Completeness and Decidability of a Fragment of Duration Calculus with Iteration

Dang Van Hung* and Dimitar P. Guelev**

International Institute for Software Technology
The United Nations University, P.O.Box 3058, Macau
email: {dvh, dg}@iist.unu.edu

Abstract. Duration Calculus with Iteration (DC*) has been used as an interface between original Duration Calculus and Timed Automata, but has not been studied rigorously. In this paper, we study a subset of DC* formulas consisting of so-called simple ones which corresponds precisely with the class of Timed Automata. We give a complete proof system and the decidability results for the subset.
Keywords: Real-Time system, formal methods, Duration Calculus, completeness, decidability.

1 Introduction

Duration Calculus (DC) was introduced by Zhou, Hoare and Ravn in 1991 as a logic to specify the requirements for real-time systems. DC has been used successfully in many case studies, see e.g. [ZZ94,YWZP94,HZ94,DW94,BHCZ94,XH95], [Dan98,ED99]. In [DW94], we have developed a method for designing a real-time hybrid system from its specification in DC. In that paper, we introduced a class of so-called simple Duration Calculus formulas with iterations which corresponds precisely with the class of real-time automata to express the design of real-time hybrid systems, and show how to derive a design in this language from a specification in the original Duration Calculus. We use the definition of semantic of our design language to reason about the correctness of our design. However, it would be more practical and interesting if the correctness of a design can be proved syntactically with a tool. Therefore, developing a proof system to assist the formal verification of the design plays an important role in making the use of formal methods for the designing process of real-time systems. This is our aim in this paper.

We achieve our aim in the following way. First we extend DC with the iteration operator (*) to obtain a logic called DC*, and define a subclass of DC* formulas called simple DC* formulas to express the designs. Secondly we develop a complete proof system for the proof of the fact that a simple DC* formula D

* On leave from the Institute of Information Technology, Hanoi, Vietnam.
** On leave from the Department of Mathematical Logic and Its Applications, Faculty of Mathematics and Informatics, Sofia University "St. Kliment Ochridski"

implies a DC formula S, meaning that any implication of this form can be proved in our proof system.

To illustrate our idea, let us consider a classical simple example Gas Burner taken from [ZHR91]. The time critical requirements of a gas burner is specified by a DC formula denoted by S, defined as $\Box(\ell > 60s \Rightarrow (20 * \int leak \leq \ell))$ which says that during the operation of the system, if the interval over which the system is observed is at least 1 min, the proportion of time spent in the leak state is not more than one-twentieth of the elapsed time.

One can design the Gas Burner as a real-time automaton depicted in Fig. 1 which expresses that any leak state must be detected and stopped within one second, and that leak must be separated by at least 30s. A natural way to express the behaviour of the automaton is to use a classical regular expression like notation

$$D \;\widehat{=}\; ((\lceil leak \rceil \wedge \ell \leq 1)^\frown(\lceil nonleak \rceil \wedge \ell \geq 30))^*$$

Here we assume that the gas burner starts from the leak state. We will see later that D is a DC formula with iteration. It expresses not only the temporal order of states but also the time constraints on the state periods.

By using our complete proof system we can show formally the implication $D \Rightarrow S$ which expresses naturally the correctness of the design.

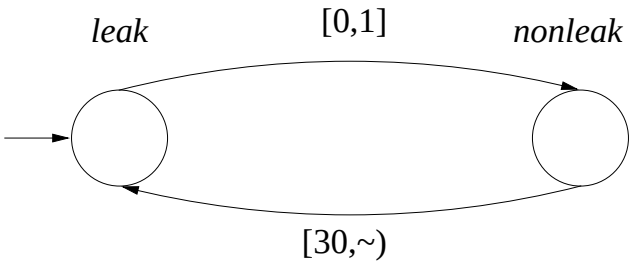

Fig. 1. Simple Design of Gas Burner

The class of simple DC* formulas has an interesting property that it is decidable, which means that we can decide if a design is implementable. Furthermore, for some class of DC formulas such as linear duration invariants (see [ZZY94,LDZ97,DP97]), the implication from a simple DC* formula to a formula in the class can be checked by a simple algorithm.

The paper is organised as follows. In the next section, we give the syntax and semantics of our Duration Calculus with Iteration. In the third section we will give a proof system for the calculus. We prove the completeness of our proof system for the class of simple DC* formulas in Section 4. The decidability of the class will be discussed in the last section.

2 Duration Calculus with Iteration

This section presents the formal definition of Duration Calculus with iteration, which is a conservative extension of Duration Calculus [ZHR91].

2.1 Language

A language for DC* is built starting from the following sets of *symbols*: a set of *constant symbols* $\{a, b, c, \ldots\}$, a set of *individual variables* $\{x, y, z, \ldots\}$, a set of *state variables* $\{P, Q, \ldots\}$, a set of *temporal variables* $\{u, v, \ldots\}$, a set of *function symbols* $\{f, g, \ldots\}$, a set of *relation symbols* $\{R, U, \ldots\}$, and a set of *propositional temporal letters* $\{A, B, \ldots\}$. These sets are required to be pairwise disjoint and disjoint with the set $\{\mathbf{0}, \bot, \neg, \vee, \frown, ^*, \exists, \int, (,)\}$. Besides, 0 should be one of the constant symbols; $+$ should be a binary function symbol; $=$ and $\leq$ should be binary relation symbols.

Given the sets of symbols, a DC* language definition is essentially that of the sets of *state expressions* S, *terms* t and *formulas* φ of the language. These sets can be defined by the following BNFs:

$$
\begin{array}{l}
S \mathrel{\widehat{=}} \mathbf{0} \mid P \mid \neg S \mid S \vee S \\
t \mathrel{\widehat{=}} c \mid x \mid u \mid \int S \mid f(t, \ldots, t) \\
\varphi \mathrel{\widehat{=}} A \mid R(t, \ldots, t) \mid \neg\varphi \mid (\varphi \vee \varphi) \mid (\varphi \frown \varphi) \mid (\varphi^*) \mid \exists x \varphi
\end{array}
$$

Terms and formulas that have no occurrences of $\frown$ (*chop*), nor of temporal variables, or $\int$, are called *rigid*.

2.2 Semantics

The linearly ordered field of the *real numbers*,

$$\langle \mathbf{R}, =_{\mathbf{R}}, 0_{\mathbf{R}}, 1_{\mathbf{R}}, +_{\mathbf{R}}, -_{\mathbf{R}}, \times_{\mathbf{R}}, /_{\mathbf{R}}, \leq_{\mathbf{R}} \rangle ,$$

is the most important component of DC semantics.

In this section, we denote by $\mathbf{I}$ the set of the bounded intervals over $\mathbf{R}$, $\{[\tau_1, \tau_2] \mid \tau_1, \tau_2 \in \mathbf{R}, \tau_1 \leq_{\mathbf{R}} \tau_2\}$. For a set $A \subseteq \mathbf{R}$, we denote by $\mathbf{I}(A)$ the set $\{[\tau_1, \tau_2] \in \mathbf{I} \mid \tau_1, \tau_2 \in A\}$ of intervals with end-points in A.

Given a DC* language $\mathcal{L}$, a model for $\mathcal{L}$ is an *interpretation* $\mathcal{I}$ of the symbols of $\mathcal{L}$ that satisfies the following conditions: $\mathcal{I}(c), \mathcal{I}(x) \in \mathbf{R}$ for constant symbols c and individual variables x; $\mathcal{I}(f) : \mathbf{R}^n \to \mathbf{R}$ for n-place function symbols f; $\mathcal{I}(v) : \mathbf{I} \to \mathbf{R}$ for temporal variables v; $\mathcal{I}(R) : \mathbf{R}^n \to \{0, 1\}$ for n-place relation symbols R; $\mathcal{I}(P) : \mathbf{R} \to \{0, 1\}$ for state variable P, and $\mathcal{I}(A) : \mathbf{I} \to \{0, 1\}$ for temporal propositional letters A. Besides, $\mathcal{I}(0) = 0_{\mathbf{R}}$, $\mathcal{I}(+) = +_{\mathbf{R}}$, $\mathcal{I}(=)$ is $=_{\mathbf{R}}$, and $\mathcal{I}(\leq)$ is $\leq_{\mathbf{R}}$. The following condition, known as *finite variability of state*, is imposed on interpretations:
For every $[\tau_1, \tau_2] \in \mathbf{I}$ *such that* $\tau_1 < \tau_2$*, and every state variable* S *there exist* $\tau'_1, \ldots, \tau'_n \in \mathbf{R}$ *such that* $\tau_1 = \tau'_1 < \ldots < \tau'_n = \tau_2$ *and* $\mathcal{I}(S)$ *is constant on the intervals* (τ'_i, τ'_{i+1})*,* $i = 1, \ldots, n-1$.

For the rest of the paper we omit the index $\cdot_{\mathbf{R}}$, that distinguishes operations on reals from the corresponding symbols.

