Realization of c-Inference as a SAT Problem

Christoph Beierle and Martin von Berg and Arthur Sanin
FernUniversität in Hagen
Faculty of Mathematics and Computer Science
Knowledge-Based Systems
58084 Hagen, Germany

Abstract
Semantically based on Spohn’s ranking functions, c-representations are special ranking models obtained by assigning individual integer impacts to the conditionals in a knowledge base R and by defining the rank of each possible world as the sum of the impacts of falsified conditionals. c-Inference is the inference relation taking all c-representations of a given knowledge base R into account. In this paper, we show that c-inference can be realized as a boolean satisfiability problem (SAT), which in turn allows c-inference to be implemented by a SAT solver. We provide a stepwise transformation of the characterization of c-inference as as constraint satisfaction problem (CSP), into a solvable-equivalent SAT problem. We present a SAT-based implementation of c-inference using the SMT solver Z3 (de Moura and Björner 2008); first evaluation results show that our SAT implementation of c-inference clearly outperforms the previous CSP-based implementation.

After briefly recalling the required basics of conditional logic and c-inference in Section 2, we develop the modeling of c-inference as a SAT problem in Section 3. Section 4 presents our implementation and first evaluation results, and Section 5 concludes and points out further work.

1 Introduction
For a knowledge base R containing qualitative conditionals of the form If A then usually B, formally denoted by (B|A), a variety of semantics have been proposed. Here, we will consider ranking functions, also called ordinal conditional functions (OCF) (Spohn 1988), which assign a degree of surprise to each world. A ranking function κ accepts (B|A) if κ considers the verification A ∧ B of the conditional strictly less surprising than its falsification A ∧ ¬B. Every OCF κ induces an inference relation where A nonmonotonically entails B if κ accepts (B|A). The so-called c-representations are special OCFs accepting all conditionals of R; they exhibit excellent inference properties (Kern-Isberner 2001; 2004). In particular, among all inference operators investigated in (Kern-Isberner, Beierle, and Brewka 2020), only inference with c-representations fully satisfies syntax splitting.

Up to now, InfOCF (Beierle, Eichhorn, and Kutsch 2017; Kutsch and Beierle 2021) provides the only implementation of c-inference which takes all c-representations of R into account (Beierle et al. 2018). In this paper, we present an alternative implementation of c-inference. While InfOCF uses the CSP characterization of c-inference given in (Beierle et al. 2018) and employs a Prolog-based CSP solver, we show how c-inference can be modelled as a SAT problem. To achieve this, we transform the CSP characterization of c-inference stepwise into a solvable-equivalent SAT problem. We implement this approach using the SMT solver Z3 (de Moura and Björner 2008); first evaluation results show that our SAT implementation of c-inference clearly outperforms the previous CSP-based implementation.

2 Conditional Logic and C-Inference
Conditional Logic and OCFs Let Σ = {v₁, ..., vₘ} be a propositional alphabet. A literal is the positive (vᵢ) or negated (¬vᵢ) form of a propositional variable, vᵢ stands for either vᵢ or ¬vᵢ. From these we obtain the propositional language L as the set of formulas of Σ closed under negation ¬, conjunction ∧, and disjunction ∨. For shorter formulas, we abbreviate conjunction by juxtaposition (i.e., AB stands for A ∧ B), and negation by overlining (i.e., A is equivalent to ¬A). Let Ω denote the set of possible worlds over L; Ω will be taken here simply as the set of all propositional interpretations over L and can be identified with the set of all complete conjunctions over Σ; we will often just write Ω instead of Ω. For ω ∈ Ω, ω |= A means that the propositional formula A ∈ L holds in the possible world ω. For A ∈ L, the set Ωₐ = {ω ∈ Ω | ω |= A} is the set of all complete conjunctions ω in which A holds.

A conditional (B|A) with A, B ∈ L encodes the defeasible rule “if A then normally B” and is a trivalent logical entity with the evaluation going back to de Finetti (1937):

\[ [(B|A)]_ω = \begin{cases} 
\text{true} & \text{iff } \omega |= AB \text{ (verification)} \\
\text{false} & \text{iff } \omega |= AB \text{ (falsification)} \\
\text{undefined} & \text{iff } \omega |= A \text{ (not applicable)} 
\end{cases} \]

An ordinal conditional function (OCF, ranking function), introduced first by Spohn (1988) in a more general form, is a function κ : Ω → N₀ U {∞} that assigns to each world ω ∈ Ω an implausibility rank κ(ω) such that at least one
world is maximally plausible, i.e., $\kappa^{-1}(0) \neq \emptyset$. The rank of a formula $A$ is $\kappa(A) = \min\{|\kappa(\omega)\mid \omega \models A\}$. An OCF $\kappa$ accepts a conditional $(B|A)$, denoted by $\kappa \models (B|A)$, iff the verification of the conditional is less than its falsification, i.e., $\kappa(AB) < \kappa(AB)$. Every OCF $\kappa$ induces a nonmonotonic inference relation $A \kappa$-entails $B$ is given by:

$$A \rightarrow^\kappa B \iff A \models_\perp \kappa(AB) < \kappa(AB)$$  \hspace{1cm} (1)

