Gabbay Separation for the Duration Calculus

Dimitar P. Guelev

Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria

Abstract

Gabbay’s separation theorem about linear temporal logic with past has proved to be one of the most useful theoretical results in temporal logic. In particular it enables a concise proof of Kamp’s seminal expressive completeness theorem for LTL. In 2000, Alexander Rabinovich established an expressive completeness result for a subset of the Duration Calculus (DC), a real-time temporal logic. DC is based on the chop binary modality, which restricts access to subintervals of the reference time interval, and is therefore regarded as introspective. The considered subset of DC is known as the $[P]$-subset in the literature. Neighbourhood Logic (NL), a system closely related to DC, is based on the neighbourhood modalities, also written $\langle A \rangle$ and $\langle \overline{A} \rangle$ in the notation stemming from Allen’s system of interval relations. These modalities are expanding as they allow writing future and past formulas to impose conditions outside the reference interval. This setting makes temporal separation relevant: is expressive power ultimately affected, if past constructs are not allowed in the scope of future ones, or vice versa? In this paper we establish an analogue of Gabbay’s separation theorem for the $[P]$-subset of the extension of DC by the neighbourhood modalities, and the $[P]$-subset of the extension of DC by the neighbourhhood modalities and chop-based analogue of Kleene star. We show that the result applies if the weak chop inverses, a pair binary expanding modalities, are given the role of the neighbourhood modalities, by virtue of the inter-expressibility between them and the neighbourhood modalities in the presence of chop.

2012 ACM Subject Classification Theory of computation → Automated reasoning; Theory of computation → Modal and temporal logics; Theory of computation → Logic and verification

Keywords and phrases Gabbay separation, Neighbourhood Logic, Duration Calculus, expanding modalities

Digital Object Identifier 10.4230/LIPIcs.TIME.2022.10

Introduction

Separation for Linear Temporal Logic (LTL, cf., e.g., [28]) was established by Dov Gabbay in [14]. Separation is about expressing temporal properties without making reference to the past in the scope of future constructs and vice versa. Gabbay proved that such a restriction does not affect the ultimate expressive power of past LTL, by a syntactically defined translation from arbitrary formulas to ones that are separated, i.e., satisfy the restriction. The applications of this theorem are numerous and important on their own right. They include a concise proof of Kamp’s seminal expressive completeness result for LTL (see, e.g., [13]), the elimination of the past modalities from LTL, which simplifies the study of extensions of LTL, c.f., e.g., [10], Fisher’s clausal normal form for past LTL [12], other normal forms [19, 15], etc. In this paper we establish an analogue of Gabbay’s separation theorem for the extension of a subset of the Duration Calculus (DC) with a pair of expanding modalities known as the neighbourhood modalities, with and without the chop-based analogue of Kleene star, which is also called iteration in DC.

The Duration Calculus (DC, [32, 30]) is an extension of real time Interval Temporal Logic (ITL), which was first proposed by Moszkowski for discrete time [24, 25, 11]. DC is a real-time interval-based predicate logic for the modeling of hybrid systems. Unlike time points, time intervals, the possible worlds in DC, have an internal structure of subintervals.
This justifies calling modalities like \textit{chop introspective} for their providing access to these subintervals only. Modalities for reaching outside the reference interval are called \textit{expanding}. Several sets of such modalities have been proposed in the literature.

In this paper we prove a separation theorem for the \([P]\)-subset of DC with the expanding \textit{neighbourhood modalities} \(\diamond_l\) and \(\diamond_r\) added to DC’s \textit{chop} and \textit{iteration}. The system based on \(\diamond_l\) and \(\diamond_r\) only, which are also written \((A)\) and \((\overline{A})\) after Allen’s interval relations [3], is called Neighbourhood Logic (NL, [4]), whereas we target DC with \(\diamond_l\) and \(\diamond_r\). Our theorem holds with \textit{iteration} included too. We write DC-NL (DC-NL\textsuperscript{*}) for DC with \(\diamond_l\) and \(\diamond_r\) (and \textit{iteration}). In separated formulas, \(\diamond_d\) cannot appear in the scope of other modalities, except \(\diamond_d, d = l, r\). \(\diamond_r\)-free formulas are regarded as \textit{past}, and \(\diamond_l\)-free formulas are \textit{future}. The \textit{strict} forms of past (future) formulas are defined by further restricting \textit{chop} and \textit{iteration} to occur only in the scope of a \(\diamond_l\) (\(\diamond_r\)). DC is a predicate logic. We prove that formulas in each of \([P]\)-subsets of DC-NL and DC-NL\textsuperscript{*} have \textit{separated} equivalents in their respective subsets. These subsets are compatible with the system of DC from Rabinovich’s expressive completeness result [29]. We also show that the \textit{weak chop inverses}, which are \textit{binary} expanding modalities, are expressible using \(\diamond_l\) and \(\diamond_r\) in the considered subset. Their use in the \textit{Mean-value Calculus}, another system from the DC family, was studied in [26]. \(\diamond_l\) and \(\diamond_r\) are definable using the weak chop inverses. Consequently, our separation theorem applies to the extensions of DC and DC\textsuperscript{*} by the weak chop inverses too.

The technique of our proofs builds on our finds from [16] which led to establishing separation for discrete time ITL.

\textbf{Structure of the paper.} Section 1 gives preliminaries on DC and DC\textsuperscript{*}, the neighbourhood modalities, the weak chop inverses, and a supplementary result on quantification over state in DC. In Section 2 we state our separation theorem for the \([P]\)-subsets of DC-NL and DC-NL\textsuperscript{*} and give a simple example application. Section 3 is dedicated to the proof. The transformations for separating DC-NL and DC-NL\textsuperscript{*} formulas are given in Sections 3.2 and 3.3, respectively, and use a lemma which is given in the preceding Section 3.1. Section 4 is about the expressibility of the weak chop inverses in the \([P]\)-subsets of DC-NL and DC-NL\textsuperscript{*}, using the lemma from Section 3.1 too. We conclude by pointing to some related work and making some comments on the relevance of the result.

