 and A2 in Table 36), and this equivalence is denoted by \(=_{AC}\). Then, each equivalence class \(s\) modulo AC of + has the following normal form

\[ s_1 + \cdots + s_k \]

with each \(s_i\) either an atomic event or of the form \(t_1 \cdot t_2\), and each \(s_i\) is called the summand of \(s\).

Now, we prove that for normal forms \(n\) and \(n'\), if \(n \sim s n'\) then \(n =_{AC} n'\). It is sufficient to induct on the sizes of \(n\) and \(n'\). We can get \(n =_{AC} n'\).

Finally, let \(s\) and \(t\) be basic terms, and \(s \sim t\), there are normal forms \(n\) and \(n'\), such that \(s = n\) and \(t = n'\). The soundness theorem of \(\text{BATC}^{\text{sat}}\) modulo step bisimulation equivalence yields \(s \sim n\) and \(t \sim n'\), so \(n \sim s n'\) and \(s = n =_{AC} n'\). Since if \(n \sim s n'\) then \(n =_{AC} n'\), \(s = n =_{AC} n'\), as desired.

2. This case can be proven similarly, just by replacement of \(\sim_s\) by \(\sim_p\).
3. This case can be proven similarly, just by replacement of \(\sim_s\) by \(\sim_{hp}\).
4. This case can be proven similarly, just by replacement of \(\sim_s\) by \(\sim_{hhp}\).

\[ \square \]

6.3. \(\text{BATC}^{\text{sat}}\) with Integration

In this subsection, we will introduce the theory \(\text{BATC}^{\text{sat}}\) with integration called \(\text{BATC}^{\text{sat}}\)I.

6.3.1. The Theory \(\text{BATC}^{\text{sat}}\)I

**Definition 6.16** (Signature of \(\text{BATC}^{\text{sat}}\)I). The signature of \(\text{BATC}^{\text{sat}}\)I consists of the signature of \(\text{BATC}^{\text{sat}}\) and the integration operator \(\int : \mathcal{P}(\mathbb{R}^2) \times \mathbb{R}^2 \cdot \mathcal{P}_{\text{abs}} \to \mathcal{P}_{\text{abs}}\).

The set of axioms of \(\text{BATC}^{\text{sat}}\)I consists of the laws given in Table 38.

The operational semantics of \(\text{BATC}^{\text{sat}}\)I are defined by the transition rules in Table 39.
It is sufficient to induct on the structure of the closed $\text{BATC}^\text{sat}$ term. Then there is a basic $\text{BATC}^\text{sat}$ term $q$ such that $\text{BATC}^\text{sat} \vdash p = q$.

**Proof.** It is sufficient to induct on the structure of the closed $\text{BATC}^\text{sat}$ term $p$. It can be proven that $p$ combined by the constants and operators of $\text{BATC}^\text{sat}$ exists an equal basic term $q$, and the other operators not included in the basic terms, such as $\vee_{\text{abs}}$ and $\forall_{\text{abs}}$ can be eliminated. \qed
6.3.3. Connections

**Theorem 6.19** (Generalization of BATC\textsuperscript{sat}I). 1. By the definitions of \( a = \int_{v \in [0, \infty)} \sigma_{v}^{\text{abs}}(\tilde{a}) \) for each \( a \in A \) and \( \delta = \int_{v \in [0, \infty)} \sigma_{v}^{\text{abs}}(\tilde{\delta}) \), BATC\textsuperscript{sat}I is a generalization of BATC.

2. BATC\textsuperscript{sat}I is a generalization of BATC\textsuperscript{sat}.

**Proof.** 1. It follows from the following two facts.
   (a) The transition rules of BATC in section 2.1 are all source-dependent;
   (b) The sources of the transition rules of BATC\textsuperscript{sat}I contain an occurrence of \( \tilde{\delta}, \tilde{a}, \sigma_{v}^{\text{abs}}, \upsilon_{v}^{\text{abs}}, \overline{\upsilon}^{\text{abs}} \) and \( \int \).

   So, BATC is an embedding of BATC\textsuperscript{sat}I, as desired.

2. It follows from the following two facts.
   (a) The transition rules of BATC\textsuperscript{sat} are all source-dependent;
   (b) The sources of the transition rules of BATC\textsuperscript{sat}I contain an occurrence of \( \int \).

   So, BATC\textsuperscript{sat} is an embedding of BATC\textsuperscript{sat}I, as desired.

\[ \square \]

6.3.4. Congruence

**Theorem 6.20** (Congruence of BATC\textsuperscript{sat}I). Truly concurrent bisimulation equivalences are all congruences with respect to BATC\textsuperscript{sat}I. That is,

- pomset bisimulation equivalence \( \sim_{p} \) is a congruence with respect to BATC\textsuperscript{sat}I;
- step bisimulation equivalence \( \sim_{s} \) is a congruence with respect to BATC\textsuperscript{sat}I;
- hp-bisimulation equivalence \( \sim_{hp} \) is a congruence with respect to BATC\textsuperscript{sat}I;
- hhp-bisimulation equivalence \( \sim_{hhp} \) is a congruence with respect to BATC\textsuperscript{sat}I.

**Proof.** It is easy to see that \( \sim_{p}, \sim_{s}, \sim_{hp} \) and \( \sim_{hhp} \) are all equivalent relations on BATC\textsuperscript{sat}I terms, it is only sufficient to prove that \( \sim_{p}, \sim_{s}, \sim_{hp} \) and \( \sim_{hhp} \) are all preserved by the operators \( \int \). It is trivial and we omit it.

\[ \square \]

6.3.5. Soundness

**Theorem 6.21** (Soundness of BATC\textsuperscript{sat}I). The axiomatization of BATC\textsuperscript{sat}I is sound modulo truly concurrent bisimulation equivalences \( \sim_{p}, \sim_{s}, \sim_{hp} \) and \( \sim_{hhp} \). That is,

1. let \( x \) and \( y \) be BATC\textsuperscript{sat}I terms. If BATC\textsuperscript{sat}I \( \vdash x = y \), then \( x \sim_{s} y \);
2. let \( x \) and \( y \) be BATC\textsuperscript{sat}I terms. If BATC\textsuperscript{sat}I \( \vdash x = y \), then \( x \sim_{p} y \);
3. let \( x \) and \( y \) be BATC\textsuperscript{sat}I terms. If BATC\textsuperscript{sat}I \( \vdash x = y \), then \( x \sim_{hp} y \);
4. let \( x \) and \( y \) be BATC\textsuperscript{sat}I terms. If BATC\textsuperscript{sat}I \( \vdash x = y \), then \( x \sim_{hhp} y \).

**Proof.** Since \( \sim_{p}, \sim_{s}, \sim_{hp} \) and \( \sim_{hhp} \) are both equivalent and congruent relations, we only need to check if each axiom in Table 38 is sound modulo \( \sim_{p}, \sim_{s}, \sim_{hp} \) and \( \sim_{hhp} \) respectively.

1. We can check the soundness of each axiom in Table 38 by the transition rules in Table 39 it is trivial and we omit them.
2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \( \cdot \)) or in concurrency (implicitly defined by \( \cdot \) and \( + \), and explicitly defined by \( \parallel \)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \( P = \{ \tilde{a}, \tilde{b} : \tilde{a} \parallel \tilde{b} \} \).
Then the pomset transition labeled by the above \( P \) is just composed of one single event transition labeled by \( \tilde{a} \) succeeded by another single event transition labeled by \( \tilde{b} \), that is, \( \frac{P}{\sim} \xrightarrow{a} \frac{P}{\sim} \xrightarrow{b} \).

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 38 is sound modulo pomset bisimulation equivalence, we omit them.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product \((C_1, f, C_2), f : C_1 \rightarrow C_2\) isomorphism. Two process terms \( s \) related to \( C_1 \) and \( t \) related to \( C_2 \), and \( f : C_1 \rightarrow C_2 \) isomorphism. Initially, \((C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)\), and \((\emptyset, \emptyset, \emptyset) \equiv_{hp}\). When \( s \xrightarrow{a} s' \) \((C_1 \xrightarrow{a} C_1')\), there will be \( t \xrightarrow{a} t' \) \((C_2 \xrightarrow{a} C_2')\), and we define \( f' = f[a \rightarrow a] \). Then, if \((C_1, f, C_2) \equiv_{hp}\), then \((C_1', f', C_2') \equiv_{hp}\).

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 38 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

4. We just need to add downward-closed condition to the soundness modulo hp-bisimulation equivalence, we omit them.

\[ \square \]

6.3.6. Completeness

**Theorem 6.22 (Completeness of BATC\textsc{sat}I).** The axiomatization of BATC\textsc{sat}I is complete modulo truly concurrent bisimulation equivalences \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hp} \). That is,

1. let \( p \) and \( q \) be closed BATC\textsc{sat}I terms, if \( p \sim_s q \) then \( p = q \);
2. let \( p \) and \( q \) be closed BATC\textsc{sat}I terms, if \( p \sim_p q \) then \( p = q \);
3. let \( p \) and \( q \) be closed BATC\textsc{sat}I terms, if \( p \sim_{hp} q \) then \( p = q \);
4. let \( p \) and \( q \) be closed BATC\textsc{sat}I terms, if \( p \sim_{hp} q \) then \( p = q \).

**Proof.** 1. Firstly, by the elimination theorem of BATC\textsc{sat}I, we know that for each closed BATC\textsc{sat}I term \( p \), there exists a closed basic BATC\textsc{sat}I term \( p' \), such that BATC\textsc{sat}I \( p = p' \), so, we only need to consider closed basic BATC\textsc{sat}I terms.

   The basic terms modulo associativity and commutativity (AC) of conflict + (defined by axioms A1 and A2 in Table 36), and this equivalence is denoted by \( =_{AC} \). Then, each equivalence class \( s \) modulo AC of + has the following normal form

\[
\sum_{i=1}^{k} s_i + \cdots + s_k
\]

with each \( s_i \) either an atomic event or of the form \( t_1 \cdot t_2 \), and each \( s_i \) is called the summand of \( s \).

Now, we prove that for normal forms \( n \) and \( n' \), if \( n \sim_s n' \) then \( n =_{AC} n' \). It is sufficient to induct on the sizes of \( n \) and \( n' \). We can get \( n =_{AC} n' \). Finally, let \( s \) and \( t \) be basic terms, and \( s \sim_s t \), there are normal forms \( n \) and \( n' \), such that \( s = n \) and \( t = n' \). The soundness theorem of BATC\textsc{sat}I modulo step bisimulation equivalence yields \( s \sim_s n \) and \( t \sim_s n' \), so \( n \sim_s n' \sim_s n' \). Since if \( n \sim_s n' \) then \( n =_{AC} n' \), \( s = n =_{AC} n' = t \), as desired.

2. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_p \).
3. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_{hp} \).
4. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_{hp} \).

\[ \square \]

6.4. Algebra for Parallelism in True Concurrency with Continuous Absolute Timing

In this subsection, we will introduce APTC\textsc{sat}.

6.4.1. Basic Definition

**Definition 6.23 (Absolute undelayable time-out).** The relative undelayable time-out \( \nu_{abs} \) of a process \( p \) behaves like the part of \( p \) that starts to perform actions at the point of time 0 if \( p \) is capable of performing actions at point of time 0; otherwise, like undelayable deadlock. And let \( \nu_{abs}^r(t) = r \nu_{abs} \).
6.4.2. The Theory APTC\textsuperscript{sat}

**Definition 6.24** (Signature of APTC\textsuperscript{sat}). The signature of APTC\textsuperscript{sat} consists of the signature of BATC\textsuperscript{sat}, and the whole parallel composition operator $\parallel : \mathcal{P}_{\text{abs}} \times \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}}$, the parallel operator $\parallel : \mathcal{P}_{\text{abs}} \times \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}}$, the communication merger operator $\mathcal{P}_{\text{abs}} \times \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}}$, the encapsulation operator $\partial_H : \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}}$ for all $H \subseteq A$, and the absolute undelayable time-out operator $\nu_{\text{abs}} : \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}}$.

The set of axioms of APTC\textsuperscript{sat} consists of the laws given in Table 40.

The operational semantics of APTC\textsuperscript{sat} are defined by the transition rules in Table 41.

6.4.3. Elimination

**Definition 6.25** (Basic terms of APTC\textsuperscript{sat}). The set of basic terms of APTC\textsuperscript{sat}, $\mathcal{B}(\text{APTC}_{\text{sat}})$, is inductively defined as follows by two auxiliary sets $\mathcal{B}_0(\text{APTC}_{\text{sat}})$ and $\mathcal{B}_1(\text{APTC}_{\text{sat}})$:

1. if $a \in A$, then $\bar{a} \in \mathcal{B}_1(\text{APTC}_{\text{sat}})$;
2. if $a \in A$ and $t \in \mathcal{B}(\text{APTC}_{\text{sat}})$, then $\bar{a} \cdot t \in \mathcal{B}_1(\text{APTC}_{\text{sat}})$;
3. if $t, t' \in \mathcal{B}_1(\text{APTC}_{\text{sat}})$, then $t + t' \in \mathcal{B}_1(\text{APTC}_{\text{sat}})$;
4. if $t, t' \in \mathcal{B}_1(\text{APTC}_{\text{sat}})$, then $t \parallel t' \in \mathcal{B}_1(\text{APTC}_{\text{sat}})$;
5. if $t \in \mathcal{B}_1(\text{APTC}_{\text{sat}})$, then $t \in \mathcal{B}_0(\text{APTC}_{\text{sat}})$;
6. if $p > 0$ and $t \in \mathcal{B}_0(\text{APTC}_{\text{sat}})$, then $\sigma^p_{\text{abs}}(t) \in \mathcal{B}_0(\text{APTC}_{\text{sat}})$;
7. if $p > 0$, $t \in \mathcal{B}_1(\text{APTC}_{\text{sat}})$ and $t' \in \mathcal{B}_0(\text{APTC}_{\text{sat}})$, then $t + \sigma^p_{\text{abs}}(t') \in \mathcal{B}_0(\text{APTC}_{\text{sat}})$;
8. if $t \in \mathcal{B}_0(\text{APTC}_{\text{sat}})$, then $\nu_{\text{abs}}(t) \in \mathcal{B}_0(\text{APTC}_{\text{sat}})$;
9. if $t \in \mathcal{B}_0(\text{APTC}_{\text{sat}})$, then $t \in \mathcal{B}(\text{APTC}_{\text{sat}})$;
10. if $t \in \mathcal{B}_0(\text{APTC}_{\text{sat}})$, then $t \in \mathcal{B}(\text{APTC}_{\text{sat}})$.

**Theorem 6.26** (Elimination theorem). Let $p$ be a closed APTC\textsuperscript{sat} term. Then there is a basic APTC\textsuperscript{sat} term $q$ such that $\text{APTC}_{\text{sat}} \vdash p \Rightarrow q$.

**Proof.** It is sufficient to induct on the structure of the closed APTC\textsuperscript{sat} term $p$. It can be proven that $p$ combined by the constants and operators of APTC\textsuperscript{sat} exists an equal basic term $q$, and the other operators not included in the basic terms, such as $\nu_{\text{abs}}, \bar{\nu}_{\text{abs}}, \dot{\nu}_{\text{abs}}, \parallel, |, \partial_H, \Theta$, and $\angle$ can be eliminated.

6.4.4. Connections

**Theorem 6.27** (Generalization of APTC\textsuperscript{sat}). 1. By the definitions of $a = \bar{a}$ for each $a \in A$ and $\delta = \bar{\delta}$, $\text{APTC}_{\text{sat}}$ is a generalization of APTC.

2. $\text{APTC}_{\text{sat}}$ is a generalization of BATC\textsuperscript{sat}

**Proof.** 1. It follows from the following two facts.

(a) The transition rules of APTC in section 6.4.1 are all source-dependent;

(b) The sources of the transition rules of APTC\textsuperscript{sat} contain an occurrence of $\bar{\delta}, \bar{\bar{a}}, \sigma^p_{\text{abs}}, \nu^p_{\text{abs}}, \bar{\nu}_{\text{abs}}$, and $\nu_{\text{abs}}$.

So, APTC is an embedding of APTC\textsuperscript{sat}, as desired.

2. It follows from the following two facts.

(a) The transition rules of BATC\textsuperscript{sat} are all source-dependent;

(b) The sources of the transition rules of APTC\textsuperscript{sat} contain an occurrence of $\dot{\delta}, \parallel, |, \Theta, \angle, \partial_H$, and $\nu_{\text{abs}}$.

So, BATC\textsuperscript{sat} is an embedding of APTC\textsuperscript{sat}, as desired.
| No. | Axiom |
|-----|-------|
| P1  | \( x \upharpoonright y = x \parallel y + x \upharpoonright y \) |
| P2  | \( x \parallel y = y \parallel x \) |
| P3  | \( (x \parallel y) \parallel z = x \parallel (y \parallel z) \) |
| P4SA | \( \tilde{a} \upharpoonright (b \cdot y) = (\tilde{a} \parallel b) \cdot y \) |
| P5SA | \( (\tilde{a} \cdot x) \parallel b = (\tilde{a} \parallel b) \cdot x \) |
| P6SA | \( (\tilde{a} \cdot x) \parallel (b \cdot y) = (\tilde{a} \parallel b) \cdot (x \parallel y) \) |
| P7  | \( x \parallel y = x \parallel z \parallel (y \parallel z) \) |
| P8  | \( (x \parallel y) \parallel z = x \parallel (z \parallel (y \parallel z)) \) |
| SAP9ID | \( \nu_{\text{abs}}(x) \parallel \delta = \sigma_{\text{abs}}(y) = \delta \) |
| SAP10ID | \( \sigma_{\text{abs}}^p(x) \parallel \nu_{\text{abs}}(y) + \delta = \delta \) |
| SAP11 | \( \sigma_{\text{abs}}^p(y) \parallel \sigma_{\text{abs}}^p(x) = \sigma_{\text{abs}}^p(x \parallel y) \) |
| PID12 | \( \delta \parallel x = \delta \) |
| PID13 | \( x \parallel \delta = \delta \) |
| C14SA | \( \tilde{a} \parallel b = \gamma(\tilde{a}, b) \) |
| C15SA | \( \tilde{a} \parallel (b \cdot y) = \gamma(\tilde{a}, b) \cdot y \) |
| C16SA | \( (\tilde{a} \cdot x) \parallel b = \gamma(\tilde{a}, b) \cdot x \) |
| C17SA | \( (\tilde{a} \cdot x) \parallel (b \cdot y) = \gamma(\tilde{a}, b) \cdot (x \parallel y) \) |
| C18 | \( (x \parallel y) \parallel z = (x \parallel z) + (y \parallel z) \) |
| C19 | \( x \parallel (y \parallel z) = (x \parallel y) + (x \parallel z) \) |
| SAC20ID | \( (\nu_{\text{abs}}(x) + \delta) \parallel \sigma_{\text{abs}}^p(y) = \delta \) |
| SAC21ID | \( \nu_{\text{abs}}(x) \parallel (\nu_{\text{abs}}(y) + \delta) = \delta \) |
| SAC22 | \( \sigma_{\text{abs}}^p(x) \parallel \sigma_{\text{abs}}^p(y) = \sigma_{\text{abs}}^p(x \parallel y) \) |
| CID23 | \( \delta \parallel \delta = \delta \) |
| CID24 | \( x \parallel \delta = \delta \) |
| CE25SA | \( \Theta(\tilde{a}) = \tilde{a} \) |
| CE26SAID | \( \Theta(\tilde{b}) = \tilde{b} \) |
| CE27 | \( \Theta(x + y) = \Theta(x) \upharpoonright y + \Theta(y) \upharpoonright x \) |
| CE28 | \( \Theta(x \cdot y) = \Theta(x) \cdot \Theta(y) \) |
| CE29 | \( \Theta(x \parallel y) = ((\Theta(x) \parallel y) \parallel y) + ((\Theta(y) \parallel x) \parallel x) \) |
| CE30 | \( \Theta(x \parallel y) = ((\Theta(x) \parallel y) \parallel y) + ((\Theta(y) \parallel x) \parallel x) \) |
| U31SAID | \( (\tilde{a}, \tilde{b}) \parallel \tilde{a} \parallel \tilde{b} = \tilde{f} \) |
| U32SAID | \( (\tilde{a} \parallel \tilde{b}, \tilde{b} \parallel \tilde{c}) \parallel \tilde{a} \parallel \tilde{c} = \tilde{a} \) |
| U33SAID | \( (\tilde{a}, \tilde{b} \parallel \tilde{c}, \tilde{b} \parallel \tilde{c}) \parallel \tilde{c} \parallel \tilde{a} = \tilde{f} \) |
| U34SAID | \( \tilde{a} \parallel \tilde{a} = \tilde{a} \) |
| U35SAID | \( \delta \parallel \tilde{a} = \tilde{a} \) |
| U36 | \( (x + y) \parallel z = (x \parallel z) + (y \parallel z) \) |
| U37 | \( (x \cdot y) \parallel z = (x \parallel z) \cdot (y \parallel z) \) |
| U38 | \( (x \parallel y) \parallel z = (x \parallel z) \parallel (y \parallel z) \) |
| U39 | \( (x \parallel y) \parallel z = (x \parallel z) \parallel (y \parallel z) \) |
| U40 | \( x \parallel (y \parallel z) = (x \parallel y) \parallel z \) |
| U41 | \( x \parallel (y \parallel z) = (x \parallel y) \parallel z \) |
| U42 | \( x \parallel (y \parallel z) = (x \parallel y) \parallel z \) |
| U43 | \( x \parallel (y \parallel z) = (x \parallel y) \parallel z \) |
| D1SAID | \( \tilde{a} \in H \parallel \tilde{H}(\tilde{a}) = \tilde{a} \) |
| D2SAID | \( \tilde{a} \in H \parallel \tilde{H}(\tilde{a}) = \tilde{a} \) |
| D3SAID | \( \tilde{H}(\delta) = \delta \) |
| 4  | \( \partial_H(x + y) = \partial_H(x) + \partial_H(y) \) |
| D5  | \( \partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y) \) |
| D6 | \( \partial_H(x \parallel y) = \partial_H(x) \parallel \partial_H(y) \) |
| SAD7 | \( \partial_H(\sigma_{\text{abs}}^p(x)) = \sigma_{\text{abs}}^p(\partial_H(x)) \) |
| SAU0 | \( \nu_{\text{abs}}(\delta) = \delta \) |
| SAU1 | \( \nu_{\text{abs}}(\tilde{a}) = \tilde{a} \) |
| SAU2 | \( \nu_{\text{abs}}(\sigma_{\text{abs}}^p(x)) = \delta \) |
| SAU3 | \( \nu_{\text{abs}}(x + y) = \nu_{\text{abs}}(x) + \nu_{\text{abs}}(y) \) |
| SAU4 | \( \nu_{\text{abs}}(x \parallel y) = \nu_{\text{abs}}(x) \parallel \nu_{\text{abs}}(y) \) |
| SAU5 | \( \nu_{\text{abs}}(x \parallel y) = \nu_{\text{abs}}(x) \parallel \nu_{\text{abs}}(y) \) |

Table 40. Axioms of APTC \(^{\text{sa}}(a, b, c \in A_\delta, p \geq 0, r > 0)\)
Table 41. Transition rules of \text{APTC}^\text{sat}(a, b, c \in a, p \geq 0, r > 0)
6.4.5. Congruence

**Theorem 6.28** (Congruence of $\text{APTC}^{\text{sat}}$). Truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$, and $\sim_{hp}$ are all congruences with respect to $\text{APTC}^{\text{sat}}$. That is,

- pomset bisimulation equivalence $\sim_p$ is a congruence with respect to $\text{APTC}^{\text{sat}}$;
- step bisimulation equivalence $\sim_s$ is a congruence with respect to $\text{APTC}^{\text{sat}}$;
- hp-bisimulation equivalence $\sim_{hp}$ is a congruence with respect to $\text{APTC}^{\text{sat}}$.

**Proof.** It is easy to see that $\sim_p$, $\sim_s$, and $\sim_{hp}$ are all equivalent relations on $\text{APTC}^{\text{sat}}$ terms, it is only sufficient to prove that $\sim_p$, $\sim_s$, and $\sim_{hp}$ are all preserved by the operators $\sigma^p_{\text{abs}}$, $\nu^p_{\text{abs}}$, $\nu^q_{\text{abs}}$, and $\nu_{\text{abs}}$. It is trivial and we omit it. □

6.4.6. Soundness

**Theorem 6.29** (Soundness of $\text{APTC}^{\text{sat}}$). The axiomatization of $\text{APTC}^{\text{sat}}$ is sound modulo truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$, and $\sim_{hp}$. That is,

1. let $x$ and $y$ be $\text{APTC}^{\text{sat}}$ terms. If $\text{APTC}^{\text{sat}} \vdash x = y$, then $x \sim_s y$;
2. let $x$ and $y$ be $\text{APTC}^{\text{sat}}$ terms. If $\text{APTC}^{\text{sat}} \vdash x = y$, then $x \sim_p y$;
3. let $x$ and $y$ be $\text{APTC}^{\text{sat}}$ terms. If $\text{APTC}^{\text{sat}} \vdash x = y$, then $x \sim_{hp} y$.

**Proof.** Since $\sim_p$, $\sim_s$, and $\sim_{hp}$ are both equivalent and congruent relations, we only need to check if each axiom in Table 41 is sound modulo $\sim_p$, $\sim_s$, and $\sim_{hp}$ respectively.