Definition 1. *Given a DC interpretation $\mathcal{I}$ for the DC* language $\mathcal{L}$, the meaning of state expressions S in $\mathcal{L}$ under $\mathcal{I}$, $S_{\mathcal{I}} : \boldsymbol{R} \to \{0,1\}$, is defined inductively as follows: for all $\tau \in \boldsymbol{R}$*

$$
\begin{array}{ll}
\mathbf{0}_{\mathcal{I}}(\tau) \mathrel{\widehat{=}} 0 & \\
P_{\mathcal{I}}(\tau) \mathrel{\widehat{=}} \mathcal{I}(P)(\tau) & \textit{for state variables } P \\
(\neg S)_{\mathcal{I}}(\tau) \mathrel{\widehat{=}} 1 - S_{\mathcal{I}}(\tau) & \\
(S_1 \vee S_2)_{\mathcal{I}}(\tau) \mathrel{\widehat{=}} \max((S_1)_{\mathcal{I}}(\tau), (S_2)_{\mathcal{I}}(\tau)) &
\end{array}
$$

Given an interval $[\tau_1, \tau_2] \in \mathbf{I}$, the meaning of a term t in $\mathcal{L}$ under $\mathcal{I}$ is a number $\mathcal{I}^{\tau_2}_{\tau_1}(t) \in \mathbf{R}$ defined inductively as follows:

$$
\begin{array}{ll}
\mathcal{I}^{\tau_2}_{\tau_1}(c) \mathrel{\widehat{=}} \mathcal{I}(c) & \textit{for constant symbols } c, \\
\mathcal{I}^{\tau_2}_{\tau_1}(x) \mathrel{\widehat{=}} \mathcal{I}(x) & \textit{for individual variables } x, \\
\mathcal{I}^{\tau_2}_{\tau_1}(v) \mathrel{\widehat{=}} \mathcal{I}(v)([\tau_1, \tau_2]) & \textit{for temporal variables } v, \\
\mathcal{I}^{\tau_2}_{\tau_1}(\int S) \mathrel{\widehat{=}} \int_{\tau_1}^{\tau_2} S_{\mathcal{I}}(\tau) d\tau & \textit{for state expressions } S, \\
\mathcal{I}^{\tau_2}_{\tau_1}(f(t_1, \ldots, t_n)) \mathrel{\widehat{=}} \mathcal{I}(f)(\mathcal{I}^{\tau_2}_{\tau_1}(t_1), \ldots, \mathcal{I}^{\tau_2}_{\tau_1}(t_n)) & \textit{for } n\textit{-place function} \\
 & \textit{symbols } f.
\end{array}
$$

The definitions given so far are relevant to the semantics of DC in general. The extension to the semantics that comes with DC* appears in the definition of the $\models$ relation below. Let us recall the following tradition relation on interpretations.

Definition 2. *Let $\mathcal{I}$, $\mathcal{J}$ be interpretations of the symbols of the same DC* language $\mathcal{L}$. Let x be a symbol in $\mathcal{L}$. The interpretation $\mathcal{I}$ x-*agrees *with the interpretation $\mathcal{J}$ iff $\mathcal{I}(s) = \mathcal{J}(s)$ for all symbols s in $\mathcal{L}$, but possibly x.*

Definition 3. *Given a DC* language $\mathcal{L}$, and an interpretation $\mathcal{I}$ of the symbols of $\mathcal{L}$. The relation $\mathcal{I}, [\tau_1, \tau_2] \models \varphi$ for $[\tau_1, \tau_2] \in \mathbf{I}$ and formulas φ in $\mathcal{L}$ is defined by induction on the construction of φ as follows:*

$$
\begin{array}{ll}
\mathcal{I}, [\tau_1, \tau_2] \not\models \bot & \\
\mathcal{I}, [\tau_1, \tau_2] \models A & \textit{iff } \mathcal{I}(A)([\tau_1, \tau_2]) = 1 \textit{ for temporal} \\
 & \textit{propositional letters } A \\
\mathcal{I}, [\tau_1, \tau_2] \models R(\tau_1, \ldots, \tau_n) & \textit{iff } \mathcal{I}(R)(\mathcal{I}^{\tau_2}_{\tau_1}(t_1), \ldots, \mathcal{I}^{\tau_2}_{\tau_1}(t_n)) = 1 \\
\mathcal{I}, [\tau_1, \tau_2] \models \neg\varphi & \textit{iff } \mathcal{I}, [\tau_1, \tau_2] \not\models \varphi \\
\mathcal{I}, [\tau_1, \tau_2] \models (\varphi \vee \psi) & \textit{iff either } \mathcal{I}, [\tau_1, \tau_2] \models \varphi \textit{ or } \mathcal{I}, [\tau_1, \tau_2] \models \psi \\
\mathcal{I}, [\tau_1, \tau_2] \models (\varphi^\frown \psi) & \textit{iff } \mathcal{I}, [\tau_1, \tau] \models \varphi \textit{ and } \mathcal{I}, [\tau, \tau_2] \models \psi \\
 & \textit{for some } \tau \in [\tau_1, \tau_2] \\
\mathcal{I}, [\tau_1, \tau_2] \models (\varphi^*) & \textit{iff either } \tau_1 = \tau_2, \textit{ or there exist } \tau'_1, \ldots, \tau'_n \in \mathbf{R} \\
 & \textit{such that } \tau_1 = \tau'_1 < \ldots < \tau'_n = \tau_2 \textit{ and} \\
 & \mathcal{I}, [\tau'_i, \tau'_{i+1}] \models \varphi \textit{ for } i = 1, \ldots, n-1 \\
\mathcal{I}, [\tau_1, \tau_2] \models \exists x \varphi & \textit{iff } \mathcal{J}, [\tau_1, \tau_2] \models \varphi \textit{ for some } \mathcal{J} \textit{ that} \\
 & x\textit{-agrees with } \mathcal{I}
\end{array}
$$

Note that only the modelling relation $\models$, and not the interpretations $\mathcal{I}$, makes the difference between DC* and DC. Besides, the clauses that define the interpretation of constructs other than * in DC* are the same as in DC. This entails that DC* is a *conservative* extension of DC.

For convenience, we introduce the following notations. Let $\mathcal{I}$ be a DC interpretation, φ be a DC* formula, and $\mathbf{J}_1, \mathbf{J}_2, \mathbf{J} \subseteq \mathbf{I}$ be sets of intervals. Let $k < \omega$. We define

$$
\begin{array}{ll}
\tilde{\mathcal{I}}(\varphi) \mathrel{\widehat{=}} \{[\tau_1, \tau_2] \in \mathbf{I} \mid \mathcal{I}, [\tau_1, \tau_2] \models \varphi\} & \\
\mathbf{J}_1 \frown \mathbf{J}_2 \mathrel{\widehat{=}} \{[\tau_1, \tau_2] \in \mathbf{I} \mid (\exists \tau \in \mathbf{R})([\tau_1, \tau] \in \mathbf{J}_1 \wedge [\tau, \tau_2] \in \mathbf{J}_2)\} & \\
\mathbf{J}^0 \mathrel{\widehat{=}} \{[\tau, \tau] \mid \tau \in \mathbf{R}\} & \\
\mathbf{J}^k \mathrel{\widehat{=}} \underbrace{\mathbf{J} \frown \ldots \frown \mathbf{J}}_{k \text{ times}} & \text{for } k > 0 \\
\mathbf{J}^* \mathrel{\widehat{=}} \bigcup_{k<\omega} \mathbf{J}^k &
\end{array}
$$

In words, $\tilde{\mathcal{I}}(\varphi)$ is the set of intervals that satisfy φ under $\mathcal{I}$, $\mathbf{J}_1 \frown \mathbf{J}_2$ is the set of intervals that are the concatenation of an interval in $\mathbf{J}_1$ and an interval in $\mathbf{J}_2$, and $\mathbf{J}^*$ is the iteration of $\mathbf{J}$ corresponding to the operation $\frown$.

The language definition in Section 2.1 introduces a minimal set of DC* syntactic elements just in order to enable the concise definition of DC* semantics in Section 2.2. In fact, a richer set is employed in the rest of the paper for its providing convenience of reading. We use the customary *infix* notation for terms with $+$, and formulas with $\leq$ and $=$ occurring in them. We introduce the constant $\top$, the boolean connectives $\wedge$, $\Rightarrow$ and $\Leftrightarrow$, the relation symbols $\neq$, $\geq$, $<$ and $>$, and the $\forall$ quantifier as abbreviations in the usual way. We assume that boolean connectives bind more tightly than $\frown$. Since $\frown$ is associative, we omit parentheses in formulas that contain consecutive occurrence of $\frown$. Besides, we use the following abbreviations, that are generally accepted in Duration Calculus:

$$
\begin{array}{ll}
\mathbf{1} \mathrel{\widehat{=}} \neg \mathbf{0} & \\
\ell \mathrel{\widehat{=}} \int \mathbf{1} & \\
[\![S]\!] \mathrel{\widehat{=}} \int S = \ell \wedge \ell \neq 0 & \\
\Diamond \varphi \mathrel{\widehat{=}} \top \frown \varphi \frown \top & \\
\Box \varphi \mathrel{\widehat{=}} \neg \Diamond \neg \varphi & \\
(\varphi^+) \mathrel{\widehat{=}} \varphi \frown (\varphi^*) & \\
\varphi^0 \mathrel{\widehat{=}} \ell = 0 & \\
\varphi^k \mathrel{\widehat{=}} \underbrace{\varphi \frown \ldots \frown \varphi}_{k \text{ times}} & \text{for } k > 0
\end{array}
$$

3 A Proof System for DC*

In this section, we propose a proof system for DC* which consists of a complete Hilbert-style proof system for first order logic (cf. e.g. [Sho67]), axioms and rules for interval logic (cf. e.g. [Dut95]), Duration Calculus axioms and rules ([HZ92]) and axioms about iteration ([Gue98b]). We assume that the readers are familiar with Hilbert-style proof systems for first order logic and do not give one here. Here follow the interval logic and DC-specific axioms and rules.