The acceptance relation is extended as usual to a set $\mathcal{R}$ of conditionals, called a knowledge base, by defining $\kappa \models \mathcal{R}$ iff $\kappa \models (B|A)$ for all $(B|A) \in \mathcal{R}$. This is synonymous to saying that $\kappa$ is admissible with respect to $\mathcal{R}$ (Goldszmidt and Pearl 1996), or that $\kappa$ is a ranking model of $\mathcal{R}$. $\mathcal{R}$ is consistent iff it has a ranking model, otherwise, $\mathcal{R}$ is inconsistent.

**C-Representations and C-Inference** Among the models of $\mathcal{R}$, $c$-representations are special ranking models obtained by assigning individual integer impacts to the conditionals in $\mathcal{R}$. For an in-depth introduction to c-representations and their use of the principle of conditional preservation we refer to (Kern-Isberner 2001; 2004). This is the central definition:

**Definition 1** (c-representation (Kern-Isberner 2001; 2004)). A $c$-representation of a knowledge base $\mathcal{R} = \{(B_1|A_1), \ldots, (B_n|A_n)\}$ is a ranking function $\kappa$ constructed from integer impacts $\eta_i \in \mathbb{N}_0$ assigned to each conditional $(B_i|A_i)$ such that $\kappa$ accepts $\mathcal{R}$ and is given by:

$$\kappa(\omega) = \sum_{1 \leq i \leq n, \omega \models A_i} \eta_i$$  \hspace{1cm} (2)

In penalty logic, violation ranks are sums of real numbers assigned to formulas (Pinkas 1995); the impacts in Def. 1 are nonnegative integers chosen such that (2) is satisfied.

C-inference takes all $c$-representations of $\mathcal{R}$ into account.

**Definition 2** (c-inference, $\models^c_\mathcal{R}$ (Beierle, Eichhorn, and Kern-Isberner 2016)). Let $\mathcal{R}$ be a knowledge base and let $A$, $B$ be formulas. $B$ is a (skeptical) c-inference from $A$ in the context of $\mathcal{R}$, denoted by $A \models^c_\mathcal{R} B$, iff $A \models^c\mathcal{R} B$ holds for all $c$-representations $\kappa$ for $\mathcal{R}$.

The set of $c$-representations of $\mathcal{R}$ can be modelled as solutions of a constraint satisfaction problem $\text{CR}(\mathcal{R})$ over $\mathbb{N}$ (see (Kern-Isberner 2004; Beierle, Eichhorn, and Kern-Isberner 2016)). The following definition sharpens $\text{CR}(\mathcal{R})$ by introducing an upper bound for the impact values $\eta_i$.

**Definition 3** (CR$^u(\mathcal{R})$ (Beierle et al. 2018)). Let $\mathcal{R} = \{(B_1|A_1), \ldots, (B_n|A_n)\}$ and $u \in \mathbb{N}$. The finite domain constraint satisfaction problem CR$^u(\mathcal{R})$ on the constraint variables $\{\eta_1, \ldots, \eta_n\}$ ranging over $\mathbb{N}$ is given by the constraints, for all $i \in \{1, \ldots, n\}$:

$$\eta_i \geq 0$$  \hspace{1cm} (3)

$$\eta_i > \min_{\omega \models A_i} \sum_{j \neq i} \eta_j - \min_{\omega \models A_i} \sum_{j \neq i} \eta_j$$  \hspace{1cm} (4)

$$\eta_i \leq u$$  \hspace{1cm} (5)

In Table 1: Verification and falsification with induced impacts for $\mathcal{R}_{\text{bird}}$ in Example 4.

A solution of $\text{CR}^u(\mathcal{R})$ is an $n$-tuple $(\eta_1, \ldots, \eta_n) \in \mathbb{N}_0^n$, its set of solutions is denoted by $\text{Sol}(\text{CR}^u(\mathcal{R}))$. For $\vec{\eta} = (\eta_1, \ldots, \eta_n) \in \text{Sol}(\text{CR}^u(\mathcal{R}))$ and $\kappa$ as in Equation (2), $\kappa$ is the OCF induced by $\vec{\eta}$, denoted by $\kappa_{\vec{\eta}}$, and the set of all induced OCFs is denoted by $\text{Sol}_{\text{OCF}}(\text{CR}^u(\mathcal{R})) = \{\kappa_{\vec{\eta}} \mid \vec{\eta} \in \text{Sol}(\text{CR}^u(\mathcal{R}))\}$.

