Multi-optional Many-sorted Past Present Future structures and its description

Sergio Miguel Tomé

Grupo de Investigación en Minería de Datos (MiDa),
Universidad de Salamanca, Salamanca, Spain
sergiom@usal.es

January 26, 2018

Abstract

The cognitive theory of true conditions (CTTC) is a proposal to describe the model-theoretic semantics of symbolic cognitive architectures and design the implementation of cognitive abilities. The CTTC is formulated mathematically using the multi-optional many-sorted past present future (MMPPF) structures. This article defines mathematically the MMPPF structures and the formal languages proposed to describe them by the CTTC.

1 Introduction

Semantics is one of the most challenging aspects of cognitive architectures. The Cognitive Theory of True Conditions (CTTC) is a proposal to describe the model-theoretic semantics of symbolic cognitive architectures and to develop decision-making processes based on model-theoretic semantics [2, 3]. The main idea behind the CTTC is that the perceptual space is a set of formal languages that denote elements of a model embedded in a quotient space of the physical space. At this moment, the mathematical formulation of the CTTC is using the multi-optional many-sorted past present future (MMPPF) structures. Also, the CTTC proposes a hierarchy of three formal languages to describe them.

This article improves the previous characterization of the MMPPF structures and the hierarchy of the formal languages [2, 3]. The article is divided in three sections. The first section gives the mathematical definitions of MMPPF structures. The second section defines a hierarchy of three formal languages to describe an MMPPF structure. The last section addresses the semantics of the three formal languages of the hierarchy.
2 Multi-optional Many-sorted Past Present Future structures

A MMPPF structure is a nested structure of possible worlds. In other words, each world of the structure also contains another possible worlds structure. Thus, they are more complex than the classical structures of possible worlds used in temporal logics. This section includes the definitions of the MMPPF structure, temporal perspective structure, and state structure and its axioms. An MMPPF structure is constructed using temporal perspectives structures, and a temporal perspective structure is constructed using state structures. Thus, we define firstly a state structure, after a temporal perspective structure and finally an MMPPF structure. After the definitions, the axioms of the MMPPF structure are provided.

A state structure is a many-sorted structure, and it is denoted by $e$. Its definition is the following:

$$e = \langle \langle U_i \rangle \langle f_i \rangle \langle R_k \rangle \rangle$$

where the domains are

$$\langle U_i \rangle = \langle O, \mathcal{P}(H), \mathcal{P}(H \times V_o), ..., \mathcal{P}(H \times V_n), A_0^p, ..., A_z^p, A_0^p, ..., A_z^p, SRA \rangle$$

- $O = \{o_1, ..., o_z\}$
- $H = \{h_1, ..., h_{z'}\} \quad z \leq z'$
- $V_p = \{\emptyset\} \cup \{(w_1, ..., w_{\text{dim}(p)}): w_1 \in W_{p,1}, ..., w_{\text{dim}(p)} \in W_{p, \text{dim}(p)}\}$

where

$$\text{dim}: P \rightarrow \mathbb{N}$$

- $A_{o_i}^p = \{a_{o_i,1}^p, ..., a_{o_i,k}^p\} \quad o_i \in O$ and $a_{o_i,k}^p = (a_{o_i,k}^p \text{.in}, a_{o_i,k}^p \text{.ext})$

where

$$a_{o_i,k}^p \text{.in} : DH_{p,k} \times DV_{p,k} \rightarrow CV \quad DH_{p,k} \subseteq H, DV_{p,k}, CV_{p,k} \subseteq V_p$$

$$a_{o_i,k}^p \text{.ext} : H_{p,k} \times DV_{p,k} \rightarrow CV \quad H_{p,k} \subseteq H, DV_{p,k}, CV_{p,k} \subseteq V_p$$

- $SRA = \{\{(s_1, c_1), ..., (s_r, c_r)\}, ..., \{(c_1, c_1), ..., (c_r, c_r)\}\} \cup \{\emptyset\}$
the functions are

- $$\langle f_i \rangle = \langle ES_j, g_{j}^{s_0}, ..., g_{j}^{s_n}, g_{j}^{0}, ..., g_{j}^{n}, \theta_{p_0}^i, ..., \theta_{p_n}^i, \odot \rangle$$

- $$ES_j: O \rightarrow \mathcal{P}(H)$$

- $$g^{p_0}: O \rightarrow \mathcal{P}(H \times V_p)$$

- $$g^p: H \rightarrow V_p$$

- $$\theta^p: O \rightarrow \cup_i \mathcal{P}(A^p_i)$$

- $$\odot: O \rightarrow SRA$$

and the relations are

- $$\langle R_k \rangle = \langle S_{j}^{s_0,0}, S_{j}^{s_0,n}, ..., S_{j}^{s_0,0}, ..., S_{j}^{s_n,n} \rangle$$

where

$$S_{j}^{s_0,0} \subseteq O$$

The definition of an *structure of temporal perspective*, $$\mathcal{P}_t$$, is

$$\mathcal{P}_t = \langle M_{\mathcal{P}_t}, T_{\mathcal{P}_t}, E_{\mathcal{P}_t}, I_{\mathcal{P}_t}, \mathcal{D}_{\mathcal{P}_t}, \mathcal{L}_{\mathcal{P}_t}, \mathcal{S}_{\mathcal{P}_t}, \odot_{\mathcal{P}_t}, \succ_{\mathcal{P}_t} \rangle$$

where its domains are the following:

- $$M_{\mathcal{P}_t} = \{ m_1, ..., m_z \}$$

Each element of $$M_{\mathcal{P}_t}$$ is denominated moment of time. Each moment of time is a set defined in the following way:

$$m_{t'} \subseteq T_{\mathcal{P}_t} \times \mathbb{C} \times \mathbb{G} \times E_{\mathcal{P}_t} \quad \mathbb{C} = \{ \downarrow, \|, \| \} \quad \mathbb{G} = \{ \downarrow, \|, \| \}$$

Being $$t$$ the constant denoted by the temporal perspective $$\mathcal{P}_t$$, the elements of a moment of time, $$m_x \in M_{\mathcal{P}_t}$$, where $$r \in m_{t'}, r = (t', \bullet, \bullet, e_y)$$, $$t' \in T_{\mathcal{P}_t}$$, $$\bullet \in \mathbb{C}$$, $$\bullet \in \mathbb{G}$$ and $$e_y \in E_{\mathcal{P}_t}$$, fulfill the following:

- If $$t' < t$$ then $$\bullet = \downarrow$$
- If $$t' = t$$ then $$\bullet = \|$$
- If $$t' > t$$ then $$\bullet = \| \downarrow$$

Each element of a moment of time is denominated *reality*. 

3
\( T_{P_t} = \{1, \ldots, z\} \) is the time set. It must be noted that the number of moment of times is the same to the number of elements that has \( T \).