\section{Preliminaries}

An in-depth presentation of DC and its extensions can be found in [30]. The syntax of the \([P]\)-subset of DC is built starting from a set \(V\) of \textit{state variables}. It includes \textit{state expressions} \(S\) and \textit{formulas} \(A\). Let \(P\) stand for a \textit{state variable}. The BNFs are:

\[
S ::= 0 \mid P \mid S \Rightarrow S \quad A ::= \bot \mid \top \mid [S] \mid A \Rightarrow A \mid A ; A
\]

\textbf{Semantics.} Given a set of state variables \(V\), the type of \textit{valuations} \(I\) is \(V \times \mathbb{R} \rightarrow \{0, 1\}\). Valuations \(I\) are required to have \textit{finite variability}.

For any \(P \in V\) and any bounded interval \([a, b] \subset \mathbb{R}\) there exists a finite sequence \(t_0 = a < t_1 < \ldots < t_n = b\) such that \(\lambda t. I(P, t)\) is constant in \((t_{i-1}, t_i), i = 1, \ldots, n\).

The \textit{value} \(I(t)(S)\) of \textit{state expression} \(S\) at \textit{time} \(t \in \mathbb{R}\) is defined by the clauses:

\[
I_t(0) \equiv 0, \quad I_t(P) \equiv I(P, t), \quad I_t(S_1 \Rightarrow S_2) \equiv \max\{I_t(S_2), 1 - I_t(S_1)\}.
\]
Satisfaction has the form $I, [a, b] \models A$, where $[a, b] \subseteq \mathbb{R}$. The defining clauses are:

$$I, [a, b] \not\models \bot, \quad I, [a, b] \models \top \quad \text{iff} \quad a = b,$$

$$I, [a, b] \models [S] \quad \text{iff} \quad a < b \text{ and } I(S) = 1 \text{ for all but finitely many } t \in [a, b],$$

$$I, [a, b] \models A \Rightarrow B \quad \text{iff} \quad I, [a, b] \models B \text{ or } I, [a, b] \not\models A,$$

$$I, [a, b] \models A; B \quad \text{iff} \quad I, [a, m] \models A \text{ and } I, [m, b] \models B \text{ for some } m \in [a, b].$$

The connectives $\neg, \land, \lor$ and $\Leftrightarrow$ are defined as usual in both state expressions and formulas. Furthermore $1 \equiv 0 \Rightarrow 0$ and $\top \equiv \bot \Rightarrow \bot$. A formula $A$ is valid in DC, written $\models A$, if $I, [a, b] \models A$ for all $I$ and all intervals $[a, b]$. In this paper we consider the extension of the $[P]$-subset of DC by the neighbourhood modalities $\diamond_d, d \in \{l, r\}$. The defining clauses for their semantics are as follows:

$$I, [a, b] \models \diamond_l A \quad \text{iff} \quad I, [a', a] \models A \text{ for some } a' \leq a,$$

$$I, [a, b] \models \diamond_r A \quad \text{iff} \quad I, [b, b'] \models A \text{ for some } b' \geq b.$$

The universal duals $\Box_d$ of $\diamond_d$ are defined by putting $\Box_d A \equiv \neg \diamond_d \neg A$, $d \in \{l, r\}$. *Chop* $A; B$ is written $A \Leftrightarrow B$ in much of the literature. We write DC-NL for the extension of DC by $\diamond_l$ and $\diamond_r$. We also consider DC-NL*, the extension of DC-NL by iteration, the chop-based form of Kleene star, included. The defining clause for this operator is

$$I, [a, b] \models A^* \quad \text{iff} \quad a = b \text{ or there exist a finite sequence } m_0 = a < m_2 < \cdots < m_n = b \text{ such that } I, [m_{i-1}, m_i] \models A \text{ for } i = 1, \ldots, n.$$ 

Iteration is interdefinable with positive iteration $A^+ \equiv A; (A^+)$, which we assume to be the derived one of the two: $\models A^+ \iff [\top] \lor A^*$.

**Predicate DC and NL** include a (defined) flexible constant $\ell$ for the length $b - a$ of reference interval $[a, b]$. Using $\ell$, *chop* can be defined in NL:

$$A; B \equiv \exists x \exists y (x + y = \ell \land \diamond_l \diamond_r (A \land \ell = x) \land \diamond_r (B \land \ell = y)).$$

This definition is not available in NL’s $[P]$-subset. Therefore we discern the $[P]$-subsets of NL and DC-NL.

**Quantification over state in DC.** Given a state variable $P$, $I, [a, b] \models \exists P A$ iff $I', [a, b] \models A$ for some $I'$ such that $I'(Q, t) = I(Q, t)$ and all $Q \in V \setminus \{P\}, t \in \mathbb{R}$. Quantification over state is expressible in the $[P]$-subset of DC*:

**Theorem 1.** For every $[P]$-formula $A$ in DC* and every state variable $P$ there exists a (quantifier-free) $[P]$-formula $B$ in DC* such that $\models B \Leftrightarrow \exists P A$.

Mind that $B$ is not guaranteed to be iteration-free, even in case $A$ is.

This theorem follows from a correspondence between stutter-invariant regular languages and the $[P]$-subset that led to the decidability of the $[P]$-subset in [31]. It is not our contribution, but the transformations from its proof supplement those from our other proofs.

**Notation.** In this paper write $\varepsilon$, possibly with subscripts, to denote optional occurrences of the negation sign $\neg$, e.g. $\varepsilon_0$ below. We write $[A/B]C$ to denote the result of simultaneously replacing all the occurrences of $B$ by $A$ in $C$, e.g., $[0/P]S$ below.
10:4 Gabbay Separation for the Duration Calculus

Proof of Theorem 1. Following [31], $A$ translates into a regular expression over the alphabet

$$\Sigma \doteq \{ \bigwedge_{Q \text{ is a state variable in } A} \varepsilon_Q : \varepsilon_Q \text{ is either } \neg \text{ or nothing} \}.$$  \hfill (1)

The translation clauses are as follows:

$$t(\bot) \doteq \emptyset \quad \quad \quad \quad t([S]) \doteq \{ \{ \sigma \in \Sigma : \models \sigma \Rightarrow S \} \}^+ \quad \quad \quad \quad t(A; B) \doteq t(A); t(B)$$

$$t([\Top]) \doteq \epsilon \quad (\text{the empty string}) \quad t(A \Rightarrow B) \doteq t(B) \cup \Sigma^* \setminus t(A) \quad \quad \quad \quad t(A^*) \doteq t(A)^*$$

