A Complete Axiomatization for Reduced Clock Constraint Specification Language

Bogdan Chornomaz, Kirill Rukkas, and Kseniia Troino

Department of Computer Science,
Kharkiv National University

Abstract. Clock Constraint Specification Language, or CCSL, is a domain-specific language designed to model distributed real-time systems in terms of logical time, that is of sequences of events. Typical application of CCSL is to serve as a specification language for verification of specified systems.
In this paper we provide a sound and complete axiomatic for propositional logic over large fragment of CCSL which we call reduced CCSL, or RCCSL. This axiomatics appears to be rather simple, thus enabling effective verification of RCCSL specifications.

Keywords: time model, verification model, propositional logic, completeness model, specification language

Key terms: Computation, FormalMethod, SpecificationProcess, MathematicalModel, DistributedSystem

1 Introduction

Models dealing with discrete logical time rather than with real-valued "physical time" are well known to computer science, one classical example being Lamport's algorithm for distributed clock synchronization [2]. In this paper we study model called Clock Constraint Specification Language (CCSL), proposed by F. Mallet in his dissertation [3]. Initially developed as UML profile for MARTE, CCSL later become domain-specific language on its own.

Constraints developed with CCSL allow some obvious logical reasoning, however the natural question arises: to which extent can this reasoning be carried out. Considerable efforts involving reasoning about CCSL constraints are put fromm the standpoint of formal verification of MARTE models with CCSL constraints. Usually this verification is carried out by transforming model into some framework which provides a model-checking ability, for example UPAAL [5] or Fiacre [1].

We take a different approach, having in mind a rather theoretical goal. We restrict ourselves with a fragment of CCSL called *Reduced CCSL* (RCCSL), for which we provide sound and complete system of axioms for propositional logic over it. The question of constructing such axiomatics for CCSL itself remains open.

ICTERI 2016, Kyiv, Ukraine, June 21-24, 2016

As we see it, this result can be interesting in two ways. First, complete and sound axiomatic gives way to effective verification of constraint system. On the other hand, lots of effort are put into extending CCSL by endowing it with clock compositions or by introducing delays, see for example [4]. These attempts reveal a demand for deeper understanding of which elementary dependencies between clock can exist and which expressive power do they bring about. The language of logic which we utilize here can be exceptionally well suited for answering such questions.

The structure of the paper is as follows: In Section 2 we introduce basic terminology from CCSL and from logic background. In Section 3 we introduce axiomatics which, as we argue later, is a complete and sound axiomatics for modeling time systems with RCCSL. In Section 4 we take first preliminary steps towards the proof of completeness. Section 5 contains the most essential part of the paper, providing the central part of the proof of completeness. Section 6 concludes the paper.

2 CCSL and RCCSL

We start with a short introduction to CCSL terminology following [6]. We define *time structure* as a tuple $(\mathcal{I}, \leq, \mathcal{C}, \pi)$ where $\mathcal{I}$ is at most countable set of *instants*, $\mathcal{C}$ is a finite set of clocks, $\leq$ is a preorder on $\mathcal{I}$ such that $(x]$ is finite for all $x \in \mathcal{I}$, where $(x]$ is a principal preorder-ideal of x, that is

$$(x] = \{y \mid y \leq x\};$$

finally, π is a function $\pi \colon \mathcal{I} \mapsto \mathcal{C}$, mapping each instant into corresponding clock, such that for every $c \in \mathcal{C}$ each preimage $\mathcal{I}_c = \pi^{-1}(c)$ is linearly ordered and nonempty. We denote an equivalence relation corresponding to $\leq$ by $\doteq$, that is, $x \doteq y$ if $x \leq y$ and $y \leq x$.

Thus defined, clock systems models a situation when we are given a set of clocks, each producing signals for which we only know their relative order of appearance. Signals in different clocks may be incomparable, or, on the contrary, may happen simultaneously. Proposition 1 in [6] can clarify this parallel.