Axioms and rules for Interval Logic

$(A1_l)$ $(\varphi^\frown\psi) \wedge \neg(\chi^\frown\psi) \Rightarrow (\varphi \wedge \neg\chi^\frown\psi)$
$(A1_r)$ $(\varphi^\frown\psi) \wedge \neg(\varphi^\frown\chi) \Rightarrow (\varphi^\frown\psi \wedge \neg\chi)$
$(A2)$ $((\varphi^\frown\psi)^\frown\chi) \Leftrightarrow (\varphi^\frown(\psi^\frown\chi))$
(R_l) $(\varphi^\frown\psi) \Rightarrow \varphi$ if φ is rigid
(R_r) $(\varphi^\frown\psi) \Rightarrow \psi$ if ψ is rigid
(B_l) $(\exists x\varphi^\frown\psi) \Rightarrow \exists x(\varphi^\frown\psi)$ if x is not free in ψ
(B_r) $(\varphi^\frown\exists x\psi) \Rightarrow \exists x(\varphi^\frown\psi)$ if x is not free in φ
$(L1_l)$ $(\ell = x^\frown\varphi) \Rightarrow \neg(\ell = x^\frown\neg\varphi)$
$(L1_r)$ $(\varphi^\frown\ell = x) \Rightarrow \neg(\neg\varphi^\frown\ell = x)$
$(L2)$ $\ell = x + y \Leftrightarrow (\ell = x^\frown\ell = y)$
$(L3_l)$ $\varphi \Rightarrow (\ell = 0^\frown\varphi)$
$(L3_r)$ $\varphi \Rightarrow (\varphi^\frown\ell = 0)$

$$(N_l) \quad \frac{\varphi}{\neg(\neg\varphi^\frown\psi)}$$

$$(N_r) \quad \frac{\varphi}{\neg(\psi^\frown\neg\varphi)}$$

$$(Mono_l) \quad \frac{\varphi \Rightarrow \psi}{(\varphi^\frown\chi) \Rightarrow (\psi^\frown\chi)}$$

$$(Mono_r) \quad \frac{\varphi \Rightarrow \psi}{(\chi^\frown\varphi) \Rightarrow (\chi^\frown\psi)}$$

Duration Calculus axioms and rules

$(DC1)$ $\int \mathbf{0} = 0$
$(DC2)$ $\int \mathbf{1} = \ell$
$(DC3)$ $\int S \geq 0$
$(DC4)$ $\int S_1 + \int S_2 = \int(S_1 \vee S_2) + \int(S_1 \wedge S_2)$
$(DC5)$ $(\int S = x^\frown\int S = y) \Rightarrow \int S = x + y$
$(DC6)$ $\int S_1 = \int S_2$ if $S_1 \Leftrightarrow S_2$ in propositional calculus.

$$(IR_1) \quad \frac{[\ell = 0/A]\varphi \;\; \varphi \Rightarrow [A^\frown\lceil S\rceil/A]\varphi \;\; \varphi \Rightarrow [A^\frown\lceil\neg S\rceil/A]\varphi}{[\top/A]\varphi}$$

$$(IR_2) \quad \frac{[\ell = 0/A]\varphi \;\; \varphi \Rightarrow [\lceil S\rceil^\frown A/A]\varphi \;\; \varphi \Rightarrow [\lceil\neg S\rceil^\frown A/A]\varphi}{[\top/A]\varphi}$$

$$(\omega) \quad \frac{\forall k < \omega \;\; [(\lceil S\rceil \vee \lceil\neg S\rceil)^k/A]\varphi}{[\top/A]\varphi}$$

Axioms about iteration

(DC_1^*) $\ell = 0 \Rightarrow \varphi^*$
(DC_2^*) $(\varphi^* \frown \varphi) \Rightarrow \varphi^*$
(DC_3^*) $(\varphi^* \wedge \psi \frown \top) \Rightarrow (\psi \wedge \ell = 0 \frown \top) \vee (((\varphi^* \wedge \neg\psi \frown \varphi) \wedge \psi) \frown \top)$.

The meaning of DC_1^* and DC_2^* is quite straightforward. As for DC_3^*, it has the following meaning: Assume that some initial subinterval of a given interval satisfies ψ, and can be chopped into finitely many parts, each satisfying φ. Then the smallest among the initial subintervals of the given one formed by these parts makes ψ hold exists which is either the 0-length initial subinterval, or otherwise consists one that does not satisfy ψ.

A restriction is made on the application of first order logic rules and axioms that involve substitution: $[t/x]\varphi$ is defined if no variable in t becomes bound due to the substitution, and either t is rigid or $\frown$ does not occur in φ.

It is known that the above proof system for interval logic is complete with respect to an abstract class of time domains in place of $\mathbf{R}$ [Dut95]. The proof system for interval logic, extended with the axioms DC_1-DC_6 and the rules IR_1, IR_2 is complete relative to the class of interval logic sentences that are valid on its real time frame [HZ92]. Taking the infinitary rule ω instead of IR_1 and IR_2 yields an ω-complete system for DC with respect to an abstract class of time domains, like that of interval logic [Gue98a]. Adding appropriate axioms about reals, and a rule like, e.g.,

$$\frac{\forall k < \omega \ \overline{k}x \leq 1}{x \leq 0}$$

where $\overline{k}$ stands for $\underbrace{1 + \ldots + 1}_{k \text{ times}}$, extends this system to one that is ω-complete with respect to the real time based semantics of DC given above.

In the rest of this section we show that adding DC_1^*-DC_3^* to the proof system of DC makes it complete for sentences where iteration is allowed only for a restricted class of formulas that we call *simple*. The following theorem gives the soundness of these axioms.

Theorem 1. *Let $\mathcal{I}$ be a Duration Calculus interpretation. Then $\mathcal{I}$ validates $DC_1^* - DC_3^*$.*

Proof. The proof about DC_1^* and DC_2^* is trivial and we omit it here. Now consider DC_3^*. Let $[\tau_1, \tau_2] \in \mathbf{I}$ be such that $\mathcal{I}, [\tau_1, \tau_2] \models \varphi^* \wedge \psi \frown \top$, and $\mathcal{I}, [\tau_1, \tau_2] \models \neg(\psi \wedge \ell = 0 \frown \top)$. We shall prove that $\mathcal{I}, [\tau_1, \tau_2] \models ((\varphi^* \wedge \neg\psi \frown \varphi) \wedge \psi) \frown \top$. We have that $[\tau_1, \tau_1] \notin \tilde{\mathcal{I}}(\psi)$, and $[\tau_1, \tau] \in \left(\tilde{\mathcal{I}}(\varphi)\right)^k \cap \tilde{\mathcal{I}}(\psi)$ for some $k < \omega$, and some $\tau \in [\tau_1, \tau_2]$. Then there exist $\tau_1', \ldots, \tau_{k+1}'$ such that $\tau_1 = \tau_1' < \ldots < \tau_{k+1}' = \tau$ and $\mathcal{I}, [\tau_i', \tau_{i+1}'] \models \varphi$ for $i = 1, \ldots, k$. Since $[\tau_1, \tau_{k+1}'] \models \psi$ and $[\tau_1, \tau_1'] \not\models \psi$ there must be $i \leq k$ for which $[\tau_1, \tau_i'] \not\models \psi$ and $[\tau_1, \tau_{i+1}'] \models \psi$. Therefore $\mathcal{I}, [\tau_1, \tau_{i+1}'] \models (\varphi^* \wedge \neg\psi \frown \varphi) \wedge \psi$, which implies that $\mathcal{I}, [\tau_1, \tau_2] \models ((\varphi^* \wedge \neg\psi \frown \varphi) \wedge \psi) \frown \top$.

Note that from the proof of the above theorem we can see that the scope of the soundness of $DC_1^* - DC_3^*$ is, in fact, interval logic. Let us prove the monotonicity of * from these axioms.

Let $\phi \Rightarrow \gamma$. We prove that $\phi^* \Rightarrow \gamma^*$.

$$\begin{array}{lll} \phi^* \wedge \neg\gamma^* \Rightarrow (\neg\gamma^* \wedge \ell = 0^\frown\top) \vee (((\phi^* \wedge \gamma^*{}^\frown\phi) \wedge \neg\gamma^*)^\frown\top) & \text{by } DC_3^* \\ \Rightarrow (((\phi^* \wedge \gamma^*{}^\frown\phi) \wedge \neg\gamma^*)^\frown\top) & \text{by } DC_1^* \\ \Rightarrow (\neg\gamma^* \wedge \gamma^*)^\frown\top & \text{by } DC_2^* \\ & \text{and } \phi \Rightarrow \gamma \\ \Rightarrow \bot & \end{array}$$

The following theorem, a proof of which is given in the Appendix, is useful in practice.