1. We only check the soundness of the non-trivial axiom SAP11 modulo $\sim_s$. Let $p, q$ be $\text{APTC}^{\text{dat}}$ processes, and $\sigma^s_{\text{abs}}(p) \parallel \sigma^s_{\text{abs}}(q) = \sigma^s_{\text{abs}}(p \parallel q)$, it is sufficient to prove that $\sigma^s_{\text{abs}}(p) \parallel \sigma^s_{\text{abs}}(q) \sim_s \sigma^s_{\text{abs}}(p \parallel q)$. By the transition rules of operator $\sigma^s_{\text{abs}}$ and $\parallel$ in Table 41 we get

$$\frac{}{(p, 0) \uparrow}$$

$$\frac{(\sigma^s_{\text{abs}}(p) \parallel \sigma^s_{\text{abs}}(q), s') \xrightarrow{\sigma^s_{\text{abs}}(p) \parallel \sigma^s_{\text{abs}}(q), s' + s}}{(p, 0) \uparrow}$$

There are several cases:

$$\frac{(p, s') \xrightarrow{a} (\sqrt{s'}, q, s') \xrightarrow{b} (\sqrt{s'}, s')}{(\sigma^s_{\text{abs}}(p) \parallel \sigma^s_{\text{abs}}(q), s' + s) \xrightarrow{a, b} (\sqrt{s'}, s' + s)}$$

$$\frac{(p, s') \xrightarrow{a} (\sqrt{s'}, q, s') \xrightarrow{b} (\sqrt{s'}, s')}{(\sigma^s_{\text{abs}}(p) \parallel \sigma^s_{\text{abs}}(q), s' + s) \xrightarrow{a, b} (\sqrt{s'}, s' + s)}$$

$$\frac{(p, s') \xrightarrow{a} (p', s') \xrightarrow{b} (\sqrt{s'}, s')}{(\sigma^s_{\text{abs}}(p) \parallel \sigma^s_{\text{abs}}(q), s' + s) \xrightarrow{a, b} (\sigma^s_{\text{abs}}(p'), s' + s)}$$

$$\frac{(p, s') \xrightarrow{a} (p', s') \xrightarrow{b} (p', s' + s) \xrightarrow{a, b} (\sigma^s_{\text{abs}}(p'), s' + s)}{(\sigma^s_{\text{abs}}(p) \parallel \sigma^s_{\text{abs}}(q), s' + s) \xrightarrow{a, b} (\sigma^s_{\text{abs}}(q'), s' + s)}$$

$$\frac{(p, s') \xrightarrow{a} (\sqrt{s'}, q, s') \xrightarrow{b} (\sqrt{s'}, s')}{(\sigma^s_{\text{abs}}(p) \parallel \sigma^s_{\text{abs}}(q), s' + s) \xrightarrow{a, b} (\sigma^s_{\text{abs}}(q'), s' + s)}$$
3. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \( \cdot \)) or in concurrency (implicitly defined by \( + \)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \( P = \{ \tilde{a}, \tilde{b} : \tilde{a} \cdot \tilde{b} \} \). Then the pomset transition labeled by the above \( P \) is just composed of one single event transition labeled by \( \tilde{a} \) succeeded by another single event transition labeled by \( \tilde{b} \), that is, \( \tilde{a} \rightarrow \tilde{b} \). Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 44 is sound modulo pomset bisimulation equivalence, we omit them.

2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \( \cdot \)) or in concurrency (implicitly defined by \( + \)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \( P = \{ \tilde{a}, \tilde{b} : \tilde{a} \cdot \tilde{b} \} \). Then the pomset transition labeled by the above \( P \) is just composed of one single event transition labeled by \( \tilde{a} \) succeeded by another single event transition labeled by \( \tilde{b} \), that is, \( \tilde{a} \rightarrow \tilde{b} \). Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 44 is sound modulo pomset bisimulation equivalence, we omit them.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product \((C_1, f, C_2), f : C_1 \rightarrow C_2 \) isomorphism. Two process terms \( s \) related to \( C_1 \) and \( t \) related to \( C_2 \) and \( f : C_1 \rightarrow C_2 \) isomorphism. Initially, \((C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)\), and \((\emptyset, \emptyset, \emptyset) \in_{hp} \). When \( s \xrightarrow{a} s' \) \( (C_1, f, C_2) \rightarrow (C_1', f', C_2') \), there will be \( t \xrightarrow{a} t' \) \( (C_2, a) \xrightarrow{a} (C_2', a) \), and we define \( f' = f[a \mapsto a] \). Then, if \( (C_1, f, C_2) \in_{hp} \), then \( (C_1', f', C_2') \in_{hp} \). Similarly to the proof of soundness modulo pemset bisimulation equivalence, we can prove that each axiom in Table 44 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

\[ \frac{\langle p, s' \rangle \xrightarrow{a} \langle q, s' \rangle \quad \langle q, s' \rangle \xrightarrow{b} \langle q', s' \rangle}{\langle p \parallel q, s' + s \rangle \xrightarrow{(a,b)} \langle p \parallel q, s' + s \rangle} \]

\[ \frac{\langle p, s' \rangle \xrightarrow{a} \langle q, s' \rangle \quad \langle q, s' \rangle \xrightarrow{b} \langle q', s' \rangle}{\langle p \parallel q, s' + s \rangle \xrightarrow{(a,b)} \langle p \parallel q, s' + s \rangle} \]

\[ \frac{\langle p, s' \rangle \xrightarrow{a} \langle q, s' \rangle \quad \langle q, s' \rangle \xrightarrow{b} \langle q', s' \rangle}{\langle p \parallel q, s' + s \rangle \xrightarrow{(a,b)} \langle p \parallel q, s' + s \rangle} \]

\[ \frac{\langle p, s' \rangle \xrightarrow{a} \langle q, s' \rangle \quad \langle q, s' \rangle \xrightarrow{b} \langle q', s' \rangle}{\langle p \parallel q, s' + s \rangle \xrightarrow{(a,b)} \langle p \parallel q, s' + s \rangle} \]

So, we see that each case leads to \( \sigma^s_{abs}(p) \parallel \sigma^s_{abs}(q) = \sigma^s_{abs}(p \parallel q) \), as desired.

6.4.7. Completeness

**Theorem 6.30** (Completeness of APTC\textsuperscript{sat}). The axiomatization of APTC\textsuperscript{sat} is complete modulo truly concurrent bisimulation equivalences \( \sim_p, \sim_s, \) and \( \sim_{hp} \). That is,

1. let \( p \) and \( q \) be closed APTC\textsuperscript{sat} terms, if \( p \sim_s q \) then \( p = q \);
2. let \( p \) and \( q \) be closed APTC\textsuperscript{sat} terms, if \( p \sim_p q \) then \( p = q \);
3. let \( p \) and \( q \) be closed APTC\textsuperscript{sat} terms, if \( p \sim_{hp} q \) then \( p = q \).
Table 42. Axioms of APTC

| No. | Axiom |
|-----|-------|
| INT13 | \( \int_{e \in V} (F(v) \parallel x) = (\int_{e \in V} F(v)) \parallel x \) |
| INT14 | \( \int_{e \in V} (x \parallel F(v)) = x \parallel (\int_{e \in V} F(v)) \) |
| INT15 | \( \int_{e \in V} F(v) \parallel x = (\int_{e \in V} F(v)) \parallel x \) |
| INT16 | \( \int_{e \in V} (x \parallel F(v)) = x \parallel (\int_{e \in V} F(v)) \) |
| INT17 | \( \partial_\mu (F(v)) = \partial_\mu (\int_{e \in V} F(v)) \) |
| INT18 | \( \int_{e \in V} \Theta(F(v)) = \Theta(\int_{e \in V} F(v)) \) |
| INT19 | \( \int_{e \in V} (F(v) \preceq x) = (\int_{e \in V} F(v)) \preceq x \) |
| SAU5 | \( v_{abs}(\int_{e \in V} F(v)) = \int_{e \in V} v_{abs}(F(v)) \) |

The set of axioms of APTC\(_{\text{sat}}\) is defined as follows by two auxiliary sets \( B_0(\text{APTC}_{\text{sat}}) \) and \( B_1(\text{APTC}_{\text{sat}}) \):

**Proof.**

1. Firstly, by the elimination theorem of APTC\(_{\text{sat}}\), we know that for each closed APTC\(_{\text{sat}}\) term \( p \), there exists a closed basic APTC\(_{\text{sat}}\) term \( p' \), such that APTC\(_{\text{sat}}\) \( \vdash p = p' \), so, we only need to consider closed basic APTC\(_{\text{sat}}\) terms.

The basic terms modulo associativity and commutativity (AC) of conflict + (defined by axioms A1 and A2 in Table 24) and associativity and commutativity (AC) of parallel || (defined by axioms P2 and P3 in Table 26), and these equivalences is denoted by =\(_{AC}\). Then, each equivalence class \( s \) modulo AC of + and || has the following normal form

\[
S_1 + \cdots + S_k
\]

with each \( S_i \) either an atomic event or of the form

\[
T_1 \cdot \cdots \cdot T_m
\]

with each \( T_j \) either an atomic event or of the form

\[
U_1 \parallel \cdots \parallel U_n
\]

with each \( U_i \) an atomic event, and each \( S_i \) is called the summand of \( S \).

Now, we prove that for normal forms \( n \) and \( n' \), if \( n \sim_s n' \) then \( n =_{AC} n' \). It is sufficient to induct on the sizes of \( n \) and \( n' \). We can get \( n =_{AC} n' \).

Finally, let \( s \) and \( t \) be basic APTC\(_{\text{sat}}\) terms, and \( s \sim s \), there are normal forms \( n \) and \( n' \), such that \( s = n \) and \( t = n' \). The soundness theorem modulo step bisimulation equivalence yields \( s \sim_s n \) and \( t \sim_s n' \), so \( n \sim_s t \sim_s n' \). Since if \( n \sim_s n' \) then \( n =_{AC} n' \), \( s \sim_{AC} n =_{AC} n' \), as desired.

2. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_p \).

3. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_{hp} \).

\[ \square \]

### 6.5. APTC\(_{\text{sat}}\) with Integration

In this subsection, we will introduce the theory APTC\(_{\text{sat}}\) with integration called APTC\(_{\text{sat}}\)I.

#### 6.5.1. The Theory APTC\(_{\text{sat}}\)I

**Definition 6.31** (Signature of APTC\(_{\text{sat}}\)I). The signature of APTC\(_{\text{sat}}\)I consists of the signature of APTC\(_{\text{sat}}\) and the integration operator \( \int : \mathcal{P}(\mathbb{R}^2) \times \mathbb{R}^2 . \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}} \).

The set of axioms of APTC\(_{\text{sat}}\)I consists of the laws given in Table 42.

The operational semantics of APTC\(_{\text{sat}}\)I are defined by the transition rules in Table 39.

#### 6.5.2. Elimination

**Definition 6.32** (Basic terms of APTC\(_{\text{sat}}\)I). The set of basic terms of APTC\(_{\text{sat}}\)I, \( B(\text{APTC}_{\text{sat}}) \), is inductively defined as follows by two auxiliary sets \( B_0(\text{APTC}_{\text{sat}}) \) and \( B_1(\text{APTC}_{\text{sat}}) \):

\[
B_0(\text{APTC}_{\text{sat}}) = \{ \text{APTC}_{\text{sat}}\text{-term} \} \setminus \{ \text{basic term} \}
\]

\[
B_1(\text{APTC}_{\text{sat}}) = \{ \text{APTC}_{\text{sat}}\text{-term} \} \setminus \{ \text{basic term} \}
\]
1. if \( a \in A \), then \( \bar{a} \in B_1(\text{APTC}^{\text{sat}1}) \);
2. if \( a \in A \) and \( t \in B(\text{APTC}^{\text{sat}1}) \), then \( \bar{a} \cdot t \in B_1(\text{APTC}^{\text{sat}1}) \);
3. if \( t, t' \in B_1(\text{APTC}^{\text{sat}1}) \), then \( t + t' \in B_1(\text{APTC}^{\text{sat}1}) \);
4. if \( t, t' \in B_1(\text{APTC}^{\text{sat}1}) \), then \( t \parallel t' \in B_1(\text{APTC}^{\text{sat}1}) \);
5. if \( t \in B_1(\text{APTC}^{\text{sat}1}) \), then \( t \in B_0(\text{APTC}^{\text{sat}1}) \);
6. if \( p > 0 \) and \( t \in B_0(\text{APTC}^{\text{sat}1}) \), then \( \sigma^p_{\text{abs}}(t) \in B_0(\text{APTC}^{\text{sat}1}) \);
7. if \( p > 0 \), \( t \in B_1(\text{APTC}^{\text{sat}1}) \) and \( t' \in B_0(\text{APTC}^{\text{sat}1}) \), then \( t + \sigma^p_{\text{abs}}(t') \in B_0(\text{APTC}^{\text{sat}1}) \);
8. if \( t \in B_0(\text{APTC}^{\text{sat}1}) \), then \( \nu_{\text{abs}}(t) \in B_0(\text{APTC}^{\text{sat}1}) \);
9. if \( t \in B_0(\text{APTC}^{\text{sat}1}) \), then \( \int_{v \in V} t \in B_0(\text{APTC}^{\text{sat}1}) \);
10. \( \delta \in B(\text{APTC}^{\text{sat}1}) \);
11. if \( t \in B_0(\text{APTC}^{\text{sat}1}) \), then \( t \in B(\text{APTC}^{\text{sat}1}) \).

**Theorem 6.33** (Elimination theorem). Let \( p \) be a closed \( \text{APTC}^{\text{sat}1} \) term. Then there is a basic \( \text{APTC}^{\text{sat}1} \) term \( q \) such that \( \text{APTC}^{\text{sat}1} \vdash p = q \).

**Proof.** It is sufficient to induct on the structure of the closed \( \text{APTC}^{\text{sat}1} \) term \( p \). It can be proven that \( p \) combined by the constants and operators of \( \text{APTC}^{\text{sat}1} \) exists an equal basic term \( q \), and the other operators not included in the basic terms, such as \( v_{\text{abs}}, \bar{v} \), \( \nu_{\text{abs}} \), \( \partial_H \), \( \Theta \) and \( < \) can be eliminated. \( \square \)

### 6.5.3. Connections

**Theorem 6.34** (Generalization of \( \text{APTC}^{\text{sat}1} \)). 1. By the definitions of \( a = \int_{v \in [0, \infty)} \sigma^v_{\text{abs}}(\bar{a}) \) for each \( a \in A \) and \( \delta = \int_{v \in [0, \infty)} \sigma^v_{\text{abs}}(\bar{\delta}) \), \( \text{APTC}^{\text{sat}1} \) is a generalization of \( \text{APTC} \).

2. \( \text{APTC}^{\text{sat}1} \) is a generalization of \( \text{APTC}^{\text{sat}} \).

**Proof.** 1. It follows from the following two facts.

(a) The transition rules of \( \text{APTC} \) in section 2.1 are all source-dependent;
(b) The sources of the transition rules of \( \text{APTC}^{\text{sat}1} \) contain an occurrence of \( \bar{\delta}, \bar{a}, \sigma^p_{\text{abs}}, v^p_{\text{abs}}, \bar{p}^p_{\text{abs}}, \nu_{\text{abs}} \), and \( \int \).

So, \( \text{APTC} \) is an embedding of \( \text{APTC}^{\text{sat}1} \), as desired.

2. It follows from the following two facts.

(a) The transition rules of \( \text{APTC}^{\text{sat}1} \) are all source-dependent;
(b) The sources of the transition rules of \( \text{APTC}^{\text{sat}1} \) contain an occurrence of \( \int \).

So, \( \text{APTC}^{\text{sat}} \) is an embedding of \( \text{APTC}^{\text{sat}1} \), as desired. \( \square \)

### 6.5.4. Congruence

**Theorem 6.35** (Congruence of \( \text{APTC}^{\text{sat}1} \)). Truly concurrent bisimulation equivalences are all congruences with respect to \( \text{APTC}^{\text{sat}1} \). That is,

- pomset bisimulation equivalence \( \sim_p \) is a congruence with respect to \( \text{APTC}^{\text{sat}1} \);
- step bisimulation equivalence \( \sim_s \) is a congruence with respect to \( \text{APTC}^{\text{sat}1} \);
- \( \text{hp} \)-bisimulation equivalence \( \sim_{hp} \) is a congruence with respect to \( \text{APTC}^{\text{sat}1} \).

**Proof.** It is easy to see that \( \sim_p, \sim_s, \sim_{hp} \) and \( \sim_{hhh} \) are all equivalent relations on \( \text{APTC}^{\text{sat}1} \) terms, it is only sufficient to prove that \( \sim_p, \sim_s, \) and \( \sim_{hp} \) are all preserved by the operators \( \int \). It is trivial and we omit it. \( \square \)
6.5.5. Soundness

**Theorem 6.36** (Soundness of $\text{APTC}^{\text{sat}}I$). The axiomatization of $\text{APTC}^{\text{sat}}I$ is sound modulo truly concurrent bisimulation equivalences $\sim_p, \sim_s, \sim_{hp}$ and $\sim_{hhp}$. That is,

1. let $x$ and $y$ be $\text{APTC}^{\text{sat}}I$ terms. If $\text{APTC}^{\text{sat}}I \vdash x = y$, then $x \sim_s y$;
2. let $x$ and $y$ be $\text{APTC}^{\text{sat}}I$ terms. If $\text{APTC}^{\text{sat}}I \vdash x = y$, then $x \sim_p y$;
3. let $x$ and $y$ be $\text{APTC}^{\text{sat}}I$ terms. If $\text{APTC}^{\text{sat}}I \vdash x = y$, then $x \sim_{hp} y$;

**Proof.** Since $\sim_p, \sim_s, \sim_{hp}$ and $\sim_{hhp}$ are both equivalent and congruent relations, we only need to check if each axiom in Table 42 is sound modulo $\sim_p, \sim_s$, and $\sim_{hp}$ respectively.

1. We can check the soundness of each axiom in Table 42 by the transition rules in Table 32, it is trivial and we omit them.
2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by $\cdot$) or in concurrency (implicitly defined by $+$, and explicitly defined by $\parallel$), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of $P = \{a, b : a \parallel b\}$. Then the pomset transition labeled by the above $P$ is just composed of one single event transition labeled by $a$ succeeded by another single event transition labeled by $b$, that is, $\frac{P}{\rightarrow} a \rightarrow b$. Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 42 is sound modulo pomset bisimulation equivalence, we omit them.
3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product $(C_1, f, C_2), f : C_1 \rightarrow C_2$ isomorphism. Two process terms $s$ related to $C_1$ and $t$ related to $C_2$, and $f : C_1 \rightarrow C_2$ isomorphism. Initially, $(C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)$, and $(\emptyset, \emptyset, \emptyset) \in \sim_{hp}$. Then the process transition labeled by the above $P$ is just composed of one single event transition labeled by $a$ succeeded by another single event transition labeled by $b$, that is, $\frac{P}{\rightarrow} a \rightarrow b$. Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 42 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.
4. We just need to add downward-closed condition to the soundness modulo hp-bisimulation equivalence, we omit them.

6.5.6. Completeness

**Theorem 6.37** (Completeness of $\text{APTC}^{\text{sat}}I$). The axiomatization of $\text{APTC}^{\text{sat}}I$ is complete modulo truly concurrent bisimulation equivalences $\sim_p, \sim_s, \sim_{hp}$ and $\sim_{hhp}$. That is,

1. let $p$ and $q$ be closed $\text{APTC}^{\text{sat}}I$ terms, if $p \sim_p q$ then $p = q$;
2. let $p$ and $q$ be closed $\text{APTC}^{\text{sat}}I$ terms, if $p \sim_p q$ then $p = q$;
3. let $p$ and $q$ be closed $\text{APTC}^{\text{sat}}I$ terms, if $p \sim_{hp} q$ then $p = q$;

**Proof.** 1. Firstly, by the elimination theorem of $\text{APTC}^{\text{sat}}I$, we know that for each closed $\text{APTC}^{\text{sat}}I$ term $p$, there exists a closed basic $\text{APTC}^{\text{sat}}I$ term $p'$, such that $\text{APTC}^{\text{sat}}I \vdash p = p'$, so, we only need to consider closed basic $\text{APTC}^{\text{sat}}I$ terms.

The basic terms modulo associativity and commutativity (AC) of conflict $+$ (defined by axioms $A1$ and $A2$ in Table 36) and associativity and commutativity (AC) of parallel $\parallel$ (defined by axioms $P2$ and $P3$ in Table 41), and these equivalences is denoted by $\sim_{AC}$. Then, each equivalence class $s$ modulo AC of $+$ and $\parallel$ has the following normal form

$$s_1 + \cdots + s_k$$

with each $s_i$ either an atomic event or of the form

$$\cdot s_1$$
\[ t_1 \cdots t_m \]

with each \( t_j \) either an atomic event or of the form

\[ u_1 \parallel \cdots \parallel u_n \]

with each \( u_i \) an atomic event, and each \( s_i \) is called the summand of \( s \).

Now, we prove that for normal forms \( n \) and \( n' \), if \( n \sim s n' \) then \( n =_{AC} n' \). It is sufficient to induct on the sizes of \( n \) and \( n' \). We can get \( n =_{AC} n' \).

Finally, let \( s \) and \( t \) be basic APTC\textsuperscript{sat}\textsuperscript{I} terms, and \( s \sim_s t \), there are normal forms \( n \) and \( n' \), such that \( s = n \) and \( t = n' \). The soundness theorem modulo step bisimulation equivalence yields \( s \sim_s n \) and \( t \sim_s n' \), so \( n \sim_s s \sim_s t \sim_s n' \). Since if \( n \sim_s n' \) then \( n =_{AC} n' \), \( s = n =_{AC} n' = t \), as desired.

2. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_p \).
3. This case can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_{hp} \).

6.6. Standard Initial Abstraction

In this subsection, we will introduce APTC\textsuperscript{sat} with standard initial abstraction called APTC\textsuperscript{sat}√.

6.6.1. Basic Definition

**Definition 6.38** (Standard initial abstraction). Standard initial abstraction \( √_s \) is an abstraction mechanism to form functions from non-negative real numbers to processes with absolute timing, that map each number \( r \) to a process initialized at time \( r \).

6.6.2. The Theory APTC\textsuperscript{sat}I√

**Definition 6.39** (Signature of APTC\textsuperscript{sat}I√). The signature of APTC\textsuperscript{sat}I√ consists of the signature of APTC\textsuperscript{sat}I, and the standard initial abstraction operator \( √_s : \mathbb{R}^≥ \mathcal{P}_{abs} \rightarrow \mathcal{P}_{abs} \). Where \( \mathcal{P}_{abs} \) is the sorts with standard initial abstraction.

The set of axioms of APTC\textsuperscript{sat}I√ consists of the laws given in Table 43. Where \( v, w, \ldots \) are variables of sort \( \mathbb{R}^≥ \), \( F, G, \ldots \) are variables of sort \( \mathbb{R}^≥ \mathcal{P}_{abs} \), \( K, L, \ldots \) are variables of sort \( \mathbb{R}^{≥} \mathbb{R}^≥ \mathcal{P}_{abs} \), and we write \( √_s v.t \) for \( √_s(v.t) \).

It sufficient to extend bisimulations \( CI/ \sim \) of APTC\textsuperscript{sat} to

\[(CI/ \sim)^* = \{ f : \mathbb{R}^≥ \rightarrow CI/ \sim | \forall v \in \mathbb{R}^≥ . f(v) = \mathcal{T}_{abs} \mathcal{T}_{abs}^v (f(v)) \}\]

and define the constants and operators of APTC\textsuperscript{sat}√ on \( (CI/ \sim)^* \) as in Table 44.

6.6.3. Connections
Table 43. Axioms of APTC

| No. | Axiom |
|-----|-------|
| SIA1 | $\sqrt{v} F(v) = \sqrt{w} F(w)$ |
| SIA2 | $\sqrt{v} w (\sqrt{v} F(v)) = \sqrt{w} w (F(p))$ |
| SIA3 | $\sqrt{v} (\sqrt{w} F(v)) = \sqrt{v} w (K(v, w))$ |
| SIA4 | $x = \sqrt{v} x$ |
| SIA5 | $(\forall w \in \mathbb{R}^2, \sqrt{w} (x) = \sqrt{w} y) \Rightarrow x = y$ |
| SIA6 | $\sqrt{v} (\sqrt{w} (\sqrt{u}) \cdot x) = \sqrt{w} (\sqrt{u} (\sqrt{v} x))$ |
| SIA7 | $\sqrt{v} w (\sqrt{v} F(v)) = \sqrt{w} w (F(0))$ |
| SIA8 | $(\sqrt{v} F(v)) + x = \sqrt{v} (F(v) + \sqrt{w} w (x))$ |
| SIA9 | $(\sqrt{v} F(v)) \cdot x = \sqrt{v} w (F(v) \cdot x)$ |
| SIA10 | $\sqrt{w} w (\sqrt{v} F(v)) = \sqrt{v} w (F(v))$ |
| SIA11 | $(\sqrt{v} F(v)) \parallel x = \sqrt{w} w (F(v) \parallel \sqrt{w} w (x))$ |
| SIA12 | $x \parallel (\sqrt{v} F(v)) = \sqrt{w} w (\sqrt{v} (w) \parallel F(v) \parallel \sqrt{w} w (x))$ |
| SIA13 | $(\sqrt{v} F(v)) \parallel x = \sqrt{v} (F(v) \parallel \sqrt{w} w (x))$ |
| SIA14 | $(\sqrt{v} F(v)) \parallel x (\sqrt{v} F(v)) = \sqrt{v} (\sqrt{w} w (x) \parallel F(v) \parallel \sqrt{w} w (x))$ |
| SIA15 | $\Theta (\sqrt{v} F(v)) = \sqrt{v} \Theta (F(v))$ |
| SIA16 | $(\sqrt{v} F(v)) < x = \sqrt{v} (F(v) < x)$ |
| SIA17 | $\delta H (\sqrt{v} F(v)) = \sqrt{v} \delta H (F(v))$ |
| SIA18 | $\nu (\sqrt{v} F(v)) = \sqrt{v} \nu (F(v))$ |
| SIA19 | $f_{\nu} (\sqrt{v} w (F(v))) = \sqrt{w} w (f_{\nu} (v) (v = w))$ |

Table 44. Definitions of APTC on $(\mathbb{C} \cup \mathbb{F})^*$

| Function | Definition |
|----------|------------|
| $\delta$ | $\lambda w. \bar{\delta}$ |
| $\tilde{a}$ | $\lambda w. \bar{a}$ (a $\in A$) |
| $\bar{f}$ | $\lambda w. \bar{f}$ (f (v)) |
| $\sigma^v_w (f)$ | $\lambda w. \bar{f}$ (f (0)) |
| $f + g$ | $\lambda w. (f (w) \parallel g (w))$ |
| $f \cdot g$ | $\lambda w. (f (w) \parallel g (w))$ |
| $\Theta (f)$ | $\lambda w. \Theta (f (w))$ |
| $\nu (f)$ | $\lambda w. \nu (f (w))$ |

Table 45. Definitions of constants and operators of APTC in APTC

Theorem 6.40 (Generalization of APTC sat $\sqrt{v}$). 1. By the definitions of constants and operators of APTC sat $\sqrt{v}$ in APTC sat $\sqrt{v}$ in Table 43, APTC sat $\sqrt{v}$ is a generalization of APTC sat $\sqrt{v}$.

2. APTC sat $\sqrt{v}$ is a generalization of BATC sat $\sqrt{v}$.

3. By the definitions of constants and operators of APTC sat $\sqrt{v}$ in APTC sat $\sqrt{v}$ in Table 43, a discretization operator $D : P^\ast sat \rightarrow P^\ast sat$ in Table 44 and Table 45, APTC sat $\sqrt{v}$ is a generalization of APTC sat $\sqrt{v}$.

Proof. 1. It follows from the following two facts. By the definitions of constants and operators of APTC sat $\sqrt{v}$ in Table 43,

(a) the transition rules of APTC sat $\sqrt{v}$ are all source-dependent;
(b) the sources of the transition rules of APTC sat $\sqrt{v}$ contain an occurrence of $\sqrt{v}, \int$, and $\nu$.

