29th International Symposium on Temporal Representation and Reasoning

TIME 2022, October 7-9,2022

Gabbay Separation for the Duration Calculus

a sequel paper of

A Separation Theorem for Discrete Time Interval Temporal Logic

JANCL, 2022, joint with Ben Moszkowski

Dimitar P. Guelev

http://www.math.bas.bg/~gelevdp

Plan of Talk

Introduction: LTL with Past and Gabbay's theorem

Preliminaries on Interval Temporal Logic (ITL, Moszkowski, Moszkowski et al, 1983-)

ITL with $\langle A\rangle$, $\langle\overline{A}\rangle$, also written $\Diamond_l$, $\Diamond_r$ in DC

The Separation Theorem in ITL [Guelev and Moszkowski, JANCL 2022]

DC and the relevant classes of formulas: (strictly) past and (strictly) future.

Key part of the proof (for both ITL and DC.)

Questions

The Grand Prototype: Separation in LTL with Past (PLTL) [Gabbay, 1989]

Set of atomic propositions AP. An interval $I \subseteq \mathbb{Z}$; $\sigma : I \to \mathcal{P}(AP)$, $i \in I$.

$$A ::= \mathit{true} \mid \underbrace{p}_{\in AP} \mid \neg A \mid A \vee A \mid \underbrace{\bigcirc A \mid A \,\mathsf{U}\, A}_{\text{not allowed in past formulas}} \mid \underbrace{\ominus A \mid A \,\mathsf{S}\, A}_{\text{not allowed in future formulas}}$$

$$\sigma, i \models \bigcirc A \quad \text{iff} \quad \sigma, i+1 \models A, \qquad \sigma, i \models \ominus A \quad \text{iff} \quad \sigma, i-1 \models A$$

$$\sigma, i \models A \,\mathsf{U}\, B \quad \text{iff} \quad \exists k (\sigma, i+k \models B \wedge \bigwedge_{j=0}^{k-1} \sigma, i+j \models A)$$

$$\sigma, i \models A \,\mathsf{S}\, B \quad \text{iff} \quad \exists k (\sigma, i-k \models B \wedge \bigwedge_{j=-k+1}^{0} \sigma, i+j \models A)$$

$\diamondminus A \mathrel{\hat{=}} \mathit{true} \,\mathsf{S}\, A$; Strictly future (past) formulas: $\bigcirc F$ ($\ominus P$).

Theorem 1 (Gabbay, 1989) *Every* LTL *formula is equivalent to a BC of past formulas, strictly future formulas and atomic propositions.*

An Example Generic Application to Synthesis

Any separated A is equivalent to a boolean combination of past and future formulas in conjunctive normal form. Let

$$A \mathrel{\hat{=}} \bigwedge_k (\underbrace{P_{k,1} \vee \cdots \vee P_{k,n_k}}_{\hat{=}\, P_{k},\text{past}} \vee \underbrace{\bigcirc F_{k,1} \vee \cdots \vee \bigcirc F_{k,m_k}}_{\hat{=}\, \bigcirc F_{k},\text{future}})$$

Then $\models A \equiv \bigwedge_k \neg P_k \supset \bigcirc F_k$, 'If $\neg P_k$ is observed, then F_k is forthcoming'.

$\mathsf{I} \mathrel{\hat{=}} \neg \ominus true$,

Consider $\Box \diamondminus (\mathsf{I} \wedge B)$; let $A \mathrel{\hat{=}} \diamondminus (\mathsf{I} \wedge B)$

Then: $\models \Box \diamondminus (\mathsf{I} \wedge B) \equiv \bigwedge_k \Box (\neg P_k \supset \bigcirc F_k)$

ITL

A vocabulary is a set of atomic propositions V.