Theorem 2.

$$\vdash_{DC^*} \Box(\varphi \Rightarrow \neg(\top^\frown\neg\alpha) \wedge \neg(\neg\beta^\frown\top)) \wedge \Box(\ell = 0 \Rightarrow \alpha \wedge \beta) \Rightarrow \varphi^* \Rightarrow \Box(\alpha^\frown\varphi^*{}^\frown\beta)\,.$$

As an example for the use of the proof system, let us prove the implication for the correctness of the simple Gas-Burner mentioned in the introduction of the paper. We have to prove that

$$((\lceil leak \rceil \wedge \ell \leq 1)^\frown(\lceil nonleak \rceil \wedge \ell \geq 30))^* \Rightarrow \Box(\ell \geq 60 \Rightarrow \textstyle\int leak \leq (1/20)\ell)\,.$$

Let us denote

$$\begin{array}{rcl} \varphi & \widehat{=} & \lceil leak \rceil \wedge \ell \leq 1^\frown\lceil \neg leak \rceil \wedge \ell \geq 30\,, \\ \alpha & \widehat{=} & \ell = 0 \vee \lceil \neg leak \rceil \vee (\lceil leak \rceil \wedge \ell \leq 1^\frown\lceil \neg leak \rceil \wedge \ell \geq 30)\,, \\ \beta & \widehat{=} & \ell = 0 \vee (\ell \leq 1 \wedge \lceil leak \rceil^\frown\ell = 0 \vee \lceil \neg leak \rceil)\,. \end{array}$$

From DC axioms it can be proved easily that $\vdash_{DC} \Box(\varphi \Rightarrow \neg(\top^\frown\neg\alpha) \wedge \neg(\neg\beta^\frown\top))$ and $\vdash_{DC} \Box(\ell = 0 \Rightarrow \alpha \wedge \beta)$. Therefore, from Theorem 2 we can complete the proof of the above if we can derive that $((\alpha^\frown\varphi^*)^\frown\beta) \Rightarrow 20\int leak \leq \ell$. This is done as follows.

1	$\alpha \Rightarrow 31\int leak \leq \ell$	DC
2	$\varphi^* \wedge 31\int leak > \ell \Rightarrow (\varphi^* \wedge 31\int leak > \ell^\frown\top)$	DC
3	$\varphi \Rightarrow 31\int leak \leq \ell$	DC
4	$(\varphi^* \wedge 31\int leak > \ell^\frown\top) \Rightarrow$ $(\ell = 0 \wedge 31\int leak > \ell^\frown\top) \vee$ $(((\varphi^* \wedge 31\int leak \leq \ell^\frown\varphi) \wedge 31\int leak > \ell)^\frown\top)$	by DC_3^*
5	$\ell = 0 \Rightarrow 31\int leak \leq \ell$	DC
6	$(\varphi^* \wedge 31\int leak > \ell^\frown\top) \Rightarrow$ $(((\varphi^* \wedge 31\int leak \leq \ell^\frown\varphi) \wedge 31\int leak > \ell)^\frown\top)$	by 4, 5, $Mono_r$
7	$(\varphi^* \wedge 31\int leak \leq \ell^\frown\varphi) \Rightarrow 31\int leak \leq \ell$	by 2, 3, DC
8	$\varphi^* \Rightarrow 31\int leak \leq \ell$	by 6, 7, $Mono_r$
9	$(\alpha^\frown\varphi^*) \Rightarrow 31\int leak \leq \ell$	by 1, 8, DC
10	$\beta \Rightarrow \int leak \leq 1$	DC
11	$((\alpha^\frown\varphi^*)^\frown\beta) \wedge \ell \geq 60 \Rightarrow 20\int leak \leq \ell$	by 9, 10, DC, arithmetic

4 Completeness of the DC* proof system for iteration of simple formulas

As said in the introduction to the paper, our purpose is to give a rigorous study of a class of DC* formulas that play an important roles in practice. The formulas in the class are called simple formulas and will be considered to be executable. In this section we extend the class of simple formulas, originally introduced in [DW94], so that conjunction is freely allowed in simple formulas. We give a rigorous proof of the completeness of the axiom system from the previous section for this class of formulas.

Definition 4. Simple *DC* formulas are defined by the following BNF:*

$$\varphi \;\widehat{=}\; \lceil S \rceil \mid a \leq \ell \mid \ell \leq a \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi^\frown\varphi) \mid \varphi^*$$

Before giving our main result on the completeness, we first explain where from and how we obtained our axioms $DC_1^* - DC_3^*$ about DC* iteration (Subsection 4.1). Then we show that given simple formula φ and DC* formula γ, DC interpretations $\mathcal{I}$ that validate

$$\begin{array}{l}(DC^*_{1,\varphi,\gamma})\ \ell = 0 \Rightarrow \gamma\\ (DC^*_{2,\varphi,\gamma})\ (\gamma^\frown\varphi) \Rightarrow \gamma\\ (DC^*_{3,\varphi,\gamma})\ (\gamma \wedge \psi^\frown\top) \Rightarrow (\psi \wedge \ell = 0^\frown\top) \vee (((\gamma \wedge \neg\psi^\frown\varphi) \wedge \psi)^\frown\top)\end{array}$$

for all DC* formulas ψ should satisfy the equality $\left(\tilde{\mathcal{I}}(\varphi)\right)^* = \tilde{\mathcal{I}}(\gamma)$. This means that the axioms $DC_1^* - DC_3^*$ enforce the clause about iteration in the DC* definition of $\models$ (Definition 3) for simple formulas φ. We do this in the following way: Given the assumption that $\left(\tilde{\mathcal{I}}(\varphi)\right)^* \neq \tilde{\mathcal{I}}(\gamma)$, we find an interval $[\tau_1, \tau_2]$ and a formula ψ that refute some of $DC^*_{1,\varphi,\gamma} - DC^*_{3,\varphi,\gamma}$ under $\mathcal{I}$. Having found an appropriate interval $[\tau_1, \tau_2]$, the formula ψ we need is a *-free one that satisfies $\tilde{\mathcal{I}}(\neg\psi) \cap \mathbf{I}([\tau_1, \tau_2]) = \left(\tilde{\mathcal{I}}(\varphi)\right)^* \cap \mathbf{I}([\tau_1, \tau_2])$.

4.1 From the propositional dynamic logic to DC*

In this subsection we point to a certain degree of semantical compatibility between interval logic frames and propositional dynamic logic (PDL) frames. We give a truth-preserving translation of PDL formulas into interval logic ones, that is based on this semantic correspondence. We apply this translation to obtain our axioms for iteration from the corresponding axioms in PDL. The readers who are not familiar with PDL can skip this section. Basic definitions about PDL can be found in, e.g., [AGM92].

Let $F = \langle\langle T, \leq\rangle, \langle D, +, 0\rangle, m\rangle$ be an interval logic frame with time domain $\langle T, \leq\rangle$, duration domain $\langle D, +, 0\rangle$, and measure function $m : \mathbf{I}(T) \to D$, where $\mathcal{I}(T) = \{[\tau_1, \tau_2] \mid \tau_1, \tau_2, \in T, \tau_1 \leq \tau_2\}$ (cf. [Dut95] for interval logic terminology). The set of time points T can be taken as the set of possible worlds of a PDL

frame. Let v be a valuation of the propositional letters $p \in P$ and the relation letters $r \in R$ of a PDL language $\mathcal{L}_{PDL}$ into such a frame, i.e. v assigns a set of possible worlds to a propositional letter, and a set of pairs of possible worlds to a relational letter. Let $v(r) \subseteq \leq$ for every relation letter r in this language. Since $Id_T \subseteq \leq$, and $R, S \subseteq \leq$ implies $R \cup S, R \circ S, R^* \subseteq \leq$, we can assume that the standard extension $\tilde{v}$ of v over relation terms gives only subrelations of $\leq$, too.

Assume that T has a least and a greatest time point, i.e. $T = [\min T, \max T]$. Let us build a language for interval logic with iteration $\mathcal{L}_{IL^*}$ with $P \cup R$ as its set of temporal propositional letters. Let us define an interpretation $\mathcal{I}$ of the temporal propositional letters of $\mathcal{L}_{IL^*}$ from $P \cup R$ on F as follows:

$\mathcal{I}(p)([\tau_1, \tau_2]) = 1$ iff $\tau_1 \in v(p)$ and $\tau_2 = \max T$ for $p \in P$;
$\mathcal{I}(r)([\tau_1, \tau_2]) = 1$ iff $\langle \tau_1, \tau_2 \rangle \in v(r)$ for $r \in R$.