\( E_{P_t} = \{e_1, \ldots, e_s\} \) is a set of states structures

\[ I_{P_t} = \bigcup_j I_{e_j} \] and \( I_{e_j} = (\theta_j^0(o_1) \times \cdots \times \theta_j^n(o_1)) \times \cdots \times (\theta_j^0(o_z) \times \cdots \times \theta_j^n(o_z)) \) \( e_j \in E_{P_t} \)

Associated with \( I_{P_t} \) we use an auxiliary function, \( \pi^o_{\alpha_n} \), which projects a part of an element \( \vec{i} \) that belongs to \( I_{P_t} \), to do definitions. The function is defined in the following way:

\[ \pi^o_{\alpha_n}(\langle \vec{a}_o^1, \ldots, \vec{a}_o^z \rangle) = \vec{a}_o^o \]

where \( \vec{i} = (\vec{a}_o^1, \ldots, \vec{a}_o^z) \).

The functions are the following:

- \( D^p_{P_t} = \bigcup_j D^p_j \) and \( D^p_j = \{S^o_{j,p} o_u : o_u \in e_j\} \)

- \( L_{P_t} = \{l^0, \ldots, l^n\} \) is the set of environmental laws of \( P_t \).

- \( l^p : \mathcal{P}(\mathcal{P}(H \times V_p) \times \mathcal{O}) \times \mathcal{P}(A^p \times \mathcal{O}) \times D^p \rightarrow \mathcal{P}(\mathcal{P}(H \times V_p)) \)

- \( SL_{P_t} = \{sl^1, \ldots, sl^r\} \)

where \( sl^k : SRA \times \mathcal{O} \times E_{P_t} \times I_{P_t} \rightarrow SRA \)

- \( &_{P_t} : E_{P_t} \times I \rightarrow E_{P_t} \cup \{\perp\} \)

where

\[ &_{P_t}(e_x, \vec{i}) \begin{cases} \perp, & \vec{i} \notin I_{e_x}; \\
\vec{e}_y \in E_{P_t}, & \vec{i} \in I_{e_x}. \end{cases} \]

- \( \circ_{P_t} : T'_t \rightarrow I \) where \( T'_t = \{t' : t' \in T \text{ and } t' \leq t\} \)

- \( Succ_{P_t} : (T \times \mathcal{C} \times \mathcal{S} \times E) \times I \rightarrow (T \times \mathcal{C} \times \mathcal{S} \times E) \)

and the relation \( \prec_{P_t} \) is defined as follows:

\[ \prec_{P_t} \subset M_{P_t} \times M_{P_t} \]

Each \( (t', \phi, \vec{e}_x), (t'', \phi, \vec{e}_x) \) \( \in \prec_{P_t} \) fulfills that \( t' < t'' \).
An MMPPF structure is formally defined as the tuple
\[ \langle U, T, E, I, D^p, \rho_{MMPPF}, (d^p_d), L, \&, \oplus \rangle \]
where its domains are

- \( U = \{ P_1, ..., P_m \} \) is the set of temporal perspectives
- \( T = \{ 1, ..., m \} \) is the time set
- \( E = \{ e_1, ..., e_s \} \) is the set of states
- \( I = \bigcup_j I_e_j \) and
  \[ I_j = (\theta_j^0(o_1) \times \cdots \times \theta_j^n(o_1)) \times \cdots \times (\theta_j^0(o_z) \times \cdots \times \theta_j^n(o_z)) \quad e_j \in E \]
- \( D^p = \bigcup_j D^p_j \) and
  \[ D^p_j = \{ S'^{p \cdot p}o_u : S'^{p \cdot p}o_u \in e_j \} \]
is the dependency set

its functions are

- \( \rho_{MMPPF} : T \rightarrow U \)
- \( d^p_{e_j} : \mathcal{P}(\mathcal{P}(H \times V_0) \times \mathcal{P}(H \times V_p) \times \mathcal{O}) \times \mathcal{P}(A^p \times \mathcal{O}) \rightarrow D^p \quad e_j \in E \)
- \( L = \{ l^0, ..., l^n \} \) is the set of environmental laws.
- \( \mathcal{L}^p : \mathcal{P}(\mathcal{P}(H \times V_p) \times \mathcal{O}) \times \mathcal{P}(A^p \times \mathcal{O}) \times D^p \rightarrow \mathcal{P}(\mathcal{P}(H \times V_p)) \)
- \( \& : E \times I \rightarrow E \cup \{ \bot \} \)

where

\[ \&(e_x, i^x) \left\{ \begin{array}{ll} = \bot, & i^x \notin I_{e_x}; \\ = e_y \in E, & i^x \in I_{e_x}. \end{array} \right. \]

- \( \oplus : T \rightarrow I \)

The following axioms define an MMPPF structure:
• First Axiom
\[ \forall e_j \in E_{P_t} \quad (g^p_j(h) \neq \emptyset \land p > 0 \Rightarrow \{h\} \subset ES_j(o_i)) \]
The first axiom determines that an essence element has assigned a value of any property if and only if is assigned to an object.

• Second Axiom
\[ \forall e_j \in E_{P_t} \quad (p > 0) \land (g^p_j(h) \neq \emptyset) \Leftrightarrow \{h\} \subset ES_j(o_{j'}) \]
The second axiom determines that a place of the space is assigned to an object if and only if it has assigned any value of any other property.

• Third Axiom
\[ \forall e_j \in E_{P_t} \quad (g^p_j(h) = x) \Leftrightarrow \exists o_{i}(h, x) \in g^{*p}_{j'}(o_i) \]
The third axiom determines the relation between \( g^p_j \) and \( g^{*p}_{j'} \).

• Fourth Axiom
\[ \forall e_j \in E_{P_t} \quad (a^p_{i,k} \in A^p_i \Rightarrow DH_{p,k} = \bigcup_j ES_j(o_i)) \]
The fourth axiom ensures that an action acts independently of the assignment by \( ES_j \).

• Fifth Axiom
\[ \forall e_j, e_{j'} \in E_{P_t} \quad (h \in ES_j(o_i) \land h \in ES_{j'}(o_u) \Leftrightarrow u = i) \]
The fifth axiom determines that an essence element is only assigned to an object.

• Sixth Axiom
\[ Succ_{P_t}(t, e_j) = e_{j'} \Leftrightarrow d^p_j((g^p_{j'}(o_i), o_i), (g^{*p}_{j'}(o_i), o_i), ((a^p_{i,k}, o_i))_i) = D^p_{j'} \]
The sixth axiom determines that the dependencies set is coherent with the changes from \( e_j \) to \( e_{j'} \).