Proposition 1 *For a time structure $(\mathcal{I}, \leq, \mathcal{C}, \pi)$, each $\mathcal{I}_c$ is well-ordered with ordinal type at most ω, where ω is first finite ordinal.*

In order to visualize time system we use a modified version of Hasse diagram, see Figure 1 below. Vertical lines depicts instants of corresponding clocks, slanted lines between clocks depict covering relation of $\leq$, finally instants equivalent with respect to $\leq$ are connected by horizontal lines.

For an instant i of $\mathcal{I}$ we define *height* $h(i)$ of i as the length of the largest increasing chain in $\mathcal{I}$, ending in i minus one. This is a standard order-theoretic definition which can be reformulated in an inductive way as follows

- $h(i) = 0$, whenever i is minimal in $\mathcal{I}$;

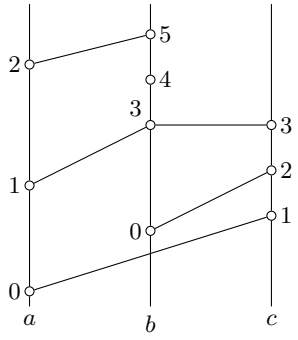

Fig. 1. Example of Hasse diagram depicting a time structure with clocks a, b and c.

– $h(i) = max\{h(j) \mid j < i\} + 1$.

Heights of instants are also depicted on Figure 1 below

We say that time system is linear if $\leq$ is a linear quasi-order, that is $\leq$ becomes a linear order after factorization by corresponding equivalence relation. When depicting a linear time system, we will omit slanted and horizontal lines on Hasse diagram, the order between instants in this case is represented solely by their relative height.

We define *run* as a time system with the set $\mathcal{I}$ defined as a custom subset of $\mathcal{C} \times \mathbb{N}$, with $\pi(a,k) = a$ and $(a,k) \leq (b,l)$ if and only if $k \leq l$, for all (a,k) and (b,l) in $\mathcal{C} \times \mathbb{N}$. In a run every set $\mathcal{I}_c$ can be treated as a finite or infinite sequence of natural numbers. Trivially, every run is a linear time system. On the other hand, every time system can be considered a run, as stated in the Proposition 2 below, we refer to [6] for proof.

Proposition 2 *For a time system $T = (\mathcal{I}, \leq, \mathcal{C}, \pi)$ define a run $L(I) = (\mathcal{I}', \mathcal{C})$ where $\mathcal{I}'$ is defined as*

$$\mathcal{I}' = \{(\pi(x), h(x)) \mid x \in \mathcal{I})\}.$$

Then $L(I)$ is a linear time system, and if I is linear, then $L(I) \cong I$.

Let us fix a potentially infinite set of clocks $\mathcal{C}$ and a set $\mathcal{S}*$ of binary relation symbols

$$\mathcal{S}^* = \{\equiv, \prec, \preceq, \subseteq, \#\},$$

called *coincidence*, *precedence*, *cause*, *subclocking* and *exclusion* correspondingly. Now we introduce *CCSL* as a propositional language over a set $\mathcal{T}$ of *terms*, where each term is defined as a triple xRy, x and y are clocks and R is a relational symbol from $\mathcal{S}^*$. Thus, $\mathcal{T} = \mathcal{C} \times \mathcal{S} \times \mathcal{C}$.

Thus, examples of terms are: $a \equiv b$, $a\#a$ or $b \prec d$; and CCSL formulas are: $a \equiv b$, $\neg(a\#b) \wedge a \prec c$, $\neg(\neg(a\#b) \wedge (b \preccurlyeq c \vee \neg(a \equiv b)))$.

Reduced CCSL, or RCCSL, is defined in a similar way, by excluding precedence from the set of possible relational symbols. That is, wee fix

$$\mathcal{S} = \{\equiv, \preceq, \subseteq, \#\}\,.$$

In the example above, $\neg(a\#b)\wedge a \prec c$ is not an RCCSL formula, but $\neg(a\#b)\wedge a \preccurlyeq c$ is.