So, APTC sat $\sqrt{v}$ is an embedding of APTC sat $\sqrt{v}$, as desired.

2. It follows from the following two facts.

(a) The transition rules of APTC sat $\sqrt{v}$ are all source-dependent;
Proof. It is easy to see that it is sufficient to prove that we omit it.

ACT C

Table 47. Axioms for discretization

\[
\begin{align*}
\bar{a} &= f_{\in(0,1)} \sigma^\text{abs}_{a}(\bar{a})(a \in A) \\
\delta &= f_{\in(0,1)} \sigma^\text{abs}_{\delta}(\delta) \\
\sigma^\text{abs}_{a}(x) &= \sigma^\text{abs}(x) \\
v^\text{abs}_{a}(x) &= v^\text{abs}(x) \\
\nu^\text{abs}_{a}(x) &= \nu^\text{abs}(x) \\
\sqrt{\mu} F(i) &= \sqrt{\nu} F(v)
\end{align*}
\]

Table 46. Definitions of constants and operators of \(ACT^{\text{dat}} \sqsupseteq \) in \(APTC^{\text{sat}} \)}

\[
\begin{align*}
\mathcal{D}(\delta) &= \hat{\delta} \\
\mathcal{D}(\bar{a}) &= \bar{\bar{a}} \\
\mathcal{D}(\sigma^D\sigma^\text{abs}(x)) &= \mathcal{D}\sigma^\text{abs}(\mathcal{D}(x)) \\
\mathcal{D}(x + y) &= \mathcal{D}(x) + \mathcal{D}(y) \\
\mu(x \cdot y) &= \mu(x) \cdot \mu(y) \\
\mathcal{D}(x \parallel y) &= \mathcal{D}(x) \parallel \mathcal{D}(y) \\
\mathcal{D}(\int F(v)) &= \int \mathcal{D}(F(v)) \\
\mathcal{D}(\sqrt{\nu} F(v)) &= \sqrt{\nu} \mathcal{D}(F(v))
\end{align*}
\]

Table 47. Transition rules of discretization \((a \in A, p \geq 0)\)

(b) The sources of the transition rules of \(APTC^{\text{sat}} \)} contain an occurrence of \(\sqrt{\cdot}\).

So, \(APTC^{\text{sat}} \)} is an embedding of \(APTC^{\text{sat}} \)} as desired.

3. It follows from the following two facts. By the definitions of constants and operators of \(ACT^{\text{dat}} \sqsupseteq \) in \(APTC^{\text{sat}} \)} in Table 47 a discretization operator \(D: P^\text{sat} \rightarrow P^\text{abs} \)} in Table 47, Table 48 and Table 49.

(a) The transition rules of \(APTC^{\text{sat}} \)} are all source-dependent;

(b) The sources of the transition rules of \(APTC^{\text{sat}} \)} contain an occurrence of \(\sqrt{\cdot}, \int \) and \(\nu^\text{abs}\).

So, \(APTC^{\text{sat}} \)} is an embedding of \(APTC^{\text{sat}} \)}, as desired. \(\square\)

6.6.4. Congruence

**Theorem 6.41** (Congruence of \(APTC^{\text{sat}} \)}). **Truly concurrent bisimulation equivalences** \(\sim_p, \sim_s\) and \(\sim_{hp}\) are all congruences with respect to \(APTC^{\text{sat}} \)}.

That is,

- **pomset bisimulation equivalence** \(\sim_p\) is a congruence with respect to \(APTC^{\text{sat}} \)};

- **step bisimulation equivalence** \(\sim_s\) is a congruence with respect to \(APTC^{\text{sat}} \)};

- **hp-bisimulation equivalence** \(\sim_{hp}\) is a congruence with respect to \(APTC^{\text{sat}} \)}.

**Proof.** It is easy to see that \(\sim_p, \sim_s\) and \(\sim_{hp}\) are all equivalent relations on \(APTC^{\text{sat}} \)} terms, it is only sufficient to prove that \(\sim_p, \sim_s\), and \(\sim_{hp}\) are all preserved by the operators \(\sigma^p, v^p\), and \(\nu^p\). It is trivial and we omit it. \(\square\)

6.6.5. Soundness

**Theorem 6.42** (Soundness of \(APTC^{\text{sat}} \)}). **The axiomatization of** \(APTC^{\text{sat}} \)} **is sound modulo truly concurrent bisimulation equivalences** \(\sim_p, \sim_s\), and \(\sim_{hp}\). That is,

\[
\begin{align*}
(x, p) \xrightarrow{\alpha} (x', p) & \quad (x, p) \xrightarrow{\int \alpha} (\int(x'), q) (q \in [p, p + 1]) \\
(x, p) \xrightarrow{\alpha} (\sqrt{\cdot}, p) & \quad (x, p) \xrightarrow{\sqrt{\cdot} \alpha} (\sqrt{\cdot}, q) (q \in [p, p + 1]) \\
(x, p) \xrightarrow{\text{act}} (x, p + r) & \quad \text{act}(x, p + r') (p + r' \in [p + r, p + r + 1]) \\
(x, p) \xrightarrow{\text{act}} (\text{act}(x), p + r') & \quad (x, p) \xrightarrow{\text{act}} (\text{act}(x), p + r) (p + r \in [p, p + 1])
\end{align*}
\]

Table 48. Transition rules of discretization \(a \in A, p, q \geq 0, r, r' > 0\)
\( D(f) = \lambda k. D(f(k)) \)

Table 49. Definitions of discretization on \(( CI/\sim_\ast )^* \)

1. Let \( x \) and \( y \) be \( \text{APTC}^\ast I_1 \) terms. If \( \text{APTC}^\ast I_1 \vdash x = y \), then \( x \sim_s y \);
2. Let \( x \) and \( y \) be \( \text{APTC}^\ast I_1 \) terms. If \( \text{APTC}^\ast I_1 \vdash x = y \), then \( x \sim_p y \);
3. Let \( x \) and \( y \) be \( \text{APTC}^\ast I_1 \) terms. If \( \text{APTC}^\ast I_1 \vdash x = y \), then \( x \sim_{hp} y \).

Proof. Since \( \sim_p, \sim_s \), and \( \sim_{hp} \) are both equivalent and congruent relations, we only need to check if each axiom in Table 49 is sound modulo step bisimulation equivalence, by \( \lambda \)-definitions in Table 44. We omit them.

1. Each axiom in Table 49 can be checked that it is sound modulo step bisimulation equivalence, by \( \lambda \)-definitions in Table 44. We omit them.
2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \( \cdot \)) or in concurrency (implicitly defined by \( + \)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \( P = \{ a, b : a \rightarrow b \} \).

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 49 is sound modulo pomset bisimulation equivalence, we omit them.

3. From the definition of \( \text{hp} \)-bisimulation, we know that \( \text{hp} \)-bisimulation is defined on the posetal product \( ( C_1, f, C_2 ), f : C_1 \rightarrow C_2 \text{ isomorphism} \). Two process terms \( s \) related to \( C_1 \) and \( t \) related to \( C_2 \), and \( f : C_1 \rightarrow C_2 \) isomorphism. Initially, \( ( C_1, f, C_2 ) = ( \emptyset, \emptyset, \emptyset ) \), and \( ( \emptyset, \emptyset, \emptyset ) \in \sim_{hp} \). When \( s \xrightarrow{a} s' ( C_1 \xrightarrow{a} C_1') \), there will be \( t \xrightarrow{a} t' ( C_2 \xrightarrow{a} C_2') \), and we define \( f' = f[a \rightarrow a] \). Then, if \( ( C_1, f, C_2 ) \in \sim_{hp} \), then \( ( C_1', f', C_2' ) \in \sim_{hp} \).

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 49 is sound modulo \( \text{hp} \)-bisimulation equivalence, we just need additionally to check the above conditions on \( \text{hp} \)-bisimulation, we omit them.

\[ \square \]

6.7. Time-Dependent Conditions

In this subsection, we will introduce \( \text{APTC}^\ast I_1 / C \) with time-dependent conditions called \( \text{APTC}^\ast I_1 / C \).

6.7.1. Basic Definition

**Definition 6.43** (Time-dependent conditions). The basic kinds of time-dependent conditions are at-time-point and at-time-point-greater-than. At-time-point \( p \) \(( p \in \mathbb{R}^2 \) is the condition that holds only at point of \( p \) and at-time-point-greater-than \( p \) \(( p \in \mathbb{R}^2 \) is the condition that holds in all point of time greater than \( p \). \( t \) is as the truth and \( f \) is as falsity.

6.7.2. The Theory \( \text{APTC}^\ast I_1 / C \)

**Definition 6.44** (Signature of \( \text{APTC}^\ast I_1 / C \)). The signature of \( \text{APTC}^\ast I_1 / C \) consists of the signature of \( \text{APTC}^\ast I_1 / \), and the at-time-point operator \( pt : \mathbb{R}^2 \rightarrow \mathbb{B}^* \), the at-time-point-greater-than operator \( pt_+ : \mathbb{R}^2 \rightarrow \mathbb{B}^* \), the logical constants and operators \( t : \mathbb{B}^* \rightarrow \mathbb{B}^* \), \( f : \mathbb{B}^* \rightarrow \mathbb{B}^* \), \( \land : \mathbb{B}^* \times \mathbb{B}^* \rightarrow \mathbb{B}^* \), \( \lor : \mathbb{B}^* \times \mathbb{B}^* \rightarrow \mathbb{B}^* \), the absolute initialization operator \( \pi_{abs} : \mathbb{R}^2 \times \mathbb{B}^* \rightarrow \mathbb{B}^* \), the standard initial abstraction operator \( \lambda : \mathbb{R}^2 \times \mathbb{B}^* \rightarrow \mathbb{B}^* \), and the conditional operator \( :: : \mathbb{B}^* \times \mathbb{P}_{abs} \rightarrow \mathbb{P}_{abs} \). Where \( \mathbb{B}^* \) is the sort of time-dependent conditions.

The set of axioms of \( \text{APTC}^\ast I_1 / C \) consists of the laws given in Table 50, Table 51 and Table 52. Where \( b \) is a condition.
Definition 6.45 (Basic terms of $\text{APTC}^{\text{sat}}I_\sqrt{C}$). The set of basic terms of $\text{APTC}^{\text{sat}}I_\sqrt{C}$, $\mathcal{B}(\text{APTC}^{\text{sat}}I_\sqrt{C}),$ is inductively defined as follows by two auxiliary sets $\mathcal{B}_0(\text{APTC}^{\text{sat}}I_\sqrt{C})$ and $\mathcal{B}_1(\text{APTC}^{\text{sat}}I_\sqrt{C})$:

1. if $a \in A$, then $\bar{a} \in \mathcal{B}_1(\text{APTC}^{\text{sat}}I_\sqrt{C});$
2. if $a \in A$ and $t \in \mathcal{B}(\text{APTC}^{\text{sat}}I_\sqrt{C})$, then $\bar{a} \cdot t \in \mathcal{B}_1(\text{APTC}^{\text{sat}}I_\sqrt{C});$
3. if $t, t' \in \mathcal{B}_1(\text{APTC}^{\text{sat}}I_\sqrt{C})$, then $t + t' \in \mathcal{B}_1(\text{APTC}^{\text{sat}}I_\sqrt{C});$
4. if $t, t' \in \mathcal{B}_1(\text{APTC}^{\text{sat}}I_\sqrt{C})$, then $t \parallel t' \in \mathcal{B}_1(\text{APTC}^{\text{sat}}I_\sqrt{C});$
5. if $t \in \mathcal{B}_1(\text{APTC}^{\text{sat}}I_\sqrt{C})$, then $t \in \mathcal{B}_0(\text{APTC}^{\text{sat}}I_\sqrt{C});$
6. if $p > 0$ and $t \in \mathcal{B}_0(\text{APTC}^{\text{sat}}I_\sqrt{C})$, then $\sigma(p)^t \in \mathcal{B}_0(\text{APTC}^{\text{sat}}I_\sqrt{C});$
7. if $p > 0$, $t \in \mathcal{B}_1(\text{APTC}^{\text{sat}}I_\sqrt{C})$ and $t' \in \mathcal{B}_0(\text{APTC}^{\text{sat}}I_\sqrt{C})$, then $t + \sigma(p)^t \in \mathcal{B}_0(\text{APTC}^{\text{sat}}I_\sqrt{C});$
8. if $t \in \mathcal{B}_0(\text{APTC}^{\text{sat}}I_\sqrt{C})$, then $\nu(t) \in \mathcal{B}_0(\text{APTC}^{\text{sat}}I_\sqrt{C});$
9. if $t \in \mathcal{B}_0(\text{APTC}^{\text{sat}}I_\sqrt{C})$, then $\int_{\nu}(t) \in \mathcal{B}_0(\text{APTC}^{\text{sat}}I_\sqrt{C});$

The operational semantics of $\text{APTC}^{\text{sat}}I_\sqrt{C}$ are defined by the transition rules in Table 53 and Table 54.

### Table 50. Axioms of logical operators

| No. | Axiom |
|-----|-------|
| CSA1 | $\text{val}_{\text{abs}}(t) = t$ |
| CSA2 | $\text{val}_{\text{abs}}(f) = f$ |
| CSA3 | $\text{val}_{\text{abs}}(\text{pt}(p)) = t$ |
| CSA4 | $\text{val}_{\text{abs}}(\text{pt}(p)) = f$ |
| CSA5 | $\text{val}_{\text{abs}}(\text{pt}(p + q)) = t$ |
| CSA6 | $\text{val}_{\text{abs}}(\text{pt},(p + q)) = f$ |
| CSA7 | $\text{val}_{\text{abs}}(\text{pt}(p + q)) = f$ |
| CSA8 | $\text{val}_{\text{abs}}(-b) = -\text{val}_{\text{abs}}(b)$ |
| CSA9 | $\text{val}_{\text{abs}}(b \land b') = \text{val}_{\text{abs}}(b) \land \text{val}_{\text{abs}}(b')$ |
| CSA10 | $\text{val}_{\text{abs}}(b \lor b') = \text{val}_{\text{abs}}(b) \lor \text{val}_{\text{abs}}(b')$ |
| CSA11 | $\sqrt{v.C}(v) = \sqrt{w.C}(v)$ |
| CSA12 | $\text{val}_{\text{abs}}(\sqrt{v.C}(v)) = \text{val}_{\text{abs}}(C(p))$ |
| CSA13 | $\sqrt{v.C}(v) = \sqrt{w.C}(w)$ |
| CSA14 | $\text{val}_{\text{abs}}(\sqrt{v.C}(v)) = \text{val}_{\text{abs}}(C(p))$ |
| CSA15 | $\sqrt{v.C}(v) = \sqrt{w.C}(w)$ |
| CSA16 | $\text{val}_{\text{abs}}(\sqrt{v.C}(v)) = \text{val}_{\text{abs}}(C(p))$ |
| CSA17 | $\text{val}_{\text{abs}}(\sqrt{v.C}(v)) = \text{val}_{\text{abs}}(C(p))$ |
| CSA18 | $\text{val}_{\text{abs}}(\sqrt{v.C}(v)) = \text{val}_{\text{abs}}(C(p))$ |

### Table 51. Axioms of conditions ($p, q \geq 0, r > 0$)

| No. | Axiom |
|-----|-------|
| R0 | $t = f$ |
| R1 | $-f = t$ |
| R2 | $-b = b$ |
| R3 | $t \lor b = t$ |
| R4 | $f \lor b = b$ |
| R5 | $b \lor b' = b' \lor b$ |
| R6 | $b \land b' = -(b \lor -b')$ |
Table 52. Axioms of conditionals ($p \geq 0$)

\[
\begin{align*}
(\langle x, p \rangle, x', p') & \overset{\Delta}{=} (\langle x', p \rangle, x, p') \\
(\langle x, p \rangle, x' \overset{\Delta}{=} (\langle x', p \rangle, x, p')) & \overset{\Delta}{=} (\langle x, p \rangle, x' + r) \\
(\langle x, p \rangle, x' \overset{\Delta}{=} (\langle x', p \rangle, x, p + r)) & \overset{\Delta}{=} (\langle x, p \rangle, x' \overset{\Delta}{=} (\langle x', p \rangle, x, p + r))
\end{align*}
\]

Table 53. Transition rules of APTC$\mathsf{sat}$I/C

10. If $s > 0$ and $t \in B_0(\text{APT} \mathsf{c} \mathsf{sat}I/C)$, then $\sqrt{s.t(s)} \in B_0(\text{APT} \mathsf{c} \mathsf{sat}I/C)$;
11. $\delta \in B(\text{APT} \mathsf{c} \mathsf{sat}I/C)$;
12. If $t \in B_0(\text{APT} \mathsf{c} \mathsf{sat}I/C)$, then $t \in B(\text{APT} \mathsf{c} \mathsf{sat}I/C)$.

Theorem 6.46 (Elimination theorem). Let $p$ be a closed APTC$\mathsf{sat}$I/C term. Then there is a basic APTC$\mathsf{sat}$I/C term $q$ such that APTC$\mathsf{sat}$I/C$\vdash p = q$.

Proof. It is sufficient to induct on the structure of the closed APTC$\mathsf{sat}$I/C term $p$. It can be proven that $p$ combined by the constants and operators of APTC$\mathsf{sat}$I/C exists an equal basic term $q$, and the other operators not included in the basic terms, such as $\nu_{abs}$, $\nu_{abs}$, $\delta$, $\parallel$, $\partial_H$, $\Theta$, $\prec$, and the constants and operators related to conditions can be eliminated. \hfill \Box

6.7.4. Congruence

Theorem 6.47 (Congruence of APTC$\mathsf{sat}$I/C). Truly concurrent bisimulation equivalences $\sim_p$, $\sim_s$ and $\sim_{hp}$ are all congruences with respect to APTC$\mathsf{sat}$I/C. That is,

- pomset bisimulation equivalence $\sim_p$ is a congruence with respect to APTC$\mathsf{sat}$I/C;
• step bisimulation equivalence \( \sim_s \) is a congruence with respect to APTC\(^{\text{sat}}\) I/C;
• hp-bisimulation equivalence \( \sim_{hp} \) is a congruence with respect to APTC\(^{\text{sat}}\) I/C.

**Proof.** It is easy to see that \( \sim_p, \sim_s, \) and \( \sim_{hp} \) are all equivalent relations on APTC\(^{\text{sat}}\) I/C terms, it is only sufficient to prove that \( \sim_p, \sim_s, \) and \( \sim_{hp} \) are all preserved by the operators \( \sigma^p_{\text{abs}}, v^p_{\text{abs}}, \) and \( \overrightarrow{p}_{\text{abs}} \). It is trivial and we omit it.

6.7.5. Soundness

**Theorem 6.48** (Soundness of APTC\(^{\text{sat}}\) I/C). The axiomatization of APTC\(^{\text{sat}}\) I/C is sound modulo truly concurrent bisimulation equivalences \( \sim_p, \sim_s, \) and \( \sim_{hp} \). That is,

1. let x and y be APTC\(^{\text{sat}}\) I/C terms. If APTC\(^{\text{sat}}\) I/C \( \vdash x = y \), then \( x \sim_s y \);
2. let x and y be APTC\(^{\text{sat}}\) I/C terms. If APTC\(^{\text{sat}}\) I/C \( \vdash x = y \), then \( x \sim_p y \);
3. let x and y be APTC\(^{\text{sat}}\) I/C terms. If APTC\(^{\text{sat}}\) I/C \( \vdash x = y \), then \( x \sim_{hp} y \).

**Proof.** Since \( \sim_p, \sim_s, \) and \( \sim_{hp} \) are all equivalent and congruent relations, we only need to check if each axiom in Table 51 and Table 52 is sound modulo \( \sim_p, \sim_s, \) and \( \sim_{hp} \) respectively.

1. Each axiom in Table 51 and Table 52 can be checked that it is sound modulo step bisimulation equivalence, by transition rules of conditionals in Table 53.
2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \( \cdot \)) or in concurrency (implicitly defined by \( \cdot \) and \( + \), and explicitly defined by \( \parallel \)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \( P = \{\bar{a}, \bar{b} : \bar{a} \parallel \bar{b}\} \). Then the pomset transition labeled by the above \( P \) is just composed of one single event transition labeled by \( \bar{a} \) succeeded by another single event transition labeled by \( \bar{b} \), that is, \( \bar{a} \parallel \bar{b} \).
3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product \((C_1, f, C_2)\), \( f : C_1 \rightarrow C_2 \) isomorphism. Two process terms \( s \) related to \( C_1 \) and \( t \) related to \( C_2 \), and \( f : C_1 \rightarrow C_2 \) isomorphism. Initially, \((C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)\), and \((\emptyset, \emptyset, \emptyset) \in \sim_{hp}\). When \( s \sim a \rightarrow t' (C'_1 \rightarrow a) \), there will be \( t \sim a \rightarrow t' (C'_2 \rightarrow a) \), and we define \( f' = f[a \rightarrow a] \). Then, if \((C_1, f, C_2) \in \sim_{hp}\), then \((C'_1, f', C'_2) \in \sim_{hp}\).

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 51 and Table 52 is sound modulo pomset bisimulation equivalence, we omit them.

6.7.6. Completeness

**Theorem 6.49** (Completeness of APTC\(^{\text{sat}}\) I/C). The axiomatization of APTC\(^{\text{sat}}\) I/C is complete modulo truly concurrent bisimulation equivalences \( \sim_p, \sim_s, \) and \( \sim_{hp} \). That is,

1. let p and q be closed APTC\(^{\text{sat}}\) I/C terms, if \( p \sim_s q \) then \( p = q \);
2. let p and q be closed APTC\(^{\text{sat}}\) I/C terms, if \( p \sim_p q \) then \( p = q \);
3. let p and q be closed APTC\(^{\text{sat}}\) I/C terms, if \( p \sim_{hp} q \) then \( p = q \).

**Proof.** 1. Firstly, by the elimination theorem of APTC\(^{\text{sat}}\) I/C, we know that for each closed APTC\(^{\text{sat}}\) I/C term \( p \), there exists a closed basic APTC\(^{\text{sat}}\) I/C term \( p' \), such that APTC\(^{\text{sat}}\) I/C \( \vdash p = p' \), so, we only need to consider closed basic APTC\(^{\text{sat}}\) I/C terms. The basic terms modulo associativity and commutativity (AC) of conflict + (defined by axioms A1 and A2 in Table 50) and associativity and commutativity (AC) of parallel \( \parallel \) (defined by axioms P2 and P3 are needed. 

Draft of Truly Concurrent Process Algebra with Timing 77
in Table 40), and these equivalences is denoted by $=_{AC}$. Then, each equivalence class $s$ modulo $AC$ of $+$ and $∥$ has the following normal form

$$s_1 + \cdots + s_k$$

with each $s_i$ either an atomic event or of the form

$$t_1 \cdot \cdots \cdot t_m$$

with each $t_j$ either an atomic event or of the form

$$u_1 \parallel \cdots \parallel u_n$$

with each $u_i$ an atomic event, and each $s_i$ is called the summand of $s$.

Now, we prove that for normal forms $n$ and $n'$, if $n \sim_s n'$ then $n =_{AC} n'$. It is sufficient to induct on the sizes of $n$ and $n'$. We can get $n =_{AC} n'$.

Finally, let $s$ and $t$ be basic $APTC^{rel}$ terms, and $s \sim_s t$, there are normal forms $n$ and $n'$, such that $s = n$ and $t = n'$. The soundness theorem modulo step bisimulation equivalence yields $s \sim_s n$ and $t \sim_s n'$, so $n \sim_s s \sim_s t \sim_s n'$. Since if $n \sim_s n'$ then $n =_{AC} n'$, $s = n =_{AC} n' = t$, as desired.

2. This case can be proven similarly, just by replacement of $\sim_s$ by $\sim_p$.
3. This case can be proven similarly, just by replacement of $\sim_s$ by $\sim_{hp}$.

\[\square\]

7. Recursion

In this section, we will introduce recursion for APTC with timing, including recursion for $APTC^{drt}$, $APTC^{dat}$, $APTC^{rel}$, and $APTC^{state}$.

7.1. Discrete Relative Timing

As recursion for APTC in section 2.1, we also need the concept of guardedness in recursion for APTC with timing. With the capabilities related timing, guardedness means that $X$ is always preceded by an action or delayed for at least one time slice.

**Definition 7.1** (Guarded recursive specification of $APTC^{drt}$ + Rec). Let $t$ be a term of $APTC^{drt}$ containing a variable $X$, an occurrence of $X$ in $t$ is guarded if $t$ has a subterm of the form $(a_1 \parallel \cdots \parallel a_k) \cdot t'(a_1, \cdots, a_k \in A, k \in \mathbb{N})$ or $\sigma^n_{rel}(t')(n > 0)$ and $t'$ is a $APTC^{drt}$ term containing this occurrence of $X$.

A recursive specification over $APTC^{drt}$ is called guarded if all occurrences of variables in the right-hand sides of its equations are guarded, or it can be rewritten to such a recursive specification using the axioms of $APTC^{drt}$ and the equations of the recursive specification.

**Definition 7.2** (Signature of $APTC^{drt}$ + Rec). The signature of $APTC^{drt}$ + Rec contains the signature of $APTC^{drt}$ extended with a constant $(X|E) :\rightarrow P_{rel}$ for each guarded recursive specification $E$ and $X \in V(E)$.

The axioms of $APTC^{drt}$ + Rec consists of the axioms of $APTC^{drt}$, and RDP and RSP in Table 6.

The additional transition rules of $APTC^{drt}$ + Rec is shown in Table 55.

**Theorem 7.3** (Generalization of $APTC^{drt}$ + Rec). By the definitions of $a_1 \parallel \cdots \parallel a_k = (X|X = a_1 \parallel \cdots \parallel a_k + \sigma^n_{rel}(X))$ for each $a_1, \cdots, a_k \in A, k \in \mathbb{N}$ and $\delta = (X|X = \delta + \sigma^n_{rel}(X))$ is a generalization of $APTC^{drt}$ + Rec.

**Proof.** It follows from the following two facts.
1. The transition rules of $APTC^{drt}$ + Rec in section 2.1 are all source-dependent;
2. The sources of the transition rules of $APTC^{drt}$ + Rec contain an occurrence of $\delta$, $a_i$, $\sigma^n_{rel}$, $v^n_{rel}$ and $\tau^n_{rel}$.
\[
\begin{align*}
&\frac{t_i((X_1|E),\ldots,(X_j|E)) \to^{(a_1,\ldots,a_k)} y}{(X_i|E) \to^{(a_1,\ldots,a_k)} y} \\
&\frac{t_i((X_1|E),\ldots,(X_j|E)) \to^{(a_1,\ldots,a_k)} y}{(X_i|E) \to^{(a_1,\ldots,a_k)} y} \\
&\frac{t_i((X_1|E),\ldots,(X_j|E)) \to^m y}{(X_i|E) \to^m y} \\
&\frac{t_i((X_1|E),\ldots,(X_j|E)) \uparrow}{(X_i|E) \uparrow}
\end{align*}
\]

Table 55. Transition rules of APTC^{drt}\_Rec \((m > 0, n \geq 0)\)

So, APTC^{drt}\_Rec is an embedding of APTC^{drt}\_Rec, as desired. □

**Theorem 7.4** (Soundness of APTC^{drt}\_Rec). Let \(x\) and \(y\) be APTC^{drt}\_Rec terms. If APTC^{drt}\_Rec \(\vdash x = y\), then

1. \(x \sim_s y\);
2. \(x \sim_p y\);
3. \(x \sim_{hp} y\).