Now consider the translation $\mathtt{t}$ of $\mathcal{L}_{PDL}$ into $\mathcal{L}_{IL}$ that is defined inductively by the following clauses:

$\mathtt{t}(\bot) \mathrel{\widehat{=}} \bot$
$\mathtt{t}(q) \mathrel{\widehat{=}} q$ for $q \in P \cup R$;
$\mathtt{t}(\varphi \vee \psi) \mathrel{\widehat{=}} \mathtt{t}(\varphi) \vee \mathtt{t}(\psi)$
$\mathtt{t}(\neg\varphi) \mathrel{\widehat{=}} \neg\mathtt{t}(\varphi)$
$\mathtt{t}(Id) \mathrel{\widehat{=}} l = 0$
$\mathtt{t}(\alpha \cup \beta) \mathrel{\widehat{=}} \mathtt{t}(\alpha) \vee \mathtt{t}(\beta)$
$\mathtt{t}(\alpha \circ \beta) \mathrel{\widehat{=}} \mathtt{t}((\alpha)^\frown\mathtt{t}(\beta))$
$\mathtt{t}(\alpha^*) \mathrel{\widehat{=}} (\mathtt{t}(\alpha))^*$
$\mathtt{t}(\langle\alpha\rangle\varphi) \mathrel{\widehat{=}} (\mathtt{t}(\alpha)^\frown\mathtt{t}(\varphi))$

The relationship between the PDL model based on T and v that we described, and the interval logic model $\langle F, \mathcal{I} \rangle$ can be expressed using $\mathtt{t}$ by the following proposition:

Proposition 1. *Let $\varphi \in \mathcal{L}_{PDL}$. Then $T, v, \min T \models \varphi$ iff $\langle F, \mathcal{I} \rangle, [\min T, \max T] \models \mathtt{t}(\varphi)$.*

Proof. Direct check by induction on the construction φ.

PDL has the following axioms for iteration in its proof system ([AGM92]):

$(*_1)$ $\quad [\alpha^*]\varphi \Rightarrow (\varphi \wedge [\alpha][\alpha^*]\varphi)$
$(*_2)$ $\quad [\alpha^*](\varphi \Rightarrow [\alpha]\varphi) \Rightarrow (\varphi \Rightarrow [\alpha^*]\varphi)$

The $\mathtt{t}$-translations of these axioms are equivalent to

$(\mathcal{I}_1)$ $\quad \psi \vee ((\alpha^\frown\alpha^*)^\frown\psi) \Rightarrow (\alpha^* {}^\frown\psi)$
$(\mathcal{I}_2)$ $\quad (\alpha^* {}^\frown(\alpha^* {}^\frown\psi) \wedge \neg\psi) \vee ((\alpha^* {}^\frown\psi) \Rightarrow \psi)$

where $\psi \mathrel{\widehat{=}} \neg\varphi$ for short.

The validity of $*_1$ for some given PDL relation term α for all possible values $\tilde{v}(\psi) \subseteq T$ for ψ enforces $(\tilde{v}(\alpha))^* \subseteq \tilde{v}(\alpha^*)$. The corresponding inclusion about

interval logic iteration can be enforced by two simpler axioms, namely DC_1^* and DC_2^* from our proof system. Similarly, the validity of $*_2$ enforces $(\tilde{v}(\alpha))^* \supseteq \tilde{v}(\alpha^*)$. The corresponding inclusion is enforced by DC_3^* in our system. Now notice that DC_3^* can be obtained from the translation $\mathcal{I}_2$ of $*_2$ by replacing subformulas with ψ of the kind $(\chi_1 \frown \psi \wedge \chi_2)$ by $(\chi_1 \wedge \psi \frown \chi_2)$, and some simple interval logic transformations. See [Gue98a] for details on the kind of convenience that this last transformation provides.

Note that, although $\mathcal{I}_1$ and $\mathcal{I}_2$ are not part of our proof system for DC*, they are valid DC* formulas, that possibly have the same expressive power as $DC_1^* - DC_3^*$ as DC*.

4.2 Local elimination of iteration from simple DC* formulas

Elimination of iteration from *timed regular expressions*, that are closely related to DC* simple formulas, has been employed earlier under various other conditions as part of model-checking algorithms by Dang and Pham[DP97], and Li, Dang and Zheng[LDZ97]. The contents of Lemma 1, Lemma 2, and Proposition 2 give a slightly stronger form of Lemma 3.6 from [LD96]. Iteration can be *locally eliminated* from a formula φ, if, for every DC interpretation $\mathcal{I}$ and every interval $[\tau_1, \tau_2] \in \mathbf{I}$, there exists a *-free formula φ' such that $\mathcal{I}, [\tau_1, \tau_2] \models \Box(\varphi \Leftrightarrow \varphi')$.

Due to the distributivity of conjunction and *chop* ($\frown$) over disjunction, simple formulas that have no occurrences of * are equivalent to disjunctions of *very simple* formulas, that are defined as follows:

Definition 5. Very simple *formulas are defined by the following BNF:*

$$\varphi \mathrel{\widehat{=}} \ell = 0 \mid \lceil\!\lceil S \rceil\!\rceil \mid a \leq \ell \mid \ell \leq a \mid (\varphi \wedge \varphi) \mid (\varphi \frown \varphi)$$

Lemma 1. *Let $\mathcal{I}$ be a DC interpretation. Let $[\tau_1, \tau_2] \in \mathbf{I}$. Let φ be a disjunction of very simple formulas that contain no subformulas of the kind $a \leq \ell$ with $a \neq 0$. Then there exists a $k < \omega$ such that $\mathcal{I}, [\tau_1, \tau_2] \models \Box(\varphi^* \Leftrightarrow \bigvee_{j=0}^{k} \varphi^j)$.*

Proof. See Appendix.

Lemma 2. *Let $\mathcal{I}$ be a DC interpretation. Let $[\tau_1, \tau_2] \in \mathbf{I}$. Let φ be a disjunction of very simple formulas. Then there exists a *-free simple formula φ' such that $\mathcal{I}, [\tau_1, \tau_2] \models \Box(\varphi^* \Leftrightarrow \varphi')$.*

Proof. See Appendix.

Lemma 3. *Let φ be a *-free simple formula. Then there exists formula φ' which is a disjunction of very simple formulas, such that $\vdash_{DC} \varphi \Leftrightarrow \varphi'$.*

Proof. The lemma follows trivially from the distributivity of the operators $\frown$ and $\wedge$ over the operator $\vee$.

Proposition 2. *Let $\mathcal{I}$ be a DC interpretation. Let $[\tau_1, \tau_2] \in \mathbf{I}$. Then for every simple formula φ there exists a *-free simple formula φ' such that $\mathcal{I}, [\tau_1, \tau_2] \models \Box(\varphi \Leftrightarrow \varphi')$.*

Proof. Proof is by induction on the number of occurrences of * in φ. Let ψ^* be a subformula of φ and let ψ be *-free. By Lemma 3, $\vdash_{DC} \psi \Leftrightarrow \psi'$ for some disjunction of very simple formulas ψ'. Now $\mathcal{I}, [\tau_1, \tau_2] \models \Box((\psi')^* \Leftrightarrow \psi'')$ for some *-free simple formula ψ'' by Lemma 2. Hence $\mathcal{I}, [\tau_1, \tau_2] \models \Box(\varphi \Leftrightarrow \varphi')$, where φ' is obtained by replacing the occurrence of ψ^* in φ by ψ''. Thus the number of the occurrences of * in φ is reduced by at least one.

4.3 Completeness of DC_1^*-DC_3^* for iteration of simple DC* formulas

In this section, we show our main result about the completeness. Namely, we prove that a formula γ is the iteration of a simple formula φ if and only if it satisfies the axioms DC_1^*, DC_2^* and DC_3^* for all DC* formulas. The following proposition has a key role in our proof.

Proposition 3. *Let $\mathcal{I}$ be a DC model that validates $DC^*_{1,\varphi,\gamma}$, $DC^*_{2,\varphi,\gamma}$ and $DC^*_{3,\varphi,\gamma}$ for some simple DC* formula φ, some arbitrary DC* formula γ, and all DC* formulas ψ. Then $\tilde{\mathcal{I}}(\gamma) = \left(\tilde{\mathcal{I}}(\varphi)\right)^*$.*

Proof. The validity of DC_1^* and DC_2^* entails that $\tilde{\mathcal{I}}(\gamma) \supseteq \left(\tilde{\mathcal{I}}(\varphi)\right)^*$. The proof of this is trivial and we omit it. For the sake of contradiction, assume that $[\tau_1, \tau_2] \in \tilde{\mathcal{I}}(\gamma) \setminus \left(\tilde{I}(\varphi)\right)^*$. By Proposition 2 there exists a simple formula φ' such that $\tilde{\mathcal{I}}(\varphi') \cap \mathbf{I}([\tau_1, \tau_2]) = \left(\tilde{\mathcal{I}}(\varphi)\right)^* \cap \mathbf{I}([\tau_1, \tau_2])$. Let $\psi \mathrel{\widehat{=}} \neg\varphi'$. Since $\mathcal{I}, [\tau_1, \tau_2] \models \gamma$ and $[\tau_1, \tau_2] \notin \left(\tilde{\mathcal{I}}(\varphi)\right)^*$, we have $\mathcal{I}, [\tau_1, \tau_2] \models \gamma \wedge \psi$, and hence $\mathcal{I}, [\tau_1, \tau_2] \models \gamma \wedge \psi^\frown\top$. Since $[\tau_1, \tau_1] \in \left(\tilde{\mathcal{I}}(\varphi)\right)^*$, $\mathcal{I}, [\tau_1, \tau_2] \not\models \psi \wedge \ell = 0^\frown\top$. Now assume that $\mathcal{I}, [\tau_1, \tau_2] \models (\gamma \wedge \neg\psi^\frown\varphi) \wedge \psi^\frown\top$. This entails that for some $\tau', \tau'' \in [\tau_1, \tau_2]$ $\mathcal{I}, [\tau_1, \tau'] \models \neg\psi$, and $\mathcal{I}, [\tau', \tau''] \models \varphi$. Then for some $k < \omega$ there exist $\tau'_1, \ldots, \tau'_{k+1}$ such that $\tau_1 = \tau'_1 < \ldots < \tau'_{k+1} = \tau''$ and $\mathcal{I}, [\tau'_i, \tau'_{i+1}] \models \varphi$ for $i = 1, \ldots, k$, and besides, $\mathcal{I}, [\tau'_1, \tau'_{k+1}] \models \psi$. This implies that $[\tau_1, \tau'_{k+1}] \in \left(\tilde{\mathcal{I}}(\varphi)\right)^k$ and $[\tau_1, \tau'_{k+1}] \in \tilde{\mathcal{I}}(\psi) \subseteq \mathcal{I} \setminus \left(\tilde{\mathcal{I}}(\varphi)\right)^*$, which is a contradiction.