• Seventh Axiom
\[ Succ_{P_t}(t, e_j) = e_{j'} \Leftrightarrow \exists (a^p_{i,k}, o_i)_i \quad \forall p ((g^p_{j'}(o_i), o_i))_i = I^p((g^{*p}_{j'}(o_i), o_i)), (a^p_{i,k}, o_i)_i, D^p_{j'}) \]
The seventh axiom determines that if state \( e_{j'} \) succeed state \( e_j \), it is because the objects can produce changes that generate \( e_{j'} \) from \( e_j \).
• Eighth Axiom
\[ \forall e \in E_{P_t}, \theta^p(o_i) \in \mathcal{P}(A^p_t) \]

The eighth axiom determines that \( \theta^p \) only assigns actions to an object when they modify that object.

• Ninth Axiom
\[
\forall t \in T \quad T_{P_t} = T
\]
\[
\forall t \in T \quad E_{P_t} = E
\]
\[
\forall t \in T \quad I_{P_t} = I
\]
\[
\forall t \in T \quad L_{P_t} = L
\]
\[
\forall t \in T \quad \&_{P_t} = \&
\]
\[
\forall t \in T \quad \mathcal{D}^p_{P_t} = \mathcal{D}^p
\]
\[
\forall t \in T \quad (d^p_{j})_{P_t} = (d^p_{j})
\]
\[
\forall P_t, P_{t'} \quad |SL_{P_t}| = |SL_{P_{t'}}|
\]
\[
\forall P_t, P_{t'} \quad st^k_{P_t} = st^k_{P_{t'}}
\]

The ninth axiom determines that domains do not change from one temporal perspective to other.

• Tenth Axiom
\[
Succ_{P_t}(a, \bullet \subsection{\bullet} e, i) = \begin{cases} 
(t' + 1, \varepsilon, \downarrow, e) & \text{if } t' + 1 < t \text{ and } \Pi_{P_t}(t') = i \text{ and } \bullet = \varepsilon \\
(t' + 1, h, \downarrow, e) & \text{if } t' + 1 < t \text{ and } \Pi_{P_t}(t') \neq i \\
(t' + 1, \varepsilon, \|, e) & \text{if } t' + 1 = t \text{ and } \Pi_{P_t}(t') = i \text{ and } \bullet = \varepsilon \\
(t' + 1, h, \|, e) & \text{if } t' + 1 = t \text{ and } \Pi_{P_t}(t') \neq i \\
(t + 1, \varepsilon, \downarrow, e) & \text{if } t = t' \text{ and } \Pi_{P_t}(t') = i \text{ and } \bullet = \varepsilon \\
(t + 1, h, \downarrow, e) & \text{if } t = t' \text{ and } \Pi_{P_t}(t') \neq i \text{ and } \bullet = h \\
(t + 1, \varepsilon, \|, e) & \text{if } t' > t \text{ and } \bullet = \varepsilon \\
(t + 1, h, \|, e) & \text{if } t' > t \text{ and } \bullet = h
\end{cases}
\]

The tenth axiom determines in what period of time a temporal moment and its reality condition are situated.

• Eleventh Axiom
\[
\forall e_j, e_{j'} \in E_{P_t} \quad Succ_{P_t}(a, \bullet \subsection{\bullet} e, i) = (t', \bullet', \|', e_{j'}) \rightarrow \\
\rightarrow \circ_{j'}(a_i) = st^k(\circ_{j'}(a_i), a_i, e_j, i)
\]

The eleventh axiom determines that the change of reward and aversion sensation is coherent with the actions that the object carries out.
3 Formal languages to the MMPPF structures

The section defines three formal languages to describe an MMPPF structure: the perceptive language, the extended perceptive language and the categorical language. They are denoted $PL_{MPF}$, $PL_{MPF}^*$ and $CL_{MPF}$ respectively.

3.1 The perceptive language of MMPPF

The elements of the alphabet of the $PL_{MPF}$ language are the following symbols:

- A constant symbol $o_i$ for each element $o_i$ of $O$
- A constant symbol $h_i$ for each element of $h_i$ of $H$
- A constant symbol $w_{p,q}^i$ for each element of $w_{p,q}^i$ of each $W_{p,q}$.
- Two constant symbols, $\varepsilon$ and $h$, for the elements of $\mathcal{C} = \{\varepsilon, h\}$
- Three hybrid operators: $E_\uparrow|, @|\|, E_\downarrow|$.
- A constant symbol $f_{i,j}^{e}$ for each $f_{i,j}^{e}$
- A constant symbol $R_{e,j}^{k}$ for each $R_{e,j}^{k}$
- Four connectives $\triangleleft, \triangleright, \triangleright>$, and $\triangle$.
- Auxiliary symbols: $[,]$, $(,)$ and $|$. 

It must be noted that in the alphabet there are neither any kind of variables nor elements to design elements of the set $T$. 
A tuple of symbols $(w_{p,1}^{0}, ..., w_{p,\text{dim}(p)}^{0})$ is denoted as $v_p$.

The language $PL_{MPF}$ has the following three kinds of atomic formulas:

- Type I: $[\oplus|\odot|o_1|P_0|,...,|P_n]$ where
  - $\oplus \in \{\varepsilon, h\}$
  - $\odot \in \{E_\uparrow|, @|\|, E_\downarrow|\}$
  - $P_p \in \mathcal{P}(H \times W_{p,1} \times \cdots \times W_{p,\text{dim}(p)})$

- Type II: $[\oplus|\odot|S_j^{o_i,p}|o_a]$ where
  - $\oplus \in \{\varepsilon, h\}$
  - $\odot \in \{E_\uparrow|, @|\|, E_\downarrow|\}$

- Type II: $[\oplus|\odot|o_i|d_o_e]$ where
  - $\oplus \in \{\varepsilon, h\}$
  - $\odot \in \{E_\uparrow|, @|\|, E_\downarrow|\}$
  - $d_o_e = (a_0^i, ..., a_n^i)$
Any atomic formula is a well formed formula (wff).

The following rules determine when a wff combined with an atomic formula constitutes a wff:

- A wff $\psi = \psi' \land p$ and an atomic formula $q$ constitute a wff $\psi \land q$ if $\circ_p = \circ_q$ and $\circ_p = \circ_q$.

- A wff $\psi = \psi' \lor \phi$ where $\circ \in \{\land, \lor, \rightarrow\}$ and an atomic formula $q$ constitute a wff $\psi \lor q$ if any of the following conditions is fulfilled:
  - $\circ_p = \underline{E}$ and $\circ_q = \underline{E}$
  - $\circ_p = \underline{E}$ and $\circ_q = \underline{E}$
  - $\circ_p = \underline{E}$ and $\circ_q = \underline{E}$
  - $\circ_p = \underline{E}$ and $\circ_q = \underline{E}$

- A set of $n$ atomic formulas $p_1, ..., p_n$ are a wff $p_1 \land \cdots \land p_n$ is a wff if there are not a $p_x \equiv [\circ_p | \circ_o | P_0, ..., P_n]$ and $p_y \equiv [\circ'_p | \circ'_o | P'_0, ..., P'_n]$ where $x \neq y$ and $o_i = o_j$.