**Example 4** ($\mathcal{R}_{\text{bird}}$). Let $\Sigma = \{b, p, f, w\}$ representing birds, penguins, flying things and winged things, and let $\mathcal{R}_{\text{bird}}$ contain $r_1 = (f|b)$, $r_2 = (\overline{f}|p)$, $r_3 = (b|p)$, and $r_4 = (w|b)$. For instance, $r_1$ expresses “birds usually fly”.

Taking the verification and falsification of these conditionals into account (cf. Table 1) CR$^u(\mathcal{R}_{\text{bird}})$ is then given as:

| $\omega$ | $r_1$: $bp$ | $r_2$: $bp$ | $r_3$: $bp$ | $r_4$: $bw$ | impact $\kappa_{\vec{\eta}_1}$ | $\kappa_{\vec{\eta}_2}$ | $\kappa_{\vec{\eta}_3}$ |
|---------|-------------|-------------|-------------|-------------|-----------------|-----------------|-----------------|
| $\Omega$ | 2 4 5       | 3 7 12      | 1 3 4       | 2 6 11       | 4 8 11          | 2 4 6           | 2 4 6           |

Table 1: Verification and falsification with induced impacts for $\mathcal{R}_{\text{bird}}$ in Example 4.

Some solutions of CR$^u(\mathcal{R}_{\text{bird}})$ and their induced ranking functions are given in Table 1. The solutions $\vec{\eta}_1$ and $\vec{\eta}_2$ are elements of $\text{Sol}(\text{CR}^u(\mathcal{R}_{\text{bird}}))$ for $u \geq 4$ while $\vec{\eta}_3$ is an element of $\text{Sol}(\text{CR}^u(\mathcal{R}_{\text{bird}}))$ only for $u \geq 7$.

While c-inference as given in Definition 2 takes all $c$-representations of a knowledge base $\mathcal{R}$ into account, c-inference defined with respect to a maximal impact value yields a kind of resource-bounded inference operation.

**Definition 5** (c-inference under maximal impact value, $\models^M_\mathcal{R}$ (Beierle et al. 2018)). Let $\mathcal{R}$ be a knowledge base, $u \in \mathbb{N}$, and let $A$, $B$ be formulas. $B$ is a (skeptical) c-inference from $A$ in the context of $\mathcal{R}$ under maximal impact value $u$, denoted by $A \models^M_\mathcal{R} B$, iff $A \models^c_\mathcal{R} B$ holds for all $c$-representations $\kappa$ with $\kappa \in \text{Sol}_{\text{OCF}}(\text{CR}^u(\mathcal{R}))$. 

Note that $A \vdash^C_R B$ implies $A \vdash^C_R u B$ and $A \vdash^{c,u'}_R B$ implies $A \vdash^{c,u}_R B$ for $u' \geq u$; these implications hold due to $\text{Sol}_{OCF}(CR^u(\mathcal{R})) \subseteq \text{Sol}_{OCF}(CR(\mathcal{R}))$ and $\text{Sol}_{OCF}(CR^u(\mathcal{R})) \subseteq \text{Sol}_{OCF}(CR^u(\mathcal{R}))$. The following proposition ensures that there is some $u$ such that $\vdash^C_R$ fully realizes c-inference.

**Proposition 6** (sufficient). For every knowledge base $\mathcal{R}$ there exists $u \in \mathbb{N}$ such that, for all formulas $A, B$, we have:

$$A \vdash^C_R B \iff A \vdash^{c,u}_R B \quad (6)$$

If (6) holds, then $CR^u(\mathcal{R})$ is called sufficient, and $u$ is sufficient for $\mathcal{R}$.

The proof of Proposition 6 relies on the fact that there are only finitely many ranking models of $\mathcal{R}$ that are pairwise not inferentially equivalent. While it is still an open problem to determine a minimally sufficient $u$ for a given $\mathcal{R}$, Komo and Beierle (2022) provide a criterion for the conditionals in $\mathcal{R}$ such that $u = |\mathcal{R}|$ is sufficient; empirical observations with the reasoning platform InfOCF suggest that $u = |\mathcal{R}|$ is indeed sufficient in most cases (Beierle and Kutsch 2019). If $u$ is sufficient for $\mathcal{R}$, c-inference can be modelled by a constraint satisfaction problem over finite domains.