Now let us state the completeness theorem for DC* with iteration of simple formulas.

Theorem 3. *Let φ be a DC* formula. Let that all of its *-subformulas be simple. Then either φ is satisfiable by some DC interpretation, or $\neg\varphi$ is derivable in our proof system.*

Proof. Assume that $\neg\varphi$ is not derivable. Let Γ be the set of all the instances of DC_1^*-DC_3^*. Then $\Gamma \cup \{\varphi\}$ is consistent, and by considering occurrences of a formula of the form ψ^* as a temporal variable, we have that $\Gamma \cup \{\varphi\}$ is a consistent set of DC formulas with temporal variables. By the ω-completeness of DC, there exists an interpretation I, and an interval $[\tau_1, \tau_2]$ such that $I, [\tau_1, \tau_2] \models \Gamma, \varphi$. Now Proposition 3 entails that $\tilde{I}(\psi^*) = \left(\tilde{I}(\psi)\right)^*$ for all ψ such that ψ^* occurs in φ, whence the modelling relation $I \models \varphi$ is as required for a DC* interpretation.

5 Decidability Results for Simple DC* and Discussion

In this section, we will discuss about the decidability of the satisfiability of simple DC* formulas and the related work.

One of the notions in the literatures that are closed to our notion of simple DC* is the notion of *Timed Regular Expressions* introduced by Asarin et al in [EPO97], a subset of which has been introduced by us earlier in [LD96]. Each simple DC* formula syntactically corresponds exactly to a timed regular expression, and their semantics coincide. Therefore, a simple DC* formula can be viewed as a timed regular expression. In [EPO97], it has been proved that from a timed regular expression E one can build a timed automaton A to recognise exactly the models of E in which the constants occurring in the constraints for the clock variables (guards, tests and invariants) are from the expression E (see [EPO97]). It is well known ([AD94]) that the emptiness of the timed automata is decidable for the case that the constants occurring in the guards and tests are integers [AD94], we can conclude that if only integer constants are allowed in the inequalities in the definition of simple DC* formulas, then the satisfiability of a simple DC* formulas is decidable.

Theorem 4. *Given a simple DC* formula φ in which all the constants occurring in the inequalities are integers. The satisfiability of φ is decidable.*

The complexity of the decidability procedure, however, is exponential in the size of the constants occurring in the clock constraints (see, e.g. [AD94]).

In [EPO97] it is also shown that from a timed automaton, one can build a timed regular expression and a renaming of the automaton states such that each model of the timed regular expression is the renaming of a behaviour of the automaton. In this sense, we can say that the expressive power of the simple DC* formulas is the same as the expressive power of the timed automata.

If we restrict ourselves to the class of *sequential* simple DC* formulas then we can have a very simple decidability procedure for the satisfiability, and some interesting results. The sequential simple DC* formulas are defined by the following BNF:

$$\varphi \mathrel{\widehat{=}} \ell = 0 \mid \llbracket S \rrbracket \mid \varphi \vee \varphi \mid (\varphi^\frown\varphi) \mid \varphi^* \mid \varphi \wedge a \leq \ell \mid \varphi \wedge \ell \leq a$$

Because the operators $^\frown$ and $\wedge$ are distributed over $\vee$, and because of the equivalence $(\varphi \vee \psi)^* \Leftrightarrow (\varphi^* {}^\frown \psi^*)^*$, each sequential simple DC* formula φ is equivalent

to a disjunction of simple formulas having no occurrences of $\vee$. Therefore φ is satisfiable iff at least one of the components of the disjunction is satisfiable. The satisfiability of sequential simple DC* formulas having no occurrence of $\vee$ is easy to decide. To each simple DC* formula φ having no occurrence of $\vee$, we can associate numbers $\min(S), \max(S) \in \mathbf{R} \cup \{\infty\}$ as follows.

$$
\begin{array}{l}
\min(\ell = 0) \mathrel{\widehat{=}} \max(\ell = 0) \mathrel{\widehat{=}} 0 \\
\min(\lceil S \rceil) \mathrel{\widehat{=}} 0 \\
\max(\lceil S \rceil) \mathrel{\widehat{=}} \infty \\
\min(\varphi_1{}^\frown\varphi_2) \mathrel{\widehat{=}} \min(\varphi_1) + \min(\varphi_2) \\
\max(\varphi_1{}^\frown\varphi_2) \mathrel{\widehat{=}} \max(\varphi_1) + \max(\varphi_2) \\
\min(\varphi^*) \mathrel{\widehat{=}} 0 \\
\text{if } \max(\varphi) > 0 \text{ then } \max(\varphi^*) \mathrel{\widehat{=}} \infty \text{ otherwise } \max(\varphi^*) \mathrel{\widehat{=}} 0 \\
\min(\varphi \wedge a \leq \ell) \mathrel{\widehat{=}} \max\{\min(\varphi), a\} \\
\max(\varphi \wedge a \leq \ell) \mathrel{\widehat{=}} \max(\varphi) \\
\min(\varphi \wedge \ell \leq a) \mathrel{\widehat{=}} \min(\varphi) \\
\max(\varphi \wedge \ell \leq a) \mathrel{\widehat{=}} \min\{\max(\varphi), a\}
\end{array}
$$

It is obvious that φ is satisfiable iff $\min(\varphi) \leq \max(\varphi)$.

In [LD96,LDZ97], we have developed some simple algorithms for checking a real-time system whose behaviour is described by a 'sequential' timed regular expression for a linear duration invariants of the form $\Box(a \leq \ell \leq b \Rightarrow \sum_{s \in S} c_s \int s \leq M)$. Because of the obvious correspondence between sequential simple DC* formulas and sequential timed regular expressions, these algorithms can be used for proving automatically the implication from a sequential simple DC* formula to a linear duration invariant. An advantage of the method is that it reduces the problem to several number of linear programming problems, which have been well understood. Because of this advantage, in [DP97], we tried to generalise the method for the general simple DC* formulas, and showed that in most cases, the method can still be used for checking the implication from a simple DC* formula to a linear duration invariant.

Together with the proof system presented in the previous sections, these decidability procedures will help to develop a tool to assist the designing and verification of real-time systems.

It seems that the with the extension of DC with the operator $*$, we can only capture the "regular" behaviour of real-time systems. In order to capture their full behaviour, we have to use the extension of DC with recursions. However, we believe that in this case the proof system would be more complicated, and would be far from to be complete.

References

[AGM92] S. Abramsky, D. Gabbay and T.S.E. Maibaum, eds. *Handbook of Logic in Computer Science*, Clarendon Press, Oxford, 1992.

[AD94] R. Alur and D.L. Dill, A Theory of Timed Automata, *Theoretical Computer Science* 126, 183-235, 1994.

[EPO97] E. Asarin, P. Caspi and O. Maler, A Kleene Theorem for Timed Automata, in G. Winskel (Ed.), Proceedings of IEEE International Symposium on *Logics in Computer Science LICS'97*, 1997, pp. 160-171.

[DW94] Dang Van Hung and Wang Ji. On The Design of Hybrid Control Systems Using Automata Models. V. Chandru and V. Vinay (eds.) LCNS 1180, *Foundations of Software Technology and Theoretical Computer Science*, 16th Conference, Hyderabad, India, December 1996, Springer, 1996.

[Dan98] Dang Van Hung. Modelling and Verification of Biphase Mark Protocols in Duration Calculus Using PVS/DC$^-$. Presented at and published in the Proceedings of the *1998 International Conference on Application of Concurrency to System Design* (CSD'98), 23-26 March 1998, Aizu-wakamatsu, Fukushima, Japan, IEEE Computer Society Press, 1998, pp. 88 - 98.