Semantics

$\sigma \mathrel{\hat{=}} \sigma^0\sigma^1 \ldots \in \mathcal{P}(V)^+ \cup \mathcal{P}(V)^\omega$ have been dubbed intervals,

These are sequences $[0, ..., |\sigma|] \rightarrow \mathcal{P}(V)$, like (not necessaryly infinite) LTL traces.

Unlike $\sigma, i \models_{\mathrm{PLTL}} \ldots$, we have $\sigma \models_{\mathrm{ITL}} \ldots$.

However, accommodating expanding modalities takes first moving to

$$\sigma, i, j \models_{\mathrm{ITL}} \ldots, \quad i < j, \quad i, j \in \operatorname{dom} \sigma$$

where $\sigma : I \rightarrow \mathcal{P}(V)$, $I \subseteq \mathbb{Z}$ - an interval.

$\models$ **for** $A ::= \mathit{false} \mid p \mid A \supset A \mid \bigcirc A \mid A;A \mid A^* \ , \quad p \in V$

$\sigma \models p$ iff $p \in \sigma^0$

next $\sigma \models \bigcirc A$ iff $|\sigma| \geq 1$ and $\sigma^{1\uparrow} \models A$

chop $\sigma \models A;B$ iff for some $k \leq |\sigma|,\ k < \omega,\ \sigma^{0..k} \models A$ and $\sigma^{k\uparrow} \models B$

chop-star $\sigma \models A^*$ iff either $|\sigma| = 0$,

or there exists a finite sequence

$k_0 = 0 < k_1 < \ldots < k_n \leq |\sigma|,\ k_n < \omega$

such that $\sigma^{k_i..k_{i+1}} \models A$ for $i = 0, \ldots, n-1,$ and $\sigma^{k_n\uparrow} \models A$,

or $|\sigma| = \omega$ and there exists an infinite sequence

$k_0 = 0 < k_1 < \ldots$ such that $\sigma^{k_i..k_{i+1}} \models A$ for all $i < \omega$.

$\sigma, i, j \models A$ **generalizes** $\sigma^{i..j} \models A$ **for the 'core' ITL operators.**

$(\sigma^0\sigma^1\ldots)^{b..e} \mathrel{\hat{=}} \sigma^b \ldots \sigma^e$, if $0 \leq b \leq e \leq |\sigma|$; $\quad (\sigma^0\sigma^1\ldots)^{k\uparrow} \mathrel{\hat{=}} (\sigma^k\sigma^{k+1}\ldots)$, if $k \leq |\sigma|$.

The Neighbourhood Modalities $\Diamond_l$, $\Diamond_r$, AKA $\langle \overline{A} \rangle$ and $\langle A \rangle$

$\sigma, i, j \models \Diamond_l A$ iff $i > -\infty$ and there exists a $k \leq i$ such that $\sigma, k, i \models A$

$\sigma, i, j \models \Diamond_r A$ iff $j < \infty$ and there exists a $k \geq j$ such that $\sigma, j, k \models A$

The Separation Theorem in ITL with $\Diamond_l$ and $\Diamond_r$

Introspective formulas C: - 'core' ITL (just chop and possibly chop-star)

Past formulas: $P ::= C \mid \neg P \mid P \vee P \mid \Diamond_l P$

Past = no $\Diamond_r$, and no $\Diamond_l$ in the scope of chop or chop-star.

Strictly past formulas: $\Diamond_l(P; skip)$

$skip \mathrel{\hat{=}} \bigcirc \neg \bigcirc true$ provides that the P-interval and the reference interval are apart.

Future formulas ($\Diamond_r$ instead of $\Diamond_l$): $F ::= C \mid \neg F \mid F \vee F \mid \Diamond_r F$.

Stricty future formulas: $\Diamond_r(skip; F)$ where F is future.

Theorem 2 (separation for ITL, Guelev and Moszkowski, JANCL 2022) *Every ITL formula is equivalent to a boolean combination of strictly past formulas, strictly future formulas and introspective formulas.*

(!) The point-based prototype's p-s become interval C-s.