**Proof.** Since \(\sim_p\), \(\sim_s\), and \(\sim_{hp}\) are both equivalent and congruent relations, we only need to check if each axiom in Table \(6\) is sound modulo step bisimulation equivalence, by transition rules in Table \(55\). We omit them.

1. Each axiom in Table \(6\) can be checked that it is sound modulo step bisimulation equivalence, by transition rules in Table \(55\). We omit them.

2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \(\cdot\)) or in concurrency (implicitly defined by \(\uparrow\)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \(\{a, b : a \cdot b\}\). Then the pomset transition labeled by the above \(P\) is just composed of one single event transition labeled by \(\bar{a}\) succeeded by another single event transition labeled by \(\bar{b}\), that is, \(\xrightarrow{a} \sim a \cdot \rightarrow \rightarrow\). Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table \(6\) is sound modulo pomset bisimulation equivalence, we omit them.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product \((C_1, f, C_2), f : C_1 \rightarrow C_2\) isomorphism. Two process terms \(s\) related to \(C_1\) and \(t\) related to \(C_2\), and \(f : C_1 \rightarrow C_2\) isomorphism. Initially, \((C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)\), and \((\emptyset, \emptyset, \emptyset) \in \sim_{hp}\). When \(s \sim s' (C_1 \rightarrow C_1')\), there will be \(t \sim t' (C_2 \rightarrow C_2')\), and we define \(f' = f [a \rightarrow a]\). Then, if \((C_1, f, C_2) \in \sim_{hp}\), then \((C_1', f', C_2') \in \sim_{hp}\). Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table \(6\) is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them. □

**Theorem 7.5** (Completeness of APTC^{drt}\_linear Rec). Let \(p\) and \(q\) be closed APTC^{drt}\_linear Rec terms, then,

1. if \(p \sim_s q\) then \(p = q\);
2. if \(p \sim_p q\) then \(p = q\);
3. if \(p \sim_{hp} q\) then \(p = q\).

**Proof.** Firstly, we know that each process term in APTC^{drt} with linear recursion is equal to a process term \(\langle X_1|E\rangle\) with \(E\) a linear recursive specification. It remains to prove the following cases.
Theorem 7.8 (Generalization of APTC\textsuperscript{dat}+Rec). By the definitions of $a_1 \parallel \cdots \parallel a_k = (X|X = a_1 \parallel \cdots \parallel a_k + \sigma_{abs}(X))$ for each $a_1, \cdots, a_k \in A, k \in \mathbb{N}$ and $\delta = (X|X = \delta + \sigma_{abs}^1(X))$ is a generalization of APTC+Rec. 

Proof. It follows from the following two facts.

1. The transition rules of APTC+Rec in section 221 are all source-dependent;
2. The sources of the transition rules of APTC\textsuperscript{dat}+Rec contain an occurrence of $\delta, a_0, \sigma_{abs}, v_{abs}$, and $\overline{v}_{abs}$.

So, APTC+Rec is an embedding of APTC\textsuperscript{dat}+Rec, as desired. 

Theorem 7.9 (Soundness of APTC\textsuperscript{dat}+Rec). Let $x$ and $y$ be APTC\textsuperscript{dat}+Rec terms. If APTC\textsuperscript{dat}+Rec $\vdash x = y$, then

1. $x \sim_s y$;
2. $x \sim_p y$;
3. $x \sim_{hp} y$.

Proof. Since $\sim_p$, $\sim_s$, and $\sim_{hp}$ are both equivalent and congruent relations, we only need to check if each axiom in Table 56 is sound modulo $\sim_p$, $\sim_s$, and $\sim_{hp}$ respectively.
1. Each axiom in Table 6 can be checked that it is sound modulo step bisimulation equivalence, by transition rules in Table 6. We omit them.

2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by ∗) or in concurrency (implicitly defined by · and +, and explicitly defined by //). Of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of P = {a, b : a, b}. Then the pomset transition labeled by the above P is just composed of one single event transition labeled by a succeeded by another single event transition labeled by b, that is, \( P \xrightarrow{a} b \).

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 6 is sound modulo pomset bisimulation equivalence, we omit them.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product \((C_1, f, C_2), f : C_1 \rightarrow C_2 \) isomorphism. Two process terms \( s \) related to \( C_1 \) and \( t \) related to \( C_2 \), and \( f : C_1 \rightarrow C_2 \) isomorphism. Initially, \((C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)\), and \((\emptyset, \emptyset, \emptyset) \in \sim_{hp}\). When \( s \sim s' (C_1 \sim C_1') \), there will be \( t \sim t' (C_2 \sim C_2') \) and we define \( f' = f[a \rightarrow a] \). Then, if \((C_1, f, C_2) \in \sim_{hp}\), then \((C_1', f', C_2') \in \sim_{hp}\).

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 6 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

\[ \square \]

**Theorem 7.10** (Completeness of APTC\(^{\text{dat}}\)+linear Rec). Let \( p \) and \( q \) be closed APTC\(^{\text{dat}}\)+linear Rec terms, then,

1. if \( p \sim q \) then \( p = q \);
2. if \( p \sim p \) then \( p = q \);
3. if \( p \sim_{hp} q \) then \( p = q \).

**Proof.** Firstly, we know that each process term in APTC\(^{\text{dat}}\) with linear recursion is equal to a process term \( \langle X_1 | E \rangle \) with \( E \) a linear recursive specification.

It remains to prove the following cases.

1. If \( \langle X_1 | E_1 \rangle \sim \langle Y_1 | E_2 \rangle \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( \langle X_1 | E_1 \rangle = \langle Y_1 | E_2 \rangle \).

   It can be proven similarly to the completeness of APTC + linear Rec, see [17].

2. If \( \langle X_1 | E_1 \rangle \sim_p \langle Y_1 | E_2 \rangle \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( \langle X_1 | E_1 \rangle = \langle Y_1 | E_2 \rangle \).

   It can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_p \), we omit it.

3. If \( \langle X_1 | E_1 \rangle \sim_{hp} \langle Y_1 | E_2 \rangle \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( \langle X_1 | E_1 \rangle = \langle Y_1 | E_2 \rangle \).

   It can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_{hp} \), we omit it.

\[ \square \]

### 7.3. Continuous Relative Timing

As recursion for APTC in section 2.1 we also need the concept of guardedness in recursion for APTC with timing. With the capabilities related timing, guardedness means that \( X \) is always preceded by an action or delayed for a period of time greater than 0.

**Definition 7.11** (Guarded recursive specification of APTC\(^{\text{sat}}\)+Rec). Let \( t \) be a term of APTC\(^{\text{sat}}\)+Rec containing a variable \( X \), an occurrence of \( X \) in \( t \) is guarded if \( t \) has a subterm of the form \( (a_1 \parallel \cdots \parallel a_k) \cdot t' a_1, \cdots, a_k \in A, k \in \mathbb{N} \) \) or \( \sigma^p_{\text{rel}}(t')(p > 0) \) and \( t' \) is a APTC\(^{\text{sat}}\)+Rec term containing this occurrence of \( X \).

A recursive specification over APTC\(^{\text{sat}}\)+Rec is called guarded if all occurrences of variables in the right-hand sides of its equations are guarded, or it can be rewritten to such a recursive specification using the axioms of APTC\(^{\text{sat}}\)+Rec and the equations of the recursive specification.
Proof.
It follows from the following two facts.

1. The transition rules of $\text{APTC}^{\text{I+Rec}}$ are either within causality relations (defined by $\parallel$) or in concurrency (implicitly defined by $\perp$ and $+$, and explicitly defined by $\uparrow$), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of $P = \{a\overset{\parallel}{\rightarrow}b\overset{\perp}{\rightarrow}a\overset{\parallel}{\rightarrow}b\}$. Then the pomset transition labeled by the above $P$ is just composed of one single event transition labeled by $a$ succeeded by another single event transition labeled by $b$, that is, $\overset{\parallel}{\rightarrow}a\overset{\perp}{\rightarrow}b$. Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 6 is sound modulo step bisimulation equivalence, as desired. 

2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by $\parallel$) or in concurrency (implicitly defined by $\perp$ and $+$, and explicitly defined by $\uparrow$), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of $P = \{a\overset{\parallel}{\rightarrow}b\overset{\perp}{\rightarrow}a\overset{\parallel}{\rightarrow}b\}$. Then the pomset transition labeled by the above $P$ is just composed of one single event transition labeled by $a$ succeeded by another single event transition labeled by $b$, that is, $\overset{\parallel}{\rightarrow}a\overset{\perp}{\rightarrow}b$. Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 6 is sound modulo pomset bisimulation equivalence, as desired.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product $(C_1, f, C_2), f : C_1 \to C_2$ isomorphism. Two process terms $s$ related to $C_1$ and $t$ related to $C_2$, and $f : C_1 \to C_2$ isomorphism. Initially, $(C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)$, and $(\emptyset, \emptyset, \emptyset) \not\sim_{\text{hp}}$. When $s \overset{a}{\rightarrow} s'$ $(C_1, a \overset{a}{\rightarrow} C_2')$, there will be $t \overset{a'}{\rightarrow} t'$ $(C_2, a' \overset{a}{\rightarrow} C_2')$, and we define $f' = f[a \mapsto a]$. Then, if $(C_1, f, C_2) \not\sim_{\text{hp}}$, then $(C_1', f', C_2') \not\sim_{\text{hp}}$.

Table 57. Transition rules of $\text{APTC}^{\text{I+Rec}}$ $(r > 0)$

| Rule                                                                 |
|----------------------------------------------------------------------|
| $t_i((X_1|E),\ldots,(X_j|E)) \overset{(a_1,\ldots,a_k)}{\rightarrow} y$ |
| $(X_i|E) \overset{\{a_1,\ldots,a_k\}}{\rightarrow} y$             |
| $t_i((X_1|E),\ldots,(X_j|E)) \overset{(a_1,\ldots,a_k)}{\rightarrow} y$ |
| $(X_i|E) \overset{\{a_1,\ldots,a_k\}}{\rightarrow} y$             |

**Definition 7.12** (Signature of $\text{APTC}^{\text{I+Rec}}$). The signature of $\text{APTC}^{\text{I+Rec}}$ contains the signature of $\text{APTC}^{\text{I+Rec}}$ extended with a constant $(X|E) \rightarrow P_{\text{rel}}$ for each guarded recursive specification $X$ and $E \in V(E)$.

The axioms of $\text{APTC}^{\text{I+Rec}}$ consists of the axioms of $\text{APTC}^{\text{I+Rec}}$, and RDP and RSP in Table 6.

The additional transition rules of $\text{APTC}^{\text{I+Rec}}$ is shown in Table 57.
\[
\frac{(t_1((X_1|E),...,X_j|E)),p_{\langle a_1,...,a_k\rangle}}{((X_1|E),p_{\langle a_1,...,a_k\rangle})} \xrightarrow{\sqrt{\cdot}} \langle \sqrt{\cdot} \rangle
\]

\[
\frac{(t_1((X_1|E),...,X_j|E)),p_{\langle a_1,...,a_k\rangle}}{((X_1|E),p_{\langle a_1,...,a_k\rangle})} \xrightarrow{y} \langle y, p \rangle
\]

\[
\frac{(t_1((X_1|E),...,X_j|E)),p_{\langle a_1,...,a_k\rangle}}{((X_1|E),p_{\langle a_1,...,a_k\rangle})} \xrightarrow{\cdot} (X_1|E),p \cdot r)
\]

\[
\frac{(t_1((X_1|E),...,X_j|E)),p_{\langle a_1,...,a_k\rangle}}{((X_1|E),p_{\langle a_1,...,a_k\rangle})} \xrightarrow{\cdot} (X_1|E),p \cdot r)
\]

Table 58. Transition rules of APTC^{sat}I+Rec \(r > 0, p \geq 0\)

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 58 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

**Theorem 7.15** (Completeness of APTC^{sat}I+linear Rec). Let \(p\) and \(q\) be closed APTC^{sat}I+linear Rec terms, then,
1. If \(p \sim_s q\) then \(p = q\);
2. If \(p \sim_p q\) then \(p = q\);
3. If \(p \sim_{hp} q\) then \(p = q\).

**Proof.** Firstly, we know that each process term in APTC^{sat} with linear recursion is equal to a process term \(\langle X_1|E \rangle\) with \(E\) a linear recursive specification. It remains to prove the following cases.

1. If \(\langle X_1|E_1 \rangle \sim_s (Y_1|E_2)\) for linear recursive specification \(E_1\) and \(E_2\), then \(\langle X_1|E_1 \rangle = \langle Y_1|E_2 \rangle\).

   It can be proven similarly to the completeness of APTC + linear Rec, see [17].

2. If \(\langle X_1|E_1 \rangle \sim_p (Y_1|E_2)\) for linear recursive specification \(E_1\) and \(E_2\), then \(\langle X_1|E_1 \rangle = \langle Y_1|E_2 \rangle\).

   It can be proven similarly, just by replacement of \(\sim_s\) by \(\sim_p\), we omit it.

3. If \(\langle X_1|E_1 \rangle \sim_{hp} (Y_1|E_2)\) for linear recursive specification \(E_1\) and \(E_2\), then \(\langle X_1|E_1 \rangle = \langle Y_1|E_2 \rangle\).

   It can be proven similarly, just by replacement of \(\sim_s\) by \(\sim_{hp}\), we omit it.


### 7.4. Continuous Absolute Timing

**Definition 7.16** (Guarded recursive specification of APTC^{sat}I+Rec). Let \(t\) be a term of APTC^{sat}I containing a variable \(X\), an occurrence of \(X\) in \(t\) is guarded if \(t\) has a subterm of the form \((a_1 || ... || a_k) \cdot t'(a_1, ..., a_k \in A, k \in \mathbb{N}), \sigma_{abs}(t')\) or \(\sigma_{abs}(s) \cdot t'(p > 0)\) and \(s, t'\) is a APTC^{sat}I term, with \(t'\) containing this occurrence of \(X\).

A recursive specification over APTC^{sat}I is called guarded if all occurrences of variables in the right-hand sides of its equations are guarded, or it can be rewritten to such a recursive specification using the axioms of APTC^{sat}I and the equations of the recursive specification.

**Definition 7.17** (Signature of APTC^{sat}I+Rec). The signature of APTC^{sat}I+Rec contains the signature of APTC^{sat}I extended with a constant \(X|E \rightarrow P_{abs}\) for each guarded recursive specification \(E\) and \(X \in V(E)\).

The axioms of APTC^{sat}I+Rec consists of the axioms of APTC^{sat}I, and RDP and RSP in Table 58. The additional transition rules of APTC^{sat}I+Rec is shown in Table 58.

**Theorem 7.18** (Generalization of APTC^{sat}I+Rec). By the definitions of \(a_1 || ... || a_k = (X|X = \int_{v \in [0,\infty)} \sigma_{abs}^v(a_1) \parallel ... \parallel \int_{v \in [0,\infty)} \sigma_{abs}^v(a_k) + \sigma_{abs}^v(X))\) for each \(a_1, ..., a_k \in A, k \in \mathbb{N}, r > 0\) and \(\delta = (X|X = \int_{v \in [0,\infty)} \sigma_{abs}^v(\delta) + \sigma_{abs}^v(X))\) \((r > 0)\) is a generalization of APTC+Rec.
Proof. It follows from the following two facts.

1. The transition rules of APTC + Rec in section 2 are all source-dependent;
2. The sources of the transition rules of APTC sat + Rec contain an occurrence of \( h \), \( a \), \( \sigma_{abs} \), \( t_{abs} \), \( \pi_{abs} \) and \( f \).

So, APTC + Rec is an embedding of APTC sat + Rec, as desired. \( \square \)

Theorem 7.19 (Soundness of APTC sat + Rec). Let \( x \) and \( y \) be APTC sat + Rec terms. If APTC sat + Rec \( \vdash x = y \), then

1. \( x \sim_s y \);
2. \( x \sim_p y \);
3. \( x \sim_{hp} y \).

Proof. Since \( \sim_p, \sim_s \), and \( \sim_{hp} \) are both equivalent and congruent relations, we only need to check if each axiom in Table 6 is sound modulo \( \sim_p, \sim_s \), and \( \sim_{hp} \) respectively.

1. Each axiom in Table 6 can be checked that it is sound modulo step bisimulation equivalence, by transition rules in Table 5. We omit them.
2. From the definition of pomset bisimulation, we know that pomset bisimulation is defined by pomset transitions, which are labeled by pomsets. In a pomset transition, the events (actions) in the pomset are either within causality relations (defined by \( \cdot \)) or in concurrency (implicitly defined by \( + \), and explicitly defined by \( \| \)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent (soundness modulo step bisimulation), so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \( \{ \tilde{a}, \tilde{b} : \tilde{a} \cdot \tilde{b} \} \). Then the pomset transition labeled by the above \( P \) is just composed of one single event transition labeled by \( \tilde{a} \) succeeded by another single event transition labeled by \( \tilde{b} \), that is, \( P = a \xrightarrow{\tilde{a}} b \).

Similarly to the proof of soundness modulo step bisimulation equivalence, we can prove that each axiom in Table 6 is sound modulo pomset bisimulation equivalence, we omit them.

3. From the definition of hp-bisimulation, we know that hp-bisimulation is defined on the posetal product \( (C_1, f, C_2), f : C_1 \rightarrow C_2 \) isomorphism. Two process terms \( s \) related to \( C_1 \) and \( t \) related to \( C_2 \), and \( f : C_1 \rightarrow C_2 \) isomorphism. Initially, \( (C_1, f, C_2) = (\emptyset, \emptyset, \emptyset) \), and \( (\emptyset, \emptyset, \emptyset) \) \( \sim_{hp} \). When \( s \xrightarrow{a} s' \) \( (C_1 \xrightarrow{a} C_1') \), there will be \( t \xrightarrow{a} t' \) \( (C_2 \xrightarrow{a} C_2') \), and we define \( f' = f[a \mapsto a] \). Then, if \( (C_1, f, C_2) \sim_{hp} \), then \( (C_1', f', C_2') \sim_{hp} \).

Similarly to the proof of soundness modulo pomset bisimulation equivalence, we can prove that each axiom in Table 6 is sound modulo hp-bisimulation equivalence, we just need additionally to check the above conditions on hp-bisimulation, we omit them.

\( \square \)

Theorem 7.20 (Completeness of APTC sat + linear Rec). Let \( p \) and \( q \) be closed APTC sat + linear Rec terms, then,

1. If \( p \sim_s q \) then \( p = q \);
2. If \( p \sim_p q \) then \( p = q \);
3. If \( p \sim_{hp} q \) then \( p = q \).

Proof. Firstly, we know that each process term in APTC sat with linear recursion is equal to a process term \( \langle X_1 | E \rangle \) with \( E \) a linear recursive specification.

It remains to prove the following cases.

1. If \( \langle X_1 | E_1 \rangle \sim_s \langle Y_1 | E_2 \rangle \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( \langle X_1 | E_1 \rangle = \langle Y_1 | E_2 \rangle \).

It can be proven similarly to the completeness of APTC + linear Rec, see [17].

2. If \( \langle X_1 | E_1 \rangle \sim_p \langle Y_1 | E_2 \rangle \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( \langle X_1 | E_1 \rangle = \langle Y_1 | E_2 \rangle \).

It can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_p \), we omit it.

3. If \( \langle X_1 | E_1 \rangle \sim_{hp} \langle Y_1 | E_2 \rangle \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( \langle X_1 | E_1 \rangle = \langle Y_1 | E_2 \rangle \).

It can be proven similarly, just by replacement of \( \sim_s \) by \( \sim_{hp} \), we omit it. \( \square \)
8. Abstraction

In this section, we will introduce something about silent step \( \tau \) and abstraction \( \tau_I \). The version of abstraction of APTC without timing, please refer to section 2.1. We will introduce APTC\(_{\tau}^{\text{drt}}\), APTC\(_{\tau}^{\text{sat}}\), APTC\(_{\tau}^{\text{ert}}\), and APTC\(_{\tau}^{\text{rst}}\) with abstraction called APTC\(_{\tau}^{\text{drt}}\), APTC\(_{\tau}^{\text{sat}}\), APTC\(_{\tau}^{\text{ert}}\), and APTC\(_{\tau}^{\text{rst}}\) with abstraction called APTC\(_{\tau}^{\text{drt}}\), respectively.

8.1. Discrete Relative Timing

**Definition 8.1** (Rooted branching truly concurrent bisimulations). The following two conditions related timing should be added into the concepts of branching truly concurrent bisimulations in section 2.1.

1. if \( C_1 \rightarrow^1 C_2 \), then there are \( C_1 \), \( C_2 \) such that \( C_1 \rightarrow C_1 \rightarrow C_2 \), and \( (C_1, C_1) \in R \) and \( (C_2, C_2) \in R \), or \( (C_1, f[\emptyset \rightarrow \emptyset], C_1) \in R \) and \( (C_2, f[\emptyset \rightarrow \emptyset], C_2) \in R \); 
2. if \( C_1 \rightarrow^1 \), then \( C_1 \rightarrow^1 \).

And the following root conditions related timing should be added into the concepts of rooted branching truly concurrent bisimulations in section 2.1.

1. if \( C_1 \rightarrow^m C_2 (m > 0) \), then there is \( C_2 \) such that \( C_1 \rightarrow C_2 \), and \( (C_2, C_2) \in R \), or \( (C_2, f[\emptyset \rightarrow \emptyset], C_2) \in R \).

**Definition 8.2** (Signature of APTC\(_{\tau}^{\text{drt}}\)). The signature of APTC\(_{\tau}^{\text{drt}}\) consists of the signature of APTC\(_{\tau}^{\text{drt}}\), and the undelayable silent step constant \( \tau \rightarrow \mathcal{P}_{\text{rel}} \), and the abstraction operator \( \tau_I : \mathcal{P}_{\text{rel}} \rightarrow \mathcal{P}_{\text{rel}} \) for \( I \subseteq A \).

The axioms of APTC\(_{\tau}^{\text{drt}}\) include the laws in Table 11 covering the case that \( a \equiv \tau \) and \( b \equiv \tau \), and the axioms in Table 59.

The additional transition rules of APTC\(_{\tau}^{\text{drt}}\) is shown in Table 60.
Definition 8.3 (Basic terms of \(\text{APTC}^{\text{drt}}\)). The set of basic terms of \(\text{APTC}^{\text{drt}}\), \(\mathcal{B}(\text{APTC}^{\text{drt}})\), is inductively defined as follows by two auxiliary sets \(\mathcal{B}_0(\text{APTC}^{\text{drt}})\) and \(\mathcal{B}_1(\text{APTC}^{\text{drt}})\):

1. if \(a \in A_\tau\), then \(a \in \mathcal{B}_1(\text{APTC}^{\text{drt}})\);
2. if \(a \in A_\tau\) and \(t \in \mathcal{B}(\text{APTC}^{\text{drt}})\), then \(a \cdot t \in \mathcal{B}_1(\text{APTC}^{\text{drt}})\);
3. if \(t, t' \in \mathcal{B}_1(\text{APTC}^{\text{drt}})\), then \(t \parallel t' \in \mathcal{B}_1(\text{APTC}^{\text{drt}})\);
4. if \(t, t' \in \mathcal{B}_1(\text{APTC}^{\text{drt}})\), then \(t \parallel t' \in \mathcal{B}_1(\text{APTC}^{\text{drt}})\);
5. if \(t \in \mathcal{B}_1(\text{APTC}^{\text{drt}})\), then \(t \in \mathcal{B}_1(\text{APTC}^{\text{drt}})\);
6. if \(n > 0\) and \(t \in \mathcal{B}_0(\text{APTC}^{\text{drt}})\), then \(\sigma^n_{\text{rel}}(t) \in \mathcal{B}_0(\text{APTC}^{\text{drt}})\);
7. if \(n > 0\), \(t \in \mathcal{B}_1(\text{APTC}^{\text{drt}})\) and \(t' \in \mathcal{B}_0(\text{APTC}^{\text{drt}})\), then \(t + \sigma^n_{\text{rel}}(t') \in \mathcal{B}_0(\text{APTC}^{\text{drt}})\);
8. \(\delta \in \mathcal{B}(\text{APTC}^{\text{drt}})\);
9. if \(t \in \mathcal{B}_0(\text{APTC}^{\text{drt}})\), then \(t \in \mathcal{B}(\text{APTC}^{\text{drt}})\).

Theorem 8.4 (Elimination theorem). Let \(p\) be a closed \(\text{APTC}^{\text{drt}}\) term. Then there is a basic \(\text{APTC}^{\text{drt}}\) term \(q\) such that \(\text{APTC}^{\text{drt}} \vdash p = q\).

Proof. It is sufficient to induct on the structure of the closed \(\text{APTC}^{\text{drt}}\) term \(p\). It can be proven that \(p\) combined by the constants and operators of \(\text{APTC}^{\text{drt}}\) exists an equal basic term \(q\), and the other operators not included in the basic terms, such as \(v_{\text{rel}}, \tau_{\text{rel}}, \top, \bot, H, \Theta, \varnothing\) and \(\tau_I\) can be eliminated. \(\square\)

8.1.1. Connections

Theorem 8.5 (Conservativity of \(\text{APTC}^{\text{drt}}\)). \(\text{APTC}^{\text{drt}}\) is a conservative extension of \(\text{APTC}^{\text{drt}}\).

Proof. It follows from the following two facts.

1. The transition rules of \(\text{APTC}^{\text{drt}}\) are all source-dependent;
2. The sources of the transition rules of \(\text{APTC}^{\text{drt}}\) contain an occurrence of \(\tau_\Xi\) and \(\tau_I\).

So, \(\text{APTC}^{\text{drt}}\) is a conservative extension of \(\text{APTC}^{\text{drt}}\), as desired. \(\square\)

8.1.2. Congruence

Theorem 8.6 (Congruence of \(\text{APTC}^{\text{drt}}\)). Rooted branching truly concurrent bisimulation equivalences \(\approx_{\text{rbp}}, \approx_{\text{rbs}},\) and \(\approx_{\text{rbhp}}\) are all congruences with respect to \(\text{APTC}^{\text{drt}}\). That is,

- rooted branching pomset bisimulation equivalence \(\approx_{\text{rbp}}\) is a congruence with respect to \(\text{APTC}^{\text{drt}}\);
- rooted branching step bisimulation equivalence \(\approx_{\text{rbs}}\) is a congruence with respect to \(\text{APTC}^{\text{drt}}\);
- rooted branching hp-bisimulation equivalence \(\approx_{\text{rbhp}}\) is a congruence with respect to \(\text{APTC}^{\text{drt}}\).