Up to equivalence, $t$ can be inverted. Regular expressions admit complementation- and $\cap$-free equivalents; hence these operations can be omitted in the converse translation $\bar{t}$:

$$\bar{t}(\emptyset) \doteq \bot \quad \bar{t}(a) \doteq [a] \quad \text{for } a \in \Sigma \quad \bar{t}(R_1 \cup R_2) \doteq \bar{t}(R_1) \lor \bar{t}(R_2) \quad \bar{t}(R^*) \doteq \bar{t}(R)^*$$

Given a regular expression $R = t(A)$, $\bar{t}(R')$ is equivalent to $A$ for any $R'$ that defines the same language as $R$. Applying $\bar{t}$ to a complementation- and $\cap$-free equivalent $R'$ to $t(A)$ produces an equivalent to $A$ with $\lor$ as the only propositional connective, except possibly inside state expressions. Given this, $\exists P$ can be eliminated from formulas of the form $\bar{t}(R')$:

$$\models \exists P \perp \iff \models \exists P [S] \iff \models \exists P [0/P]S \lor [1/P]S)^+ \models \exists P (A_1; A_2) \iff \exists P A_1; \exists P A_2$$

$$\models \exists P [\Top] \iff \models \exists P (A_1 \lor A_2) \iff \exists P A_1 \lor \exists P A_2 \models \exists P A^* \iff (\exists P A)^*.$$  \hfill ▷

The equivalence $\exists P [S]$ above hinges on the finite variability of $I_t(P)$.  

The weak chop inverses $A/B$ and $A\setminus B$, cf., e.g., [26], are defined by the clauses:

$$I, [a, b] \models A/B \iff \text{for all } r \geq b, \text{ if } I, [b, r] \models A \text{ then } I, [a, r] \models A.$$

$$I, [a, b] \models A\setminus B \iff \text{for all } l \leq a, \text{ if } I, [l, a] \models B \text{ then } I, [l, b] \models A.$$

$\Diamond A$ and $\Diamond A$ can be defined as $\neg(\bot \setminus A)$ and $\neg(\bot / A)$, respectively. In Section 4 we show how $A/B$ and $A\setminus B$ can be expressed using $\Diamond$ and $\Diamond$, too for $[P]$-formulas $A$ and $B$, but with the expressing formulas built in a more complex way.

Separation as Known for LTL. We relate the setting and statement of Gabbay’s separation theorem about past LTL as our work builds in the example of this theorem. Let $p$ stand for an atomic proposition. Discrete time LTL formulas with past have the syntax:

$$A ::= \bot | p | A \Rightarrow A | \Diamond A | A U A | \Diamond A | A S A$$

$\Diamond$ and $S$ are the past mirror operators of $\Diamond$ and $U$. $\Diamond$- and $S$-free formulas are called future formulas, and $\Diamond$- and $U$-free formulas are called past. Formulas of the form $\Diamond F$ where $F$ is future are called strictly future. In [14], Dov Gabbay demonstrated that any formula in LTL with past is equivalent to a Boolean combination of past and strictly future formulas for flows of time which are either finite or infinite, in either the future or thepast, or both.

Modal heights $h_{\Diamond}(.), h_{\Diamond}(.),$ and $h(.)$ of formulas wrt the neighbourhood modalities and iteration appear in our inductive reasoning below. In general, $h(A)$ denotes the length of the longest chain of $A$’s subformulas, including possibly $A$, with the main connective being the specified modality wrt the (transitive closure of) the subformula relation.
2 The Separation Theorem

In this section we formulate the main contribution of the paper, Theorems 2 and 3, which are separation theorems for the \([P]\)-subsets of DC-NL and DC-NL*, and use Theorem 2 to demonstrate the expressibility of an interval-based version of the “past-forgetting” operator from [18] as a simple example application.

We call DC-NL (DC-NL*) formula \(F\) (non-strictly) future if it has the syntax

\[ F ::= C \mid \neg F \mid F \lor F \mid O_r F \]

where \(C\) stands for a DC (DC*) formula, where \(chop\) (and \(iteration\)) are the only modalities. Non-strictly past formulas are defined similarly, with \(O_l\) instead of \(O_r\). A separated formula is a Boolean combination of past and future formulas.

Following the example of LTL, we call Boolean combinations of \(O_l\), resp. \(O_r\)-formulas with non-strict past, resp. future operands strictly past, resp. strictly future formulas. Such formulas can impose no conditions on the reference interval; they only refer to the adjacent past and future intervals along the timeline. These adjacent intervals still include the respective endpoints of the reference interval. However the \([P]\) construct cannot tell apart interpretations \(I\) of the state variables such that \(\lambda t.I(P,t)\) varies only at finitely many time points \(t\). Unlike that, in discrete time an extra step away from the present time using \(\sqcap\), resp., \(\sqcup\) is necessary to prevent a formula from imposing conditions on the reference time point or the reference interval’s respective endpoint. This shared time point causes strictly past and strictly future formulas to be defined differently in discrete time ITL. Separated formulas can also be defined as Boolean combinations of strictly past formulas, strictly future formulas and introspective, i.e., just DC (DC*), formulas, where the only modalities are \(chop\) (and \(iteration\)), that are known as introspective too.

\[\begin{align*}
\text{Theorem 2.} & \quad \text{Let } A \text{ be a } [P]\text{-formula in DC-NL (DC-NL*)}. \text{ Then there exists a separated } [P]\text{-formula } A' \text{ in DC-NL (DC-NL*) such that } \models A \iff A'.
\end{align*}\]

In Section 4 we demonstrate the inter-expressibility between \((\ldots)\) and \((\ldots)\), and \(O_l\) and \(O_r\), respectively. This implies that Theorem 2 holds for the weak chop inverses instead of the respective \(O_d\), \(d \in \{l, r\}\) wrt a corresponding notion of separated formula too:

\[\begin{align*}
\text{Theorem 3.} & \quad \text{Let } A \text{ be a } [P]\text{-formula in the extension of DC (DC*) by } (\ldots) \text{ and } (\ldots). \text{ Then there exists a separated } [P]\text{-formula } A' \text{ in DC (DC*) with } (\ldots) \text{ and } (\ldots) \text{ such that } \models A \iff A'.
\end{align*}\]