CCSL terms can be interpreted on time systems with the set of clocks $\mathcal{C}$ as follows:

- $a \equiv b \Leftrightarrow$ for any $x \in \mathcal{I}_a$ there is $y \in \mathcal{I}_b$) with $x \doteq y$ and vice versa;
- $a \prec b \Leftrightarrow$ there is a strict extensive $h\colon \mathcal{C}_a \to \mathcal{C}_b$, that is, $x < h(x)$, for all $x \in \mathcal{C}_a$;
- $a \preceq b \Leftrightarrow$ there is an extensive $h\colon \mathcal{C}_a \to \mathcal{C}_b$, that is, $x \leq h(x)$, for all $x \in \mathcal{C}_a$;
- $a \subseteq b \Leftrightarrow$ for any $x \in \mathcal{I}_a$ there is $y \in \mathcal{I}_b$) with $x \doteq y$;
- $a \# b \Leftrightarrow x \neq y$ for all $x \in \mathcal{C}_a$, $y \in C_b$.

Figure 2 illustrates the interpretation of CCSL terms on time systems.

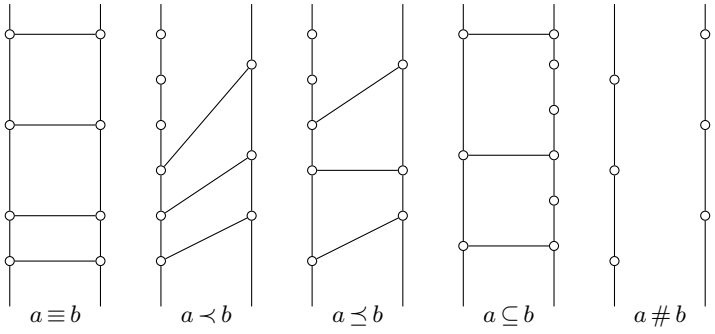

Fig. 2. Cause and subclocking relation on clocks a and b

After we interpret all CCSL terms, the interpretation of CCSL formulas on time systems is straightforward. For example, time system on Figure 1 satisfies formula $(a \prec b) \wedge (\neg(b\#c) \vee \neg(a\#c))$.

3 Axiomatics

Notice, that not all CCSL formulas, satisfiable as propositional formulas, can be satisfied on time system. For example, a formula $\neg(a \equiv a)$ is clearly satisfiable if we put $(a \equiv a) = False$. On the other hand, this formula can hold on no time structure. In fact, the following formulas, which we call axioms, hold on any time structure.

Axiomatics A1 ($\mathbf{A}_0$)

1. $\equiv$ *is an equivalence relation, which is congruent with respect to every other relation in* $\mathcal{S}$*, i.e.*

$$\forall * \in \mathcal{S} \forall a, b, a', b' \in \mathcal{C}, a \equiv a', b \equiv b' : a * b \Leftrightarrow a' * b';$$

2. $\preceq$ *and* $\subseteq$ *are quasiorders (i.e. reflexive and transitive) sharing associated equivalence relation* $\equiv$*;*
3. $a \subseteq b \Rightarrow b \preceq a$*;*
4. $\#$ *is irreflexive and symmetrical;*
5. $a \subseteq b, b \# c \Rightarrow a \# c$*.*

We say that a CCSL formula is *valid* if it holds under any interpretation on time structures. We say that an axiomatics is *sound* if all its axioms are valid formulas. Similarly, we call axiomatics *complete*, if any valid formula can be inferred from it. Throughout the paper, we consider all propositional axioms and propositional inference rules over CCSL terms to hold.

We denote Axiomatics A1 by $\mathbf{A}_0$ and refer to [6] for its soundness. In fact, in the following two sections we will show that this axiom set is also complete. Each of the axioms in $\mathbf{A}_0$ is not a singular propositional axiom, but rather a set of axioms, described in generally used terminology.

Define *relation structure* as a pair $(\mathcal{C}, R)$, where R is a subset in $\mathcal{T}$, which we treat as a valuation on a set of CCSL terms on $\mathcal{C}$. Usually we will write relation structure simply as R. For each relation symbol $*$ we define its corresponding relation in a relation structure:

$$*_R = \{(a, b) \mid a, b \in \mathcal{C}; (a, *, b) \in R\}$$

For a set of propositional formulas F over T we write $R \models F$ iff all formulas in F hold under truth assignment R, and say that R *comply* with F. Given a time structure T we define $R(T)$ as a valuation of terms given by their interpretation on T. We say that time structure T complies with F, denoted $T \models F$, if $R(T)$ does.