Proof. It is easy to see that \(\approx_{\text{rbp}}, \approx_{\text{rbs}},\) and \(\approx_{\text{rbhp}}\) are all equivalent relations on \(\text{APTC}^{\text{drt}}\) terms, it is only sufficient to prove that \(\approx_{\text{rbp}}, \approx_{\text{rbs}},\) and \(\approx_{\text{rbhp}}\) are all preserved by the operators \(\tau_I\). It is trivial and we omit it. \(\square\)

8.1.3. Soundness

Theorem 8.7 (Soundness of \(\text{APTC}^{\text{drt}}\)). The axiomatization of \(\text{APTC}^{\text{drt}}\) is sound modulo rooted branching truly concurrent bisimulation equivalences \(\approx_{\text{rbp}}, \approx_{\text{rbs}},\) and \(\approx_{\text{rbhp}}\). That is,

1. let \(x\) and \(y\) be \(\text{APTC}^{\text{drt}}\) terms. If \(\text{APTC}^{\text{drt}} \vdash x = y\), then \(x \approx_{\text{rbs}} y\);
2. let \(x\) and \(y\) be \(\text{APTC}^{\text{drt}}\) terms. If \(\text{APTC}^{\text{drt}} \vdash x = y\), then \(x \approx_{\text{rbp}} y\);
3. let \(x\) and \(y\) be \(\text{APTC}^{\text{drt}}\) terms. If \(\text{APTC}^{\text{drt}} \vdash x = y\), then \(x \approx_{\text{rbhp}} y\).
Proof. Since \( \equiv_{rhp} \), \( \equiv_{rhbs} \), and \( \equiv_{rhhbp} \) are both equivalent and congruent relations, we only need to check if each axiom in Table 59 is sound modulo \( \equiv_{rhp} \), \( \equiv_{rhbs} \), and \( \equiv_{rhhbp} \) respectively.

1. Each axiom in Table 59 can be checked that it is sound modulo rooted branching step bisimulation equivalence, by transition rules in Table 59. We omit them.

2. From the definition of rooted branching pomset bisimulation \( \equiv_{rhp} \), we know that rooted branching pomset bisimulation \( \equiv_{rhp} \) is defined by weak pomset transitions, which are labeled by pomsets with \( \equiv \). In a weak pomset transition, the events in the pomset are either within causality relations (defined by \( \cdot \)) or in concurrency (implicitly defined by \( + \) and \( \cdot \), and explicitly defined by \( \|$). Of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent, so, only need to prove the case of events in causality. Without loss of generality, we take a pomset of \( P = \langle a, b \rangle \). Then the weak pomset transition labeled by the above \( P \) is just composed of one single event transition labeled by \( a \) preceded by another single event transition labeled by \( b \), that is, \( P = \frac{a}{E} \rightarrow \frac{b}{E} \). Similarly to the proof of soundness modulo rooted branching step bisimulation \( \equiv_{rhp} \), we can prove that each axiom in Table 59 is sound modulo rooted branching pomset bisimulation \( \equiv_{rhp} \), we omit them.

3. From the definition of rooted branching hp-bisimulation \( \equiv_{rhhbp} \), we know that rooted branching hp-bisimulation \( \equiv_{rhhbp} \) is defined on the weakly posetal product \( (\hat{C}_1, f, C_2) \). Two process terms \( s \) related to \( C_1 \) and \( t \) related to \( C_2 \), and \( f : \hat{C}_1 \rightarrow \hat{C}_2 \) isomorphism. Initially, \( (\hat{C}_1, f, C_2) = (\varnothing, \varnothing, \varnothing) \) and \( (\varnothing, \varnothing, \varnothing) \) is \( \equiv_{rhhbp} \). Then, if \( (C_1, f, C_2) \in \equiv_{rhhbp} \), then \( (C_1', f', C_2') \in \equiv_{rhhbp} \). Similarly to the proof of soundness modulo rooted branching pomset bisimulation equivalence, we can prove that each axiom in Table 59 is sound modulo rooted branching hp-bisimulation equivalence, we just need additionally to check the above conditions on rooted branching hp-bisimulation, we omit them.

8.1.4. Completeness

For APTC\( _{\tau} \) + Rec, it is similar to APTC\( ^{dirt} \) + Rec, except that \( \tau \cdot X \) is forbidden in recursive specifications for the sake of fairness. Like APTC, the proof of completeness need the help of CFAR (see section 2.1).

Theorem 8.8 (Completeness of APTC\( ^{dirt} \) + CFAR + guarded linear Rec). The axiomatization of APTC\( ^{dirt} \) + CFAR + guarded linear Rec is complete modulo rooted branching truly concurrent bisimulation equivalences \( \equiv_{rhbs} \), \( \equiv_{rhp} \), and \( \equiv_{rhhbp} \). That is,

1. let \( p \) and \( q \) be closed APTC\( ^{dirt} \) + CFAR + guarded linear Rec terms, if \( p \equiv_{rhbs} q \) then \( p = q \);
2. let \( p \) and \( q \) be closed APTC\( ^{dirt} \) + CFAR + guarded linear Rec terms, if \( p \equiv_{rhp} q \) then \( p = q \);
3. let \( p \) and \( q \) be closed APTC\( ^{dirt} \) + CFAR + guarded linear Rec terms, if \( p \equiv_{rhhbp} q \) then \( p = q \).

Proof. Firstly, we know that each process term in APTC\( ^{dirt} \) + CFAR + guarded linear Rec is equal to a process term \( \langle X_1 | E \rangle \) with \( E \) a linear recursive specification.

It remains to prove the following cases.

1. If \( \langle X_1 | E_1 \rangle \equiv_{rhbs} \langle Y_1 | E_2 \rangle \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( \langle X_1 | E_1 \rangle = \langle Y_1 | E_2 \rangle \). It can be proven similarly to the completeness of APTC\( _{\tau} \) + CFAR + linear Rec, see [17].
2. If \( \langle X_1 | E_1 \rangle \equiv_{rhp} \langle Y_1 | E_2 \rangle \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( \langle X_1 | E_1 \rangle = \langle Y_1 | E_2 \rangle \). It can be proven similarly, just by replacement of \( \equiv_{rhbs} \) by \( \equiv_{rhp} \), we omit it.
3. If \( \langle X_1 | E_1 \rangle \equiv_{rhhbp} \langle Y_1 | E_2 \rangle \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( \langle X_1 | E_1 \rangle = \langle Y_1 | E_2 \rangle \). It can be proven similarly, just by replacement of \( \equiv_{rhbs} \) by \( \equiv_{rhhbp} \), we omit it.
Table 61. Additional axioms of $\text{APTC}_\tau^\text{dat} \ (a \in A_{\tau,\delta}, n \geq 0)$

Table 62. Transition rule of $\text{APTC}_\tau^\text{dat} \ (a \in A_{\tau, m > 0}, n \geq 0)$

8.2. Discrete Absolute Timing

Definition 8.9 (Rooted branching truly concurrent bisimulations). The following two conditions related timing should be added into the concepts of branching truly concurrent bisimulations in section [2, 7].

1. if $C_1 \rightarrow^1 C_2$, then there are $C'_1, C'_2$ such that $C'_1 \Rightarrow C'_2$ and $(C_1, C'_1) \in R$ and $(C_2, C'_2) \in R$, or $(C_1, f[\emptyset \rightarrow \emptyset], C'_1) \in R$ and $(C_2, f[\emptyset \rightarrow \emptyset], C'_2) \in R$.

2. if $C_1 \uparrow$, then $C'_1 \uparrow$.

And the following root conditions related timing should be added into the concepts of rooted branching truly concurrent bisimulations in section [2, 7].

1. if $C_1 \rightarrow^m C_2 \ (m > 0)$, then there is $C'_2$ such that $C'_1 \Rightarrow C'_2$, and $(C_2, C'_2) \in R$, or $(C_2, f[\emptyset \rightarrow \emptyset], C'_2) \in R$.

Definition 8.10 (Signature of $\text{APTC}_\tau^\text{dat}$). The signature of $\text{APTC}_\tau^\text{dat}$ consists of the signature of $\text{APTC}_\tau^\text{dat}$, and the undelayable silent step constant $\tau \colon P_{\text{abs}} \rightarrow P_{\text{abs}}$ for $I \in A$.

The axioms of $\text{APTC}_\tau^\text{dat}$ include the laws in Table [15] covering the case that $a \equiv \tau$ and $b \equiv \tau$, and the axioms in Table [61].

The additional transition rules of $\text{APTC}_\tau^\text{dat}$ is shown in Table [62].

Definition 8.11 (Basic terms of $\text{APTC}_\tau^\text{dat}$). The set of basic terms of $\text{APTC}_\tau^\text{dat}$, $B(\text{APTC}_\tau^\text{dat})$, is inductively defined as follows by two auxiliary sets $B_0(\text{APTC}_\tau^\text{dat})$ and $B_1(\text{APTC}_\tau^\text{dat})$:

1. if $a \in A_{\tau}$, then $\underline{a} \in B_1(\text{APTC}_\tau^\text{dat})$;
2. if $a \in A_{\tau}$ and $t \in B(\text{APTC}_\tau^\text{dat})$, then $\underline{a} \cdot t \in B_1(\text{APTC}_\tau^\text{dat})$;
3. if $t, t' \in B_1(\text{APTC}_\tau^\text{dat})$, then $t + t' \in B_1(\text{APTC}_\tau^\text{dat})$;
4. if $t, t' \in B_1(\text{APTC}_\tau^\text{dat})$, then $t \parallel t' \in B_1(\text{APTC}_\tau^\text{dat})$;
5. if \( t \in B_1(\text{APT}C^\text{dat}_\tau) \), then \( t \in B_0(\text{APT}C^\text{dat}_\tau) \);
6. if \( n > 0 \) and \( t \in B_0(\text{APT}C^\text{dat}_\tau) \), then \( \sigma^n_{\text{abs}}(t) \in B_0(\text{APT}C^\text{dat}_\tau) \);
7. if \( n > 0 \), \( t \in B_1(\text{APT}C^\text{dat}_\tau) \) and \( t' \in B_0(\text{APT}C^\text{dat}_\tau) \), then \( t + \sigma^n_{\text{abs}}(t') \in B_0(\text{APT}C^\text{dat}_\tau) \);
8. \( \delta \in B(\text{APT}C^\text{dat}_\tau) \);
9. if \( t \in B_0(\text{APT}C^\text{dat}_\tau) \), then \( t \in B(\text{APT}C^\text{dat}_\tau) \).

**Theorem 8.12** (Elimination theorem). Let \( p \) be a closed \( \text{APT}C^\text{dat}_\tau \) term. Then there is a basic \( \text{APT}C^\text{dat}_\tau \) term \( q \) such that \( \text{APT}C^\text{dat}_\tau \vdash p = q \).

**Proof.** It is sufficient to induct on the structure of the closed \( \text{APT}C^\text{dat}_\tau \) term \( p \). It can be proven that \( p \) combined by the constants and operators of \( \text{APT}C^\text{dat}_\tau \) exists an equal basic term \( q \), and the other operators not included in the basic terms, such as \( v_{\text{abs}}, \overline{v}_{\text{abs}}, \frac{q}{\cdot}, \frac{q}{\mid}, \partial_H, \Theta, \angle \) and \( \tau_I \) can be eliminated. \( \square \)

8.2.1. Connections

**Theorem 8.13** (Conservativity of \( \text{APT}C^\text{dat}_\tau \)). \( \text{APT}C^\text{dat}_\tau \) is a conservative extension of \( \text{APT}C^\text{dat} \).

**Proof.** It follows from the following two facts.

1. The transition rules of \( \text{APT}C^\text{dat} \) are all source-dependent;
2. The sources of the transition rules of \( \text{APT}C^\text{dat} \) contain an occurrence of \( \tau \) and \( \tau_I \).

So, \( \text{APT}C^\text{dat} \) is a conservative extension of \( \text{APT}C^\text{dat} \), as desired. \( \square \)

8.2.2. Congruence

**Theorem 8.14** (Congruence of \( \text{APT}C^\text{dat}_\tau \)). Rooted branching truly concurrent bisimulation equivalences \( \equiv_{\text{rbp}}, \equiv_{\text{rbs}}, \) and \( \equiv_{\text{rhp}} \) are all congruences with respect to \( \text{APT}C^\text{dat}_\tau \). That is,

- rooted branching pomset bisimulation equivalence \( \equiv_{\text{rbp}} \) is a congruence with respect to \( \text{APT}C^\text{dat} \);
- rooted branching step bisimulation equivalence \( \equiv_{\text{rbs}} \) is a congruence with respect to \( \text{APT}C^\text{dat}_\tau \);
- rooted branching hp-bisimulation equivalence \( \equiv_{\text{rhp}} \) is a congruence with respect to \( \text{APT}C^\text{dat}_\tau \).

**Proof.** It is easy to see that \( \equiv_{\text{rbp}}, \equiv_{\text{rbs}}, \) and \( \equiv_{\text{rhp}} \) are all equivalent relations on \( \text{APT}C^\text{dat}_\tau \) terms, it is only sufficient to prove that \( \equiv_{\text{rbp}}, \equiv_{\text{rbs}}, \) and \( \equiv_{\text{rhp}} \) are all preserved by the operators \( \tau_I \). It is trivial and we omit it. \( \square \)

8.2.3. Soundness

**Theorem 8.15** (Soundness of \( \text{APT}C^\text{dat}_\tau \)). The axiomatization of \( \text{APT}C^\text{dat}_\tau \) is sound modulo rooted branching truly concurrent bisimulation equivalences \( \equiv_{\text{rbp}}, \equiv_{\text{rbs}}, \) and \( \equiv_{\text{rhp}} \). That is,

1. let \( x \) and \( y \) be \( \text{APT}C^\text{dat}_\tau \) terms. If \( \text{APT}C^\text{dat}_\tau \vdash x = y \), then \( x \equiv_{\text{rbs}} y \);
2. let \( x \) and \( y \) be \( \text{APT}C^\text{dat}_\tau \) terms. If \( \text{APT}C^\text{dat}_\tau \vdash x = y \), then \( x \equiv_{\text{rhp}} y \);
3. let \( x \) and \( y \) be \( \text{APT}C^\text{dat}_\tau \) terms. If \( \text{APT}C^\text{dat}_\tau \vdash x = y \), then \( x \equiv_{\text{rhp}} y \).

**Proof.** Since \( \equiv_{\text{rbp}}, \equiv_{\text{rbs}}, \) and \( \equiv_{\text{rhp}} \) are both equivalent and congruent relations, we only need to check if each axiom in Table [51] is sound modulo \( \equiv_{\text{rbp}}, \equiv_{\text{rbs}}, \) and \( \equiv_{\text{rhp}} \) respectively.

1. Each axiom in Table [61] can be checked that it is sound modulo rooted branching step bisimulation equivalence, by transition rules in Table [62]. We omit them.
2. From the definition of rooted branching pomset bisimulation \( \equiv_{\text{rbp}} \), we know that rooted branching pomset bisimulation \( \equiv_{\text{rbp}} \) is defined by weak pomset transitions, which are labeled by pomsets with \( \tau \). In a weak pomset transition, the events in the pomset are either within causality relations (defined by \( \cdot \)) or in concurrency (implicitly defined by \( \cdot \) and \( + \), and explicitly defined by \( \angle \)), of course, they are pairwise
consistent (without conflicts). We have already proven the case that all events are pairwise concurrent, so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of $P = \{a \cdot b : a \cdot b\}$. Then the weak pomset transition labeled by the above $P$ is just composed of one single event transition labeled by $a$ succeeded by another single event transition labeled by $b$ that is, $P \Rightarrow a \Rightarrow b$.

Similarly to the proof of soundness modulo rooted branching step bisimulation $\approx_{rbp}$, we can prove that each axiom in Table 61 is sound modulo rooted branching pomset bisimulation $\approx_{rbhp}$, we omit them.

3. From the definition of rooted branching hp-bisimulation $\approx_{rbhp}$, we know that rooted branching hp-bisimulation $\approx_{rbhp}$ is defined on the weakly posetal product $(C_1, f, C_2, \tau)$ isomorphism. Two process terms $s$ related to $C_1$ and $t$ related to $C_2$, and $f : \hat{C}_1 \rightarrow \hat{C}_2$ isomorphism. Initially, $(C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)$, and $(\emptyset, \emptyset, \emptyset) \approx_{rbhp}$ when $s \Rightarrow s' (C_1 \Rightarrow C_1')$, there will be $t \Rightarrow t' (C_2 \Rightarrow C_2')$, and we define $f' = f[a \Rightarrow a]$. Then, if $(C_1, f, C_2) \approx_{rbhp}$, then $(C_1', f', C_2') \approx_{rbhp}$.

Similarly to the proof of soundness modulo rooted branching pomset bisimulation equivalence, we can prove that each axiom in Table 61 is sound modulo rooted branching hp-bisimulation equivalence, we just need additionally to check the above conditions on rooted branching hp-bisimulation, we omit them.

8.2.4. Completeness

For APTC$_{\text{dat}}$ + Rec, it is similar to APTC$_{\text{dat}}$ + Rec, except that $\tau \cdot X$ is forbidden in recursive specifications for the sake of fairness. Like APTC, the proof of completeness need the help of CFAR (see section 2.1).

**Theorem 8.16** (Completeness of APTC$_{\text{dat}}$ + CFAR + guarded linear Rec). The axiomatization of APTC$_{\text{dat}}$ + CFAR + guarded linear Rec is complete modulo rooted branching truly concurrent bisimulation equivalences $\approx_{rbhs}$, $\approx_{rbhp}$, and $\approx_{rbhp}$. That is,

1. let $p$ and $q$ be closed APTC$_{\text{dat}}$ + CFAR + guarded linear Rec terms, if $p \approx_{rbhs} q$ then $p = q$;
2. let $p$ and $q$ be closed APTC$_{\text{dat}}$ + CFAR + guarded linear Rec terms, if $p \approx_{rbhp} q$ then $p = q$;
3. let $p$ and $q$ be closed APTC$_{\text{dat}}$ + CFAR + guarded linear Rec terms, if $p \approx_{rbhp} q$ then $p = q$.

**Proof.** Firstly, we know that each process term in APTC$_{\text{dat}}$ + CFAR + guarded linear Rec is equal to a process term $(X_1|E)$ with $E$ a linear recursive specification.

It remains to prove the following cases.

1. If $(X_1|E_1) \approx_{rbhs} (Y_1|E_2)$ for linear recursive specification $E_1$ and $E_2$, then $(X_1|E_1) = (Y_1|E_2)$. It can be proven similarly to the completeness of APTC$_{\tau}$ + CFAR + linear Rec, see [17].
2. If $(X_1|E_1) \approx_{rbhp} (Y_1|E_2)$ for linear recursive specification $E_1$ and $E_2$, then $(X_1|E_1) = (Y_1|E_2)$. It can be proven similarly, just by replacement of $\approx_{rbhs}$ by $\approx_{rbhp}$, we omit it.
3. If $(X_1|E_1) \approx_{rbhp} (Y_1|E_2)$ for linear recursive specification $E_1$ and $E_2$, then $(X_1|E_1) = (Y_1|E_2)$. It can be proven similarly, just by replacement of $\approx_{rbhs}$ by $\approx_{rbhp}$, we omit it.

8.3. Continuous Relative Timing

**Definition 8.17** (Rooted branching truly concurrent bisimulations). The following two conditions related timing should be added into the concepts of branching truly concurrent bisimulations in section 2.1.

1. if $C_1 \Rightarrow C_2 (r > 0)$, then either there are $C_1', C_2', C_1''$ and $r' : 0 < r' < r$ such that $C_1' \Rightarrow C_1'' \Rightarrow C_2'$ and $C_2' \Rightarrow r' C_2'' (C_1', C_2' \in R)$ and $(C_1', C_1'') \in R$ and $(C_2', C_2'') \in R$, or $(C_1, f[\emptyset \Rightarrow \emptyset], C_1') \in R$ and $(C_2, f[\emptyset \Rightarrow \emptyset], C_2') \in R$; or there are $C_1', C_2'$ such that $C_1' \Rightarrow C_1'', C_2' \Rightarrow r' C_2'', (C_1', C_2') \in R$ and $(C_2', C_2'') \in R$; or there are $C_1', C_2'$ such that $C_1' \Rightarrow C_1'' \Rightarrow r' C_2'$, and $(C_1, C_1') \in R$ and $(C_2, C_2') \in R$,
2. if $C_1 \uparrow$, then $C_1' \uparrow$.

And the following root conditions related timing should be added into the concepts of rooted branching truly concurrent bisimulations in section 2.1.
10. If Theorem 8.20

SRT B1 \( x \cdot (\tilde{\tau} \cdot (\nu_{rel}(y) + z + \tilde{\delta}) + \nu_{rel}(y)) = x \cdot (\nu_{rel}(y) + z + \tilde{\delta}) \)

SRT B2 \( x \cdot (\tilde{\tau} \cdot (\nu_{rel}(y) + z + \tilde{\delta}) + z) = x \cdot (\nu_{rel}(y) + z + \tilde{\delta}) \)

SRT B3 \( x \cdot (\sigma_{rel}(\tilde{\tau} \cdot (y + \tilde{\delta}) + \nu_{rel}(z))) = x \cdot (\sigma_{rel}(y + \tilde{\delta}) + \nu_{rel}(z)) \)

B3 \( \alpha \parallel \beta = \alpha \)

T10 \( \tau_I(\tilde{\delta}) = \tilde{\delta} \)

T11 \( \alpha \in I \quad \tau_I(\tilde{\delta}) = \tilde{\alpha} \)

T12 \( \alpha \in I \quad \tau_I(\tilde{\alpha}) = \tilde{\alpha} \)

SRT1 \( \tau_I(\sigma_{rel}(x)) = \sigma_{rel}(\tau_I(x)) \)

T14 \( \tau_I(x + y) = \tau_I(x) + \tau_I(y) \)

T15 \( \tau_I(x \cdot y) = \tau_I(x) \cdot \tau_I(y) \)

T16 \( \tau_I(x \parallel y) = \tau_I(x) \parallel \tau_I(y) \)

Table 63. Additional axioms of APTC_{sr}^{\tau} (a \in A_\tau, p \geq 0, r > 0)

\[ \frac{x \stackrel{a}{\to}}{\tau_I(x) \stackrel{\tau_I(\tilde{\alpha})}{\to}} \quad \frac{x \stackrel{\alpha}{\overleftarrow{\to}}}{\tau_I(x) \stackrel{a}{\overleftarrow{\to}} \tau_I(x')} \quad \frac{x \stackrel{a}{\to}}{\tau_I(x) \stackrel{a}{\to} \tau_I(x')} \]

Table 64. Transition rule of APTC_{sr}^{\tau} (a \in A_\tau, r > 0, p \geq 0)

1. if \( C_1 \rightarrow^r C_2 (r > 0) \), then there is \( C_1' \rightarrow^r C_2' \), such that \( C_1' \rightarrow^r C_2' \) and \( (C_2, C_2') \in R \), or \( (C_2, f[\varnothing \mapsto \varnothing], C_2') \in R \).

**Definition 8.18** (Signature of APTC_{sr}^{\tau}). The signature of APTC_{sr}^{\tau} consists of the signature of APTC_{sr}, and the undelayable silent step constant \( {\tilde{\tau}} : \text{P}_{\text{rel}} \rightarrow \text{P}_{\text{rel}} \) for \( I \subseteq A \).

The axioms of APTC_{sr}^{\tau} include the laws in Table 63 covering the case that \( a \equiv \tau \) and \( b \equiv \tau \), and the axioms in Table 63.

The additional transition rules of APTC_{sr}^{\tau} is shown in Table 64.

**Definition 8.19** (Basic terms of APTC_{sr}^{\tau}). The set of basic terms of APTC_{sr}^{\tau}, \( B(\text{APTC}_{sr}^{\tau}) \), is inductively defined as follows by two auxiliary sets \( B_0(\text{APTC}_{sr}^{\tau}) \) and \( B_1(\text{APTC}_{sr}^{\tau}) \):

1. if \( a \in A_\tau \), then \( \tilde{a} \in B_1(\text{APTC}_{sr}^{\tau}) \);
2. if \( a \in A_\tau \) and \( t \in B(\text{APTC}_{sr}^{\tau}) \), then \( \tilde{a} \cdot t \in B_1(\text{APTC}_{sr}^{\tau}) \);
3. if \( t, t' \in B_1(\text{APTC}_{sr}^{\tau}) \), then \( t + t' \in B_1(\text{APTC}_{sr}^{\tau}) \);
4. if \( t \in B_1(\text{APTC}_{sr}^{\tau}) \), then \( t \parallel t' \in B_1(\text{APTC}_{sr}^{\tau}) \);
5. if \( t \in B_1(\text{APTC}_{sr}^{\tau}) \), then \( t \in B_0(\text{APTC}_{sr}^{\tau}) \);
6. if \( p > 0 \) and \( t \in B_0(\text{APTC}_{sr}^{\tau}) \), then \( \sigma_{rel}(p, t) \in B_0(\text{APTC}_{sr}^{\tau}) \);
7. if \( p > 0 \), \( t \in B_1(\text{APTC}_{sr}^{\tau}) \) and \( t' \in B_0(\text{APTC}_{sr}^{\tau}) \), then \( t + \sigma_{rel}(p, t') \in B_0(\text{APTC}_{sr}^{\tau}) \);
8. if \( t \in B_0(\text{APTC}_{sr}^{\tau}) \), then \( \nu_{rel}(t) \in B_0(\text{APTC}_{sr}^{\tau}) \);
9. if \( \tilde{\delta} \in B(\text{APTC}_{sr}^{\tau}) \);
10. if \( t \in B_0(\text{APTC}_{sr}^{\tau}) \), then \( t \in B(\text{APTC}_{sr}^{\tau}) \).

**Theorem 8.20** (Elimination theorem). Let \( p \) be a closed APTC_{sr}^{\tau} term. Then there is a basic APTC_{sr}^{\tau} term \( q \) such that \( \text{APTC}_{sr}^{\tau} \vdash p = q \).
Proof. It is sufficient to induct on the structure of the closed \(\text{APTC}_{\tau}^{\text{crt}}\) term \(p\). It can be proven that \(p\) combined by the constants and operators of \(\text{APTC}_{\tau}^{\text{crt}}\) exists an equal basic term \(q\), and the other operators not included in the basic terms, such as \(v_{\text{rel}}, \pi_{\text{rel}}, \gamma, \partial_{H}, \Theta, <\) and \(\tau_{I}\) can be eliminated.