- A set of $n$ atomic formulas $p_1, ..., p_n$ are a wff $p_1 \land \cdots \land p_n$ is a wff if there are not a $p_x \equiv [\circ_p | \circ_o | (a_{i,k}', ..., a_{i,k})]$ and $p_y \equiv [\circ'_p | \circ'_o | (a_{j,l}', ..., a_{j,l})]$ where $o_i = o_j$.

Finally, $\psi \land \phi$ is a wff if $\psi$ and $\phi$ are wffs.

### 3.2 The extended perceptive language of MMPPF

$PL^*_{MMPF}$ is a language of description to MMPPF structures with a higher level of abstraction than $PL_{MMPF}$. The definition of the $PL^*_{MMPF}$ requires the definition of the alphabets of metainformation $\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4, \Pi_1$ and $\Pi_2$. These alphabets can be assigned to an object from its description in $PL_{MMPF}$. Thus, the atomic formulas of $PL^*_{MMPF}$ can be built from formulas of $PL_{MMPF}$. The rules to built a wff of $PL^*_{PPFM}$ are the same that the rules defined in $PL_{PPFM}$. 
The metainformation alphabet $\Sigma_1$

The metainformation alphabet $\Sigma_1$ has two elements that are denoted by 0 and 1. Thus,

$$\Sigma_1 = \{0, 1\}$$

The elements of the metainformation alphabets are relations. Thus, their definitions are the following:

- $\forall o_i, p, P_p \quad 0(o_i, p, P_p) \iff P_p = \emptyset$
- $\forall o_i, p, P_p \quad 1(o_i, p, P_p) \iff P_p \neq \emptyset$

Each element of $\Sigma_1$ is named momentary state. Using $\Sigma_1$, a qualitative state is given to an object in a moment of time to the $p$-property. It describes whether an object has a specific quality without naming the specific value. Then, a functor of the language $PL_{MMPF}^*$ is defined, which maps each object in a moment of time and in relation to a property into the metainformation alphabet $\Sigma_1$. It is denominated momentary state of the $p$-property and denoted by $ms_p$, which is defined in the following way:

$$ms^p(o_i, \psi) = \begin{cases} 0, & \text{if } P_p = \emptyset \\ 1, & \text{if } P_p \neq \emptyset \end{cases}$$

where $\psi \equiv [\lozenge |\square |o_i|P_o|, ..., |P_n]|$ and $\square$ a temporal situator.

The following formula can be built using the functor $em_p$:

$$\psi^* \equiv [\lozenge |\square |ms^0(o_i, \psi)|, ..., |ms^n(o_i, \psi)|]$$

Thus, $\psi^*$, which belongs to $PL_{MMPF}^*$, is built from $\psi$.

The metainformation alphabet $\Sigma_2$

The metainformation alphabet $\Sigma_2$ has three elements that are denoted by $\lozenge$, $\beta_1$ y $\beta_2$. Thus,

$$\Sigma_2 = \{\lozenge, \beta_1, \beta_2\}$$

As the elements of the metainformation alphabets are relations, their definitions are the following:

- $\forall P_p, P'_p \quad \beta_1(P_p, P'_p) \iff P_p = P'_p \land \phi$
- $\forall P_p, P'_p \quad \beta_2(P_p, P'_p) \iff P_p \neq P'_p \land \phi$
- $\forall P_p, P'_p \quad \lozenge (P_p, P'_p) \iff P_p = \emptyset \lor P'_p = \emptyset$

where

$$P_p, P'_p \in \mathcal{P}(H \times W_{p,1} \times \cdots \times W_{p,dim(p)}) = \mathcal{P}(H \times V_p)$$
and

\[ \phi \equiv P_p \neq \emptyset \land P'_p \neq \emptyset \]

It is denominated \textit{temporal state of the p-property} to the functor \( tsp^p \) that is defined in the following way:

\[
tsp^p(o_i, \varphi) = \begin{cases} 
\beta_1, & \text{if } P_p = P'_p \land \phi \\
\beta_2, & \text{if } P_p \neq P'_p \land \phi \\
l, & \text{if } ms_p(o_i, \psi) = 0 \lor ms_p(o_i, \psi') = 0 
\end{cases}
\]

where

\[ \varphi \equiv \psi \bowtie \psi' \]

and

\[ \psi \equiv \{ \multimap | o_i | P_0 |, ..., | P_{dim(p)} | \} \]

\[ \psi' \equiv \{ \multimap | o_i | P'_0 |, ..., | P'_{dim(p)} | \} \]

and

\[ \phi \equiv ms^p(o_i, \psi) \neq 0 \land ms^p(o_i, \varphi) \neq 0 \]

Using the functor \( tsp^p \), the following formula can be built:

\[ \varphi^* \equiv \{ \multimap | o_i | tsp^0(o_i, \varphi) |, ..., | tsp^n(o_i, \varphi) \} \]

Thus, \( \varphi^* \), which belongs to \( PL_{PPFM}^* \), is built from \( \varphi \).

The metainformation alphabet \( \Sigma_3 \)

The metainformation alphabet \( \Sigma_3 \) has three elements that are denoted by \( l \), \( \gamma_1 \) and \( \gamma_2 \). Thus,

\[ \Sigma_3 = \{ l, \gamma_1, \gamma_2 \} \]

The elements of the metainformation alphabets are relations. Thus, their definitions are the following:

- \( \forall P_{p,q}, P'_{p,q} \gamma_1 \langle P_{p,q}, P'_{p,q} \rangle \iff P_{p,q} = P'_{p,q} \)
- \( \forall P_{p,q}, P'_{p,q} \gamma_2 \langle P_{p,q}, P'_{p,q} \rangle \iff P_{p,q} \neq P'_{p,q} \)
- \( \forall P_{p,q}, P'_{p,q} \ l \langle P_{p,q}, P'_{p,q} \rangle \iff P_{p,q} = \emptyset \lor P'_{p,q} = \emptyset \)

where

\[ P_{p,q} = \{(h, w_q)\} \ h \in H w_k \in W^{p,q} \text{ where } V_p = W_{p,1} \times \cdots \times W_{p,q} \times \cdots \times W_{dim(p)} \]

\[ P'_{p,q} = \{(h', w'_q)\} \ h' \in H w'_k \in W^{p,q'} \text{ where } V'_p = W'_{p,1} \times \cdots \times W'_{p,q} \times \cdots \times W'_{dim(p)} \]

Previously to define the function that assigns elements of \( \Sigma_3 \) to the objects, it is necessary to define a function that assigns to each element essence the value
that has in a specific dimension of a specific property. The function is denoted by \( c_{j}^{p,q} \) and its definition is the following:

\[
\begin{align*}
c_{j}^{p,q} : & \quad H \rightarrow W_{p,q} \\
h_i & \mapsto w_q
\end{align*}
\]

where \( g_{j}^{p}(h_i) = (w_1, \ldots, w_{\text{dim}(p)}) \)