The Theorem Applies to the Weak Binary Chop Inverses

$\sigma, i, j \models A/B$ iff for all $k \geq j,$ if $\sigma, j, k \models B$ then $\sigma, i, k \models A$.

$\sigma, i, j \models A \backslash B$ iff for all $k \leq i,$ if $\sigma, k, i \models B$ then $\sigma, k, j \models A$.

Interestingly, some of the technique for proving separation helps establishing:

$\mathrm{ITL} + \Diamond_r = \mathrm{ITL} + (./.); \ \mathrm{ITL} + \Diamond_l = \mathrm{ITL} + (.\backslash.)$

Past chop, signed chop, embedding all reasoning in formulas that are evaluated at infinite intervals.

The Prototype's Applications

These, I believe, can be ported from the LTL case; that automatically leads to stronger results, given the greater expressive power of ITL.

Separation at Work in Branching Time Logics with Past

The key observation looks next to trivial but saves a lot of hassle:

$\sigma, i \models \exists A$ iff a σ' exists (in the model) s.t. $\sigma'|_{\{\ldots,i\}} = \sigma|_{\{\ldots,i\}}$ and $\sigma, i \models A$

Now, A may be imposing restrictions on both $\sigma|_{\{\ldots,i\}}$ and $\sigma|_{\{i,\ldots\}}$.

If, e.g. $\models A \equiv P \wedge F$, then $\models \exists(P \wedge F) \equiv P \wedge \exists F$.

Hence restricting to only Fs in the scope of $\exists$ WL of expressiveness.

$\exists$ is CTL^*'s branching time construct; other BT constructs admit the same transformations.

The same applies to branching time systems that have an interval-based set of (linear time) connectives. Cf. e.g. Cong Tian and Zhenhua Duan's Interval-based ATL [ICFEM 2010]. **Enter interval-based separation!**

The $\lceil P \rceil$-subset of DC

Vocabulary: sets V of state variables P, Q, $\ldots$

Models: $I : V \times \mathbb{R} \rightarrow \{0, 1\}$

Finite Variability: For every $P \in V$ and every $[a, b] \subset \mathbb{R}$ 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), $i = 1, \ldots, n$.

Syntax: state expressions S and formulas A:

$$S ::= \mathbf{0} \mid P \mid S \Rightarrow S$$

$$A ::= \mathit{false} \mid \lceil\rceil \mid \lceil S \rceil \mid A \Rightarrow A \mid A ; A$$

Semantics: $I_t(S)$ and $I, [a,b] \models A$

$S ::= \mathbf{0} \mid P \mid S \Rightarrow S \qquad A ::= \mathit{false} \mid \lceil\rceil \mid \lceil S \rceil \mid A \Rightarrow A \mid A ; A$

$I_t(\mathbf{0}) \mathrel{\hat{=}} 0, \quad I_t(P) \mathrel{\hat{=}} I(P, t), \quad I_t(S_1 \Rightarrow S_2) \mathrel{\hat{=}} \max\{I_t(S_2), 1 - I_t(S_1)\}.$

$I, [a,b] \not\models \mathit{false}, \qquad I, [a,b] \models \lceil\rceil$ iff $a = b$

$I, [a,b] \models \lceil S \rceil$ iff $a < b$ and $\{t \in [a,b] : I_t(S) = 0\}$ is finite

$I, [a,b] \models A \Rightarrow B$ iff $I, [a,b] \models B$ or $I, [a,b] \not\models A$

$I, [a,b] \models A ; B$ iff $I, [a,m] \models A$ and $I, [m,b] \models B$ for some $m \in [a,b]$

Abbreviations: $\top$, $\neg$, $\wedge$, $\vee$ and $\Leftrightarrow$ are defined as usual.