**Proposition 7** ($CR^u(\mathcal{R}, A, B)$). Let $\mathcal{R} = \{ (B_1|A_1), \ldots, (B_n|A_n) \}$ be a consistent knowledge base, $CR^u(\mathcal{R})$ sufficient, and $A, B$ formulas. Then for

$$CR^u(\mathcal{R}, A, B) = CR^u(\mathcal{R}) \cup \{ \neg CR_R(B|A) \} \quad (7)$$

with $\neg CR_R(B|A)$ being the constraint

$$\min_{\omega \models AB} \sum_{1 \leq i \leq n} \eta_i \geq \min_{\omega \models AB} \sum_{1 \leq i \leq n} \eta_i$$

the following holds:

$$A \vdash^C_R B \iff CR^u(\mathcal{R}, A, B) \text{ is not solvable} \quad (9)$$

Proving Proposition 7 is analogous to the proof for the unbounded $CR_R(\mathcal{R})$ given in (Beiere, Eichhorn, and Kern-Ishberner 2016, Proposition 8), relying on that $CR^u(\mathcal{R})$ is sufficient. Since $\neg CR_R(B|A)$ does not introduce any variables not already in $CR^u(\mathcal{R})$, also $CR^u(\mathcal{R}, A, B)$ is a constraint satisfaction problem over finite domains. Thus, with a sufficient $CR^u(\mathcal{R})$, we can exploit techniques developed for finite domain constraint solvers, as they are available e.g. in constraint logic programming, for computing the inference $A \vdash^C_R B$ as done in InfOCF (Kutsch and Beierle 2021).

### 3 Modelling C-inference as a SAT Problem

In order to model c-inference as a Boolean satisfiability problem (SAT), we will refer to the following definition of constraint satisfaction problems.

**Definition 8** ($\text{CSP}_b$). A $\text{CSP}_b$ (Constraint Satisfaction Problem with Booleans) is defined as a tuple $(\mathcal{V}, L, U, B, S)$ where

- $\mathcal{V}$ is a finite subset of an infinite set of integer variables,
- $L : \mathcal{V} \rightarrow \mathbb{Z}$ represents the lower bound of $x \in \mathcal{V}$,
- $U : \mathcal{V} \rightarrow \mathbb{Z}$ represents the upper bound of $x \in \mathcal{V}$,
- $B$ is a finite subset of an infinite set of Boolean variables,
- $S$ is a finite set of clauses representing the constraint to be satisfied. Clauses over $\mathcal{V}$ and $B$ are disjunctions of literals which are either Boolean variables, negations thereof or comparisons of linear expressions (of the form $\sum a_i x_i$ with integer constants $a_i$ and integer variables $x_i$) with constants, variables and other linear expressions. In addition to linear expressions, we allow elements of clauses containing minima of a set of such expressions.

An assignment of a CSP is a pair $(\alpha, \beta)$ with $\alpha : \mathcal{V} \rightarrow \mathbb{Z}$ and $\beta : B \rightarrow \{ \text{true, false} \}$. A clause $C \in S$ is satisfied by $(\alpha, \beta)$, denoted as $(\alpha, \beta) \models C$, if $C$ evaluates to true under $(\alpha, \beta)$ and $L(x) \leq \alpha(x) \leq U(x)$ for all $x \in \mathcal{V}$. $C$ is satisfiable if there is an assignment that satisfies it.

Given this definition of a $\text{CSP}_b$, SAT can be defined as follows:

**Definition 9** (SAT). A Boolean Satisfiability Testing Problem (SAT) is a $\text{CSP}_b$ of the form $(\emptyset, \emptyset, \emptyset, B, S)$.

On the other hand, $CR^u(\mathcal{R}, A, B)$ can be modelled directly as a $\text{CSP}_b$ without Boolean variables.

**Definition 10** ($\text{CSP}_1(\mathcal{R}, A, B)$). Let $CR^u(\mathcal{R}, A, B)$ as in Proposition 7. The $\text{CSP}_1$-modelling of c-inference is

$$\text{CSP}_1(\mathcal{R}, A, B) = (V_\eta, L_\eta, U_\eta, \emptyset, S_\eta) \quad (10)$$

with $V_\eta = \{ \eta_1, \ldots, \eta_n \}$, $L_\eta(\eta_i) = 0$ and $U_\eta(\eta_i) = u$ for $i \in \{ 1, \ldots, n \}$, and $S_\eta$ is the set of constraints given by (4) in Definition 3 and (8) in Proposition 7.

Due to Proposition 7, we immediately get the following.

**Proposition 11.** If $u$ is sufficient for $\mathcal{R}$ and $A, B$ are formulas then $A \vdash^C_R B \iff \text{CSP}_1(\mathcal{R}, A, B)$ is not solvable.

Later, we will see how the complex terms in $S_\eta$ containing minimum functions can be transformed into so-called primitive comparisons of the form $x \leq c$ where $x$ is an integer variable and $c$ an integer constant.

Various different kinds of encodings of CSP into SAT exist, namely direct encoding, support encoding, log encoding, order encoding and its further development compact order encoding (Tanjo, Tamura, and Banbara 2011). Evaluations on different kinds of problems have demonstrated the general superiority of order encoding in terms of size of the generated SAT and performance of solving these; compact order encoding shows further improvements especially for CSP with larger domain size (Tanjo, Tamura, and Banbara 2011). Hereinafter, we will explicate order encoding, demonstrate by an example how a translation of $\text{CSP}_1(\mathcal{R}, A, B)$ is achieved by utilizing order encoding, and give a short overview of the idea behind compact order encoding, which we will use for evaluation purposes.