It is denominated temporal state of the \( q \)-component of the \( p \)-property to the functor \( tscp^{p,q} \) that is defined in the following way:

\[
tscp^{p,q}(o_i, \varphi) = \begin{cases} 
\gamma_1, \text{if } \phi \land \phi' \\
\gamma_2, \text{if } \phi \land \phi'' \\
1, \text{if } ms^{p}(o_i, \psi) = 0 \lor ms^{p}(o_i, \psi') = 0 
\end{cases}
\]

where

\[
\varphi \equiv \psi \triangleright \psi'
\]

and

\[
\begin{align*}
\psi & \equiv \llbracket \top | o_i | P_0 |, \ldots, | P_n] \\
\psi' & \equiv \llbracket \top | o_i | P'_0 |, \ldots, | P'_n] 
\end{align*}
\]

Using the functor \( tscp^{p,q} \), the following formula can be built:

\[
\varphi^* \equiv \llbracket \top | tscp^{p,q}(o_i, \varphi), \ldots, | tscp^{p,q}(o_i, \varphi)]
\]

Thus, \( \varphi^* \), which belongs to \( PL_{P,PP,F,M,M}^{*} \), is built from \( \varphi \).

The metainformation alphabet \( \Sigma_4 \)

The metainformation alphabet \( \Sigma_4 \) has three elements that are denoted by \( \triangleright \), \( \delta_1 \) and \( \delta_2 \). Thus,

\[
\Sigma_4 = \{ \triangleright, \delta_1, \delta_2 \}
\]

The elements of the metainformation alphabets are relations. Thus, their definitions are the following:

\[
\begin{align*}
\forall C_{p,q}, C'_{p,q} & \quad \delta_1 (C_{p,q}, C'_{p,q}) \iff \\
\forall (h_j, w_k) & \in C_{p,q} \Rightarrow C'_{p,q} h_j \in C'_{p,q} h_j \rightarrow w_k \prec w_{k'}
\end{align*}
\]
\[
\forall C_{p,q}, C'_{p,q} \quad \delta_1 \langle C_{p,q}, C'_{p,q} \rangle \iff \\
\forall (h_j, w_k) \in C_{p,q} \quad (h_{j'}', w_{k'}) \in C'_{p,q} \quad h_j = h_{j'}' \rightarrow w_k \succ w_{k'}
\]

\[
\forall C_{p,q}, C'_{p,q} \quad \iota \langle C_{p,q}, C'_{p,q} \rangle \iff C_q = \emptyset \lor C'_q = \emptyset
\]

where
\[C_{p,q} = \{(h_j, w_k) \mid h_j \in H, w_k \in W_{p,q} \mid V_p = W_1 \times \cdots \times W_q \times \cdots \times W_{\text{dim}(p)}\} \]
\[C'_{p,q} = \{(h_{j'}, w_{k'}) \mid h_j \in H, w_k \in W_{p,q} \mid V_p = W_1 \times \cdots \times W_q \times \cdots \times W_{\text{dim}(p)}\} \]

It is denominated *temporal order state of the q-component of the p-property* to the functor \(tscp^{p,q}\) that is defined in the following way:

\[
tscp^{p,q}(o_i, \varphi) = \begin{cases} 
\gamma_1, & \text{if } \phi \land \phi' \\
\gamma_2, & \text{if } \phi \land \phi'' \\
\iota, & \text{if } ms^p(o_i, \psi) = 0 \lor ms^p(o_i, \psi') = 0
\end{cases}
\]

where
\[\varphi \equiv \psi \triangleright \psi'\]

and
\[\psi \equiv [\llbracket o_i | g_j^{s_0}(o_i), ..., | g_j^{s_n}(o_i) \rrbracket] \]
\[\psi' \equiv [\llbracket o_i | g_j'^{s_0}(o_i), ..., | g_j'^{s_n}(o_i) \rrbracket] \]

and
\[\phi \equiv ms^p(o_i, \psi) \neq 0 \land ms^p(o_i, \psi') \neq 0 \]
\[\phi' \equiv \forall h_i \in ES_j(o_i) \cap ES_{j'}(o_i), c_j^{s^p,q}(h_i) < c_{j'}^{s^p,q}(h_i) \]
\[\phi'' \equiv \forall h_i \in ES_j(o_i) \cap ES_{j'}(o_i), c_j^{s^p,q}(h_i) > c_{j'}^{s^p,q}(h_i) \]

Using the functor \(tscp^{p,q}\), the following formula can be built:

\[\varphi^* \equiv [\llbracket o_i | tscp^{p,q}(o_i, \varphi), ..., | tscp^{p,q}(o_i, \varphi) \rrbracket] \]

Thus, \(\varphi^*\), which belongs to \(PL_{PPFM}^*\), is built from \(\varphi\).
The metainformation alphabet $\Pi_1$

The metainformation alphabet $\Pi_1$ has two elements that are denoted by $\kappa_1$ and $\kappa_2$. Thus,

$$\Pi_1 = \{\kappa_1, \kappa_2\}$$

It is denominated *relational state of the p-property* to the functor $rs^p$ that is defined in the following way:

$$rs^p(o_i, o_u, \psi) = \begin{cases} 
\kappa_1, & \text{if } \nexists k \psi_k \equiv \phi \\
\kappa_2, & \text{if } \exists k \psi_k \equiv \phi 
\end{cases}$$

where

$$\psi \equiv \psi_1 \triangleright \cdots \triangleright \psi_n$$

$$\phi \equiv \Box [S_j^{o_i-o_u}]$$

Using the functor $rs^p$, the following formula can be built:

$$\psi^* \equiv [\Box] [o_i] [o_u] [rs^p(o_i, \psi)]$$

Thus, $\psi^*$, which belongs to $PL_{PPFM}^*$, is built from $\psi$.