$\mathbf{1} \mathrel{\hat{=}} \mathbf{0} \Rightarrow \mathbf{0} \qquad \Diamond A \mathrel{\hat{=}} \top ; A ; \top \qquad \Box A \mathrel{\hat{=}} \neg \Diamond \neg A \ \ldots$

$A ; B$ is written $A \frown B$ in much of the literature on DC.

Validity: $\models A$, if $I, [a,b] \models A$ for all I and all intervals $[a,b]$.

The Defining Clauses for $\Diamond_l$ and $\Diamond_r$ Are the Same

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

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

In $\Diamond_l$ and $\Diamond_r$, $_l$ and $_r$ stand for left (past) and right (future), respectively.

DC-NL $\hat{=} \mathrm{DC} + \Diamond_l + \Diamond_r$.

Iteration: DC's chop-based Form of Kleene Star is the Natural Counterpart of chop-star Too

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

Positive iteration A^+ and *iteration* are interdefinable:

$A^+ \mathrel{\hat{=}} A; (A^*)$, $\models A^* \Leftrightarrow \lceil\rceil \vee A^+$.

$\mathrm{DC}^* \mathrel{\hat{=}} \mathrm{DC} + \mathit{iteration}$.

DC-NL* $\mathrel{\hat{=}} \mathrm{DC} + \Diamond_l + \Diamond_r + \mathit{iteration}$.

Separation in DC-NL and DC-NL*

DC-NL (resp. DC-NL*) introspective, future and past formulas are like in ITL:

$$C ::= \mathit{false} \mid \lceil\rceil \mid \lceil S \rceil \mid C \Rightarrow C \mid C;C \mid C^*$$

$$P ::= C \mid \neg P \mid P \vee P \mid \Diamond_l P, \quad F ::= C \mid \neg F \mid F \vee F \mid \Diamond_r F.$$

Strict Forms of Future and Past Formulas Are DC-Specific

A strictly past (strictly future) formula is a boolean combination of $\Diamond_l$ ($\Diamond_r$) formulas whose operands are non-strictly past (non-strictly future):

$$SP ::= \Diamond_l P \mid SP \Rightarrow SP \qquad SF ::= \Diamond_l F \mid SF \Rightarrow SF$$

$\lceil S \rceil$ is not affected by varying $I_t(S)$ at single time instants, such as the midpoint in DC's chop. Given $I : V \times \mathbb{R} \to \{0, 1\}$,

$I, [a, b] \models C$ is a condition on $I_{V \times [a,b]}$.

$I, [a, b] \models SF$ is a condition on $I|_{V \times [b,+\infty)}$

$I, [a, b] \models SP$ is a condition on $I|_{V \times (-\infty,a]}$

The Separation Theorem for DC-NL and DC-NL*

A separated formula A is a boolean combination of strictly past, strictly future and introspective formulas:

$$A ::= C \mid SP \mid SF \mid A \Rightarrow A$$

In separated formulas,

$\Diamond_l$ is not allowed in the scope of chop, iteration and $\Diamond_r$;

$\Diamond_r$ is not allowed in the scope of chop, iteration and $\Diamond_l$.

Theorem: Every formula in the $\lceil P \rceil$-subset of DC-NL (DC-NL*) is equivalent to a separated formula in the $\lceil P \rceil$-subset of DC-NL (DC-NL*).

The Companion Result: Expressive Completeness [Rabinovich, LICS 2000]

The LTL prototype is known to be related with expressive completeness.

The same subset of DC was proven expressive complete by Rabinovich wrt a corresponding monadic second order theory. (LTL's is first order.)

In principle, a proof of separation using expressive completeness is doable in this setting.

Such a proof seems to be no less trivial than the one on the example of the discrete time ITL proof. It may as well be publishable. . .

The Proof: A collection of valid equivalences to apply as transformation rules!

Two collections of equivalences:

for the particular cases of extracting $\Diamond_l$, $\Diamond_r$ from the scope of other operators, and

for a transformation that recurs in the them:

$A_1, \dots, A_n$ is a full system, if $\models \bigvee\limits_{k=1}^{n} A_k$ and $\models \neg(A_{k_1} \wedge A_{k_2})$ for $k_1 \neq k_2$.