[DP97] Dang Van Hung and Pham Hong Thai. On Checking Parallel Real-Time Systems for Linear Duration Invariants, in Bernd Kramer, Naoshi Uchihita, Peter Croll and Stefano Russo (eds.) Proceedings of the International Symposium of Software Engineering for Parallel and Distributed Systems (PDSE'98), 20-21 April, 1998, Kyoto, Japan. IEEE Computer Society Press, 1998, pp. 61-71.

[Dut95] B. Dutertre. On First Order Interval Temporal Logic Report no. CSD-TR-94-3 Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, England, 1995.

[Gue98a] Dimitar P. Guelev. A Calculus of Durations on Abstract Domains: Completeness and Extensions. Technical Report 139, UNU/IIST, P.O.Box 3058, Macau, May 1998.

[Gue98b] Dimitar P. Guelev. Iteration of Simple Formulas in Duration Calculus. Technical Report 141, UNU/IIST, P.O.Box 3058, Macau, June 1998.

[HZ92] M. R. Hansen and Zhou Chaochen. Semantics and Completeness of Duration Calculus. In: *Real-Time: Theory and Practice*, LNCS 600, Springer-Verlag, 1992, pp. 209-225.

[HZ97] Michael R. Hansen and Zhou Chaochen. Duration Calculus: Logical Foundations. *Formal Aspects of Computing*, 9, 283-330 , 1997.

[HZ94] He Weidong and Zhou Chaochen. A Case Study of Optimization. *The Computer Journal*, Vol. 38, No. 9, pp. 734-746, 1995.

[LD96] Li Xuan Dong and Dang Van Hung. Checking Linear Duration invariants by Linear Programming. Joxan Jaffar and Roland H. C. Yap (Eds.), *Concurrency and Palalellism, Programming, Networking, and Security* LNCS 1179, Springer-Verlag, Dec 1996, pp. 321-332.

[LDZ97] Li Xuan Dong, Dang Van Hung, and Zheng Tao. Checking Hybrid Automata for Linear Duration Invariants. R.K.Shamasundar, K.Ueda (eds.), *Advances in Computing Science*, LNCS 1345, Springer-Verlag, 1997, pp.166-180.

[ED99] E. Pavlova and Dang Van Hung. A Formal Specification of the Concurrency Control in Real-Time Databases. Technical Report 152, UNU/IIST, P.O.Box 3058, Macau, January 1999.

[Sho67] J. Shoenfield. *Mathematical logic*. Addison-Wesley, Reading, Massachusetts, 1967.

[BHCZ94] Belawati H. Widjaja, He Weidong, Chen Zongji, and Zhou Chaochen. A Cooperative Design for Hybrid Systems. *Logic and Software Engineering International Workshop in Honor of Chih-Sung Tang*, pp. 127-150, Edited by A. Pnueli and H. Lin World Scientific, 1996.

[XH95] Xu Qiwen and He Weidong. Hierarchical Design of a Chemical Concentration Control System. Proceedings of *Hybrid Systems III: Verification and Control*, U.S.A., LNCS 1066, Springer–Verlag, 1995, pp. 270-281.

[YWZP94] Yu Xinyao, Wang Ji, Zhou Chaochen, and Paritosh K. Pandya. Specification of an Adaptive Control System. Research Report 19, UNU/IIST, P.O.Box 3058, Macau, 1. April 1994. Published in: *Formal Techniques in Real-Time and Fault-Tolerant systems*, LNCS 863, 1994, pp. 738–755.

[ZZ94] Zheng Yuhua and Zhou Chaochen. A Formal Proof of a Deadline Driven Scheduler. *Formal Techniques in Real-Time and Fault-Tolerant Systems*, LNCS 863, 1994, pp. 756–775.

[ZHR91] Zhou Chaochen, C. A. R. Hoare and A. P. Ravn. A Calculus of Durations. *Information Processing Letters*, 40(5):269-276, 1991

[ZZY94] Zhou Chaochen, Zhang Jingzhong, Yang Lu, and Li Xiaoshan. Linear Duration Invariants. *Formal Techniques in Real-Time and Fault-Tolerant systems*, LNCS 863, 1994.

A Proof of Theorem 2

We shall prove that

$$\varphi \Rightarrow \neg(\top^\frown\neg\alpha), \varphi \Rightarrow \neg(\neg\beta^\frown\top), \ell = 0 \Rightarrow \alpha, \ell = 0 \Rightarrow \beta \vdash_{DC^*} \varphi^* \Rightarrow \Box(\alpha^\frown\varphi^*{}^\frown\beta).$$

Then the theorem will follow by the deduction theorem for DC [HZ97].

1	$\varphi \Rightarrow \neg(\top^\frown\neg\alpha)$	asumption
2	$\varphi \Rightarrow \neg(\top^\frown\neg(\alpha^\frown\ell = 0))$	by 1
3	$\ell = 0 \Rightarrow \varphi^*$	by DC_1^*
4	$\varphi \Rightarrow \neg(\top^\frown\neg(\alpha^\frown\varphi^*))$	by 2, 3, $Mono_r$
5	$\ell = 0 \Rightarrow \alpha$	asumption
6	$\ell = 0 \Rightarrow (\ell = 0^\frown\ell = 0)$	$L2$
7	$\ell = 0 \Rightarrow (\alpha^\frown\varphi^*))$	by 5, 6, DC_1^*, $Mono_l$, $Mono_r$
8	$(\top^\frown\neg(\alpha^\frown\varphi^*)) \Rightarrow \neg\ell = 0$	by 7, DC
9	$\neg((\top^\frown\neg(\alpha^\frown\varphi^*)) \wedge \ell = 0^\frown\top)$	by 8, N_l
10	$(\varphi^* \wedge (\top^\frown\neg(\alpha^\frown\varphi^*))) \Rightarrow$ $((\top^\frown\neg(\alpha^\frown\varphi^*)) \wedge \ell = 0^\frown\top)\vee$ $((\varphi^* \wedge \neg(\top^\frown\neg(\alpha^\frown\varphi^*))^\frown\varphi)\wedge$ $(\top^\frown\neg(\alpha^\frown\varphi^*))^\frown\top)$	by DC_3^*
11	$(\varphi^* \wedge \neg(\top^\frown\neg(\alpha^\frown\varphi^*))^\frown\varphi)\wedge$ $(\top^\frown\neg(\alpha^\frown\varphi^*)) \Rightarrow$ $(\top^\frown(\alpha^\frown\varphi^*{}^\frown\varphi) \wedge \neg(\alpha^\frown\varphi^*))\vee$ $(\varphi \wedge (\top^\frown\neg(\alpha^\frown\varphi^*))))$	DC
12	$(\alpha^\frown\varphi^*{}^\frown\varphi) \Rightarrow (\alpha^\frown\varphi^*)$	by DC_2^*, $Mono_r$
13	$\neg((\alpha^\frown\varphi^*{}^\frown\varphi) \wedge \neg(\alpha^\frown\varphi^*))$ $\vee(\varphi \wedge (\top^\frown\neg(\alpha^\frown\varphi^*)))$	by 4, $Mono_r$, 14
14	$\neg(\top^\frown((\alpha^\frown\varphi^*{}^\frown\varphi) \wedge \neg(\alpha^\frown\varphi^*))$ $\vee(\varphi \wedge (\top^\frown\neg(\alpha^\frown\varphi^*))))$	by 13, N_r