Encoding $\text{CSP}_1(\mathcal{R}, A, B)$ as a SAT with order encoding starts with introducing axiom clauses built of a Boolean variable $p_{\eta,j}$ for every admissible $i$ of every integer variable $\eta$ plus an additional variable $p_{\eta,-1}$. Applied to our CSP $CR^u(\mathcal{R}, A, B)$, we have $L(\eta_i) = 0$ for all $\eta_i$’s and a maximal impact constant $u$ yielding the propositional variables

$$B_\eta = \bigcup_{i=1}^n \{ p_{\eta,j} \mid j \in \{ -1, \ldots, u \} \}.$$
The variable \( p_{n-1} \) is used for encoding that the lower bound of \( \eta \) is always 0. For modeling upper and lower bounds and the order relation (encoding \( a < a + 1, a \in \mathbb{Z} \)), axiom clauses

\[
C^{ax}_n = \bigcup_{i=1}^{n} \left\{ \left\{ -p_{n, L(n) - 1} \right\}, \left\{ p_{n, U(n)} \right\} \right\} \\
\cup \bigcup_{j=0}^{U(n)} \left\{ -p_{n,j-1} \vee p_{n,j} \right\}
\]

are introduced.

After introduction of the axiom clauses, the clauses have to be transformed so that they only contain propositional variables. For this purpose, the complex terms with minima and sums of several integer variables first must be reduced to simpler comparisons. A \textit{primitive comparison} is a comparison in the form of \( x \leq c \) where \( x \) is an integer variable and \( c \) is an integer satisfying \( L(x) - 1 \leq c \leq U(x) \). For a proof of the following proposition, we refer to (Tamura et al. 2009).

**Proposition 12 (Conversion into primitive comparisons).** Let \( \langle V, L, U, B, S \rangle \) be a CSP, then for any assignment \( (\alpha, \beta) \) of the CSP, for any linear expression \( \sum_{i=1}^{n} a_{i}x_{i} \) with \( a_{i} \in \{-1, 1\} \), and for any integer \( c \geq (L(x_{1}) + \cdots + L(x_{n})) \) the following holds

\[
(\alpha, \beta) \models \sum_{i=1}^{n} a_{i}x_{i} \leq c \iff (\alpha, \beta) \models \bigwedge_{i=1}^{n} b_{i} = c-n+1 \bigvee_{i} (a_{i}x_{i} \leq b_{i}) \quad (11)
\]

where the parameters \( b_{i} \) range over \( \mathbb{Z} \) satisfying \( \sum_{i=1}^{n} b_{i} = c+n+1 \) and \( L(x_{i}) - 1 \leq b_{i} \leq U(x_{i}) \) for all \( i \), and where the translation \((\cdot)\) is defined as follows:

\[
(ax \leq b)(\cdot) = \begin{cases} 
\frac{b}{a} & \text{iff } (a = 1) \\
-(x \leq \frac{b}{a} - 1) & \text{iff } (a = -1)
\end{cases} \quad (13)
\]

We define \( C^{min}_{v,i} \) analogously to \( C^{min}_{v,i} \) and \( C^{min}_{v,q} \). \( C^{min}_{v,i} \) are analogously defined for the query conditional \((B_{i} \mid A_{i})\). Further, let \( S_{v} \subseteq \mathbb{Z} \) be a set of clauses obtained by replacing \( S_{q} \) each expression \( V_{min}, F_{min}, V_{min,q}, F_{min,q} \) with the corresponding variables \( x_{v, i}^{k}, x_{v, q}^{k}, x_{v, i}^{k+1}, x_{v, q}^{k+1} \) respectively. Then CSP\(^{a}_{2}(R, A, B) \) is given by \( \langle V_{X}, L_{0}, U_{0}, 0, S_{X} \rangle \) with

\[
V_{X} = V_{0} \cup V_{0'} \cup V_{f} \cup V_{0''} \cup V_{f} \cup S_{X} \\
S_{X} = S_{v} \cup C_{v,i} \cup C_{f,i} \cup C_{f,q} \cup C_{v,q} \cup C_{v,min} \cup C_{v,q} \cup C_{v,min} \cup C_{v, q} \cup C_{v,min}.
\]

Given the intermediate CSP from Definition 14, we are now able to define the SAT representation of c-inference. Our final transformation in the transition from CSP to SAT is performed by reduction of all comparisons into primitive comparisons and conversion of the integer variables representing the minima into propositional variables.