Using completeness of propositional logic we infer following general fact, which is essential for our proof of completeness

Proposition 3 $\mathbf{A}$ *is complete iff there is a model for a relation structure* R *whenever* R *complies with* $\mathbf{A}$*.*

Proof. ($\Rightarrow$) : Let R comply with $\mathbf{A}$ but does not have a model. Let F_R be a propositional formula which holds only for R. As there is no model for R, $\neg F_R$ is a valid formula and thus $\mathbf{A} \vdash \neg F_R$. By propositional inference rules this is equivalent to the formula $\neg \mathbf{A} \vee \neg F_R$ being propositionally valid, but it does not hold on R, a contradiction.

($\Leftarrow$) : Let F be a valid formula not inferred from $\mathbf{A}$. Then there is a propositional structure R such that R complies with $\mathbf{A}$ but not with F. By assumption, there is a time structure T with $R(T) = R$. But then F does not hold on T, which means F is not valid, a contradiction. ■

4 Completeness: preliminary reduction

Our first goal is to eliminate relations $\equiv$ and $\#$. We say that relational structure R is *clarified* if $\equiv$ is an equivalence relation. We say that time structure is clarified if its relational structure is.

For a relational structure $(\mathcal{C}_0, R_0)$ we define its *factorization* as a relational structure $(\mathcal{C}_e, R_e)$, denoted $(\mathcal{C}_e, R_e) = {}^{(\mathcal{C}_0,R_0)}/_{\equiv_0}$, such that:

- $\mathcal{C}_e$ is a set of equivalence classes of $\mathcal{C}_0$ by $\equiv_0$;
- $R_e = \{([a]_{\equiv_0}, *, [b]_{\equiv_0}) \mid a, b \in \mathcal{C}_0, * \in \mathcal{S}; (a, *, b) \in R_0\}$.

The fact that $\equiv_0$ is a congruence guarantees that the definition of R_e is consistent. Obviously, R_e is clarified for any R_0. Let us now define simplified axiom system $\mathbf{A}_e$, which defines axiomatics for clarified time systems.

Axiomatics A2 ($\mathbf{A}_e$)

1. *$\equiv$ is an equity;*
2. *$\preceq$ and $\subseteq$ are partial orders;*

3 - 5. *same as in* $\mathbf{A}_0$

To justify passing from $\mathbf{A}_0$ to $\mathbf{A}_e$ let us prove the following two easy lemmas:

Lemma 1 *If a relational model R_A complies with $\mathbf{A}_0$ then ${}^{R_A}/_{\equiv_A}$ complies with $\mathbf{A}_e$.*

Proof. Obvious. ■

Lemma 2 *Given a relational model R_A, if there is model for ${}^{R_A}/_{\equiv_A}$ then there is model for R_A.*

Proof. Let $\mathcal{C}$ be clocks of ${}^{R_A}/_{\equiv_A}$ and let T' be a model for ${}^{R_A}/_{\equiv_A}$. Then clocks from $\mathcal{C}$ are equivalence classes of clocks from $\mathcal{C}_A$. Define time system T with clocks $\mathcal{C}_A$ such that $\mathcal{I}^T_a = \mathcal{I}^{T'}_{[a]}$. Now it is a trivial fact to check that T is a model for R_A. ■

As our next step we relax restrictions on $\#$ relation. From $\mathbf{A}_e$ we can easily deduce that

$$\exists c\colon c \subseteq a, c \subseteq b \Rightarrow \neg a \# b.$$

Indeed

$$a\#b, c \subseteq a, c \subseteq b \Rightarrow c\#b, c \subseteq b \Rightarrow c\#c,$$

a contradiction.

It would be convenient for us if this implication would work the other way as well, i.e. if $\exists c\colon c \subseteq a, c \subseteq b \Leftrightarrow \neg a\#b$. However it is easy to construct a counterexample to this statement, on the other hand, it is always possible to "extend" the temporal structure by adding a clock (or several), so that it become true, see Figure 3 below:

We want to make the same trick, but with relation structures rather than with temporal structures.