15 $\neg((\varphi^* \wedge \neg(\top^\frown\neg(\alpha^\frown\varphi^*))^\frown\varphi)\wedge$
$(\top^\frown\neg(\alpha^\frown\varphi^*)))$ by 11, 14
16 $\varphi^* \Rightarrow \neg(\top^\frown\neg(\alpha^\frown\varphi^*)))$ by 9, 10, 15, $Mono_l$
17 $(\alpha^\frown\varphi^*) \wedge (\neg(\alpha^\frown\varphi^*{}^\frown\beta)^\frown\top) \Rightarrow$
$(\alpha \wedge (\neg(\alpha^\frown\varphi^*{}^\frown\beta)^\frown\top)^\frown\top)\vee$
$(\alpha^\frown(\varphi^* \wedge (\neg(\varphi^*{}^\frown\beta)^\frown\top)))$ DC
18 $\ell = 0 \Rightarrow \beta$ assumption
19 $\ell = 0 \Rightarrow ((\varphi^*{}^\frown\beta)^\frown\top)$ by 6, DC_1^*, 18, $Mono_l$, $Mono_r$
20 $\alpha \Rightarrow (\alpha^\frown\ell = 0)$ by $L3_r$
21 $\alpha \Rightarrow ((\alpha^\frown\varphi^*{}^\frown\beta)^\frown\top)$ by 19, 20, $Mono_r$
22 $(\alpha^\frown\varphi^*) \wedge (\neg(\alpha^\frown\varphi^*{}^\frown\beta)^\frown\top) \Rightarrow$
$(\alpha^\frown(\varphi^* \wedge (\neg(\varphi^*{}^\frown\beta)^\frown\top)))$ by 17, 21, $Mono_l$
23 $\varphi^* \wedge (\neg(\varphi^*{}^\frown\beta)^\frown\top) \Rightarrow$
$(\varphi^* \wedge (\neg(\varphi^*{}^\frown\beta)^\frown\top)^\frown\top)$ by $Mono_r$
24 $\varphi^* \wedge (\neg(\varphi^*{}^\frown\beta)^\frown\top) \Rightarrow$
$(\ell = 0 \wedge (\neg(\varphi^*{}^\frown\beta)^\frown\top)^\frown\top)\vee$
$(((\varphi^* \wedge \neg(\neg(\varphi^*{}^\frown\beta)^\frown\top))^\frown\varphi)\wedge$
$(\neg(\varphi^*{}^\frown\beta)^\frown\top)^\frown\top)$ by DC_3^*
25 $\ell = 0 \Rightarrow (\varphi^*{}^\frown\beta)$ by 6, DC_1^*, 18, $Mono_l$, $Mono_r$
26 $(\neg(\varphi^*{}^\frown\beta)^\frown\top) \Rightarrow \ell \neq 0$ by 25, $Mono_l$
27 $((\varphi^* \wedge \neg(\neg(\varphi^*{}^\frown\beta)^\frown\top))^\frown\varphi)$
$\wedge(\neg(\varphi^*{}^\frown\beta)^\frown\top) \Rightarrow$
$(\varphi^*{}^\frown\varphi \wedge (\neg\beta^\frown\top))$ DC
28 $\varphi \Rightarrow \neg(\neg\beta^\frown\top)$ assumption
29 $\neg(\varphi^*{}^\frown\varphi \wedge (\neg\beta^\frown\top))$ by 28, N_r
30 $\neg(((\varphi^* \wedge \neg(\neg(\varphi^*{}^\frown\beta)^\frown\top))^\frown\varphi)\wedge$
$(\neg(\varphi^*{}^\frown\beta)^\frown\top)^\frown\top)$ by 29, N_l
31 $\varphi^* \Rightarrow \neg(\neg(\varphi^*{}^\frown\beta)^\frown\top)$ by 23, 24, 26, 30
32 $\neg(\alpha^\frown(\varphi^* \wedge (\neg(\varphi^*{}^\frown\beta)^\frown\top)))$ by 31, N_r
33 $(\alpha^\frown\varphi^*) \Rightarrow \neg(\neg(\alpha^\frown(\varphi^*{}^\frown\beta))^\frown\top)$ by 22, 32
34 $\varphi^* \Rightarrow \neg((\top^\frown\neg(\alpha^\frown(\varphi^*{}^\frown\beta)))^\frown\top)$ by 16, 33, $Mono_l$, $Mono_r$

B Proof of Lemma 1

For $[\tau_1', \tau_2'] \subset [\tau_1, \tau_2]$, by the definition of the semantics of φ^*, we have that $\mathcal{I}, [\tau_1', \tau_2'] \models \varphi^*$ iff there exists a $n < \omega$ such that $\mathcal{I}, [\tau_1', \tau_2'] \models \varphi^n$. We shall prove that there exist k such that for all $[\tau_1', \tau_2'] \subset [\tau_1, \tau_2]$, m we have $\mathcal{I}, [\tau_1', \tau_2'] \models \varphi^m \Rightarrow \bigvee_{j=0}^{k} \varphi^j$.

Let $\varphi \mathrel{\widehat{=}} \bigvee_{i=1}^{p} \alpha_i$, where α_i are very simple formulas. By the finite variability there exist $\sigma_1, \ldots, \sigma_r \in [\tau_1, \tau_2]$ such that $\tau_1 = \sigma_1 < \ldots < \sigma_r = \tau_2$ and for every $i = 1, \ldots, r-1$ and every state expression S that occurs in φ either $\mathcal{I}, [\sigma_i, \sigma_{i+1}] \models \lceil S \rceil$, or $\mathcal{I}, [\sigma_i, \sigma_{i+1}] \models \lceil \neg S \rceil$. Let $b_0 = \min(\{b \mid \ell \leq b \text{ occurs in } \varphi \text{ and } b > 0\} \cup \{\infty\})$. Let $d = \left\lceil \frac{\tau_2 - \tau_1}{b_0} \right\rceil + 1$. Let $\mathcal{I}, [\tau_1', \tau_2'] \models \varphi^m$ for some m with $0 < m < \omega$ and $\tau_1' < \tau_2'$. This implies that there exist $\xi_1, \ldots, \xi_{n+1} \in [\tau_1, \tau_2]$ and $\beta_1, \ldots, \beta_n \in$

$\{\alpha_1, \ldots \alpha_p\}$ such that $n \leq m$ $\tau_1' = \xi_1 < \ldots < \xi_{n+1} = \tau_2'$ and $\mathcal{I}, [\xi_i, \xi_{i+1}] \models \beta_i$ for $i = 1, \ldots, n$. There are at most $r-1$ such indices i for which there exist $j \leq r-1$ satisfying $\xi_i < \sigma_j < \xi_{i+1}$. For all other values of i there exists $j \leq r-1$ such that $[\xi_i, \xi_{i+1}] \subset [\sigma_j, \sigma_{j+1}]$. Therefore, we can find l indexes $1 = i_1 < \ldots < i_l = n$, $l \leq 2 * r$ such that for $1 \leq s \leq l-1$ either $i_{s+1} = i_s + 1$ or $[\xi_{i_s}, \xi_{i_{s+1}}] \subseteq [\sigma_j, \sigma_{j+1}]$ for some $j \leq r$. Hence, $\mathcal{I}, [\xi_{i_s}, \xi_{i_{s+1}}] \models \beta_{i_s}$ holds for the former case. Consider the latter case, i.e. $[\xi_{i_s}, \xi_{i_{s+1}}] \subseteq [\sigma_j, \sigma_{j+1}]$. Let β_{i_s} contain subformulas of the kind $\ell \leq b$. Since $b \geq b_0$, since for every state expression S that occurs in β_{i_s} $[\![S]\!]$ holds in all nonpoint subintervals of $[\sigma_j, \sigma_{j+1}]$, since there is no subformula of the form $a \leq \ell$ with $a > 0$ in β_{i_s}, and since $\mathcal{I}, [\xi_{i_s}, \xi_{i_s+1}] \models \beta_{i_s}$, we have that any subinterval of $[\sigma_j, \sigma_{j+1}]$ with the length less than $(\tau_2 - \tau_1)/b_0$ should satisfy β_{i_s}. Hence, because $\sigma_{j+1} - \sigma_j \leq \tau_2 - \tau_1$, for any $[\xi, \eta] \subset [\sigma_j, \sigma_{j+1}]$, $\xi \neq \eta$, $\mathcal{I}, [\xi, \eta] \models \beta_{i_s}^{(\lceil \frac{\tau_2 - \tau_1}{b_0} \rceil + 1)}$. In case no formula of the kind $\ell \leq b$ occurs in β_{i_s}, the same holds trivially.

Consequently, for all $1 \leq s \leq l-1$ either $\mathcal{I}, [\xi_{i_s}, \xi_{i_{s+1}}] \models \varphi$ or $\mathcal{I}, [\xi_{i_s}, \xi_{i_{s+1}}] \models \varphi^d$ holds. Therefore, for $k = (2r-1)d$ we have $\mathcal{I}, [\xi_{i_s}, \xi_{i_{s+1}}] \models \bigvee_{j=0}^{k} \varphi^j$ holds.

Now obviously the existence of an n such that $\mathcal{I}, [\tau_1', \tau_2'] \models \varphi^n$ entails that $\mathcal{I}, [\tau_1', \tau_2'] \models \bigvee\limits_{i=0}^{k} \varphi^i$, and the lemma follows immediately.

C Proof of Lemma 2

Let $\varphi \mathrel{\widehat{=}} \bigvee\limits_{i=1}^{p} \alpha_i \vee \bigvee\limits_{j=1}^{q} \beta_j$, where α_i, $i = 1, \ldots, p$, contain no subformulas of the kind $a \leq \ell$, $a \neq 0$, and β_j does contain such occurrences for every $j = 1, \ldots, q$. The case in which there are no βs has been dealt with in Lemma 1. Let $A \mathrel{\widehat{=}} \bigvee\limits_{i=1}^{p} \alpha_i$, $B \mathrel{\widehat{=}} \bigvee\limits_{j=1}^{q} \beta_j$ and $a_0 = \min\{a \mid a \leq \ell \text{ occurs in } B \text{ and } a > 0\}$. By the property of $\beta_j' s$ it must be that $a_0 > 0$. Then obviously $\mathcal{I}, [\tau_1, \tau_2] \not\models B^k$ for $k \geq \left\lceil \frac{\tau_2 - \tau_1}{a_0} \right\rceil$. Hence,

$$\mathcal{I}, [\tau_1, \tau_2] \models \Box \left(\varphi \Leftrightarrow \bigvee_{i=0}^{\lceil \frac{\tau_2 - \tau_1}{a_0} \rceil} (A^* \frown (B \frown A^*)^i) \right).$$

By Lemma 1, there exists a simple formula A' with no occurrences of *, such that $\mathcal{I}, [\tau_1, \tau_2] \models \Box(A \Leftrightarrow A')$. Hence

$$\mathcal{I}, [\tau_1, \tau_2] \models \Box \left(\varphi \Leftrightarrow \bigvee_{i=0}^{\lceil \frac{\tau_2 - \tau_1}{a_0} \rceil} (A' \frown (B \frown A')^i) \right),$$

which completes the proof.