8.3.1. Connections

**Theorem 8.21** (Conservativity of \(\text{APTC}_{\tau}^{\text{crt}}\)). \(\text{APTC}_{\tau}^{\text{crt}}\) is a conservative extension of \(\text{APTC}_{\tau}^{\text{crt}}\).

Proof. It follows from the following two facts.

1. The transition rules of \(\text{APTC}_{\tau}^{\text{crt}}\) are all source-dependent;
2. The sources of the transition rules of \(\text{APTC}_{\tau}^{\text{crt}}\) contain an occurrence of \(\hat{\tau}\) and \(\tau_{I}\).

So, \(\text{APTC}_{\tau}^{\text{crt}}\) is a conservative extension of \(\text{APTC}_{\tau}^{\text{crt}}\), as desired.

8.3.2. Congruence

**Theorem 8.22** (Congruence of \(\text{APTC}_{\tau}^{\text{crt}}\)). Rooted branching truly concurrent bisimulation equivalences \(\sim_{\text{rbp}}\), \(\sim_{\text{rbs}}\), and \(\sim_{\text{rhh}}\) are all congruences with respect to \(\text{APTC}_{\tau}^{\text{crt}}\). That is,

- rooted branching pomset bisimulation equivalence \(\sim_{\text{rbp}}\) is a congruence with respect to \(\text{APTC}_{\tau}^{\text{crt}}\);
- rooted branching step bisimulation equivalence \(\sim_{\text{rbs}}\) is a congruence with respect to \(\text{APTC}_{\tau}^{\text{crt}}\);
- rooted branching hp-bisimulation equivalence \(\sim_{\text{rhh}}\) is a congruence with respect to \(\text{APTC}_{\tau}^{\text{crt}}\).

Proof. It is easy to see that \(\sim_{\text{rbp}}, \sim_{\text{rbs}},\) and \(\sim_{\text{rhh}}\) are all equivalent relations on \(\text{APTC}_{\tau}^{\text{crt}}\) terms, it is only sufficient to prove that \(\sim_{\text{rbp}}, \sim_{\text{rbs}},\) and \(\sim_{\text{rhh}}\) are all preserved by the operators \(\tau_{I}\). It is trivial and we omit it.

8.3.3. Soundness

**Theorem 8.23** (Soundness of \(\text{APTC}_{\tau}^{\text{crt}}\)). The axiomatization of \(\text{APTC}_{\tau}^{\text{crt}}\) is sound modulo rooted branching truly concurrent bisimulation equivalence \(\sim_{\text{rbp}}, \sim_{\text{rbs}},\) and \(\sim_{\text{rhh}}\). That is,

1. Each axiom in Table 63 can be checked that it is sound modulo rooted branching step bisimulation equivalence, by transition rules in Table 64.
2. From the definition of rooted branching pomset bisimulation \(\sim_{\text{rbp}}\), we know that rooted branching pomset bisimulation \(\sim_{\text{rbp}}\) is defined by weak pomset transitions, which are labeled by pomsets with \(\hat{\tau}\). In a weak pomset transition, the events in the pomset are either within causality relations (defined by \(\cdot\)) or in concurrency (implicitly defined by \(\cdot\) and \(\tau\), and explicitly defined by \(\gamma\)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent, so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \(P = \{\hat{a}, b; \hat{a} \cdot b\}\). Then the weak pomset transition labeled by the above \(P\) is just composed of one single event transition labeled by \(\hat{a}\) succeeded by another single event transition labeled by \(\hat{b}\), that is, \(P = \hat{a} \cdot b\). Similarly to the proof of soundness modulo rooted branching step bisimulation \(\sim_{\text{rbs}}\), we can prove that each axiom in Table 63 is sound modulo rooted branching pomset bisimulation \(\sim_{\text{rbp}}\), we omit them.
3. From the definition of rooted branching hp-bisimulation \(\sim_{\text{rhh}}\), we know that rooted branching hp-bisimulation \(\sim_{\text{rhh}}\) is defined on the weakly posetal product \((C_{1}, f, C_{2}), f : \hat{C}_{1} \rightarrow \hat{C}_{2}\) isomorphism. Two process terms \(s\) related to \(C_{1}\) and \(t\) related to \(C_{2}\), and \(f : \hat{C}_{1} \rightarrow \hat{C}_{2}\) isomorphism. Initially,
(C_1, f, C_2) = (∅, ∅, ∅), and (∅, ∅, ∅) ≈_{r_{bhp}}. When s \xrightarrow{a} s' (C_1 \xrightarrow{a} C'_1), there will be t \xrightarrow{a} t' (C_2 \xrightarrow{a} C'_2), and we define f' = f[a \mapsto a]. Then, if \( (C_1, f, C_2) \approx_{r_{bhp}} \), then \( (C'_1, f', C'_2) \approx_{r_{bhp}} \).

Similarly to the proof of soundness modulo rooted branching pomset bisimulation equivalence, we can prove that each axiom in Table 65 is sound modulo rooted branching hp-bisimulation equivalence, we just need additionally to check the above conditions on rooted branching hp-bisimulation, we omit them.

\[ \square \]

### 8.3.4. Completeness

For APTC^{\tau}_r + Rec, it is similar to APTC^{\tau}_r + Rec, except that \( \tilde{\tau} \cdot X \) is forbidden in recursive specifications for the sake of fairness. Like APTC, the proof of completeness need the help of CFAR (see section 2.1).

**Theorem 8.24** (Completeness of APTC^{\tau}_r + CFAR + guarded linear Rec). The axiomatization of APTC^{\tau}_r + CFAR + guarded linear Rec is complete modulo rooted branching truly concurrent bisimulation equivalences \( \approx_{rbs}, \approx_{r_{bhp}}, \) and \( \approx_{r_{bhp}} \). That is,

1. let \( p \) and \( q \) be closed APTC^{\tau}_r + CFAR + guarded linear Rec terms, if \( p \approx_{rbs} q \) then \( p = q \);
2. let \( p \) and \( q \) be closed APTC^{\tau}_r + CFAR + guarded linear Rec terms, if \( p \approx_{r_{tb}} q \) then \( p = q \);
3. let \( p \) and \( q \) be closed APTC^{\tau}_r + CFAR + guarded linear Rec terms, if \( p \approx_{r_{bhp}} q \) then \( p = q \).

**Proof.** Firstly, we know that each process term in APTC^{\tau}_r + CFAR + guarded linear Rec is equal to a process term \( (X|E) \) with \( E \) a linear recursive specification.

It remains to prove the following cases.

1. If \( (X|E_1) \approx_{rbs} (Y|E_2) \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( (X|E_1) = (Y|E_2) \).
   It can be proven similarly to the completeness of APTC^{\tau}_r + CFAR + linear Rec, see [17].
2. If \( (X|E_1) \approx_{r_{tb}} (Y|E_2) \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( (X|E_1) = (Y|E_2) \).
   It can be proven similarly, just by replacement of \( \approx_{rbs} \) by \( \approx_{r_{bhp}} \), we omit it.
3. If \( (X|E_1) \approx_{r_{bhp}} (Y|E_2) \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( (X|E_1) = (Y|E_2) \).
   It can be proven similarly, just by replacement of \( \approx_{rbs} \) by \( \approx_{r_{bhp}} \), we omit it.

\[ \square \]

### 8.4. Continuous Absolute Timing

**Definition 8.25** (Rooted branching truly concurrent bisimulations). The following two conditions related timing should be added into the concepts of branching truly concurrent bisimulations in section 2.1.

1. if \( C_1 \Rightarrow C_2 (r > 0) \), then either there are \( C'_1, C'_2, C''_2 \) and \( r' : 0 < r' < r \) such that \( C'_1 \Rightarrow C''_2 \) and \( C'_2 \Rightarrow r' C'_2 \), and \((C_1, C'_1) \in R \) and \((C_2, C'_2) \in R \), or \( (C_1, f[\emptyset \Rightarrow \emptyset], C'_1) \in R \) and \((C_2, f[\emptyset \Rightarrow \emptyset], C''_2) \in R \); or there are \( C'_1, C'_2 \) such that \( C'_1 \Rightarrow C'_2 \Rightarrow C'_2 \), and \((C_1, C'_1) \in R \) and \((C_2, C'_2) \in R \), or \((C_1, f[\emptyset \Rightarrow \emptyset], C'_1) \in R \) and \((C_2, C'_2) \in R \).
2. if \( C_1 \uparrow \), then \( C'_1 \uparrow \).

And the following root conditions related timing should be added into the concepts of rooted branching truly concurrent bisimulations in section 2.7.

1. if \( C_1 \Rightarrow C_2 (r > 0) \), then there is \( C''_2 \) such that \( C'_1 \Rightarrow C''_2 \), and \((C_2, C''_2) \in R \), or \((C_2, f[\emptyset \Rightarrow \emptyset], C''_2) \in R \).

**Definition 8.26** (Signature of APTC^{\tau}_r). The signature of APTC^{\tau}_r consists of the signature of APTC^{\tau}, and the undelayable silent step constant \( \tau : \mathcal{P}_{abs} \rightarrow \mathcal{P}_{abs} \) for \( I \subseteq A \).

The axioms of APTC^{\tau}_r include the laws in Table 60 covering the case that \( a \equiv \tau \) and \( b \equiv \tau \), and the axioms in Table 65.

The additional transition rules of APTC^{\tau}_r is shown in Table 66.
Theorem 8.28 (Elimination theorem). Let \( p \) be a closed \( \text{APTC}_{\tau}^{\text{sat}} \) term. Then there is a basic \( \text{APTC}_{\tau}^{\text{sat}} \) term \( q \) such that \( \text{APTC}_{\tau}^{\text{sat}} \vdash p = q \).

**Proof.** It is sufficient to induct on the structure of the closed \( \text{APTC}_{\tau}^{\text{sat}} \) term \( p \). It can be proven that \( p \) combined by the constants and operators of \( \text{APTC}_{\tau}^{\text{sat}} \) exists an equal basic term \( q \), and the other operators not included in the basic terms, such as \( \nu_{\text{abs}}, \sigma_{\text{abs}}, \ll, \|, \partial_H, \Theta, \prec \) and \( \tau_I \) can be eliminated.

### 8.4.1 Connections

**Theorem 8.29** (Conservativity of \( \text{APTC}_{\tau}^{\text{sat}} \)). \( \text{APTC}_{\tau}^{\text{sat}} \) is a conservative extension of \( \text{APTC}_{\tau}^{\text{sat}} \).

**Proof.** It follows from the following two facts.

1. The transition rules of \( \text{APTC}_{\tau}^{\text{sat}} \) are all source-dependent;
2. The sources of the transition rules of $\text{APTC}_{\tau}^{\text{sat}}$ contain an occurrence of $\hat{\tau}$, and $\tau_f$.

So, $\text{APTC}_{\tau}^{\text{sat}}$ is a conservative extension of $\text{APTC}_{\tau}^{\text{sat}}$, as desired.

\begin{proof}
\end{proof}

8.4.2. Congruence

\textbf{Theorem 8.30 (Congruence of $\text{APTC}_{\tau}^{\text{sat}}$).} Rooted branching truly concurrent bisimulation equivalences $\approx_{\text{rbp}}$, $\approx_{\text{rbs}}$, and $\approx_{\text{rbhp}}$ are all congruences with respect to $\text{APTC}_{\tau}^{\text{sat}}$. That is,

- rooted branching pomset bisimulation equivalence $\approx_{\text{rbp}}$ is a congruence with respect to $\text{APTC}_{\tau}^{\text{sat}}$;
- rooted branching step bisimulation equivalence $\approx_{\text{rbs}}$ is a congruence with respect to $\text{APTC}_{\tau}^{\text{sat}}$;
- rooted branching hp-bisimulation equivalence $\approx_{\text{rbhp}}$ is a congruence with respect to $\text{APTC}_{\tau}^{\text{sat}}$.

\textbf{Proof.} It is easy to see that $\approx_{\text{rbp}}$, $\approx_{\text{rbs}}$, and $\approx_{\text{rbhp}}$ are all equivalent relations on $\text{APTC}_{\tau}^{\text{sat}}$ terms, it is only sufficient to prove that $\approx_{\text{rbp}}$, $\approx_{\text{rbs}}$, and $\approx_{\text{rbhp}}$ are all preserved by the operators $\tau_f$. It is trivial and we omit it.

\begin{proof}
\end{proof}

8.4.3. Soundness

\textbf{Theorem 8.31 (Soundness of $\text{APTC}_{\tau}^{\text{sat}}$).} The axiomatization of $\text{APTC}_{\tau}^{\text{sat}}$ is sound modulo rooted branching truly concurrent bisimulation equivalence, by transition rules in Table 65. We omit them.

1. Each axiom in Table 65 can be checked that it is sound modulo rooted branching step bisimulation equivalence, by transition rules in Table 66. We omit them.

2. From the definition of rooted branching pomset bisimulation $\approx_{\text{rbp}}$, we know that rooted branching pomset bisimulation $\approx_{\text{rbp}}$ is defined by weak pomset transitions, which are labeled by pomsets with $\hat{\tau}$. In a weak pomset transition, the events in the pomset are either within causality relations (defined by $\cdot$) or in concurrency (implicitly defined by $\cdot$ and $+$, and explicitly defined by $\|$), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent, so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of $P = \{\hat{a}, \hat{b}; \hat{a} \cdot \hat{b}\}$. Then the weak pomset transition labeled by the above $P$ is just composed of one single event transition labeled by $\hat{a}$ succeeded by another single event transition labeled by $\hat{b}$, that is, $\overset{P}{\Rightarrow} \overset{\hat{a}}{\Rightarrow} \overset{\hat{b}}{\Rightarrow}$. Similarly to the proof of soundness modulo rooted branching step bisimulation $\approx_{\text{rbs}}$, we can prove that each axiom in Table 65 is sound modulo rooted branching pomset bisimulation $\approx_{\text{rbp}}$, we omit them.

3. From the definition of rooted branching hp-bisimulation $\approx_{\text{rbhp}}$, we know that rooted branching hp-bisimulation $\approx_{\text{rbhp}}$ is defined on the weakly posetal product $(C_1, f, C_2, f : C_1 \to C_2)$ isomorphism. Two process terms $s$ related to $C_1$ and $t$ related to $C_2$, and $f : C_1 \to C_2$ isomorphism. Initially, $(C_1, f, C_2) = (\varnothing, \varnothing, \varnothing)$, and $(\varnothing, \varnothing, \varnothing) \in \approx_{\text{rbhp}}$. When $s \overset{\hat{a}}{\Rightarrow} s' (C_1 \overset{\hat{a}}{\Rightarrow} C'_1)$, there will be $t \overset{\hat{a}}{\Rightarrow} t' (C_2 \overset{\hat{a}}{\Rightarrow} C'_2)$, and we define $f' = f[a \mapsto a]$. Then, if $(C_1, f, C_2) \in \approx_{\text{rbhp}}$, then $(C'_1, f', C'_2) \in \approx_{\text{rbhp}}$.

Similarly to the proof of soundness modulo rooted branching pomset bisimulation equivalence, we can prove that each axiom in Table 65 is sound modulo rooted branching hp-bisimulation equivalence, we just need additionally to check the above conditions on rooted branching hp-bisimulation, we omit them.

\begin{proof}
\end{proof}

8.4.4. Completeness

For $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec}$, it is similar to $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec}$, except that $\hat{\tau} \cdot X$ is forbidden in recursive specifications for the sake of fairness. Like APTC, the proof of completeness need the help of $\text{CFAR}$ (see section 2.21).
Theorem 8.32 (Completeness of APTC\textsuperscript{sat} + CFAR + guarded linear Rec). The axiomatization of APTC\textsuperscript{dat} + CFAR + guarded linear Rec is complete modulo rooted branching truly concurrent bisimulation equivalences \( \approx_{rbx}, \approx_{rhp}, \) and \( \approx_{rbhp} \). That is,

1. let \( p \) and \( q \) be closed APTC\textsuperscript{sat} + CFAR + guarded linear Rec terms, if \( p \approx_{rbx} q \) then \( p = q \);
2. let \( p \) and \( q \) be closed APTC\textsuperscript{sat} + CFAR + guarded linear Rec terms, if \( p \approx_{rhp} q \) then \( p = q \);
3. let \( p \) and \( q \) be closed APTC\textsuperscript{sat} + CFAR + guarded linear Rec terms, if \( p \approx_{rbhp} q \) then \( p = q \).

Proof. Firstly, we know that each process term in APTC\textsuperscript{sat} + CFAR + guarded linear Rec is equal to a process term \( (X_1|E) \) with \( E \) a linear recursive specification.

It remains to prove the following cases.

1. If \( (X_1|E_1) \approx_{rbx} (Y_1|E_2) \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( (X_1|E_1) = (Y_1|E_2) \).

   It can be proven similarly to the completeness of APTC\textsuperscript{sat} + CFAR + linear Rec, see \cite{17}.

2. If \( (X_1|E_1) \approx_{rhp} (Y_1|E_2) \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( (X_1|E_1) = (Y_1|E_2) \).

   It can be proven similarly, just by replacement of \( \approx_{rbx} \) by \( \approx_{rhp} \), we omit it.

3. If \( (X_1|E_1) \approx_{rbhp} (Y_1|E_2) \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( (X_1|E_1) = (Y_1|E_2) \).

   It can be proven similarly, just by replacement of \( \approx_{rbx} \) by \( \approx_{rbhp} \), we omit it.

9. Applications

APTC with timing provides a formal framework based on truly concurrent behavioral semantics, which can be used to verify the correctness of system behaviors with timing. In this section, we choose one protocol verified by APTC \cite{17} – alternating bit protocol (ABP) \cite{22}.

The ABP protocol is used to ensure successful transmission of data through a corrupted channel. This success is based on the assumption that data can be resent an unlimited number of times, which is illustrated in Fig 4 we alter it into the true concurrency situation.

1. Data elements \( d_1, d_2, d_3, \ldots \) from a finite set \( \Delta \) are communicated between a Sender and a Receiver.
2. If the Sender reads a datum from channel \( A_1 \), then this datum is sent to the Receiver in parallel through channel \( A_2 \).
3. The Sender processes the data in \( \Delta \), forms new data, and sends them to the Receiver through channel \( B \).
4. And the Receiver sends the datum into channel \( C \).
5. If channel \( B \) is corrupted, the message communicated through \( B \) can be turn into an error message \( \bot \).
6. Every time the Receiver receives a message via channel \( B \), it sends an acknowledgement to the Sender via channel \( D \), which is also corrupted.
7. Finally, then Sender and the Receiver send out their outputs in parallel through channels \( C_1 \) and \( C_2 \).

In the truly concurrent ABP, the Sender sends its data to the Receiver; and the Receiver can also send its data to the Sender, for simplicity and without loss of generality, we assume that only the Sender sends its data and the Receiver only receives the data from the Sender. The Sender attaches a bit 0 to data elements \( d_{2k-1} \) and a bit 1 to data elements \( d_{2k} \), when they are sent into channel \( B \). When the Receiver reads a datum, it sends back the attached bit via channel \( D \). If the Receiver receives a corrupted message, then it sends back the previous acknowledgement to the Sender.

9.1. Discrete Relative Timing

Time is divided into slices, \( t_1, t_2 \) are the times that it takes the different processes to send, receive, etc. \( t'_1 \) is the time-out time of the sender and \( t'_2 \) is the time-out of the receiver.

Then the state transition of the Sender can be described by APTC\textsuperscript{drt} + Rec as follows.
Fig. 1. Alternating bit protocol

\[ S_b = \sum_{d \in \Delta} r_{A_1}(d) \cdot T_{db} \]
\[ T_{db} = (\sum_{d' \in \Delta} (s_B(d', b) \cdot s_{C_1}(d')) + s_B(\bot)) \cdot \sigma_{rel}^l(U_{db}) + \sigma_{rel}^l(T_{db}) \]
\[ U_{db} = \sum_{k \in \Delta} \sigma_{rel}^k(r_D(b)) \cdot S_{1-b} + \sum_{k \in \Delta} \sigma_{rel}^k((r_D(1-b) + r_D(\bot))) \cdot \sigma_{rel}^l(T_{db}) \]

where \( s_B \) denotes sending data through channel \( B \), \( r_D \) denotes receiving data through channel \( D \), similarly, \( r_{A_1} \) means receiving data via channel \( A_1 \), \( s_{C_1} \) denotes sending data via channel \( C_1 \), and \( b \in \{0, 1\} \). And the state transition of the Receiver can be described by \( \text{APTC}_{\tau}^{\text{drt}} + \text{Rec} \) as follows.

\[ R_b = \sum_{d \in \Delta} r_{A_2}(d) \cdot R_b' \]
\[ R_b' = \sum_{d' \in \Delta} \{r_B(d', b) \cdot \sigma_{rel}^l(s_{C_2}(d')) \cdot Q_b + r_B(d', 1-b) \cdot Q_{1-b} + r_B(\bot) \cdot Q_{1-b} + \sigma_{rel}^l(R_b') \} \]
\[ Q_b = \sigma_{rel}^l(s_D(b) + s_D(\bot)) \cdot R_{1-b} \]

where \( s_B \) denotes sending data via channel \( A_2 \), \( r_B \) denotes receiving data via channel \( B \), \( s_{C_2} \) denotes sending data via channel \( C_2 \), \( s_D \) denotes sending data via channel \( D \), and \( b \in \{0, 1\} \).

The send action and receive action of the same data through the same channel can communicate each other, otherwise, a deadlock \( \delta \) will be caused. We define the following communication functions.

Draft of Truly Concurrent Process Algebra with Timing
We get the time-out time of the sender and theorem 9.1 (Correctness of the ABP protocol with discrete relative timing). Time is divided into slices, \( S \) exhibits desired external behaviors with discrete relative timing.

Then the state transition of the Sender can be described by \( \text{APT}_C \) where

\[
\gamma(s_B(d',b), r_B(d',b)) = c_B(d',b)\\
\gamma(s_B(\bot), r_B(\bot)) = c_B(\bot)\\
\gamma(s_D(b), r_D(b)) = c_D(b)\\
\gamma(s_D(\bot), r_D(\bot)) = c_D(\bot)
\]

Let \( R_0 \) and \( S_0 \) be in parallel, then the system \( R_0S_0 \) can be represented by the following process term.

\[
\tau_I(\partial_H(\Theta(R_0 \parallel S_0))) = \tau_I(\partial_H(R_0 \parallel S_0))
\]

where \( H = \{s_B(d',b), r_B(d',b), s_D(b), r_D(b)|d' \in \Delta, b \in \{0,1\}\} \)

\[
\{s_B(\bot), r_B(\bot), \{s_D(b), r_D(b)\}\}
\]

\[
I = \{c_B(d',b), c_D(b)|d' \in \Delta, b \in \{0,1\}\} \cup \{c_B(\bot), c_D(\bot)\}.
\]

Then we get the following conclusion.

**Theorem 9.1** (Correctness of the ABP protocol with discrete relative timing). The ABP protocol \( \tau_I(\partial_H(R_0 \parallel S_0)) \) exhibits desired external behaviors with discrete relative timing.

**Proof.** We get \( \tau_I(\partial_H(R_0 \parallel S_0)) = \sum_{d',d \in \Delta}(r_{A_1}(d) \parallel r_{A_2}(d')) \cdot (s_{C_1}(d') \parallel s_{C_2}(d')) \cdot \tau_I(\partial_H(R_0 \parallel S_0)) \). So, the ABP protocol \( \tau_I(\partial_H(R_0 \parallel S_0)) \) exhibits desired external behaviors with discrete relative timing. \( \blacksquare \)

### 9.2. Discrete Absolute Timing

Time is divided into slices, \( t_1, t_2 \) are the times that it takes the different processes to send, receive, etc. \( t'_1 \) is the time-out time of the sender and \( t'_2 \) is the time-out of the receiver.

Then the state transition of the Sender can be described by \( \text{APT}_C^\text{dat} + \text{Rec} \) as follows.

\[
S_b = \sum_{d \in \Delta} r_{A_1}(d) \cdot T_{db}
\]

\[
T_{db} = \sum_{d' \in \Delta} (s_B(d',b) \cdot s_{C_1}(d')) + s_B(\bot) \cdot \sigma^{t_1}_{\text{abs}}(U_{db}) + \sigma^1_{\text{abs}}(T_{db})
\]

\[
U_{db} = \sum_{k \in \Delta} \sigma^k_{\text{abs}}(r_{D}(b)) \cdot S_{1-b} + \sum_{k \in \Delta} \sigma^k_{\text{abs}}(r_{D}(1-b)) \cdot \sigma^{t_1}_{\text{abs}}(T_{db})
\]

where \( s_B \) denotes sending data through channel \( B \), \( r_D \) denotes receiving data through channel \( D \). Similarly, \( r_{A_1} \) means receiving data via channel \( A_1 \), \( s_{C_1} \) denotes sending data via channel \( C_1 \), and \( b \in \{0,1\} \).

And the state transition of the Receiver can be described by \( \text{APT}_C^\text{dat} + \text{Rec} \) as follows.

\[
R_b = \sum_{d \in \Delta} r_{A_2}(d) \cdot R'_b
\]

\[
R'_b = \sum_{d' \in \Delta} \{r_B(d',b) \cdot \sigma^t_{\text{abs}}(s_{C_2}(d')) \cdot Q_b + r_B(d',1-b) \cdot Q_{1-b} + r_B(\bot) \cdot Q_{1-b} \} + r_B(\bot) \cdot s_{D}(\bot) + \sigma^t_{\text{abs}}(R'_b)
\]

\[
Q_b = \sigma^t_{\text{abs}}(s_D(b) + s_D(\bot)) \cdot R_{1-b}
\]

where \( r_{A_2} \) denotes receiving data via channel \( A_2 \), \( r_B \) denotes receiving data via channel \( B \), \( s_{C_2} \) denotes sending data via channel \( C_2 \), \( s_D \) denotes sending data via channel \( D \), and \( b \in \{0,1\} \).

The send action and receive action of the same data through the same channel can communicate each other, otherwise, a deadlock \( \delta \) will be caused. We define the following communication functions.
\[
\gamma(s_B(d',b), r_B(d',b)) \triangleq c_B(d',b)
\]
\[
\gamma(s_B(\overline{1}), r_B(\overline{1})) \triangleq c_B(\overline{1})
\]
\[
\gamma(s_D(b), r_D(b)) \triangleq c_D(b)
\]
\[
\gamma(s_D(\overline{1}), r_D(\overline{1})) \triangleq c_D(\overline{1})
\]

Let \( R_0 \) and \( S_0 \) be in parallel, then the system \( R_0 S_0 \) can be represented by the following process term.

\[
\tau_I(\partial_H(\Theta(R_0 \parallel S_0))) = \tau_I(\partial_H(R_0 \parallel S_0))
\]

where \( H = \{s_B(d',b), r_B(d',b), s_D(b), r_D(b) | d' \in \Delta, b \in \{0,1\}\} \)
\[
I = \{c_B(\overline{1}), c_D(b) | b \in \{0,1\}\} \cup \{c_B(\overline{1}), c_D(\overline{1})\}.
\]

Then we get the following conclusion.