**Definition 15 (SAT\(^{a}_{2}(R, A, B) \)).** Let CSP\(^{a}_{2}(R, A, B) = \langle V_{X}, L_{0}, U_{0}, 0, S_{X} \rangle \) be as in Definition 14. Let \( S_{v} \) be the set of clauses resulting from the transformation of all comparison clauses from \( S_{v} \) into clauses only containing primitive comparisons, following the transformation rules for primitive comparisons from Proposition 12. Let \( B_{min} \) be new additional propositional variables representing the new integer variables for the minima given by

\[
B_{min} = \{ p_{v,q,j} \mid j \in \{-1, \ldots, u\} \}
\]
Let $C_{\text{ax}}$ be the axiom clauses for $B_{\text{min}}$, generated analogously to $C_{\text{ax}}$. Then SAT$^a(\mathcal{R}, A, B) = (\emptyset, \emptyset, B_\emptyset, S_\emptyset)$ with

$$B_\emptyset = B_\emptyset \cup B_{\text{min}} \text{ and } S_\emptyset = C_{\text{ax}} \cup C_{\text{min}} \cup S_{\psi}.$$  

Since CR$^a(\mathcal{R}, A, B)$ is satisfiable iff SAT$^a(\mathcal{R}, A, B)$ is satisfiable, we can generalize Proposition 11 and can reduce c-inference to a SAT problem.

**Proposition 16** (SAT$^a(\mathcal{R}, A, B)$). If $u$ is sufficient for $\mathcal{R}$ and $A, B$ are formulas then $A \models^c B$ iff SAT$^a(\mathcal{R}, A, B)$ is not solvable.

Before presenting our first evaluation results exploiting the SAT realization of c-inference given by Proposition 16, we illustrate the order encoding with Example 4.

**Example 17** ($\mathcal{R}_{\text{bird}}$ continued). The constraint $\eta_3 > \min\{\eta_1, \eta_2\} - 0$ from Example 4 is equivalent to $\min\{\eta_1, \eta_2\} - \eta_3 < 0$. By using the transformation rule for the less-relation, we get $\min\{\eta_1, \eta_2\} - \eta_3 + 1 \leq 0$, and using the rule for minima, the resulting clauses are

$$\left\{\{x_{\min} - \eta_3 \leq -1\}, \right.$$  
$$\{x_{\min} \geq \eta_1\}, \right.$$  
$$\{x_{\min} \geq \eta_2\},$$  
$$\{x_{\min} \leq \eta_1\}, \{x_{\min} \leq \eta_2\}\}.$$  

Converting comparisons into primitive comparisons and replacing these with the matching propositional variables yields

$$\left( p_{v,0} \lor \neg p_{\eta_3,0} \right) \land \left( p_{v,0} \lor \neg p_{\eta_3,1} \right) \land \left( p_{v,0} \lor \neg p_{\eta_3,2} \right)$$  
$$\land \left( p_{v,2} \lor \neg p_{\eta_3,3} \right) \land \left( p_{\eta_1,1} \lor \neg p_{v,1} \lor \neg p_{\eta_3,1} \right)$$  
$$\land \left( p_{\eta_1,0} \lor \neg p_{v,0} \right) \land \left( p_{\eta_1,1} \lor \neg p_{v,1} \lor \neg p_{\eta_3,1} \right)$$  
$$\land \left( p_{\eta_2,0} \lor \neg p_{v,0} \right) \land \left( p_{v,1} \lor \neg p_{\eta_1,1} \right)$$  
$$\land \left( p_{v,0} \lor \neg p_{\eta_1,0} \right) \land \left( p_{v,1} \lor \neg p_{\eta_2,1} \right) \land \left( p_{v,0} \lor \neg p_{\eta_2,0} \right) \right).$$

Note that the propositional formula from the preceding example encoding the original complex comparison is not yet in clausal normal form (CNF) and must, in order to be accepted by a SAT solver, be converted into CNF. The formula can be transformed in an equi-satisfiable CNF formula by introduction of new propositional variables (Prestwich 2009). The magnitude of the generated SAT problem becomes large with large domain size of the original CSP. For example, the number of SAT clauses while encoding an $n$-ary constraint $\sum_{i=1}^n a_i x_i \leq b$ will be $O(d^{n-1})$ with $d$ being the maximum domain size of the $x_i$. The idea behind compact order encoding, which we actually use in our implementation, is to use a numeric system of base $D \geq 2$ to keep the domain size smaller so that each integer variable $x$ is represented by a sum $\sum_{i=0}^{m-1} D^i x_i$ where $m = \lceil \log_D d \rceil$ and $0 \leq x_i < D$ for all $x_i$ (Tanjo, Tamura, and Banbara 2011).

## 4 First Evaluation Results

For the evaluation of the aforementioned approach to modeling c-inference as a SAT problem, we build upon the existing library InfOCF-Lib (Kutsch and Beierle 2021). The hitherto existing implementation utilizes the clp(FD) library of SICStus Prolog (Carlsson, Otter, and Carlson 1997) for calculating c-inference. For encoding c-inference as a SAT, we used Java-based SAT encoder Azucar (Tanjo, Tamura, and Banbara 2012). As Azucar is designed to be used with any external SAT solver, its output can be further processed with the satisfiability modulo theories (SMT) solver Z3 (de Moura and Björner 2008). SMT generalizes SAT; therefore, the widely recognized Z3 can also be used for the problem at hand, which we did for our empirical evaluation.