The Key Lemma. Let A be a $\lceil P \rceil$-formula in DC (DC^*). Then there exists an $n < \omega$ and some DC (DC^*) $\lceil P \rceil$-formulas A_k, A'_k, $k = 1, \dots, n$, such that $A_1, \dots, A_n$ is a full system and

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

Let $h_*(A)$ be the *-height of A. Then, furthermore, $h_*(A_k) \leq h_*(A)$ and $h_*(A'_k) \leq h_*(A)$.

Proof of the Key Lemma

$\models \bot \;\Leftrightarrow\; (\top;\bot)$ $\qquad\qquad \models \lceil\rceil \;\Leftrightarrow\; (\lceil\rceil;\lceil\rceil) \vee (\neg\lceil\rceil;\bot)$

$\models \lceil P\rceil \;\Leftrightarrow\; (\lceil P\rceil;(\lceil P\rceil \vee \lceil\rceil)) \vee (\lceil\rceil;\lceil P\rceil) \vee (\neg(\lceil\rceil \vee \lceil P\rceil);\bot)$

Let $B_1,\ldots,B_n$, $B'_1,\ldots,B'_n$, $C_1,\ldots,C_m$, $C'_1,\ldots,C'_m$ satisfy (1) for B and C, respectively. Then:

$$\models B \mathrel{op} C \Leftrightarrow \bigvee_{k=1}^{n}\bigvee_{l=1}^{m}(B_k \wedge C_l);(B'_k \mathrel{op} C'_l),\;\; op \in \{\Rightarrow,\vee,\wedge,\Leftrightarrow\}$$

$$\models B;C \Leftrightarrow \bigvee_{\substack{k=1,\ldots,n\\ X\subseteq\{1,\ldots,m\}}} \left(B_k \wedge \bigwedge_{l\in X}(B;C_l) \wedge \bigwedge_{l\notin X}\neg(B;C_l)\right);\left((B'_k;C) \vee \bigvee_{l\in X} C'_l\right)$$

For the equivalence about iteration, let $C_1,\ldots,C_m$, and $C'_1,\ldots,C'_m$ satisfy (1) for $C \mathrel{\hat{=}} B \vee \lceil\rceil$. Then $B^* \Leftrightarrow C^*$, and:

$$\models B^* \;\;\Leftrightarrow\;\; \bigvee_{X\subseteq\{1,\ldots,m\}} \left(\bigwedge_{l\in X}(B^*;C_l) \wedge \bigwedge_{l\notin X}\neg(B^*;C_l)\right);\left(\bigvee_{l\in X}(C'_l;B^*)\right)$$

Mirror Statements

All the technicalities in the proof come in pairs: along with every statement, its time mirror holds too.

The validity of the time mirrors of valid statements follows from the time symmetry in the semantics of chop, iteration, $\Diamond_l$ and $\Diamond_r$.

Mirror statements are obtained by

exchanging the operands of chop;

replacing $\Diamond_l$ by $\Diamond_r$ and vice versa.

E.g., the mirror statement of the Key Lemma is

Mirror Key Lemma. Let A be a $\lceil P \rceil$-formula in DC (DC^*). Then there exists an $n < \omega$ and some DC (DC^*) $\lceil P \rceil$-formulas A_k, A'_k, $k = 1, \ldots, n$, such that $A_1, \ldots, A_n$ is a full system and

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

Separating $\Diamond_l$-formulas

Consider $\Diamond_l A$, where A is already separated.

A can be assumed to be in DNF.