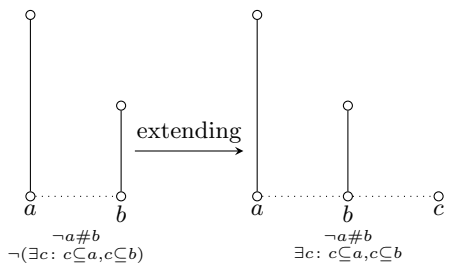

Fig. 3. Extension of a temporal structure

For a relation structure $(\mathcal{C}_A, R_A)$ and a set of clocks $\mathcal{C}_B \subseteq \mathcal{C}_A$ by *restriction* of relation structure R_A to $\mathcal{C}_B$, denoted $R_A|_{\mathcal{C}_B}$, we understand a relation structure $(\mathcal{C}_B, R_B)$, where:

$$R_B = R_A \bigcap \mathcal{C}_B \times \mathcal{S} \times \mathcal{C}_B.$$

Next, by *extension* of a relation structure R_A we understand a relation structure R_B such that $R_A = R_B|_{\mathcal{C}_A}$. We say that relation structure is *subclock-closed*, iff for it holds

$$\exists c\colon c \subseteq a, c \subseteq b \Leftrightarrow \neg a \# b \qquad (\mathfrak{S})$$

The time structure is *subclock-closed* iff its relation structure is subclock-closed. Theorem 1 allows us to consider only subclock-closed relation structures:

Theorem 1 *For each relation structure satisfying* $\mathbf{A}_e$ *there is a subclock-closed extension satisfying* $\mathbf{A}_e$.

Proof. Take a non subclock-closed relation structure R_A satisfying $\mathbf{A}_e$. Let $\lhd$ be some strict linear order on the set $\mathcal{C}_A$. Define a set R as:

$$R = \{c_{ab} \mid a, b \in \mathcal{C}_A, a \lhd b; \neg a \# b, \neg\,(\exists c\colon c \subseteq a, c \subseteq b)\}.$$

Define the set of clocks $\mathcal{C}_B$ of our to-be-constructed system as:

$$\mathcal{C}_B = \mathcal{C}_A \cup R.$$

Now we need to define relations R_B in three cases: for pair of old clocks, for pair of new clocks and for a pair of an old and a new clock. In the first case we simply put $R_B|_{\mathcal{C}_A} = R_A$, which automatically assures that R_B is an extension of R_A.

In case of two elements from R we put:

$$c_{ab} \preccurlyeq c_{de} \Leftrightarrow c_{ab} = c_{de};$$
$$c_{ab} \subseteq c_{de} \Leftrightarrow c_{ab} = c_{de};$$
$$c_{ab} \# c_{de} \Leftrightarrow c_{ab} \neq c_{de}.$$

Finally, when elements are from different sets, put:

$$\forall a, b, c\colon c_{ab} \not\preccurlyeq d, d \not\subseteq c_{ab}$$

and

$$\begin{aligned} &d \preccurlyeq c_{ab} \Leftrightarrow d \preccurlyeq a \text{ or } d \preccurlyeq b;\\ &c_{ab} \subseteq d \Leftrightarrow a \subseteq d \text{ or } b \subseteq d;\\ &c_{ab} \# d \Leftrightarrow d \# c_{ab} \Leftrightarrow \neg c_{ab} \subseteq d \Leftrightarrow a \not\subseteq d \text{ and } b \not\subseteq d \end{aligned}$$

Generally, for a pair of clocks a, b from $\mathcal{C}_A$ such that $\neg a \# b, \neg (\exists c \in \mathcal{C}_A\colon c \subseteq a, c \subseteq b)$ by $\overline{c_{ab}}$ we understand element c_{ab} in case when $a \triangleleft b$ and element c_{ba} in case when $b \triangleleft a$.