First, a knowledge base file provided in a specified format is parsed by InfOCF-Lib and together with a given query processed into a text file representing CR$^a(\mathcal{R}, A, B)$ in abridged XCSP 2.1 format (Roussel and Lecoutre 2009). This file is then translated by the Azucar library through compact order encoding with a base $D = \lceil u^{\frac{1}{2}} \rceil$ resulting in a representation of all integer variables by at most two digits. Azucar outputs a propositional formula in CNF; furthermore, it provides information about which propositional variable encodes which integer variable so that a satisfying assignment for the original CSP can be reconstructed through analysis of a satisfying assignment for the SAT. Finally, the CNF formula is passed to Z3 in order to check for satisfiability. An Intel CPU with 3.6 Ghz has been used for all computations with 4GB of 2667 Mhz DDR4 RAM provided for use within the programs.

Our first evaluation results (Table 2) show an increase in computation speed by up to 500% for the smaller knowledge bases with a number of signature elements and conditionals of four to five. Even more interesting are the results for the larger knowledge bases, for which the former approach utilizing Prolog resulted in either a timeout (which we set at 60 minutes) or memory errors because of buffer overflows. For our largest knowledge base $\mathcal{R}_9$, the SAT approach shows a jump in the time needed for computation, due to the increased number of comparisons with growing number of conditionals on the one hand and the exponential increase in considered worlds with growing number of signature elements on the other hand.

## 5 Conclusions and Further Work

We have seen a considerable speed-up in the computation of c-inference when using an implementation based on a SAT encoding. Knowledge bases with a signature size and number of conditionals for which c-inference could not even be computed by the previous approach are now accessible in reasonable time. While these results are encouraging, investigations of scalability and complexity of our SAT approach are still to be performed, as well as further empirical evaluation. In our current work, we are evaluating our approach with other SAT
schedulers. Our current work also includes finding an encoding specifically tailored to c-inference; recent developments show the possibility to find such a problem-specific tailoring in an automated way (Gorjiiara, Xu, and Demsky 2020). Furthermore, we will extend our approach to also cover credulous and weakly skeptical c-inference (Beierle et al. 2021) and conditional descriptor revision (Sauerwald et al. 2020; Haldimann et al. 2021).

Table 2: Evaluation of CSP- vs. SAT-approach with (variants of) knowledge bases from (Haldimann, Osiak, and Beierle 2020).

| Conditionals | Query | | CSP | | variables | | clauses | | literals |
|-------------|-------|-------|-------|-------|-------|-------|-------|
| | | | | | | | | | | |
| R₁ | (u|m), (v|n), (m|s), (v|s) | (u|s) | 4 | 4 | 1.546s | 0.504s | 105 | 314 | 901 |
| R₂ | (c|h), (v|e), (v|e), (v|e) | (v|l|c) | 4 | 4 | 1.470s | 0.447s | 64 | 191 | 549 |
| R₃ | (c|h), (v|b), (v|c), (v|e), (v|e) | (v|b|c) | 5 | 5 | 2.445s | 0.466s | 188 | 920 | 3024 |
| R₄ | (n|b), (m|b), (b|l), (m|l), (g|l) | (m|g|l) | 5 | 5 | timeout | 0.530s | 392 | 1440 | 4204 |
| R₅ | (v|m), (c|m), (m|e), (v|e), (m|k), (v|e) | (v|e) | 6 | 6 | timeout | 0.568s | 623 | 2498 | 7279 |
| R₆ | (a|l), (l|h), (v|l), (v|l), (l|z), (a|z) | (a|z) | 6 | 6 | mem err | 0.561s | 887 | 5383 | 17333 |
| R₇ | (v|m), (c|m), (u|m), (m|e), (v|e), (m|k), (v|e) | (v|e) | 6 | 7 | mem err | 0.666s | 1626 | 8676 | 27052 |
| R₈ | (i|w), (w|m), (v|m), (c|m), (m|e), (v|e), (m|k), (v|e), (u|l), (g|l), (s|b), (f|b) | (v|e) | 7 | 8 | mem err | 0.842s | 4371 | 29515 | 94490 |
| R₉ | (i|w), (w|m), (v|m), (c|m), (m|e), (v|e), (m|k), (v|e), (u|l), (g|l), (s|b), (f|b) | (v|e) | 11 | 12 | mem err | 83.22s | 86370 | 989267 | 3277125 |

References

Beierle, C., and Kutsch, S. 2019. Computation and comparison of nonmonotonic skeptical inference relations induced by sets of ranking models for the realization of intelligent agents. Applied Intelligence 49(1):28–43.