An Example Application: Expressing the \(N\) operator. The temporal operator \(N\) (“now”) was proposed for past LTL in [18], see also [17], as a means for “preventing access” into the past beyond the time of applying \(N\). Assuming \(\sigma = \sigma_0 \sigma_1 \ldots\) to be a sequence of states

\[\sigma, i \models_{\text{LTL}} NA \iff \sigma^i \sigma^{i+1} \ldots, 0 \models_{\text{LTL}} A.\]

If an arbitrary closed interval \(D \subseteq \mathbb{R}\), and not only the whole of \(\mathbb{R}\), is allowed to be the time domain, \(N\) can be defined for (real-time) DC-NL too. With such time domains, the endpoints of “all time” can be identified, because, e.g., \(D, I, [a, b] \models O_l [ ]\) iff \(a = \min D\). (Since the \([P]\)-subset of DC-NL is merely topological, as opposed to metric, it cannot distinguish open time domains from \(\mathbb{R}\).) We can define \(N\) on intervals by putting:

\[\begin{align*}
D, I, [a, b] & \models N_l A \iff \{x \in D : x \geq a\}, I, [a, b] \models A \\
D, I, [a, b] & \models N_r A \iff \{x \in D : x \leq b\}, I, [a, b] \models A
\end{align*}\]

Theorem 2 entails that \(N_l\) and \(N_r\) are expressible in DC-NL:
Proposition 4. DC-NL + N_i, N_r has the same expressive power as DC-NL.

Proof. Let A' be a separated equivalent of A. Then |= N_d A ↔ [◊_d (B ∧ [])]/◊_d B : B ∈ Subf(A')]A', d ∈ {l, r}, where Subf(F) stands for the set of the subformulas of F, including F.

3 The Proof of Separation for DC-NL and DCNL*

In this section we propose a set of valid equivalences which, if appropriately used as transformation rules starting from some arbitrary given formula from the [P]-subset of DC-NL*, lead to a separated formula in DC-NL*. If the given formula is iteration-free, i.e., in DC-NL, then so is the separated equivalent. This amounts to proving Theorem 2.

Our key observation is that formulas which are supposed to be evaluated at intervals that extend some given interval into either the future or the past have equivalents which consist of subformulas to be evaluated at the given interval and subformulas to be evaluated at intervals which are adjacent to it, these two subintervals being appropriately referenced using chop as parts of the enveloping interval. In our proof of separation, this observation is referred to as a lemma that states the possibility to express any introspective formula as a case distinction of subformulas to be evaluated at the given interval and subformulas to be evaluated at intervals extend some given interval into either the future or the past have equivalents which consist of chop-formulas with the LHS (RHS) operands of chop forming a full system. The lemma can be seen as a generalization of the guarded normal form, which is ubiquitous in process logics, with the full systems of guards describing a primitive opening move replaced by full systems of interval-based temporal conditions to be satisfied at whatever prefixes (suffixes) of the lemma that states the possibility to express any introspective formula as a case distinction of subformulas to be evaluated at the given interval and subformulas to be evaluated at intervals extend some given interval into either the future or the past have equivalents which consist of chop-formulas with the LHS (RHS) operands of chop forming a full system. The lemma can be seen as a generalization of the guarded normal form, which is ubiquitous in process logics, with the full systems of guards describing a primitive opening move replaced by full systems of interval-based temporal conditions to be satisfied at whatever prefixes (suffixes) of the reference runs necessary. Later on we use the lemma in expressing (\cdot/\cdot) ((\cdot\cdot\cdot)) in terms of ◊_r (◊_l) too.

3.1 The Key Lemma

A finite set of formulas A_1, ..., A_n is a full system, if |= \bigwedge_{k=1}^{n} A_k and, given 1 ≤ k_1 < k_2 ≤ n, |= ¬(A_{k_1} ∧ A_{k_2}).

Lemma 5. Let A be a [P]-formula in DC (DC*). Then there exists an n < ω and some DC [DC*] [P]-formulas A_k, A'_k, k = 1, ..., n, such that A_1, ..., A_n form a full system and

|= A ⇔ \bigvee_{k=1}^{n} A_k; A'_k and |= A ⇔ \bigwedge_{k=1}^{n} ¬(A_k; ¬A'_k). (2)

Furthermore, h∗(A_k) ≤ h∗(A) and h∗(A'_k) ≤ h∗(A).

Informally, this means that, I, [a, b] |= A iff whenever m ∈ [a, b] and I, [a, m] |= A_k, I, [m, b] |= A'_k holds. Furthermore, for every m ∈ [a, b] there is a unique k such that I, [a, m] |= A_k. Interestingly, the construct ¬(F; ¬G) used in the second equivalence (2) is regarded as a form of temporal implication, written F ⇒ G, in ITL [23, 5]. This construct is akin to suffix implication [2], see also [1]. It requires the suffix of an interval to satisfy B, if the complementing prefix satisfies A. Much like ⇒’s being the right adjoint of ∧, ⇒ is the right adjoint of chop:

|= (A ⇒ (B ⇒ C)) ⇔ ((A; B) ⇒ C).

In this paper we stick to the notation in terms of chop for both ⇒ and its mirror ¬(¬G; F).
Proof of Lemma 5. Induction on the construction of $A$. For $\bot$, $[\bot]$ and $[P]$, we have:

$$\models \bot \iff (\top; \bot) \quad \models [\bot] \iff ([\bot]; [\bot]) \lor (\neg [\bot]; \bot)$$
$$\models [P] \iff ([P]; ([P] \lor [\bot]) \lor ([\bot]; [P]) \lor (\neg ([\bot] \lor [P]); \bot))$$

Let $B_1, \ldots, B_n$, $B'_1, \ldots, B'_n$ satisfy (2) for $B$ and $C_1, \ldots, C_m$, $C'_1, \ldots, C'_m$ satisfy (2) for $C$. Then:

$$\models B \circ P C \iff \bigvee_{k=1}^n \bigwedge_{i=1}^m (B_k \land C_i); (B'_k \circ P C'_i), \; \circ \in \{\Rightarrow, \land, \iff\}$$

$$\models B; C \iff \bigvee_{k=1}^n \bigwedge_{x \subseteq \{1, \ldots, m\}} \left( \bigwedge_{\ell \in X} (B \land C \land \neg(B; C_i)); \left( (B'_k \land C); \bigvee_{\ell \in X} C'_i \right) \right)$$