The metainformation alphabet $\Pi_2$

The metainformation alphabet $\Pi_2$ has four elements that are denoted by $\tau_1$, $\tau_2$, $\tau_3$ and $\tau_4$. Thus,

$$\Pi_2 = \{\tau_1, \tau_2, \tau_3, \tau_4\}$$

It is denominated *temporal relational state of the p-property* to the functor $rs^p$ that is defined in the following way:

$$rs^p(o_i, o_u, \varphi) = \begin{cases} 
\tau_1, & \text{if } \nexists k, k' \psi_k \equiv \phi \land \psi_k' \equiv \phi' \\
\tau_2, & \text{if } \nexists k \exists k' \psi_k \equiv \phi \land \psi_k' \equiv \phi' \\
\tau_3, & \text{if } \exists k \nexists k' \psi_k \equiv \phi \land \psi_k' \equiv \phi' \\
\tau_4, & \text{if } \exists k, k' \psi_k \equiv \phi \land \psi_k' \equiv \phi'
\end{cases}$$

where

$$\varphi \equiv \psi_1 \triangleright \cdots \triangleright \psi_n \triangleright \psi' \cdots \triangleright \psi_n$$

$$\phi \equiv \Box [S_j^{o_i-o_u}]$$

$$\phi' \equiv \Box [S_j^{o_i-o_u}]$$

Using the functor $rs^p$, the following formula can be built:

$$\psi^* \equiv [\Box] [o_i] [o_u] [rs^p(o_i, \varphi)]$$

Thus, $\psi^*$, which belongs to $PL_{PPFM}^*$, is built from $\psi$. 
3.3 The categorical language of MMPPF

$CL_{MMPPF}$ is at the top of the hierarchy of languages. It uses the metainformation alphabets of $PL_{MMPPF}^*$, but the atomic formulas qualify the intervals of time. It is abstracted from the duration of the time interval. There are four kinds of atomic formulas as are in $PL_{MMPPF}$. The atomic formulas can be created with elements that belong to sets called categories. There are five categories: objects, patterns of objects, conditions of reality, temporal situators, and registers of states.

- Objects $C_O = \{o_1, ..., o_z\}$
- Patterns of objects $C_{PO} = \{\lambda o. \varphi_i, ...\}$ where $\varphi_i$ is a formula that is true if the object fulfills a specific feature.
- Conditions of Reality $C_{CR} = \{\varepsilon, h\}$
- Temporal situators $C_{TS} = \{\updownarrow, ||, \updownarrow\}$
- Registers of states $C_{RS}$ whose elements are functions that produce atomic formulas. There are three kinds of functions.
  - $\lambda tp\lambda o_i[tp|o_i|p|x] \quad x \in \Sigma_1 \cup \Sigma_2$
  - $\lambda tp\lambda o_i[tp|o_i|p|q|x] \quad x \in \Sigma_3 \cup \Sigma_4$
  - $\lambda tp\lambda o_i[tp|o_i|o_u|x] \quad x \in \prod_2$

where $p$ determines a property, $q$ determines a dimension of a property, and $tp$ a determines temporal positioner.

Each element of a category is named atom. An atomic well-formed formula (wff) of $CL_{MMPPF}$ can be generated by applying $\beta$-reduction to the atoms according to their types. The compound formulas of $CL_{MMPPF}$ have $\uparrow$, $\updownarrow$, and $\triangle$ as connective. However, the meaning of the connectives is different in $CL_{MMPPF}$ from $PL_{MMPPF}$ and $PL_{MMPPF}^*$ because atomic formulas are not about moments of time but intervals of time. Thus, the connectives relate intervals of time.

4 Semantics

We have defined a class of mathematical structures and three formal languages to describe structures of that class. We need to know when a formula describe rightly a structure. Since, the formal languages is a hierarchy, the true of a formula of a language can be derived from the true of a formula of a language of a lower level. Thus, to provide a semantics for the hierarchy of languages, firstly, it must be defined a satisfiability for each formula of $PL_{MMPPF}$. To do that we need an interpretation to each symbol that composes each atomic formula. Because the alphabet of $PL_{MMPPF}$ does not have variables we do not
need an assignation function. Given a \( r_c = (\mathcal{R}, \mathcal{S}, t, e_i) \), the interpretation function \( I^{r_c} \) fulfills the following \( I^{e_i} \subset I^{r_c} \). Thus, for example, the following is fulfilled:

\[
I^{r_j}(S_{oi}) = I^{e_i}(S_{oi,p}) = S_{oi,p}
\]

We will use the letters: \( p \) to designate atomic formulas of type I and II, \( i \) to designate atomic formulas of type III, and \( \varphi, \psi, \ldots \) to designate formulas of \( PL_{MMPPF} \).

### 4.1 The satisfiability relation of \( PL_{MMPPF} \)

The satisfiability relation of a formula of \( PL_{MMPPF} \) with \( \downarrow \) or \( \triangleright \) is not only about the satisfiability of the two sides of the connective in theirs moments of time also those connectives mean that there is a transformation from one to the other. For example, if \( p \) is true in \( m_t \) and \( q \) is true in \( m_{t+1} \) we cannot ensure that \( p \downarrow q \) is true. Only if there is a path from any of the realities in which \( p \) is true to any of the realities of the next moment of time in which \( q \) is true, then \( p \downarrow q \) is true. Due to that fact, the definition of satisfiability relation to the connectives \( \downarrow \) and \( \triangleright \) is complex.

Being \( \mathfrak{M} \) a \( MMPPF \) structure, the satisfiability relation \( \vdash \) to the formulas of \( PL_{MMPPF} \) is defined as follows:

\[
\mathfrak{M}, \mathcal{P}_t \vdash \phi \text{ if } \mathfrak{M}, \mathcal{P}_t, \min(T) \vdash^\leq \phi
\]

\[
\mathfrak{M}, \mathcal{P}_t, t' \vdash^\leq p_1 \uparrow \cdots \uparrow p_n \uparrow i_1 \uparrow \cdots \uparrow i_n' \uparrow \varphi \text{ iff there exist } t'', t' \leq t'' \text{ and } \mathfrak{M}, \mathcal{P}_t, t'' \mathfrak{M}, \mathcal{P}_t' \vdash \varphi \text{ where } t''' = t'' + 1 \text{ and } \mathfrak{M}, \mathcal{P}_t, t' \vdash i_1 \uparrow \cdots \uparrow i_n' \text{ and } \mathfrak{M}, \mathcal{P}_t, t_1 \vdash i_1 \uparrow \cdots \uparrow i_n' \text{ and } \pi_{\alpha_{i_1}}(\vec{i}) = I^{r_c}(\vec{a}_{o_{i_1}}) \text{ and } \cdots \text{ and } \pi_{\alpha_{i_n'}}(\vec{i}) = I^{r_c}(\vec{a}_{o_{i_n'}})
\]

16
\[\mathcal{M}, \mathcal{P}_t, t', RI \models \phi \iff p_1 \land \cdots \land p_n \land i_1 \land \cdots \land i_{n'} \land \phi \iff \text{iff} \]
\[\text{iff } R'' \neq \emptyset \text{ and } \mathcal{M}, \mathcal{P}_t, t', RI \models \phi \text{ where} \]
where \( R'' = \{ r_v : r_v \in R', (r_u, I') \in RI, i_w \in I \text{ where } \& (e_x, i_w) = e_y \} \neq \emptyset \) and
and \( t'' = t' + 1 \) and
and \( R' = \{ r_i : r_i \in m_v \} \) and \( \mathcal{M}, \mathcal{P}_t, r_i \models p_1 \land \cdots \land p_n \) and \( \mathcal{M}, \mathcal{P}_t, r_i \models i_1 \land \cdots \land i_{n'} \) and
and \( r_u = (t_\alpha, \bullet, \mathbf{[]}, e_x) \) and \( r_v = (t', \bullet, \mathbf{[]}, e_y) \) and
and \( \text{and } RI' = \{ (r_v, I') : r_v \in R'' \} \text{ and } I'' = \{ \iota : \iota \in I x \pi_{\alpha_1}(\tilde{\iota}) = I^{x_\alpha}(\alpha_{o_1}) \} \) and...