Observe, that R_B is subclock-closed, indeed:

$$\begin{aligned} a, b \in \mathcal{C}_A, \neg a \# b, \neg (\exists c \in \mathcal{C}_A\colon c \subseteq a, c \subseteq b) \Rightarrow \exists c = \overline{c_{ab}} \in \mathcal{C}_B\colon c \subseteq a, c \subseteq b\\ a \in \mathcal{C}_A, c_{de} \in R, \neg a \# c_{de} \Rightarrow c_{de} \subseteq a;\\ c_{ab}, c_{de} \in R, \neg c_{ab} \# c_{de} \Rightarrow c_{ab} = c_{de}. \end{aligned}$$

So what is left to check is that R_B satisfy $\mathbf{A}_e$, let us do it.

1. $\preceq$ and $\subseteq$ are partial orders:
   Check the transitivity of $\preceq$: if $f \preccurlyeq e \preccurlyeq a$, $a \neq e$, $e \neq f$ then, as all elements in R are not larger than any element of $\mathcal{C}_A$, we have two possibilities: either $a, e, f \in \mathcal{C}_A$, in which case the transitivity is trivial, or $a = c_{b,d} \in R, e, f \in \mathcal{C}_A$, but then:

   $$e \preccurlyeq c_{b,d} \Leftrightarrow e \preccurlyeq b \text{ or } e \preccurlyeq d \Rightarrow f \preccurlyeq b \text{ or } f \preccurlyeq d \Leftrightarrow f \preccurlyeq c_{b,d}.$$

   The reflexivity and the fact that associated equivalence relation is an equity are trivial. The proof for $\subseteq$ is similar.
2. $a \subseteq b \Rightarrow b \preceq a$: obvious.
3. $\#$ is irreflexive and symmetrical: obvious.
4. $a \subseteq b, b \# c \Rightarrow a \# c$:
   follows from the fact that R_B is subclock-closed and that $\subseteq$ is a partial order, indeed let $\neg a \# c$ then $\exists d\colon d \subseteq a, d \subseteq c$. But then $d \subseteq a \subseteq b$ and so $\neg b \# c$, a contradiction.

■

Now, if we fix axiom system $\mathbf{A}_F$, which is a proper subset of $\mathbf{A}_e$,

Axiomatics A3 ($\mathbf{A}_F$)

1. $\preceq$ and $\subseteq$ are partial orders;
2. $a \subseteq b \Rightarrow b \preceq a$;

then Theorem 1 together with Lemmas 1 and 2 yield the following corollary

Corollary 1 *If every subclock-closed relational structure compliant with $\mathbf{A}_F$ can be realized by clarified subclock-closed time system, then every relational structure compliant with $\mathbf{A}_0$ can be realized by some time system.*

Proof. Let R be a relational structure compliant with $\mathbf{A}_0$. Then by Lemma 1 $R_E = {}^{R}\!/_{\equiv}$ is compliant with $\mathbf{A}_e$. Take a subclock-closed extension R_F of R_E, which exists by Theorem 1. Now R_F satisfy $\mathbf{A}_e$ and thus $\mathbf{A}_F$ and by the hypothesis of the corollary there is subclock-closed time system T_F such that $R_F = R(T_F)$.

Notice that while $R(T_F)$ contains only interpretations of formulas with $\preccurlyeq$ and $\subseteq$, by $\mathfrak{S}$ it can be extended in a straightforward fashion to formulas with $\#$ and $\equiv$. Thus, T_F restricted to $\mathcal{C}_F$ is a model for R_F. The claim of the corollary now follows by Lemma 2. ■

5 Completeness: modelling $\subseteq$ and $\preccurlyeq$

Theorem 2 *Take a set of clocks $\mathcal{C}$ and a pair of partial orders $\preccurlyeq$ and $\subseteq$ on it, such that $a \subseteq b \Rightarrow b \preccurlyeq a$. Then there is a subclock-closed time structure T over the same clocks such that $\subseteq_T = \subseteq$ and $\preccurlyeq_T = \preccurlyeq$.*

Proof. Let $n = |\mathcal{C}|$. Fix some linear order $\lhd$ on $\mathcal{C}$ and denote by c_i the i-th clock in $\mathcal{C}$ relative to this order. Fix p linear orders $\pi_1 \dots \pi_n$ on $\mathcal{C}$ so that

1. each π_i is an extension of $\preccurlyeq$;
2. $\bigcap_{i=1\dots p} \pi_i = \preccurlyeq$

Clearly, such orders can be found and p can always be chosen so that $p \leq n$.