**Theorem 9.2** (Correctness of the ABP protocol with discrete absolute timing). The ABP protocol \( \tau_I(\partial_H(R_0 \parallel S_0)) \) exhibits desired external behaviors with discrete absolute timing.

**Proof.** We get \( \tau_I(\partial_H(R_0 \parallel S_0)) = \sum_{d,d' \in \Delta} (r_{A_2}(d) \parallel r_{A_2}(d')) \cdot (s_{C_2}(d') \parallel s_{C_2}(d')) \cdot \tau_I(\partial_H(R_0 \parallel S_0)) \). So, the ABP protocol \( \tau_I(\partial_H(R_0 \parallel S_0)) \) exhibits desired external behaviors with discrete absolute timing.

### 9.3. Continuous Relative Timing

Time is denoted by time point, \( t_1, t_2 \) are the time points that it takes the different processes to send, receive, etc. \( t'_1 \) is the time-out time of the sender and \( t'_2 \) is the time-out of the receiver.

Then the state transition of the Sender can be described by \( \text{APTC}_{\tau}^{\text{art}} + \text{Rec} \) as follows.

\[
S_b = \sum_{d \in \Delta} \overline{r_{A_1}(d)} \cdot T_{db}
\]

\[
T_{db} = (\sum_{d' \in \Delta} (s_B(d',b) \cdot s_{C_1}(d')) + s_B(\overline{1})) \cdot \sigma_{rel}(U_{db}) + \sigma_{rel}(T_{db})
\]

\[
U_{db} = \sum_{k \in t'_1} \sigma_{rel}(r_D(b)) \cdot S_{1-b} + \sum_{k \in t'_1} \sigma_{rel}((r_D(1-b) + r_D(\overline{1}))) \cdot \sigma_{rel}(T_{db})
\]

where \( s_B \) denotes sending data through channel \( B \), \( r_D \) denotes receiving data through channel \( D \), similarly, \( r_{A_1} \) means receiving data via channel \( A_1 \), \( s_{C_1} \) denotes sending data via channel \( C_1 \), and \( b \in \{0,1\} \).

And the state transition of the Receiver can be described by \( \text{APTC}_{\tau}^{\text{art}} + \text{Rec} \) as follows.

\[
R_b = \sum_{d \in \Delta} \overline{r_{A_2}(d)} \cdot R'_b
\]

\[
R'_b = \sum_{d' \in \Delta} (r_B(d',b) \cdot \sigma_{rel}(s_{C_2}(d')) \cdot Q_b + r_B(\overline{1} - b) \cdot Q_{1-b}) + r_B(\overline{1}) \cdot Q_{1-b} + \sigma_{rel}(R'_b)
\]

\[
Q_b = \sigma_{rel}(s_D(b) \cdot s_D(\overline{1})) \cdot R_{1-b}
\]

where \( r_{A_2} \) denotes receiving data via channel \( A_2 \), \( r_B \) denotes receiving data via channel \( B \), \( s_{C_2} \) denotes sending data via channel \( C_2 \), \( s_D \) denotes sending data via channel \( D \), and \( b \in \{0,1\} \).

The send action and receive action of the same data through the same channel can communicate each other, otherwise, a deadlock \( \delta \) will be caused. We define the following communication functions.
Proof.
We get

Theorem 9.3
(Correctness of the ABP protocol with continuous relative timing)

\[
\tau_I(\partial_H(\Theta(R_0 \upharpoonright S_0))) = \tau_I(\partial_H(R_0 \upharpoonright S_0))
\]

where \( H = \{ (s_B(d', b), r_B(d', b), s_D(b), r_D(b)) \mid d' \in \Delta, b \in \{0, 1\} \} \)

\( I = \{ (s_B(d', b), c_D(b)) \mid d' \in \Delta, b \in \{0, 1\} \cup \{ r_B(1), c_D(1) \} \} \).

Then we get the following conclusion.

**Theorem 9.3** (Correctness of the ABP protocol with continuous relative timing). The ABP protocol \( \tau_I(\partial_H(R_0 \upharpoonright S_0)) \) exhibits desired external behaviors with continuous relative timing.

**Proof.** We get \( \tau_I(\partial_H(R_0 \upharpoonright S_0)) = \sum_{d, d' \in \Delta} (s_{A_1}(d) \parallel s_{A_2}(d')) \cdot (s_{C_1}(d') \parallel s_{C_2}(d')) \cdot \tau_I(\partial_H(R_0 \upharpoonright S_0)) \). So, the ABP protocol \( \tau_I(\partial_H(R_0 \upharpoonright S_0)) \) exhibits desired external behaviors with continuous relative timing. \( \square \)

### 9.4. Continuous Absolute Timing

Time is divided into slices, \( t_1, t_2 \) are the times that it takes the different processes to send, receive, etc. \( t'_1 \) is the time-out time of the sender and \( t'_2 \) is the time-out of the receiver.

Then the state transition of the Sender can be described by \( \text{APTC}_\text{dat} + \text{Rec} \) as follows.

\[
S_b = \sum_{d \in \Delta} r_{A_1}(d) \cdot T_{db}
\]

\[
T_{db} = (\sum_{d' \in \Delta} (s_B(d', b) \cdot s_{C_1}(d')) + s_B(1)) \cdot \sigma_{abs}^{t_1}(U_{db}) + \sigma_{abs}^r(T_{db})
\]

\[
U_{db} = \sum_{k \in \mathcal{C}_1} \sigma_{abs}^k(r_D(b)) \cdot S_{1-b} + \sum_{k \in \mathcal{C}_2} \sigma_{abs}^k(r_D(1-b) + r_D(1)) \cdot \sigma_{abs}^{t_1}(T_{db})
\]

where \( s_B \) denotes sending data through channel \( B \), \( r_D \) denotes receiving data through channel \( D \), similarly, \( r_{A_1} \) means receiving data via channel \( A_1 \), \( s_{C_1} \) denotes sending data via channel \( C_1 \), and \( b \in \{0, 1\} \).

And the state transition of the Receiver can be described by \( \text{APTC}_\text{dat} + \text{Rec} \) as follows.

\[
R_b = \sum_{d \in \Delta} r_{A_2}(d) \cdot R'_b
\]

\[
R'_b = \sum_{d' \in \Delta} (r_B(d', b) \cdot \sigma_{abs}^{t_2}(s_{C_2}(d')) \cdot Q_b + r_B(d', 1-b) \cdot Q_{1-b} + r_B(1) \cdot Q_{1-b} + \sigma_{abs}^r(R'_b))
\]

\[
Q_b = \sigma_{abs}^{t_2}(s_D(b) + s_D(1)) \cdot R_{1-b}
\]

where \( r_{A_2} \) denotes receiving data via channel \( A_2 \), \( r_B \) denotes receiving data via channel \( B \), \( s_{C_2} \) denotes sending data via channel \( C_2 \), \( s_D \) denotes sending data via channel \( D \), and \( b \in \{0, 1\} \).

The send action and receive action of the same data through the same channel can communicate each other, otherwise, a deadlock \( \delta \) will be caused. We define the following communication functions.
Proof. We get\( \text{ABP protocol } \tau \)

Theorem 9.4

\begin{align*}
\text{Table 67. Additional axioms of renaming operator } & (a \in A_{\tau}, n \geq 0) \\
\gamma(s_B(d',b), r_B(d',b)) & \equiv c_B(d',b) \\
\gamma(s_B(\bot), r_B(\bot)) & \equiv c_B(\bot) \\
\gamma(s_D(b), r_D(b)) & \equiv c_D(b) \\
\gamma(s_D(\bot), r_D(\bot)) & \equiv c_D(\bot)
\end{align*}

Let \( R_0 \) and \( S_0 \) be in parallel, then the system \( R_0S_0 \) can be represented by the following process term.

\[ \tau_I(\partial_H(\Theta(R_0 \parallel S_0))) = \tau_I(\partial_H(R_0 \parallel S_0)) \]

where \( H = \{ s_B(d',b), r_B(d',b), s_D(b), r_D(b) \mid d', b \in \{0,1\} \} \)

\[ \{ s_B(\bot), r_B(\bot), s_D(\bot), r_D(\bot) \} \]

\[ I = \{ c_B(d',b), c_D(b) \mid d', b \in \{0,1\} \} \cup \{ \overline{c_B(\bot)}, \overline{c_D(\bot)} \}. \]

Then we get the following conclusion.

**Theorem 9.4** (Correctness of the ABP protocol with continuous absolute timing). The ABP protocol \( \tau_I(\partial_H(R_0 \parallel S_0)) \) exhibits desired external behaviors with continuous absolute timing.

**Proof.** We get \( \tau_I(\partial_H(R_0 \parallel S_0)) = \sum_{d,d' \in \Delta} (r_{A_1}(d) \parallel r_{A_2}(d')) \cdot (s_{C_1}(d') \parallel s_{C_2}(d')) \cdot \tau_I(\partial_H(R_0 \parallel S_0)) \). So, the ABP protocol \( \tau_I(\partial_H(R_0 \parallel S_0)) \) exhibits desired external behaviors with continuous absolute timing.

10. Extensions

In the above sections, we have already seen the modular structure of APTC with timing by use of the concepts of conservative extension and generalization, just like APTC [17] and ACP [4]. New computational properties can be extended elegantly based on the modular structure.

In this section, we show the extension mechanism of APTC with timing by extending a new renaming property. We will introduce \( \text{APTC}_{\tau}^{\text{drt}} \) with renaming called \( \text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming} \), \( \text{APTC}_{\tau}^{\text{dat}} + \text{Rec} + \text{renaming} \), \( \text{APTC}_{\tau}^{\text{est}} + \text{Rec} + \text{renaming} \), and \( \text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming} \), respectively.

10.1. Discrete Relative Timing

**Definition 10.1** (Signature of \( \text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming} \)). The signature of \( \text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming} \) consists of the signature of \( \text{APTC}_{\tau}^{\text{drt}} + \text{Rec} \), and the renaming operator \( \rho_f : \mathcal{P}_{\text{rel}} \rightarrow \mathcal{P}_{\text{rel}} \).

The axioms of \( \text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming} \) include the laws of \( \text{APTC}_{\tau}^{\text{drt}} + \text{Rec} \), and the axioms of renaming operator in Table 67.

The additional transition rules of renaming operator is shown in Table 68.
$$\begin{align*}
x \overset{a}{\rightarrow} y & \\
\rho_f(x) \overset{f(a)}{\rightarrow} \rho_f(x) & \\
x \overset{m}{\rightarrow} y & \\
\rho_f(x) \overset{m}{\rightarrow} \rho_f(x) & \\
x \uparrow & \\
\rho_f(x) \uparrow &
\end{align*}$$

Table 68. Transition rule of renaming operator ($a \in A, m > 0, n \geq 0$)

**Theorem 10.2** (Elimination theorem). Let $p$ be a closed $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$ term. Then there is a basic $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec}$ term $q$ such that $\text{APTC}_{\tau}^{\text{drt}} + \text{renaming} \vdash p = q$.

**Proof.** It is sufficient to induct on the structure of the closed $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$ term $p$. It can be proven that $p$ combined by the constants and operators of $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$ exists an equal basic $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec}$ term $q$, and the other operators not included in the basic terms, such as $\rho_f$ can be eliminated. □

10.1.1. Connections

**Theorem 10.3** (Conservativity of $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$). $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$ is a conservative extension of $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec}$.

**Proof.** It follows from the following two facts.

1. The transition rules of $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec}$ are all source-dependent;
2. The sources of the transition rules of $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$ contain an occurrence of $\rho_f$.

So, $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$ is a conservative extension of $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec}$, as desired. □

10.1.2. Congruence

**Theorem 10.4** (Congruence of $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$). Rooted branching truly concurrent bisimulation equivalences $\approx_{\text{rbp}}, \approx_{\text{rbs}}$ and $\approx_{\text{rbhp}}$ are all congruences with respect to $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$. That is,

- rooted branching pomset bisimulation equivalence $\approx_{\text{rbp}}$ is a congruence with respect to $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$;
- rooted branching step bisimulation equivalence $\approx_{\text{rbs}}$ is a congruence with respect to $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$;
- rooted branching hp-bisimulation equivalence $\approx_{\text{rbhp}}$ is a congruence with respect to $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$.

**Proof.** It is easy to see that $\approx_{\text{rbp}}, \approx_{\text{rbs}}$, and $\approx_{\text{rbhp}}$ are all equivalent relations on $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$ terms, it is only sufficient to prove that $\approx_{\text{rbp}}, \approx_{\text{rbs}}$, and $\approx_{\text{rbhp}}$ are all preserved by the operators $\rho_f$. It is trivial and we omit it. □

10.1.3. Soundness

**Theorem 10.5** (Soundness of $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$). The axiomatization of $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$ is sound modulo rooted branching truly concurrent bisimulation equivalences $\approx_{\text{rbp}}, \approx_{\text{rbs}}$, and $\approx_{\text{rbhp}}$. That is,

1. let $x$ and $y$ be $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$ terms. If $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming} \vdash x = y$, then $x \approx_{\text{rbs}} y$;
2. let $x$ and $y$ be $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming}$ terms. If $\text{APTC}_{\tau}^{\text{drt}} + \text{Rec} + \text{renaming} \vdash x = y$, then $x \approx_{\text{rbp}} y$;
3. Let \(x\) and \(y\) be APTC\(_{\tau}^{\text{drt}}\) + Rec + renaming terms. If APTC\(_{\tau}^{\text{drt}}\) + Rec + renaming \(\vdash x = y\), then \(x \equiv_{\text{rhp}} y\).

**Proof.** Since \(\equiv_{\text{rhp}}, \equiv_{\text{rbs}},\) and \(\equiv_{\text{rhp}}\) are both equivalent and congruent relations, we only need to check if each axiom in Table 67 is sound modulo \(\equiv_{\text{rhp}}, \equiv_{\text{rbs}},\) and \(\equiv_{\text{rhp}}\) respectively.

1. Each axiom in Table 67 can be checked that it is sound modulo rooted branching step bisimulation equivalence, by transition rules in Table 68. We omit them.

2. From the definition of rooted branching pomset bisimulation \(\equiv_{\text{rhp}}\), we know that rooted branching pomset bisimulation \(\equiv_{\text{rhp}}\) is defined by weak pomset transitions, which are labeled by pomsets with \(\tau\). In a weak pomset transition, the events in the pomset are either within causality relations (defined by \(\cdot\)) or in concurrency (implicitly defined by \(\cdot\) and +, and explicitly defined by \(\downarrow\)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent, so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \(P = \langle a, b : a + b \rangle\). Then the weak pomset transition labeled by the above \(P\) is just composed of one single event transition labeled by \(q\) succeeded by another single event transition labeled by \(b\), that is, \(P \Rightarrow a \Downarrow b\).

3. Similarly to the proof of soundness modulo rooted branching step bisimulation \(\equiv_{\text{rbs}},\) we can prove that each axiom in Table 67 is sound modulo rooted branching pomset bisimulation \(\equiv_{\text{rhp}},\) we omit them.

From the definition of rooted branching hp-bisimulation \(\equiv_{\text{rhp}}\), we know that rooted branching hp-bisimulation \(\equiv_{\text{rhp}}\) is defined on the weakly posetal product \((C_1, f, C_2), f : C_1 \rightarrow C_2\) isomorphism. Two process terms \(s\) related to \(C_1\) and \(t\) related to \(C_2\), and \(f : C_1 \rightarrow C_2\) isomorphism. Initially, \((C_1, f, C_2) = (\emptyset, \emptyset, \emptyset)\) and \((\emptyset, \emptyset, \emptyset) \equiv_{\text{rhp}}\). When \(s \Rightarrow s' (C_1 \xrightarrow{a} C'_1)\), there will be \(t \Rightarrow t' (C_2 \xrightarrow{a} C'_2)\), and we define \(f' = f[a \mapsto a]\). Then, if \((C_1, f, C_2) \equiv_{\text{rhp}}\), then \((C'_1, f', C'_2) \equiv_{\text{rhp}}\).

Similarly to the proof of soundness modulo rooted branching pomset bisimulation equivalence, we can prove that each axiom in Table 67 is sound modulo rooted branching hp-bisimulation equivalence, we just need additionally to check the above conditions on rooted branching hp-bisimulation, we omit them.

\[\square\]

### 10.1.4. Completeness

**Theorem 10.6** (Completeness of APTC\(_{\tau}^{\text{drt}}\) + CFAR + guarded linear Rec + renaming). The axiomatization of APTC\(_{\tau}^{\text{drt}}\) + CFAR + guarded linear Rec + renaming is complete modulo rooted branching truly concurrent bisimulation equivalences \(\equiv_{\text{rbs}}, \equiv_{\text{rhp}},\) and \(\equiv_{\text{rhp}}\). That is,

1. Let \(p\) and \(q\) be closed APTC\(_{\tau}^{\text{drt}}\) + CFAR + guarded linear Rec + renaming terms, if \(p \equiv_{\text{rbs}} q\) then \(p = q\);
2. Let \(p\) and \(q\) be closed APTC\(_{\tau}^{\text{drt}}\) + CFAR + guarded linear Rec + renaming terms, if \(p \equiv_{\text{rhp}} q\) then \(p = q\);
3. Let \(p\) and \(q\) be closed APTC\(_{\tau}^{\text{drt}}\) + CFAR + guarded linear Rec + renaming terms, if \(p \equiv_{\text{rhp}} q\) then \(p = q\).

**Proof.** Firstly, we know that each process term in APTC\(_{\tau}^{\text{drt}}\) + CFAR + guarded linear Rec + renaming is equal to a process term \((X_1|E)\) with \(E\) a linear recursive specification.

It remains to prove the following cases.

1. If \((X_1|E_1) \equiv_{\text{rbs}} (Y_1|E_2)\) for linear recursive specification \(E_1\) and \(E_2\), then \((X_1|E_1) = (Y_1|E_2)\).
   
   It can be proven similarly to the completeness of APTC\(_{\tau} + \text{CFAR} + \text{linear Rec} + \text{renaming}, see [17].

2. If \((X_1|E_1) \equiv_{\text{rhp}} (Y_1|E_2)\) for linear recursive specification \(E_1\) and \(E_2\), then \((X_1|E_1) = (Y_1|E_2)\).
   
   It can be proven similarly, just by replacement of \(\equiv_{\text{rbs}}\) by \(\equiv_{\text{rhp}}\), we omit it.

3. If \((X_1|E_1) \equiv_{\text{rhp}} (Y_1|E_2)\) for linear recursive specification \(E_1\) and \(E_2\), then \((X_1|E_1) = (Y_1|E_2)\).
   
   It can be proven similarly, just by replacement of \(\equiv_{\text{rbs}}\) by \(\equiv_{\text{rhp}}\), we omit it.

\[\square\]

### 10.2. Discrete Absolute Timing

**Definition 10.7** (Signature of APTC\(_{\tau}^{\text{drat}}\) + Rec + renaming). The signature of APTC\(_{\tau}^{\text{drat}}\) + Rec + renaming consists of the signature of APTC\(_{\tau}^{\text{drat}}\) + Rec, and the renaming operator \(\rho_f : \mathcal{P}_{\text{abs}} \rightarrow \mathcal{P}_{\text{abs}}\).
\textbf{Table 69. Additional axioms of renaming operator} \((a \in A, \delta, n \geq 0)\)

\begin{align*}
\rho_f(\omega) &= \omega \\
\rho_f(\delta) &= \delta \\
\rho_f(\tau) &= \tau \\
\rho_f(\sigma_{\text{abs}}(x)) &= \sigma_{\text{abs}}(\rho_f(x)) \\
\rho_f(x + y) &= \rho_f(x) + \rho_f(y) \\
\rho_f(x \cdot y) &= \rho_f(x) \cdot \rho_f(y) \\
\rho_f(x \parallel y) &= \rho_f(x) \parallel \rho_f(y)
\end{align*}

\textbf{Table 70. Transition rule of renaming operator} \((a \in A, m > 0, n \geq 0)\)

\begin{align*}
\langle x, n \rangle &\xrightarrow{a} \langle \sqrt{x}, n \rangle \\
\langle \rho_f(x), n \rangle &\xrightarrow{f(a)} \langle \rho_f(x), n \rangle \\
\langle x, n \rangle &\xrightarrow{m} \langle x, n + m \rangle \\
\langle \rho_f(x), n \rangle &\xrightarrow{m} \langle \rho_f(x), n + m \rangle \\
\langle x, n \rangle &\xrightarrow{\uparrow} \langle \rho_f(x), n \rangle \\
\rangle &\xrightarrow{\uparrow}
\end{align*}

The axioms of \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec + renaming}\) include the laws of \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec}\), and the axioms of renaming operator in Table 69.

The additional transition rules of renaming operator is shown in Table 70.

\textbf{Theorem 10.8} (Elimination theorem). Let \(p\) be a closed \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec + renaming}\) term. Then there is a basic \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec}\) term \(q\) such that \(\text{APTC}_{\tau}^{\text{dat}} + \text{renaming} \vdash p = q\).

\textit{Proof.} It is sufficient to induct on the structure of the closed \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec + renaming}\) term \(p\). It can be proven that \(p\) combined by the constants and operators of \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec + renaming}\) exists an equal basic \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec}\) term \(q\), and the other operators not included in the basic terms, such as \(\rho_f\) can be eliminated.

\textbf{10.2.1. Connections}

\textbf{Theorem 10.9} (Conservativity of \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec + renaming}\)). \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec + renaming}\) is a conservative extension of \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec}\).

\textit{Proof.} It follows from the following two facts.

1. The transition rules of \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec}\) are all source-dependent;
2. The sources of the transition rules of \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec + renaming}\) contain an occurrence of \(\rho_f\).

So, \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec + renaming}\) is a conservative extension of \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec}\), as desired.

\textbf{10.2.2. Congruence}

\textbf{Theorem 10.10} (Congruence of \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec + renaming}\)). Rooted branching truly concurrent bisimulation equivalences \(\approx_{\text{rbp}}, \approx_{\text{rbs}}\) and \(\approx_{\text{rbhp}}\) are all congruences with respect to \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec + renaming}\). That is,

- rooted branching pomset bisimulation equivalence \(\approx_{\text{rbp}}\) is a congruence with respect to \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec + renaming}\);
- rooted branching step bisimulation equivalence \(\approx_{\text{rbs}}\) is a congruence with respect to \(\text{APTC}_{\tau}^{\text{dat}} + \text{Rec + renaming}\);
• rooted branching hp-bisimulation equivalence \( \approx_{rbhp} \) is a congruence with respect to \( \text{APTC}^{\text{dat}}_{\tau} + \text{Rec} + \text{renaming} \).

Proof. It is easy to see that \( \approx_{rbp} \), \( \approx_{rbs} \), and \( \approx_{rbhp} \) are all equivalent relations on \( \text{APTC}^{\text{dat}}_{\tau} + \text{Rec} + \text{renaming} \) terms, it is only sufficient to prove that \( \approx_{rbp} \), \( \approx_{rbs} \), and \( \approx_{rbhp} \) are all preserved by the operators \( \rho_f \). It is trivial and we omit it.

10.2.3. Soundness

Theorem 10.11 (Soundness of \( \text{APTC}^{\text{dat}}_{\tau} + \text{Rec} + \text{renaming} \)). The axiomatization of \( \text{APTC}^{\text{dat}}_{\tau} + \text{Rec} + \text{renaming} \) is sound modulo rooted branching truly concurrent bisimulation equivalences \( \approx_{rbp} \), \( \approx_{rbs} \), and \( \approx_{rbhp} \). That is,

1. let \( x \) and \( y \) be \( \text{APTC}^{\text{dat}}_{\tau} + \text{Rec} + \text{renaming} \) terms. If \( \text{APTC}^{\text{dat}}_{\tau} + \text{Rec} + \text{renaming} \vdash x = y \), then \( x \approx_{rbs} y \);
2. let \( x \) and \( y \) be \( \text{APTC}^{\text{dat}}_{\tau} + \text{Rec} + \text{renaming} \) terms. If \( \text{APTC}^{\text{dat}}_{\tau} + \text{Rec} + \text{renaming} \vdash x = y \), then \( x \approx_{rbp} y \);
3. let \( x \) and \( y \) be \( \text{APTC}^{\text{dat}}_{\tau} + \text{Rec} + \text{renaming} \) terms. If \( \text{APTC}^{\text{dat}}_{\tau} + \text{Rec} + \text{renaming} \vdash x = y \), then \( x \approx_{rbhp} y \).

Proof. Since \( \approx_{rbp} \), \( \approx_{rbs} \), and \( \approx_{rbhp} \) are both equivalent and congruent relations, we only need to check if each axiom in Table 69 is sound modulo \( \approx_{rbp} \), \( \approx_{rbs} \), and \( \approx_{rbhp} \) respectively.

1. Each axiom in Table 69 can be checked that it is sound modulo rooted branching step bisimulation equivalence, by transition rules in Table 70. We omit them.

2. From the definition of rooted branching pomset bisimulation \( \approx_{rbp} \), we know that rooted branching pomset bisimulation \( \approx_{rbp} \) is defined by weak pomset transitions, which are labeled by pomsets with \( \bot \). In a weak pomset transition, the events in the pomset are either within causality relations (defined by \( \cdot \)) or in concurrency (implicitly defined by \( \cdot \) and \( + \), and explicitly defined by \( \hat{\cdot} \)), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent, so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of \( P = \{ a, b, a \rightarrow b \} \). Then the weak pomset transition labeled by the above \( P \) is just composed of one single event transition labeled by \( a \) succeeded by another single event transition labeled by \( b \) that is, \( \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow \). Similarly to the proof of soundness modulo rooted branching step bisimulation \( \approx_{rbs} \), we can prove that each axiom in Table 69 is sound modulo rooted branching pomset bisimulation \( \approx_{rbp} \), we omit them.