\[\mathcal{M}, \mathcal{P}_t, r_e \models p_1 \land \cdots \land p_n \text{ if } \mathcal{M}, \mathcal{P}_t, r_e \models p_1 \text{ and } \]

... and \( \mathcal{M}, \mathcal{P}_t, r_e \models p_n \)

\[\mathcal{M}, \mathcal{P}_t, r_e \models i_1 \land \cdots \land i_{n'} \text{ if } \mathcal{M}, \mathcal{P}_t, r_e \models i_1 \text{ and } \]

... and \( \mathcal{M}, \mathcal{P}_t, r_e \models i_{n'} \)

\[\mathcal{M}, \mathcal{P}_t, t' \models \phi \text{ if there exists } t\text{'' and } t' \leq t\text{'' and } RT^{t'} \neq \emptyset \text{ and } \mathcal{M}, \mathcal{P}_t, t'' \text{, } RT^{t''} \models \phi \text{ where} \]
\[\text{where } t'' = t' + 1 \text{ and} \]
and \( RT^{t''} = \{ (r_v, I') : r_v \in m_v \} \) and \( \mathcal{M}, \mathcal{P}_t, r_e \models p_1 \land \cdots \land p_n \) and \( \mathcal{M}, \mathcal{P}_t, r_e \models i_1 \land \cdots \land i_{n'} \) and
and \( I' = \{ \tilde{i} : \tilde{i} \in I x \pi_{\alpha_1}(\tilde{i}) = I^{x_\alpha}(\alpha_{o_1}) \} \) and...

\[\mathcal{M}, \mathcal{P}_t, t', RI \models \phi \text{ if } \mathcal{M}, \mathcal{P}_t, t'' \text{, } RI'' \models \phi \text{ where } t'' = t' + 1 \text{ and} \]
\[\text{and } R'' = \{ r_v : r_v \in m_v \} \text{ and } \mathcal{M}, \mathcal{P}_t, r_e \models p_1 \land \cdots \land p_n \text{ and } \mathcal{M}, \mathcal{P}_t, r_e \models i_1 \land \cdots \land i_{n'} \} \]
and \( \text{and } RI'' = \{ (r_v, I'') : r_v \in R', (r_u, I') \in RI, i_w \in I _e \text{ where } \& (e_x, i_w) = e_y \} \) and
and \( I'' = \{ \tilde{i} : \tilde{i} \in I e x \pi_{\alpha_1}(\tilde{i}) = I^{x_\alpha}(\alpha_{o_1}) \} \) and...

\[\mathbf{17}\]
\[\mathcal{M}, \mathcal{P}, t', RI \models p_1 \cdot \cdots \cdot p_n \cdot i_1 \cdot \cdots \cdot i_m \not\models \varphi \text{ iff there exists} t'', \text{ and } t' < t'' \text{ and } R' \neq \emptyset \text{ and } \mathcal{M}, \mathcal{P}, t'', RI'_{\mathcal{P}} \models \varphi \text{ where} \]

where \(RI'_{\mathcal{P}} = \{ r_v : r_v \in m_e \}, \) and \(\mathcal{M}, \mathcal{P}, r_v \models p_1 \cdot \cdots \cdot p_n \) and \(\mathcal{M}, \mathcal{P}, r_v \models i_1 \cdot \cdots \cdot i_m \) and there and there exist an \((i_1, \ldots, i_m)\) where \(t'' - t' = m\) and \(i_1 \in I'\) and \((r_u, I') \in RI\) and

and \&(&(&(...(&e_x, i_1), \ldots), i_{m-1}), i_m) = e_y\} and \(t'' = t'+1\) and \(r_v = (t', \downarrow, \bullet^t, e_y)\) and \(r_u = (t_o, \downarrow, \bullet^t, e_x)\)

\[\mathcal{M}, \mathcal{P}, t' \not\models \varphi \triangle \psi \text{ iff } \mathcal{M}, \mathcal{P}, t' \models \varphi \text{ and } \mathcal{M}, \mathcal{P}, \min(T) \not\models \psi\]

\[\mathcal{M}, \mathcal{P}, r_e = (t', \downarrow, \bullet^t, e_x) \models [\mathcal{R}][\mathcal{S}][o_i[P_0], \ldots, [P_n] \text{ iff} \]

if \(r_e \in m_e \in M_{\mathcal{P}}\) and

and \(\mathcal{I}^e(\mathcal{R}) = \downarrow \text{ and } \mathcal{I}^e(\mathcal{S}) = \bullet^t \text{ and for all } P_p, \mathcal{I}^e(P_p) = \mathcal{g}_p^e(\mathcal{I}^e(o_i))\)

\[\mathcal{M}, \mathcal{P}, r_e = (t', \downarrow, \bullet^t, e_x) \models [\mathcal{R}][\mathcal{S}][S^{o^t}][o_u] \text{ iff} \]

if \(r_e \in m_e \in M_{\mathcal{P}}\) and

and \(\mathcal{I}^e(\mathcal{R}) = \downarrow \text{ and } \mathcal{I}^e(\mathcal{S}) = \bullet^t \text{ and } \mathcal{I}^e(o_u) \in \mathcal{I}^e(S^{o^t})\)

\[\mathcal{M}, \mathcal{P}, r_e = (t', \downarrow, \bullet^t, e_x) \models [\varepsilon][\mathcal{S}][o_i](a_{1,k}^0, \ldots, a_{n,k}^0) \text{ where } t' \leq t \text{ iff} \]

if \(r_e \in m_e \in M_{\mathcal{P}}\) and \(\downarrow_{i}(t') = (\mathcal{I}^e(a_i^0), \ldots, \mathcal{I}^e(a_i^n))\) and

and \(\mathcal{I}^e(\mathcal{S}) = \bullet^t \text{ where } r_e = (t', \downarrow, \bullet^t, e_x)\)

\[\mathcal{M}, \mathcal{P}, r_e = (t', \downarrow, \bullet^t, e_x) \models [\varepsilon|[\mathcal{S}][o_i](a_{1,k}^0, \ldots, a_{n,k}^0) \text{ where } t' > t \text{ iff} \]

if \(r_e \in m_e \in M_{\mathcal{P}}\) and \(\mathcal{I}^e(a_0^0) \in \theta_x^u(\mathcal{I}^e(o_i))\) and \(\ldots\), and \(\mathcal{I}^e(a_0^n) \in \theta_x^u(\mathcal{I}^e(o_i))\) and

and \(\mathcal{I}^e(\mathcal{S}) = \bullet^t \text{ where } r_e = (t', \downarrow, \bullet^t, e_x)\)

\[\mathcal{M}, \mathcal{P}, r_e = (t', \downarrow, \bullet^t, e_x) \models [b][\mathcal{S}][o_i](a_{1}^0, \ldots, a_{n}^0) \text{ iff} \]

if \(r_e \in m_e \in M_{\mathcal{P}}\) and \(\mathcal{I}^e(a_0^0) \in \theta_x^u(\mathcal{I}^e(o_i))\) and \(\ldots\), and \(\mathcal{I}^e(a_0^n) \in \theta_x^u(\mathcal{I}^e(o_i))\) and

and \(\mathcal{I}^e(\mathcal{S}) = \bullet^t \text{ where } r_e = (t', \downarrow, \bullet^t, e_x)\)