For the equivalence on the left in (2) about $\bot$, $\bot$, and $[\bot]$. Let $I, [a, b] \models B; C$, $m \in [a, b]$, and $I, [a, m] \models B$ and $I, [m, b] \models C$. Let $t \in [a, b]$. If $t \in [a, m]$, then $I, [a, t] \models B_k$ for some unique $k$. If $t \in [m, b]$, then a unique $X \subseteq \{1, \ldots, m\}$ exists such that $I, [a, t] \models B; C_i$ holds iff $\ell \in X$. The conjunctions of $B_k \land \bigwedge_{\ell \in X} (B \land C \land \neg(B; C_i)); k = 1, \ldots, n, X \subseteq \{1, \ldots, m\}$ form a full system because both the $B_k$s, and the conjunctions $\bigwedge_{\ell \in X} (B \land C \land \neg(B; C_i)); X \subseteq \{1, \ldots, m\}$. Since $I, [a, m] \models B$ and $I, [m, b] \models C$, for an $[a, t]$ satisfying the member of this full system for any given $k$ and $X$, we can conclude that $I, [a, t] \models (B'_k \land C \land \neg(B; C_i))$ from the assumptions on the $B'_k$s and the $C'_i$s. For the converse implication ($\iff$), let $[a, b]$ be an arbitrary interval, $t \in [a, b]$, and let $I, [a, t] \models B_k \land \bigwedge_{\ell \in X} (B \land C \land \neg(B; C_i))$, which is bound to be true for some unique pair $k, X$. Then, $I, [t, b] \models B'_k; C$ implies $I, [a, b] \models B_k; B'_k; C$; and $I, [t, b] \models C'_i$ implies $I, [a, b] \models B; C_i; C'_i$ for any $\ell \in X$. In both cases $I, [a, b] \models B; C$ follows because $I, [t, b] \models B_k \land C'_i \Rightarrow B$ and $C \models C'_i \Rightarrow C$. The LHS equivalence (2) about $B^*$ is established similarly, with the use of $C$ facilitating a uniform handling of the case of $B^*$ holding trivially at $0$-length intervals. The RHS equivalences (2) follow from the LHS ones by the assumption that the $A_k$s form a full system.

Observe that the equivalence (3) about $A = B^*$ satisfies $h_\cdot(A_k) \leq h_\cdot(A)$ and $h_\cdot(A'_k) \leq h_\cdot(A)$. The non-increase of $h_\cdot(\cdot)$ also holds for the rest of the equivalences, which, despite not featuring iteration explicitly, may become used for transforming formulas with iteration. Hence, $h_\cdot(A_k) \leq h_\cdot(A)$ and $h_\cdot(A'_k) \leq h_\cdot(A)$ for all $A$. □

The time mirror image of Lemma 5 holds too, with the time mirror of (2) reading

$$\models A \iff \bigvee_{k=1}^n A'_k; A_k \quad \text{and} \quad \models A \iff \bigvee_{k=1}^n \neg(A'_k; A_k).$$

The proof is no different because all the modalities are symmetrical wrt the direction of time. For this reason, in the sequel we omit “mirror” statements and their proofs.
On the complexity of the transformations from Lemma 5. Interestingly, a peak (exponential) blowup in the transformations from Lemma 5’s proof occurs in the clause for $\text{chop}$ and not the clause for $\neg$, the typical source of such blowups. However, a closer look at the inductive assumptions shows that the pairwise inconsistency achieved at the cost of using $A_k \land \bigwedge_{l \in X} (A; B_l) \land \bigwedge_{l \in X} \neg (A; B_l)$ for all $k \in \{1, \ldots, m\}$ and the $2^n$ different $X \subseteq \{1, \ldots, m\}$ in the required full system is instrumental for the correctness of the clause about the binary Boolean connectives, where negation is obtained for $op \Rightarrow$ and $B = \bot$. Hence this blowup can be linked to the alternation of $\neg$ and monotone operators such as $\text{chop}$ that is common in proofs of the non-elementariness of the blowup upon reaching normal forms.

Lemma 5 admits an automata-theoretic proof, along the lines of the proof of Theorem 1. We have sketched such a proof for discrete time ITL in [16]. That proof leads to different $A_k$ and $A'_k$ satisfying (2) for the same $A$, and allows a non-elementary upper bound on the length of these formulas to be established using the size of a deterministic FSM recognizing $A$. Unlike the automata-based proof, the equivalences of this proof suggest transformations that are valuable for their compositionality and their validity in DC in general, and not just for the $[P]$-subset. Furthermore, the proof given here facilitates establishing that $^*$-height is not increased upon moving to the RHSs of (2).

3.2 Separating the Neighbourhood Modalities in DC-NL

In this section we prove Theorem 2 by showing how occurrences of $\Diamond_d$ can be taken out of the scope of $\text{chop}$ and $\Diamond_3$, $d \in \{l, r\}$, $\bar{l} \equiv r$, $\bar{r} \equiv l$. The transformations that we propose are supposed to be applied bottom up, on formulas with $\text{chop}$ or $\Diamond_d$, $d \in \{l, r\}$, as the main connective, assuming that the operands of are already separated. If the main connective is $\Diamond_d$, then we need to target only the $\Diamond_3$-subformulas in $\Diamond_d$’s operand, possibly at the cost of introducing some $\Diamond_3$-subformulas in the scope of $\text{chop}$, to be subsequently extracted from there too.

To show that the above transformations combine into a terminating procedure which produces a separated formula, for DC-NL, we reason by induction on the $\Diamond_d$-height of the relevant formulas. In the case of DC-NL*, which is the topic of Section 3.2, we also keep track of $^*$-height. It is not increased upon applying Lemma 5, nor by the transformations for separating formulas with $\Diamond_3$, $\Diamond_r$ or $\text{chop}$ as the main connective. The effect on $^*$-height of eliminating some quantification over state which appears at an intermediate stage of the transformations by an application of Theorem 1 on $^*$-height is irrelevant because it involves only introspective, i.e., DC*, formulas. In most cases, we give detail only on the extracting of $\Diamond_r$-subformulas, because of the time symmetry.