Since

$$\models \Diamond_l(A_1 \vee A_2) \Leftrightarrow \Diamond_l A_1 \vee \Diamond_l A_2,$$

A can be assumed to be a conjunction of possibly negated non-strictly past formulas P and strictly future formulas $\varepsilon_k \Diamond_r F_k$. We have

$$\models \Diamond_l\left(P \wedge \bigwedge_{k=1}^{n} \varepsilon_k \Diamond_r F_k\right) \Leftrightarrow \Diamond_l P \wedge \bigwedge_{k=1}^{n} ((\lceil\rceil \wedge \varepsilon_k \Diamond_r F_k); \top)\ .$$

Hence separating $\Diamond_l A$ boils down to separating the chop formulas $((\lceil\rceil \wedge \varepsilon \Diamond_r F_k); \top)$.

Separating chop-formulas

Again, since $\models (L_1 \vee L_2); R \Leftrightarrow (L_1; R) \vee (L_2; R)$ and

$$\models L; (R_1 \vee R_2) \Leftrightarrow (L; R_1) \vee (L; R_2),$$

we need to do only conjunctions of introspective formulas and possibly negated past $\Diamond_l$-formulas or future $\Diamond_r$-formulas.

Past $\Diamond_l$-formulas (future $\Diamond_r$-formulas) can be extracted from the left (right) operand of chop using

$\models (L \wedge \varepsilon\Diamond_l P); R \Leftrightarrow (L; R) \wedge \varepsilon\Diamond_l P$ and $\models L; (R \wedge \varepsilon\Diamond_r F) \Leftrightarrow (L; R) \wedge \varepsilon\Diamond_r F$.

It remains to do $(L \wedge \bigwedge_{k=1}^{n} \varepsilon_k \Diamond_r F_k); R$.

The mirror transformations work for $P; (R \wedge \bigwedge_{k=1}^{n} \varepsilon_k \Diamond_l P_k)$.

Separating $(P \wedge \bigwedge_{k=1}^{n} \varepsilon_k \Diamond_r F_k); R$

Consider $(L \wedge \varepsilon \Diamond_r F); R$ where $\varepsilon \Diamond_r F \mathrel{\hat{=}} \varepsilon_1 \Diamond_r F_1$ and $L \mathrel{\hat{=}} P \wedge \bigwedge_{k=2}^{n} \varepsilon_k \Diamond_r F_k$.

Again F of $\varepsilon \Diamond_r F$ can be assumed to be a conjunct (of a DNF).

Let F be $C \wedge G$ where C is introspective and G is strictly future.

Let C_k, C'_k, $k = 1, \ldots, n$, satisfy the Key Lemma for C. Then

$$\models (L \wedge \Diamond_r(\underbrace{C \wedge G}_{=F})); R \Leftrightarrow (L; (R \wedge (\underbrace{C \wedge G}_{=F}; true))) \vee \bigvee_{k=1}^{n} (L; (R \wedge C_k)) \wedge \Diamond_r(C'_k \wedge G)$$

$$\models (L \wedge \neg\Diamond_r(\underbrace{C \wedge G}_{=F})); R \Leftrightarrow \bigvee_{k=1}^{n} (L; (R \wedge C_k \wedge \neg((\underbrace{C \wedge G}_{=F}); true))) \wedge \neg\Diamond_r(C'_k \wedge G).$$

To finish the separation, the blue occurrences of G must be extracted from the scope of chop. This is possible because G's $\Diamond_r$-height is lower than F's.

Separating iteration formulas in DC-NL*

Separating iteration formulas in DC-NL* can be done using

(1) quantification over state in DC

and

(2) the fact that quantification over state can be eliminated in the $\lceil P \rceil$-subset of DC.

$I, [a, b] \models \exists\, P\, A$ 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 $\lceil P \rceil$-subset of DC^*:

Theorem: For every $\lceil P \rceil$-formula A in DC^* and every state variable P there exists a (quantifier-free) $\lceil P \rceil$-formula B in DC^* such that $\models B \Leftrightarrow \exists\, P\, A$.

Importantly, B is not guaranteed to be iteration-free, even in case A is.