3. From the definition of rooted branching hp-bisimulation \( \approx_{rbhp} \), we know that rooted branching hp-bisimulation \( \approx_{rbhp} \) is defined on the weakly posetal product \((C_1, f, C_2) \rightarrow C_1 \rightarrow C_2 \) isomorphism. Two process terms \( s \) related to \( C_1 \) and \( t \) related to \( C_2 \), and \( f \) : \( C_1 \rightarrow C_2 \) isomorphism. Initially, \((C_1, f, C_2) = (\emptyset, \emptyset, \emptyset) \) \( \approx_{rbhp} \). When \( s \overset{a}{\rightarrow} s' \) \( (C_1 \overset{a}{\rightarrow} C_1') \), there will be \( t \overset{a}{\rightarrow} t' \) \( (C_2 \overset{a}{\rightarrow} C_2') \), and we define \( f' = f[a \mapsto a] \). Then, if \((C_1, f, C_2) \approx_{rbhp} \), then \((C_1', f', C_2') \approx_{rbhp} \). Similarly to the proof of soundness modulo rooted branching pomset bisimulation equivalence, we can prove that each axiom in Table 69 is sound modulo rooted branching hp-bisimulation equivalence, we just need additionally to check the above conditions on rooted branching hp-bisimulation, we omit them.

10.2.4. Completeness

Theorem 10.12 (Completeness of \( \text{APTC}^{\text{dat}}_{\tau} + \text{CFAR} + \text{guarded linear Rec} + \text{renaming} \)). The axiomatization of \( \text{APTC}^{\text{dat}}_{\tau} + \text{CFAR} + \text{guarded linear Rec} + \text{renaming} \) is complete modulo rooted branching truly concurrent bisimulation equivalences \( \approx_{rbs} \), \( \approx_{rbp} \), and \( \approx_{rbhp} \). That is,

1. let \( p \) and \( q \) be closed \( \text{APTC}^{\text{dat}}_{\tau} + \text{CFAR} + \text{guarded linear Rec} + \text{renaming} \) terms, if \( p \approx_{rbs} q \) then \( p = q \);
2. let \( p \) and \( q \) be closed \( \text{APTC}^{\text{dat}}_{\tau} + \text{CFAR} + \text{guarded linear Rec} + \text{renaming} \) terms, if \( p \approx_{rbp} q \) then \( p = q \);
3. let \( p \) and \( q \) be closed \( \text{APTC}^{\text{dat}}_{\tau} + \text{CFAR} + \text{guarded linear Rec} + \text{renaming} \) terms, if \( p \approx_{rbhp} q \) then \( p = q \).

Proof. Firstly, we know that each process term in \( \text{APTC}^{\text{dat}}_{\tau} + \text{CFAR} + \text{guarded linear Rec} + \text{renaming} \) is equal to a process term \( (X_1|E) \) with \( E \) a linear recursive specification. It remains to prove the following cases.
Table 71. Additional axioms of renaming operator \((a \in A, s, p \geq 0)\)

| No. | Axiom | 
|-----|-------|
| SRTRN1 | \(\rho_f(\tilde{a}) = f(\tilde{a})\) |
| SRTRN2 | \(\rho_f(\delta) = \delta\) |
| SRTRN3 | \(\rho_f(\tilde{\delta}) = \tilde{\delta}\) |
| SRTRN | \(\rho_f(\sigma_{s,\delta}(x)) = \sigma_{s,\delta}(\rho_f(x))\) |
| RN3 | \(\rho_f(x + y) = \rho_f(x) + \rho_f(y)\) |
| RN4 | \(\rho_f(x \cdot y) = \rho_f(x) \cdot \rho_f(y)\) |
| RN5 | \(\rho_f(x \parallel y) = \rho_f(x) \parallel \rho_f(y)\) |

Table 72. Transition rule of renaming operator \((a \in A, r > 0, p \geq 0)\)

1. If \(X_1|E_1 \equiv_{rbs} Y_1|E_2\) for linear recursive specification \(E_1\) and \(E_2\), then \(X_1|E_1 = Y_1|E_2\).
   It can be proven similarly to the completeness of APTC + CFAR + linear Rec + renaming, see [17].
2. If \(X_1|E_1 \equiv_{rhp} Y_1|E_2\) for linear recursive specification \(E_1\) and \(E_2\), then \(X_1|E_1 = Y_1|E_2\).
   It can be proven similarly, just by replacement of \(s_{rbs}\) by \(s_{rhp}\), we omit it.
3. If \(X_1|E_1 \equiv_{bbp} Y_1|E_2\) for linear recursive specification \(E_1\) and \(E_2\), then \(X_1|E_1 = Y_1|E_2\).
   It can be proven similarly, just by replacement of \(s_{rbs}\) by \(s_{bbp}\), we omit it.

10.3. Continuous Relative Timing

**Definition 10.13** (Signature of APTC\(_{srt}\) + Rec + renaming). The signature of APTC\(_{srt}\) + Rec + renaming consists of the signature of APTC\(_{srt}\) + Rec, and the renaming operator \(\rho_f : P_{rel} \rightarrow P_{rel}\).

The axioms of APTC\(_{srt}\) + Rec + renaming include the laws of APTC\(_{srt}\) + Rec, and the axioms of renaming operator in Table 7[1.

The additional transition rules of renaming operator is shown in Table 7[2.

**Theorem 10.14** (Elimination theorem). Let \(p\) be a closed APTC\(_{srt}\) + Rec + renaming term. Then there is a basic APTC\(_{srt}\) + Rec term \(q\) such that APTC\(_{srt}\) + renaming \(\vdash p = q\).

**Proof.** It is sufficient to induct on the structure of the closed APTC\(_{srt}\) + Rec + renaming term \(p\). It can be proven that \(p\) combined by the constants and operators of APTC\(_{srt}\) + Rec + renaming exists an equal basic APTC\(_{srt}\) + Rec term \(q\), and the other operators not included in the basic terms, such as \(\rho_f\) can be eliminated.

10.3.1. Connections

**Theorem 10.15** (Conservativity of APTC\(_{srt}\) + Rec + renaming). APTC\(_{srt}\) + Rec + renaming is a conservative extension of APTC\(_{srt}\) + Rec.

**Proof.** It follows from the following two facts.

1. The transition rules of APTC\(_{srt}\) + Rec are all source-dependent;
2. The sources of the transition rules of APTC$^{\text{ext}}$ + Rec + renaming contain an occurrence of $\rho_f$.

So, APTC$^{\text{ext}}$ + Rec + renaming is a conservative extension of APTC$^{\text{ext}}$ + Rec, as desired.

10.3.2. Congruence

**Theorem 10.16** (Congruence of APTC$^{\text{ext}}$ + Rec + renaming). Rooted branching truly concurrent bisimulation equivalences $\approx_{r\text{bp}}$, $\approx_{r\text{bs}}$ and $\approx_{r\text{bhp}}$ are all congruences with respect to APTC$^{\text{ext}}$ + Rec + renaming. That is,

- rooted branching pomset bisimulation equivalence $\approx_{r\text{bp}}$ is a congruence with respect to APTC$^{\text{ext}}$ + Rec + renaming;
- rooted branching step bisimulation equivalence $\approx_{r\text{bs}}$ is a congruence with respect to APTC$^{\text{ext}}$ + Rec + renaming;
- rooted branching hp-bisimulation equivalence $\approx_{r\text{bhp}}$ is a congruence with respect to APTC$^{\text{ext}}$ + Rec + renaming.

**Proof.** It is easy to see that $\approx_{r\text{bp}}$, $\approx_{r\text{bs}}$, and $\approx_{r\text{bhp}}$ are all equivalent relations on APTC$^{\text{ext}}$ + Rec + renaming terms, it is only sufficient to prove that $\approx_{r\text{bp}}$, $\approx_{r\text{bs}}$, and $\approx_{r\text{bhp}}$ are all preserved by the operators $\rho_f$. It is trivial and we omit it.

10.3.3. Soundness

**Theorem 10.17** (Soundness of APTC$^{\text{ext}}$ + Rec + renaming). The axiomatization of APTC$^{\text{ext}}$ + Rec + renaming is sound modulo rooted branching truly concurrent bisimulation equivalences $\approx_{r\text{bp}}$, $\approx_{r\text{bs}}$, and $\approx_{r\text{bhp}}$. That is,

1. let $x$ and $y$ be APTC$^{\text{ext}}$ + Rec + renaming terms. If APTC$^{\text{ext}}$ + Rec + renaming $\vdash x = y$, then $x \approx_{r\text{bs}} y$;
2. let $x$ and $y$ be APTC$^{\text{ext}}$ + Rec + renaming terms. If APTC$^{\text{ext}}$ + Rec + renaming $\vdash x = y$, then $x \approx_{r\text{bp}} y$;
3. let $x$ and $y$ be APTC$^{\text{ext}}$ + Rec + renaming terms. If APTC$^{\text{ext}}$ + Rec + renaming $\vdash x = y$, then $x \approx_{r\text{bhp}} y$.

**Proof.** Since $\approx_{r\text{bp}}$, $\approx_{r\text{bs}}$, and $\approx_{r\text{bhp}}$ are both equivalent and congruent relations, we only need to check if each axiom in Table 72 is sound modulo rooted branching step bisimulation equivalence, by transition rules in Table 72. We omit them.

1. Each axiom in Table 72 can be checked that it is sound modulo rooted branching step bisimulation equivalence, by transition rules in Table 72. We omit them.
2. From the definition of rooted branching pomset bisimulation $\approx_{r\text{bp}}$, we know that rooted branching pomset bisimulation $\approx_{r\text{bp}}$ is defined by weak pomset transitions, which are labeled by pomsets with $\hat{\tau}$. In a weak pomset transition, the events in the pomset are either within causality relations (defined by $\cdot$) or in concurrency (implicitly defined by $\cdot$ and $+$, and explicitly defined by $\parallel$), of course, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent, so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of $P = \{\tilde{a}, \tilde{b} : \tilde{a} \cdot \tilde{b}\}$. Then the weak pomset transition labeled by the above $P$ is just composed of one single event transition labeled by $\hat{a}$ succeeded by another single event transition labeled by $\hat{b}$, that is, $\overset{\tilde{a} \cdot \tilde{b}}{\overset{\hat{a} \parallel \hat{b}}{\hat{a}}}$. Similarly to the proof of soundness modulo rooted branching step bisimulation $\approx_{r\text{bs}}$, we can prove that each axiom in Table 72 is sound modulo rooted branching pomset bisimulation $\approx_{r\text{bp}}$, we omit them.
3. From the definition of rooted branching hp-bisimulation $\approx_{r\text{bhp}}$, we know that rooted branching hp-bisimulation $\approx_{r\text{bhp}}$ is defined on the weakly posetal product $(C_1, f, C_2, f : C_1 \rightarrow C_2)$ isomorphism. Two process terms $s$ related to $C_1$ and $t$ related to $C_2$, and $f : C_1 \rightarrow C_2$ isomorphism. Initially, $(C_1, f, C_2) = (\varnothing, \varnothing, \varnothing)$, and $(\varnothing, \varnothing, \varnothing) \approx_{r\text{bhp}}$. When $s \underset{a}{\overset{a}{\Rightarrow}} s'$ $(C_1 \overset{\alpha}{\rightarrow} C_1')$, there will be $t \overset{\alpha}{\Rightarrow} t'$ $(C_2 \overset{\alpha}{\rightarrow} C_2')$, and we define $f' = f[a \mapsto a]$. Then, if $(C_1, f, C_2) \approx_{r\text{bhp}}$, then $(C_1', f', C_2') \approx_{r\text{bhp}}$. Similarly to the proof of soundness modulo rooted branching pomset bisimulation equivalence, we can prove that each axiom in Table 72 is sound modulo rooted branching hp-bisimulation equivalence, we just need additionally to check the above conditions on rooted branching hp-bisimulation, we omit them.
Table 73. Additional axioms of renaming operator \((a \in A, p \geq 0)\)

| No. | Axiom |
|-----|-------|
| SATRN1 | \(\rho_f(\bar{a}) = f(\bar{a})\) |
| SATRN2 | \(\rho_f(\delta) = \delta\) |
| SATRN3 | \(\rho_f(\bar{y}) = \bar{y}\) |
| SATRN | \(\rho_f(\sigma^p_{abs}(x)) = \sigma^p_{abs}(\rho_f(x))\) |
| RN3 | \(\rho_f(x + y) = \rho_f(x) + \rho_f(y)\) |
| RN4 | \(\rho_f(x \cdot y) = \rho_f(x) \cdot \rho_f(y)\) |
| RN5 | \(\rho_f(x \parallel y) = \rho_f(x) \parallel \rho_f(y)\) |

10.3.4. Completeness

**Theorem 10.18** (Completeness of \(\APTCT\) + CFAR + guarded linear Rec + renaming). *The axiomatization of \(\APTCT\) + CFAR + guarded linear Rec + renaming is complete modulo rooted branching truly concurrent bisimulation equivalences \(\approx_{rbs}, \approx_{rbp},\) and \(\approx_{rbhp}\). That is,

1. Let \(p\) and \(q\) be closed \(\APTCT\) + CFAR + guarded linear Rec + renaming terms, if \(p \approx_{rbs} q\) then \(p = q\);
2. Let \(p\) and \(q\) be closed \(\APTCT\) + CFAR + guarded linear Rec + renaming terms, if \(p \approx_{rbp} q\) then \(p = q\);
3. Let \(p\) and \(q\) be closed \(\APTCT\) + CFAR + guarded linear Rec + renaming terms, if \(p \approx_{rbhp} q\) then \(p = q\).

**Proof.** Firstly, we know that each process term in \(\APTCT\) + CFAR + guarded linear Rec + renaming is equal to a process term \((X_1|E)\) with \(E\) a linear recursive specification.

It remains to prove the following cases.

1. If \((X_1|E_1) \approx_{rbs} (Y_1|E_2)\) for linear recursive specification \(E_1\) and \(E_2\), then \((X_1|E_1) = (Y_1|E_2)\).

   It can be proven similarly to the completeness of \(\APTCT\) + CFAR + linear Rec + renaming, see \[17\].
2. If \((X_1|E_1) \approx_{rbp} (Y_1|E_2)\) for linear recursive specification \(E_1\) and \(E_2\), then \((X_1|E_1) = (Y_1|E_2)\).

   It can be proven similarly, just by replacement of \(\approx_{rbs}\) by \(\approx_{rbp}\), we omit it.
3. If \((X_1|E_1) \approx_{rbhp} (Y_1|E_2)\) for linear recursive specification \(E_1\) and \(E_2\), then \((X_1|E_1) = (Y_1|E_2)\).

   It can be proven similarly, just by replacement of \(\approx_{rbs}\) by \(\approx_{rbhp}\), we omit it.

\(\square\)

10.4. Continuous Absolute Timing

**Definition 10.19** (Signature of \(\APTCS\) + Rec + renaming). *The signature of \(\APTCS\) + Rec + renaming consists of the signature of \(\APTCE\) + Rec, and the renaming operator \(\rho_f: P_{abs} \rightarrow P_{abs}\).*

The axioms of \(\APTCS\) + Rec + renaming include the laws of \(\APTCS\) + Rec, and the axioms of renaming operator in Table 73.

The additional transition rules of renaming operator is shown in Table 74.

**Theorem 10.20** (Elimination theorem). *Let \(p\) be a closed \(\APTCE\) + Rec + renaming term. Then there is a basic \(\APTCE\) + Rec term \(q\) such that \(\APTCE + \text{renaming} \vdash p = q\).*
Proof. It is sufficient to induct on the structure of the closed $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ term $p$. It can be proven that $p$ combined by the constants and operators of $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ exists an equal basic $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec}$ term $q$, and the other operators not included in the basic terms, such as $\rho_f$ can be eliminated.

10.4.1 Connections

**Theorem 10.21** (Conservativity of $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$). $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ is a conservative extension of $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec}$.

**Proof.** It follows from the following two facts.

1. The transition rules of $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec}$ are all source-dependent;
2. The sources of the transition rules of $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ contain an occurrence of $\rho_f$.

So, $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ is a conservative extension of $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec}$, as desired.

10.4.2 Congruence

**Theorem 10.22** (Congruence of $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$). Rooted branching truly concurrent bisimulation equivalences $\approx_{rbp}$, $\approx_{rbs}$ and $\approx_{rbhp}$ are all congruences with respect to $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$.

That is,

- rooted branching pomset bisimulation equivalence $\approx_{rbp}$ is a congruence with respect to $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$;
- rooted branching step bisimulation equivalence $\approx_{rbs}$ is a congruence with respect to $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$;
- rooted branching hp-bisimulation equivalence $\approx_{rbhp}$ is a congruence with respect to $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$.

**Proof.** It is easy to see that $\approx_{rbp}$, $\approx_{rbs}$, and $\approx_{rbhp}$ are all equivalent relations on $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ terms, it is only sufficient to prove that $\approx_{rbp}$, $\approx_{rbs}$, and $\approx_{rbhp}$ are all preserved by the operators $\rho_f$. It is trivial and we omit it.

10.4.3 Soundness

**Theorem 10.23** (Soundness of $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$). The axiomatization of $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ is sound modulo rooted branching truly concurrent bisimulation equivalences $\approx_{rbp}$, $\approx_{rbs}$, and $\approx_{rbhp}$. That is,

1. let $x$ and $y$ be $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ terms. If $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ $\vdash x = y$, then $x \approx_{rbs} y$;
2. let $x$ and $y$ be $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ terms. If $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ $\vdash x = y$, then $x \approx_{rbp} y$;
3. let $x$ and $y$ be $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ terms. If $\text{APTC}_{\tau}^{\text{sat}} + \text{Rec} + \text{renaming}$ $\vdash x = y$, then $x \approx_{rbhp} y$.

**Proof.** Since $\approx_{rbp}$, $\approx_{rbs}$, and $\approx_{rbhp}$ are both equivalent and congruent relations, we only need to check if each axiom in Table 73 is sound modulo $\approx_{rbp}$, $\approx_{rbs}$, and $\approx_{rbhp}$ respectively.

1. Each axiom in Table 73 can be checked that it is sound modulo rooted branching step bisimulation equivalence, by transition rules in Table 74. We omit them.
2. From the definition of rooted branching pomset bisimulation $\approx_{rbp}$, we know that rooted branching pomset bisimulation $\approx_{rbp}$ is defined by weak pomset transitions, which are labeled by pomsets with $\tau$. In a weak pomset transition, the events in the pomset are either within causality relations (defined by $\cdot$) or in concurrency (implicitly defined by $\cdot$ and $+$, and explicitly defined by $\frac{1}{\cdot}$), so, they are pairwise consistent (without conflicts). We have already proven the case that all events are pairwise concurrent, so, we only need to prove the case of events in causality. Without loss of generality, we take a pomset of $P = \{\hat{a}, b: \hat{a} \cdot b\}$. Then the weak pomset transition labeled by the above $P$ is just composed of one single event transition labeled by $\hat{a}$ succeeded by another single event transition labeled by $b$, that is, $\Rightarrow$.
Similarly to the proof of soundness modulo rooted branching step bisimulation \( \approx_{rbp} \), we can prove that each axiom in Table 73 is sound modulo rooted branching pomset bisimulation \( \approx_{rhp} \), we omit them.

3. From the definition of rooted branching hp-bisimulation \( \approx_{rhp} \), we know that rooted branching hp-bisimulation \( \approx_{rhp} \) is defined on the weakly posetal product \((C_1, f, C_2, f) : \hat{C}_1 \rightarrow \hat{C}_2\) isomorphism. Two process terms \( s \) related to \( C_1 \) and \( t \) related to \( C_2 \), and \( f : \hat{C}_1 \rightarrow \hat{C}_2 \) isomorphism. Initially, \((C_1, f, C_2) = (s, \phi, \phi)\), and \((\phi, \phi, \phi) \in \approx_{rhp}.\) When \( s \xrightarrow{a} s' \) \((C_1 \xrightarrow{a} C_1')\), there will be \( t \xrightarrow{a} t' \) \((C_2 \xrightarrow{a} C_2')\), and we define \( f' = f[a \mapsto a] \). Then, if \((C_1, f, C_2) \approx_{rhp}, \) then \((C_1', f', C_2') \approx_{rhp}.\)

Similarly to the proof of soundness modulo rooted branching pomset bisimulation equivalence, we can prove that each axiom in Table 73 is sound modulo rooted branching hp-bisimulation equivalence, we just need additionally to check the above conditions on rooted branching hp-bisimulation, we omit them.

\[ \Box \]

10.4.4. Completeness

**Theorem 10.24** (Completeness of APTC\textsuperscript{at} + CFAR + guarded linear Rec + renaming). The axiomatization of APTC\textsuperscript{at} + CFAR + guarded linear Rec + renaming is complete modulo rooted branching truly concurrent bisimulation equivalences \( \approx_{rbp} \), \( \approx_{rhp} \), and \( \approx_{rhp}. \) That is,

1. let \( p \) and \( q \) be closed APTC\textsuperscript{at} + CFAR + guarded linear Rec + renaming terms, if \( p \approx_{rbp} q \) then \( p = q; \)
2. let \( p \) and \( q \) be closed APTC\textsuperscript{at} + CFAR + guarded linear Rec + renaming terms, if \( p \approx_{rhp} q \) then \( p = q; \)
3. let \( p \) and \( q \) be closed APTC\textsuperscript{at} + CFAR + guarded linear Rec + renaming terms, if \( p \approx_{rhp} q \) then \( p = q. \)

*Proof.* Firstly, we know that each process term in APTC\textsuperscript{at} + CFAR + guarded linear Rec + renaming is equal to a process term \((X_1|E)\) with \( E \) a linear recursive specification.

It remains to prove the following cases.

1. If \( \langle X_1|E_1 \rangle \approx_{rbs} \langle Y_1|E_2 \rangle \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( \langle X_1|E_1 \rangle = \langle Y_1|E_2 \rangle. \)

   It can be proven similarly to the completeness of APTC\textsuperscript{at} + CFAR + linear Rec + renaming, see [17].

2. If \( \langle X_1|E_1 \rangle \approx_{rhp} \langle Y_1|E_2 \rangle \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( \langle X_1|E_1 \rangle = \langle Y_1|E_2 \rangle. \)

   It can be proven similarly, just by replacement of \( \approx_{rbs} \) by \( \approx_{rhp} \), we omit it.

3. If \( \langle X_1|E_1 \rangle \approx_{rhp} \langle Y_1|E_2 \rangle \) for linear recursive specification \( E_1 \) and \( E_2 \), then \( \langle X_1|E_1 \rangle = \langle Y_1|E_2 \rangle. \)

   It can be proven similarly, just by replacement of \( \approx_{rbs} \) by \( \approx_{rhp} \), we omit it.

\[ \Box \]

11. Conclusions

Our previous work on truly concurrent process algebra APTC [17] is an axiomatization for true concurrency. There are correspondence between APTC and process algebra ACP [3]. In this paper, we extend APTC with timing related properties. Just like ACP with timing [23] [24] [25], APTC with timing also has four parts: discrete relative timing, discrete absolute timing, continuous relative timing and continuous absolute timing.

APTC with timing is formal theory for a mixture of true concurrency and timing, which can be used to verify the correctness of systems in a true concurrency flavor with timing related properties support.

References

[1] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. J. ACM, 1985, 32, 137-161.
[2] R. Milner. Communication and concurrency. Prentice Hall, 1989.
[3] R. Milner. A calculus of communicating systems. LNCS 92, Springer, 1980.
[4] W. Fokkink. Introduction to process algebra 2nd ed. Springer-Verlag, 2007.
[5] M. Nielsen, G. D. Plotkin, and G. Winskel. Petri nets, event structures and domains, Part I. Theoret. Comput. Sci. 1981, 13, 85-108.
[6] G. Winskel. Event structures. In Petri Nets: Applications and Relationships to Other Models of Concurrency, Wilfried Brauer, Wolfgang Reisig, and Grzegorz Rozenberg, Eds., Lecture Notes in Computer Science, 1987, vol. 255, Springer, Berlin, 325-392.
G. Winskel and M. Nielsen. Models for concurrency. In Samson Abramsky, Dov M. Gabbay, and Thomas S. E. Maibaum, Eds., Handbooks of Logic in Computer Science, 1995, vol. 4, Clarendon Press, Oxford, UK.

M. A. Bednarczyk. Hereditary history preserving bisimulations or what is the power of the future perfect in program logics. Tech. Rep. Polish Academy of Sciences. 1991.

S. B. Fröschle and T. T. Hildebrandt. On plain and hereditary history-preserving bisimulation. In Proceedings of MFCS'99, Mirosław Kutyłowski, Leszek Pacholski, and Tomasz Wierzbicki, Eds., Lecture Notes in Computer Science, 1999, vol. 1672, Springer, Berlin, 354-365.

J. Bradfield and C. Stirling. Modal mu-calculi. In Handbook of Modal Logic, Patrick Blackburn, Johan van Benthem, and Frank Wolter, Eds., Elsevier, Amsterdam, The Netherlands, 2006, 721-756.

I. Phillips and I. Ulidowski. Reverse bisimulations on stable configuration structures. In Proceedings of SOS'09, B. Klin and P. Sobociński, Eds., Electronic Proceedings in Theoretical Computer Science, 2010, vol. 18, Elsevier, Amsterdam, The Netherlands, 62-76.

I. Phillips and I. Ulidowski. A logic with reverse modalities for history-preserving bisimulations. In Proceedings of EXPRESS'11, Bas Luttik and Frank Valencia, Eds., Electronic Proceedings in Theoretical Computer Science, 2011, vol. 64, Elsevier, Amsterdam, The Netherlands, 104-118.

J. Gutierrez. On bisimulation and model-checking for concurrent systems with partial order semantics. Ph.D. dissertation. LFCS-University of Edinburgh, 2011.

P. Baldan and S. Crafa. A logic for true concurrency. In Proceedings of CONCUR’10, Paul Gastin and François Laroussinie, Eds., Lecture Notes in Computer Science, 2010, vol. 6269, Springer, Berlin, 147-161.

Y. Wang. Weakly true concurrency and its logic. 2016, Manuscript, arXiv:1606.06422.

Y. Wang. Algebraic laws for true concurrency. Submitted to JACM, 2016. arXiv:1611.09035.

Y. Wang. A calculus for true concurrency. Submitted to ACM TOCL, 2017. arxiv:1703.00159.

Y. Wang. A calculus of truly concurrent mobile processes. Manuscript, arXiv:1704.07774.

F.W. Vaandrager. Verification of two communication protocols by means of process algebra. Report CS-R8608, CWI, Amsterdam, 1986.

R. Glabbeek and U. Goltz. Refinement of actions and equivalence notions for concurrent systems. Acta Inf. 2001, 37, 4/5, 229-327.

K.A. Bartlett, R.A. Scantlebury, and P.T. Wilkinson. A note on reliable full-duplex transmission over half-duplex links. Communications of the ACM, 12(5):260-261, 1969.

J. C. M. Baeten and J. A. Bergstra. Real time process algebra. Formal Aspects of Computing 3.2(1991):142-188.

J. C. M. Baeten and J. A. Bergstra. Discrete time process algebra. CONCUR ’92. Springer Berlin Heidelberg, 1992.

J. C. M. Baeten and C. A. Middelburg. Process Algebra with Timing. Springer Science and Business Media, 2002.