Separating $\Diamond_d$-formulas. Let $d = l$; the case of $d = r$ is its mirror. Since

$$\models \Diamond_3(A_1 \lor A_2) \iff \Diamond_3A_1 \lor \Diamond_3A_2,$$

the availability of DNF for $A$ of $\Diamond_3A$ makes it sufficient to consider the case of $A$ of the form $P \land \bigwedge_{k=1}^n \varepsilon_k \Diamond_r F_k$ where $P$ is (non-strictly) past and $F_1, \ldots, F_n$ are future. Observe that

$$\models \Diamond_3\left(P \land \bigwedge_{k=1}^n \varepsilon_k \Diamond_r F_k\right) \iff \Diamond_3P \land \bigwedge_{k=1}^n (\langle \top \rangle \land \varepsilon_k \Diamond_r F_k); \top).$$

Using (4) and (5) does not increase $\Diamond_3$-height and implies that separating $\Diamond_3A$ reduces to separating $(\langle \top \rangle \land \varepsilon \Diamond_r F_k); \top)$, which are $\text{chop}$-formulas. Here follow the transformations for doing this.
Separating chop-formulas. We need to consider only chop applied to conjunctions of
introspective formulas and possibly negated past $\diamond_r$-formulas or future $\diamond_r$-formulas because
\[ (L_1 \lor L_2); R \iff (L_1; R) \lor (L_2; R) \text{ and } \models L; (R_1 \lor R_2) \iff (L; R_1) \lor (L; R_2) \]
Past $\diamond_r$-formulas (future $\diamond_r$-formulas) can be extracted from the left (right) operand of chop
using that
\[ (L \land \varepsilon \diamond_1 P); R \iff (L; R) \land \varepsilon \diamond_1 P \text{ and } \models L; (R \land \varepsilon \diamond_r F) \iff (L; R) \land \varepsilon \diamond_r F. \quad (6) \]
Much like (4), this does not affect $\diamond_d$-height. It remains to consider $(L \land \bigwedge_{k=1}^n \varepsilon_k \diamond_r F_k); R,$
which, by virtue of the time symmetry, will explain separating $(L \land \bigwedge_{k=1}^n \varepsilon_k \diamond_1 P_k)$ too.

The transformations of formulas of the form $(L \land \varepsilon \diamond_r F); R$ below are about the designated
$\varepsilon \diamond_r F$ only, and are supposed to be used repeatedly, if $L$ has more conjuncts of this form. By
(4), $F$ can be assumed to be a conjunction $C \land G$ where $C$ is introspective and $G$ is strictly
future. Let $C_k, C'_k, k = 1, \ldots, n,$ satisfy Lemma 5 for $C$. We do the cases of $(L \land \diamond_r F); R$
and $(L \land \neg \diamond_r F); R$ separately.

$(L \land \diamond_r F); R$: Observe that
\[ (L \land \diamond_r (C \land G)); R \iff (L; (R \land ((C \land G); \top))) \lor \bigvee_{k=1}^n (L; (R \land C_k)) \land \diamond_r (C'_k \land G) \]
and further process the RHS of $\iff$ in it. The two disjuncts on the RHS above correspond
to $F$ being satisfied at an interval which is shorter, or the same length, or longer than the
one which presumably satisfies $R$. Since $C_k$ and $C'_k$ are introspective, the newly introduced
formulas $\diamond_r (C'_k \land G)$ on the RHS of $\iff$ are separated. $G$ can be extracted from the scope of
chop in $(L; (R \land ((C \land G); \top)))$ too, because $h_{\diamond_r} (G) < h_{\diamond_r} ((L \land \diamond_r F); R)$.

$(L \land \neg \diamond_r F); R$: Satisfying $(L \land \neg \diamond_r (C \land G)); R$ requires $\neg (C \land G)$ to hold at all the
intervals which start at the right end of the one where $L$ presumably holds. Therefore we
can use that
\[ (L \land \neg \diamond_r (C \land G)); R \iff \bigvee_{k=1}^n (L; (R \land C_k \land \neg ((C \land G); \top))) \land \neg \diamond_r (C'_k \land G). \]
Again, $G$ must be extracted from the scope of chop in the newly introduced $L; (R \land C_k \land
\neg ((C \land G); \top))$ on the RHS of the equivalence. This can be accomplished because $h_{\diamond_r} (G) < h_{\diamond_r} ((L \land \neg \diamond_r F); R)$.

The transformations above are sufficient for establishing Theorem 2 about DC-NL. By
Lemma 5, these transformations do not cause $^*\text{-height}$ to increase. This is relevant in
separating formulas in DC-NL$^*$, which is explained next.

3.3 Separating iteration formulas
To extract $\diamond_1$ and $\diamond_r$ from the scope of iteration, we use the inter-expressibility between
iteration and quantification over state, and the expressibility of quantification over state in the $[P]$-subset of DC$^*$ (Theorem 1). Consider $B^*$ where $B$ is a separated formula. Without
loss of generality, $B$ can be assumed to be $\bigvee_{i=1}^r B_s$ where
\[ B_s \doteq H_s \land \bigwedge_{i=1}^n \varepsilon_{s,i} \diamond_1 P_i \land \bigwedge_{j=1}^n \varepsilon_{s,j} \diamond_r F_j, \]
$H_s, s = 1, \ldots, t$ are introspective, $P_i, i = 1, \ldots, u$, are past formulas, and $F_j, j = 1, \ldots, v$, are future formulas. Furthermore, $P_i, i = 1, \ldots, u$, $(F_j, j = 1, \ldots, v)$ can be assumed to be conjunctions of introspective and strictly past (strictly future) formulas by (4) and its mirror equivalence.