Next, define function $f\colon \mathcal{C} \to \mathbb{N}^+$ as:

$$f(x) = \begin{cases} 1 & \forall y \neq x\colon y \not\preccurlyeq x \\ \displaystyle\sum_{y \preccurlyeq x, y \neq x} f(y) & \text{otherwise} \end{cases}$$

It is clear that, although this definition is "recursive", the recursion is only seeming: for bottom elements with regards to $\preccurlyeq$, i.e. for elements of height 1, the sum is empty, and so f equals 1; for elements of height 2 f is defined via elements of height 1, etc., see Figure 4 below.

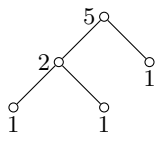

Fig. 4. Function f recursively defined for a partial order $\preccurlyeq$

Let $F = \sum_x f(x)$. Choose $l_1, \ldots, l_p \in \mathbb{N}$ as:

$$\begin{aligned} l_1 &= 1; \\ l_i &= F * (l_1 + \cdots + l_{i-1}) + 1. \end{aligned}$$

Or, using a direct formula:

$$l_i = (F+1)^{i-1}.$$

Now, define $\mathcal{I}$ to be the chain with $F * (l_1 + \cdots + l_p)$ elements:

$$\mathcal{I} = \{(i, c, j) \mid i = 1 \ldots p; c \in \mathcal{C}; j = 1 \ldots l_i * f(c)\}$$

with order given by

$$(i,c,j) \leq (q,d,r) \Leftrightarrow \begin{bmatrix} i < q \\ i = q, c >_{\pi_i} d \\ i = q, c = d, j \leq r \end{bmatrix}$$

So this order is "almost" lexicographic, except that the second letter is each time ordered differently, depending of the first one.

Define clock c_T in T as:

$$c_T = \{(i, d, j) \in \mathcal{I} \mid d \subseteq c\}.$$

We claim that thus defined time structure T satisfies the requirements of the theorem.

From the definition of c_T it is obvious that T is subclock-closed, and that it satisfies $\subseteq_T = \subseteq$. The nontrivial part is to show that $\preceq_T = \preceq$, which we will do now by separately showing that $\preceq_T \subseteq \preceq$ and $\preceq \subseteq \preceq_T$.

1. $\preceq_T \subseteq \preceq$:
   Let $a, b \in \mathcal{C}, a \preccurlyeq b$. Then for each i we have $a \leq_{\pi_i} b$. Define the function $h : a_T \to b_T$ as:
   $$h(i, x, j) = (i, b, g_a(i, x, j))$$
   where
   $$g_a(i, x, j) = \left| \{(i, x', j') \in a_T \mid (i, x', j') \leq (i, x, j)\} \right|.$$
   Observe, that to assure that h is correctly defined, we must check that $g_a(i, x, j) \leq l_i * f(b)$, but indeed:
   $$\begin{aligned} g_a(i, x, j) &= \left| \{(i, x', j') \in a_T \mid (i, x', j') \leq (i, x, j)\} \right| \\ &\leq \left| \{(i, x', j') \in a_T\} \right| \\ &= \sum_{x' \subseteq a} f(x') * l_i \leq \sum_{x' \preccurlyeq a} f(x') * l_i \\ &\leq \sum_{x' \preccurlyeq b, x' \neq b} f(x') * l_i = l_i * f(b). \end{aligned}$$
   So h is defined correctly, it is obviously strictly increasing and from $(i, x, j) \in a_T$ follows $x \preccurlyeq a \preccurlyeq b$, and $\forall w$: $(i, x, j) > (i, b, w)$ yields $h(i, x, j) \leq (i, x, j)$.