However introducing fresh occurrences of iteration upon quantifier elimination is used if iteration already occurs in the formula to be separated.

Extracting $\Diamond_l$- and $\Diamond_r$-formulas from the scope of iteration

Let B of B^* be $\bigvee_{s=1}^{t} B_s$ where $B_s \mathrel{\hat{=}} H_s \wedge \bigwedge_{i=1}^{u} \varepsilon^p_{s,i} \Diamond_l P_i \wedge \bigwedge_{j=1}^{v} \varepsilon^f_{s,j} \Diamond_r F_j$.

Then B^* is equivalent to

$$\exists T \exists S^p_1 \dots S^p_u \exists S^f_1 \dots S^f_v \left((\lceil T \rceil ; \lceil \neg T \rceil) \wedge \bigvee_{s=1}^{t} \left(B_s \wedge \bigwedge_{i=1}^{u} \lceil \varepsilon^p_{s,i} S^p_i \rceil \wedge \bigwedge_{j=1}^{v} \lceil \varepsilon^f_{s,j} S^f_j \rceil \right) \right)^*$$

The satisfying assignment of $T, S^p_1, \dots, S^p_u, S^f_1, \dots, S^f_v$ is such that

(1) the left endpoints of the maximal $T \wedge \varepsilon^p_{s,i} S^p_i$-subintervals are the left endpoints of the intervals which must satisfy $\varepsilon^p_{s,i} \Diamond_l P_i$ for B_s to hold,

and

(2) the right endpoints of the maximal $\neg T \wedge \varepsilon^f_{s,j} S^f_j$-subintervals are the right endpoints of the intervals which must satisfy $\varepsilon^f_{s,i} \Diamond_r F_j$ for B_s to hold.

Separating iteration formulas in DC-NL*

The correspondence between the assignments of $\Diamond_r F_j$, and T and S_j^f can be expressed by the formulas

$$\varphi_j \mathrel{\hat{=}} \left(\begin{array}{l} (\mathit{true}; \lceil S_j^f \rceil) \Rightarrow \Diamond_r F_j \wedge \neg((\mathit{true}; \lceil S_j^f \wedge \neg T \rceil); ((\lceil T \rceil; \mathit{true}) \wedge \neg((\Diamond_r F_j \wedge \lceil\rceil); \mathit{true}))) \wedge \\ (\mathit{true}; \lceil \neg S_j^f \rceil) \Rightarrow \neg \Diamond_r F_j \wedge \neg((\mathit{true}; \lceil \neg S_j^f \wedge \neg T \rceil); ((\lceil T \rceil; \mathit{true}) \wedge ((\Diamond_r F_j \wedge \lceil\rceil); \mathit{true}))) \end{array} \right)$$

and their past mirrors π_i, for the correspondence between $\Diamond_l F_i$, and T and S_i^p. Hence B^* is equivalent to

$$\exists T \exists S_1^p \ldots \exists S_u^p \exists S_1^f \ldots \exists S_v^f \left(\begin{array}{l} \left((\lceil T \rceil; \lceil \neg T \rceil) \wedge \bigvee\limits_{s=1}^{t} H_s \wedge \bigwedge\limits_{i=1}^{u} \lceil \varepsilon_{s,i}^p S_i^p \rceil \wedge \bigwedge\limits_{j=1}^{v} \lceil \varepsilon_{s,j}^f S_j^f \rceil \right)^* \\ \wedge \bigwedge\limits_{i=1}^{u} \pi_i \wedge \bigwedge\limits_{j=1}^{v} \varphi_j \end{array} \right).$$

The separation procedure can now be concluded by

– separating π_i and φ_j;

– taking the $\Diamond_l$- and the $\Diamond_r$-subformulas of the separated equivalents of π_i and φ_j out of the scope of the quantifier prefix;

– eliminating the quantifier prefix from the remaining introspective formula.

The End