Let $T, S^p_i$, $i = 1, \ldots, u$, and $S^f_j, j = 1, \ldots, v$, be fresh state variables. Then

$$\models B^* \Leftrightarrow \exists T \exists S^p_1 \ldots \exists S^p_u \exists S^f_1 \ldots \exists S^f_v \left( \left( [T]; [\neg T] \right) \land \bigvee_{s=1}^{t} \left( B_s \land \bigwedge_{i=1}^{u} [\varepsilon^p_{s,i} S^p_i] \land \bigwedge_{j=1}^{v} [\varepsilon^f_{s,j} S^f_j] \right) \right)^* ,$$

This equivalence states that an interval $[a, b]$ such that $I, [a, b] \models B^*$ can be partitioned into subintervals $[m_0, e_1], \ldots, [m_{d-1}, e_d]$ so that each subinterval satisfies $B_s$ for some $s \in \{1, \ldots, t\}$, and an assignment of $T, S^p_1, \ldots, S^p_u$ and $S^f_1, \ldots, S^f_v$ can be chosen so that, for $d = 1, \ldots, e$, $[m_{d-1}, e_d]$ is a maximal $[T]; [\neg T]$-interval, and for some $s \in \{1, \ldots, t\}$ such that $I, [m_{d-1}, e_d] \models B_s$, $I, [m_{d-1}, e_d] \models [\varepsilon^p_{s,i} S^p_i]$ iff $I, [m_{d-1}, e_d] \models e^p_{s,i} P_i$, $i = 1, \ldots, u$, and $I, [m_{d-1}, e_d] \models [\varepsilon^f_{s,j} S^f_j]$ if $I, [m_{d-1}, e_d] \models e^f_{s,j} F_j, j = 1, \ldots, v$.

Now observe that $I, [m_{d-1}, e_d] \models B_s$ would follow if $I, [m_{d-1}, e_d] \models H_s$, and for some appropriate $a' \leq m_{d-1}, I, [a', m_{d-1}] \models [\varepsilon^p_{s,i} P_i]$, $i = 1, \ldots, u$, and, for some appropriate $b' \geq m_d, I, [m_k, b'] \models [\varepsilon^f_{s,j} F_j], i = 1, \ldots, v$. Here appropriate stands for all $b' \geq m_d (a' \leq m_{d-1})$, if $e^f_{s,j} (e^p_{s,i})$ is $\neg$; otherwise it stands for some $b' \geq m_d (a' \leq m_{d-1})$. Furthermore, the $m_d$ such that $I, [m, b'] \models [\varepsilon^f_{s,j} F_j]$ is required for all (some) $b' \geq m_d$ can be identified by the condition that $\neg T \land \varepsilon^f_{s,j} S^f_j$ holds in a left neighbourhood of $m_d$ and $T$ holds in a right neighbourhood of $m_d$, for $d = 1, \ldots, e - 1$. For $d = e$, $m_d = b$, and, unless $a = b$, $\neg T \land \varepsilon^f_{s,j} S^f_j$ holds in a left neighbourhood of $m_d$. The mirror conditions allow identifying the $m_{d-1}$ for which $I, [a', m_{d-1}] \models [\varepsilon^p_{s,i} P_i]$ is required, for either some or all $a' \leq m_{d-1}$, depending on $\varepsilon^p_{s,i}, d = 1, \ldots, e$, with $m_0$ similarly handled separately.

Given the possibility to identify the relevant $m_d$ as observed, $I, [m_d, b'] \models [\varepsilon^f_{s,j} F_j]$ for the required $b' \geq m_d$ can be expressed as $I, [a, b] \models \psi_j$ where

$$\psi_j \equiv \left( (T; [S^f_j]) \Rightarrow \bigcirc_F j \land \neg ((T; [S^f_j] \land \neg T); (([T]; T) \land \neg (\bigcirc_F j \land [\neg T])); (([T]; [\neg T]) \land ((\bigcirc_F j \land [\neg T])); (([T]; T) \land ((\bigcirc_F j \land [\neg T])); ) \right). \quad (7)$$

The time mirrors of $\psi_j$ can be used to enforce $I, [a', m_{d-1}] \models [\varepsilon^p_{s,i} P_i]$, for the required $a' \leq m_{d-1}, i = 1, \ldots, u$. Let these formulas be $\pi_i, i = 1, \ldots, u$. Then $B^*$ is equivalent to

$$\exists T \exists S^p_1 \ldots \exists S^p_u \exists S^f_1 \ldots \exists S^f_v \left( \left( [T]; [\neg T] \right) \land \bigvee_{s=1}^{t} H_s \land \bigwedge_{i=1}^{u} [\varepsilon^p_{s,i} S^p_i] \land \bigwedge_{j=1}^{v} [\varepsilon^f_{s,j} S^f_j] \right)^* . \quad (8)$$

$\bigcirc_r F_j$ occurs in the left operand of $chop$ in $\psi_j$. As mentioned above, by the mirror equivalence of (4), $F_j$ can be assumed to be the conjunction of some introspective $C_j$ and some strictly future $G_j$. Let $C_{j,k}$ and $C'_{j,k}$, $k = 1, \ldots, u$, satisfy Lemma 5 for $C_j$. Then

$$\models ((\bigcirc_r F_j \land [\neg T]) \land ([T]; T) \lor \bigwedge_{k=1}^{n} C_{j,k} \Rightarrow \bigcirc_r (C'_{j,k} \land G_j)). \quad (9)$$

Since $h_{\bigcirc_r} (G_j) < h_{\bigcirc_r} (B)$ and $h_{\bigcirc_r} (G_j) < h_{\bigcirc_r} (B)$, $G_j$ can be extracted from the left operand of $chop$ in the RHS of (9). This produces a (non-strictly) future formula which is equivalent to $((\bigcirc_r F_j \land [\neg T]) \land ([T]; T)$ by this future formula in (7), the $\bigcirc_r$-subformulas of this future formula and the formulas $\bigcirc_r (C'_{j,k} \land G_j)$ can be further extracted
from the right operand of chop in (7) using the right equivalence of (6). This leads to a future equivalent of \( \varphi_j \), by which we replace \( \varphi_j \) in (8), \( j = 1, \ldots, v \). We use the time mirror of (9) and the left equivalence of (6) to similarly replace \( \pi_i \), \( i = 1, \ldots, u \), by some appropriate past equivalents. This leads to a separated formula as the operand of \( \exists T \exists S_\sigma^p \ldots \exists S_u^p \exists S_1^f \ldots \exists S_v^f \) in (8).