The connectives \(\blacktriangleleft\) and \(\blacktriangleleft\) are commutative while \(\blacktriangleleft\) and \(\blacktriangleright\) are noncommutative.
4.2 The satisfiability relation of $PL_{\text{MMPPF}}^*$

Once the satisfiability relation of $PL_{\text{MMPPF}}$ has been defined, using a translation function we define the satisfiability relation of $PL_{\text{MMPPF}}^*$. The translation function is $Tr_1$ defined in the following way:

$$Tr_1 : PL_{\text{MMPPF}} \rightarrow PL_{\text{MMPPF}}^*$$

The translation function $Tr_1$ uses the functions defined in the subsection 3.2. Thus, we define the satisfiability relation of $PL_{\text{MMPPF}}^*$ as follows:

$$\mathcal{M}, \mathcal{P}_t \models \phi \text{ iff there exists an } \varphi \in L_{\text{MMPPF}} \text{ where } Tr_1(\varphi) = \phi \text{ and } \mathcal{M}, \mathcal{P}_t, \min(T) \models ^{\leq} \varphi$$

4.3 The satisfiability relation of $LC_{\text{MMPPF}}$ and the recognizer grammars of true conditions

Finally, once defined the satisfiability relation of $PL_{\text{MMPPF}}^*$ we can define the satisfiability relation of $LC_{\text{MMPPF}}$ with the same method. Using a translation function, $Tr_2$, defined in the following way:

$$Tr_2 : PL_{\text{MMPPF}}^* \rightarrow LC_{\text{MMPPF}}$$

Thus, we define the satisfiability relation of $LC_{\text{MMPPF}}$ in the following way:

$$\mathcal{M}, \mathcal{P}_t \models \phi \text{ iff there exists an } \varphi \in PL_{\text{MMPPF}}^* \text{ where } Tr(\varphi) = \phi \text{ and } \mathcal{M}, \mathcal{P}_t, \min(T) \models ^{\leq} \varphi$$

The important issue about the satisfiability relation of $LC_{\text{MMPPF}}$ is how define $Tr_2$ because building a formula of $CL_{\text{MMPPF}}$ from $PL_{\text{MMPPF}}^*$ is a much more complicated task because we have many formulas of $PL_{\text{MMPPF}}^*$ that must be associated with only one formula of $CL_{\text{MMPPF}}$. This is caused because $CL_{\text{MMPPF}}$ bringing the qualification on time, so there is not a specific duration associated with a formula of $CL_{\text{MMPPF}}$. Thus, we need a mechanism that allows many-1 translations. It involves recognizing formulas of $PL_{\text{MMPPF}}^*$ and generating formulas of $CL_{\text{MMPPF}}$.

A formal grammar is a mechanism that allows recognition of an infinite number of expressions with a finite number of rules. However, this is not sufficient because we need the capacity to associate expressions of one language with expressions of another language. Fortunately, there is a mechanism that has been studied that has both features, syntax directed translation scheme(SDTS) [1]. The interesting issue of the SDTSs is that they permit many-1 translations, which is the feature needed to resolve the problem. We use the SDTS to define a mechanism named recognizer grammar of true conditions(RGTC). The RGTC is the mechanism that the CTTC uses to build a formula of $CL_{\text{MMPPF}}$ from $PL_{\text{MMPPF}}^*$. Thus, we begin with defining translation and the SDTS.
A translation is a subset of $\Sigma^* \times \Upsilon^*$ for finite alphabets $\Sigma$ and $\Upsilon$. An SDTS is a system that generalizes the notion of a context free grammar to generate a translation. It is denoted $G = (N, \Sigma, \Upsilon, R, S)$, where $N$, $\Sigma$, $\Upsilon$ are finite sets of nonterminal symbols, input symbols, and output symbols, respectively. $N$ is disjoint from $\Sigma \cup \Upsilon$. $S \in N$ is the start symbol. $R$ is the finite set of rules. A rule is an object $A \rightarrow (\alpha, \gamma, \Pi)$, where $A$ is a nonterminal symbol, where $\alpha \in (N \cup \Sigma)^*$, $\gamma \in (N \cup \Upsilon)^*$ and $\Pi$ is a permutation.

An SDTS $G = (V, \Sigma, \Upsilon, R, S)$ is simple if for all $A \rightarrow (\alpha, \gamma, \Pi) \in R$, $\Pi$ is an identity permutation (i.e., $\Pi(i) = i$ for all $i$). Since the permutation portion of a rule is irrelevant for a simple SDTS, it is deleted from all rules.

An RGTC is an SDTS with production rules that also have a set of assignments. The RGTC is a simple SDTS whose right sides of its production rules have at most one nonterminal symbol in $\alpha$ and $\gamma$. This is because the translations from $PL_{MMPPF}^*$ to $CL_{MMPPF}$ can be done sequentially because each formula of $PL_{MMPPF}^*$ is a sequence of atomic formulas that describes an object in a temporal way. This choice is because psychological time flows in only one direction. However, the RGTC is not only a simple SDTS because each rule also contains a set of assignments. This happens because the elements generated in the translation are $\lambda$-terms. Thus, $\gamma$ allows a set of assignments $\Lambda$. The general form for the rule of an RGTC is as follows:

$$A \rightarrow (\alpha, \gamma, \Lambda)$$

So, given $\gamma = \lambda x. M$ and $\Lambda = x := c$, the following process is done:

$$A \rightarrow \lambda x. M \{x := c\} \rightarrow_\beta M(c)$$

The first transformation is generated by the rule of the RGTC, and the second is done by $\beta$-reduction.

The important issue of the assignments is that they can delay a decision about the translation until move forward in the input formula.

References

[1] Aho, A., Ullman, J.: Properties of syntax directed translations. Journal of Computer and System Sciences 3(3), 319–334 (1969)

[2] Miguel-Tomé, S.: Principios matemáticos del pensamiento natural: Teoría cognitiva de condiciones de verdad. Gráficas Quintanilla (2006)

[3] Miguel-Tomé, S.: Principios matemáticos del comportamiento natural. Ph.D. thesis, Universidad de Salamanca (2017)