Beierle, C.; Eichhorn, C.; Kern-Iserbern, G.; and Kutsch, S. 2018. Properties of skeptical c-inference for conditional knowledge bases and its realization as a constraint satisfaction problem. Ann. Math. Artif. Intell. 83(3-4):247–275.

Beierle, C.; Eichhorn, C.; Kern-Iserbern, G.; and Kutsch, S. 2021. Properties and interrelationships of skeptical, weakly skeptical, and credulous inference induced by classes of minimal models. Artificial Intelligence 297.

Beierle, C.; Eichhorn, C.; and Kern-Iserbern, G. 2016. Skeptical inference based on c-representations and its characterization as a constraint satisfaction problem. In FoIKS 2016, volume 9616 of LNCS, 65–82. Springer.

Beierle, C.; Eichhorn, C.; and Kutsch, S. 2017. A practical comparison of qualitative inferences with preferred ranking models. KI – Künstliche Intelligenz 31(1):41–52.

Carlsson, M.; Ottosson, G.; and Carlson, B. 1997. An open-ended finite domain constraint solver. In PLILP’97, volume 1292 of LNCS, 191–206. Springer.

de Finetti, B. 1937. La prévision, ses lois logiques et ses sources subjectives. Ann. Inst. H. Poincaré 7(1):1–68. Engl. transl. The Theory of Probability, J. Wiley & Sons, 1974.

de Moura, L. M., and Bjørner, N. 2008. Z3: an efficient SMT solver. In TACAS 2008, volume 4963 of LNCS, 337–340. Springer.

Goldszmidt, M., and Pearl, J. 1996. Qualitative probabilities for default reasoning, belief revision, and causal modeling. Artificial Intelligence 84(1-2):57–112.

Gorjiiara, H.; Xu, G. H.; and Demsky, B. 2020. Satune: synthesizing efficient SAT encoders. Proc. ACM Program. Lang. 4(OOPSLA):146:1–146:32.

Haldimann, J.; Sauerwald, K.; von Berg, M.; Kern-Iserbern, G.; and Beierle, C. 2021. Conditional descriptor revision and its modelling by a CSP. In JELIA 2021, volume 12678 of LNCS, 35–49. Springer.

Haldimann, J.; Osiak, A.; and Beierle, C. 2020. Modelling and reasoning in biomedical applications with qualitative conditional logic. In KI 2020, volume 12325 of LNCS, 283–289. Springer.

Kern-Iserbern, G.; Beierle, C.; and Brewka, G. 2020. Syntax splitting = relevance + independence: New postulates for nonmonotonic reasoning from conditional belief bases. In KR-2020, 560–571.

Kern-Iserbern, G. 2001. Conditionals in nonmonotonic reasoning and belief revision, volume 2087 of LNAI. Springer.

Kern-Iserbern, G. 2004. A thorough axiomatization of a principle of conditional preservation in belief revision. Ann. Math. Artif. Intell. 40(1-2):127–164.

Komo, C., and Beierle, C. 2022. Nonmonotonic reasoning from conditional knowledge bases with system W. Ann. Math. Artif. Intell. 90(1):107–144.

Kutsch, S., and Beierle, C. 2021. InfOCF-Web: An online tool for nonmonotonic reasoning with conditionals and ranking functions. In IJCAI 2021, 4996–4999. ijcai.org.

Pinkas, G. 1995. Reasoning, nonmonotonicity and learning in connectionist networks that capture propositional knowledge. Artif. Intell. 77(2):203 – 247.

Prestwich, S. D. 2009. CNF encodings. In Handbook of Satisfiability. IOS Press. 75–97.

Roussel, O., and Lecoutre, C. 2009. XML representation of constraint networks: Format XCSP 2.1. CoRR abs/0902.2362.

Sauerwald, K.; Haldimann, J.; von Berg, M.; and Beierle, C. 2020. Descriptor revision for conditionals: Literal descriptors and conditional preservation. In KI-2020, volume 12325 of LNCS, 204–218. Springer.

Spohn, W. 1988. Ordinal conditional functions: a dynamic theory of epistemic states. In Causation in Decision, Belief Change, and Statistics, II. Kluwer Academic Publishers. 105–134.

Tamura, N.; Taga, A.; Kitagawa, S.; and Banbara, M. 2009. Compiling finite linear CSP into SAT. Constraints An Int. J. 14(2):254–272.

Tanjo, T.; Tamura, N.; and Banbara, M. 2011. A compact and efficient SAT-encoding of finite domain CSP. In SAT 2011, volume 6965 of LNCS, 375–376. Springer.

Tanjo, T.; Tamura, N.; and Banbara, M. 2012. Azucar: A SAT-based CSP solver using compact order encoding. In SAT 2012, volume 7317 of LNCS, 456–462. Springer.