In order to obtain a separated equivalent to \( B^* \), we need to eliminate this quantifier prefix. To this end, observe that the \( \Diamond_l \)- and \( \Diamond_r \)-subformulas which appear in the separated equivalents of \( \pi_i \), \( i = 1, \ldots, u \), and \( \varphi_j \), \( j = 1, \ldots, v \), have no occurrences of \( T, S_\sigma^p, S_u^p, S_1^f, \ldots, S_v^f \), and are linked with the remaining introspective subformulas in the scope of \( \exists T \exists S_\sigma^p \ldots \exists S_u^p \exists S_1^f \ldots \exists S_v^f \), which may have such occurrences, by Boolean connectives only. Hence the \( \Diamond_l \)- and \( \Diamond_r \)-subformulas can be extracted using the De Morgan laws and

\[
\models \exists S (X \vee Y) \Leftrightarrow \exists S X \vee \exists S Y, \quad \text{and, for S-free } X, \quad \models \exists S (X \wedge Y) \Leftrightarrow X \wedge \exists S Y,
\]

Then the quantifier prefix can be eliminated by Theorem 1, which is about introspective formulas only. Hence Theorem 2 holds about DC-NL* too.

4 Expressing the Weak Chop Inverses by the Neighbourhood Modalities and Separation for the Weak Chop Inverses

In this section we prove that the weak chop inverses are expressible in DC-NL, which means that separation applies to DC with these expanding modalities instead of \( \Diamond_l \) and \( \Diamond_r \) too.

Suppose that \( A_1, A_2, B \) are separated formulas in DC-NL (DC-NL*). Then the availability of conjunctive normal forms and the validity of the equivalences

\[
(A_1 \wedge A_2)/B \Leftrightarrow A_1/B \wedge A_2/B
\]

entails that we need to consider only formulas \( A/B \) where \( A \) is a disjunction of introspective formulas, strictly future formulas and strictly past formulas. Strictly past disjuncts \( P \) in the left operand of \((/.)\) can be extracted using the validity of

\[
(A \vee P)/B \Leftrightarrow P \vee A/B.
\]

The following proposition shows how to express \( A/B \) in case \( A \) is a disjunction of introspective and possibly negated \( \Diamond_r \)-formulas.

\begin{proposition}
Let \( A \) be a \( \lfloor P \rfloor \)-formula in DC (DC*) and \( A_k, A_k^* \), \( k = 1, \ldots, n \) satisfy Lemma 5 for \( A \). Let \( B \) be a \( \lfloor P \rfloor \)-formula in DC-NL*. Let \( F \) be a strictly future formula. Then

\[
\models (A \vee F)/B \Leftrightarrow \bigvee_{k=1}^n A_k \wedge \Box_r(B \Rightarrow (A_k^* \vee F)). \tag{10}
\]

\end{proposition}

\begin{proof} (\( \Rightarrow \)): Let \( I, [a, b] \) satisfy the RHS of (10). Consider an arbitrary \( r \geq b \) such that

\[
I, [b, r] \models B.
\]

Then \( I, [a, r] \models A \vee F \). There is a (unique) \( k \in \{1, \ldots, n\} \) such that \( I, [a, b] \models A_k \). Hence \( I, [b, r] \models A_k^* \vee F \) follows from \( I, [a, r] \models A \vee F \) and \( \models A \Rightarrow (A_k; \neg A_k^*) \), which follows from Lemma 5. The \( (\Leftarrow) \) direction is trivial to check and we omit it.

The formula for \( A/B \) in terms of \( \Diamond_l \) and \( \Diamond_r \) in the RHS of (10) can be further separated to extract past subformulas of \( B \) from the scope of \( \Box_r \) as in DC-NL (DC-NL*). The above argument shows that \((/.)\)-formulas whose operands are in the \( \lfloor P \rfloor \)-subset of DC-NL (DC-NL*)
have equivalents in the $[P]$-subset of DC-NL (DC-NL*) themselves. Observe that, in the presence of chop, it takes only $\Diamond_1$ to eliminate $(\setminus /)$. Similarly, $(\setminus,)$, which is about looking to the left of reference interval, can be eliminated using only chop and $\Diamond_1$. As mentioned in the Preliminaries section, expressing $\Diamond_1$ and $\Diamond_2$ by means of $(\setminus,)$ and $(\setminus /)$ is straightforward. This concludes our reduction of the $[P]$-subset of DC-NL (DC-NL*) with the weak chop inverses to the $[P]$-subset of DC-NL (DC-NL*), and entails that separation applies to that system too as stated in Theorem 3.

Concluding Remarks

In this paper we have shown how separation after Gabbay applies to the $[P]$-subsets of DC-NL and DC-NL*, the extensions of DC by the neighbourhood modalities. These subsets correspond to the subset of DC whose expressive completeness was demonstrated in [29].

The $[S]$-construct, which is definitive for the $[P]$-subsets of DC-NL and DC-NL*, has a considerable similarity with the homogeneity principle which is known from studies on neighbourhood logics of discrete time. That principle was proposed in [22, 20] and was adopted in a number of more recent works such as [7, 8, 9]. Unlike the locality principle from Moszkowski’s (standard) discrete time ITL, where the satisfaction of an atomic proposition $p$ is determined by the labeling of the initial state of the reference interval, homogeneity means that atomic proposition $p$ must label all the states in the reference interval for $p$ to hold at that interval as a formula. The two variants are ultimately interdefinable, but facilitate applications in a slightly different way. Homogeneity can be compared with DC’s $[P]$ because $[P]$ means that $P$ is supposed to hold “almost everywhere” in the reference interval. The main difference is that varying valuations at zero-length interval is negligible in real-time NL and DC, whereas the labeling of the only point in such intervals can be referred to in discrete time. This leads to different notions of strictly past and strictly future formulas. It is known that past expanding modalities increase the ultimate expressive power of discrete time ITL [21], and not just its succinctness, the latter being the case in past LTL. This adds to the relevance of algorithmic methods for interval-based expanding modalities in general.

Providing a separation theorem to the $[P]$-subset of DC-NL improves our understanding of the logic and may facilitate further results. One obvious avenue of future study would be to consider interval-based variants of the applications of separation that are known about point-based past LTL. In particular, one rather straightforward application would be to simplify the theoretical considerations that are needed for the study of extensions, especially branching time ones such as [27], by making it sufficient to consider future-only formulas, while still enjoying the succinctness contributed by the availability of past operators.

References

D. P. Guelev 10:13


Antonio Cau, Ben Moszkowski, and Hussein Zedan. ITL web pages. URL: http://www.antonio-cau.co.uk/ITL/.


Gabbay Separation for the Duration Calculus