2. $\preceq \subseteq \preceq_T$:
Take $a, b \in \mathcal{C}, a \not\preccurlyeq b$, and take k so that

$$a \not\leq_{\pi_k} b \Leftrightarrow b \leq_{\pi_k} a.$$

If $a \preccurlyeq_T b$ then there is an increasing function $h\colon a_T \to b_T$ such that $\forall w\colon h(w) \leq w$. Observe that $f(a) * l_k$ elements represented as (k, a, u) do not belong to b_T, from which we conclude:

$$\begin{aligned}
f(a) * l_k \leq & \left|\{(i,x,j) \in b_T \mid (i,x,j) < (k,a,f(a) * l_k)\}\right| \\
= & \left|\{(i,x,j) \in b_T \mid (i,x,j) < (k,a,1)\}\right| \\
\leq & \left|\{(k,x,j) \in b_T \mid (k,x,j) < (k,a,1)\}\right| \\
& + \left|\{(i,x,j) \in b_T \mid i <= k-1\}\right| \\
\leq & \left|\{(k,x,j) \in b_T \mid a \leq_{\pi_k} x \leq_{\pi_k} b\}\right| + \left|\{(i,x,j) \in \mathcal{I} \mid i <= k-1\}\right| \\
= & \; 0 + F * (l_1 + \cdots + l_{k-1}) = l_k - 1,
\end{aligned}$$

a contradiction.

■

Corollary 2 *Every subclock-closed relational structure compliant with* $\mathbf{A}_F$ *can be realized by clarified subclock-closed time system*

Combining Corollaries 1, 2 and Proposition 3 we obtain

Theorem 3 *Axiom system* $\mathbf{A}_0$ *is complete and sound axiom system with time systems as its models.*

As a spin-off, let us notice that time system constructed in Theorem **??** is a run and is finite. These properties are also preserved by extension in Theorem 1 and by factorization in Lemma 1. Thus, we have the following propositions.

Statement 1 *Axiom system* $\mathbf{A}_0$ *is complete and sound axiom system with runs as its models.*

Statement 2 *Axiom system* $\mathbf{A}_0$ *has finite model property.*

6 Conclusion and future work

We had constructed sound and complete axiomatics for propositional logic over RCCSL, with completeness being the nontrivial part of this construction. We hope that the proposed model-theoretical approach would help to define canonical clock constraints that would be essential from theoretical perspective. As

an example, this paper shows that when arguing about clock constrains, coincidence could be easily removed from consideration, which is obvious. What is not so obvious is that arguing about exclusion could be replaced by arguing about subclocking.

Natural question that arises from this perspective is to extend our result to wider fragments of logic over CCSL, we formulate it as a series of problems.

Problem 1 *What is the complete and sound system of axioms for propositional logic over CCSL.*

Problem 2 *What is the complete and sound system of axioms for propositional logic over CCSL with clock compositions.*

Problem 3 *What is the complete and sound system of axioms for propositional logic over CCSL with delays.*

Our preliminary research shows that axiom system for complete CCSL might be much more complicated than the one for RCCSL. On the other hand, augmenting CCSL with composition might quite naturally fall into our approach of extending time systems.

References

1. Dhaussy, P., Menad, M.: A transformation approach for multiform time requirements. Software Engineering and Formal Methods, volume 8137 of Lecture Notes in Computer Science, 1630, (2013) Commun. ACM, 21(7):558565, 1978.
2. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. Commun. ACM, 21(7):558565, 1978.
3. Mallet, F.: Logical Time @ Work for the Modeling and Analysis of Embedded Systems, Habilitation thesis. LAMBERT Academic Publishing (2011).
4. Mallet, F., Millo J.-V., Romenska, Y.: State-based representation of CCSL operators. Research Report RR-8334, INRIA (2013).
5. Mallet, F., Petterson, P., Seceleanu, C., Suryadevara, J.: Verifying MARTE/CCSL mode behaviors using UPPAAL. Software Engineering and Formal Methods, volume 8137 of Lecture Notes in Computer Science, 115, (2013)
6. Mallet, F.,Zaretska, I., Zholtkevych, G., Zholtkevych, G.: Clocks Model for Specification and Analysis of Timing in Real-Time Embedded Systems. ICTERI:475-489 (2013)