Logique du temps linéaire et théorème de Kamp

Peter Schmitt (trad. J. Goubault-Larrecq)

28 avril 2004

Résumé

Note du traducteur : ces notes sont une traduction du début du chapitre 4 d'un texte sur les logiques non classiques dû au professeur Peter H. Schmitt (*Nichtklassische Logiken*, mars 2004, `http://i12www.ira.uka.de/~pschmitt/NKL/NKLSkript.ps`), de l'université de Karlsruhe, que j'ai eu l'occasion de lire en 1996. J'ai rajouté à l'occasion des définitions en notes en bas de pages, ou des commentaires. — Jean Goubault-Larrecq.

1 Logique temporelle

1.1 Logiques temporelles linéaires

Si l'on comprend les mondes d'un cadre de Kripke (G, R)[1] comme les états d'une portion du monde à différents instants, et R comme une relation de prédécesseur, on a obtenu une logique temporelle simple, sans changer notre syntaxe et notre sémantique précédentes[2]. La relation $R(s, t)$ sera ainsi lue *s se trouve avant t dans le temps* et nous écrivons $s < t$ au lieu de $R(s, t)$. Si la relation de satisfaction $(\mathcal{K}, t) \models A$ est vraie pour $\mathcal{K} = (G, <, v)$ et $t \in G$, alors nous disons *la formule A est vraie à l'instant t*. En particulier on lira $(\mathcal{K}, t) \models \Diamond A$: *A sera vraie à un instant futur* et de même $(\mathcal{K}, t) \models \Box A$ *A est vraie à tous les instants futurs*. Nous voulons rendre visibles ces changements de signification aussi dans la notation et nous écrivons $\mathsf{sif}\ A$ ("sometime in the future") au lieu de $\Diamond A$ et $\mathsf{aif}\ A$ ("always in the future") pour $\Box A$. Ce que nous appelions jusqu'ici un cadre s'appelle maintenant une structure temporelle et les structures de Kripke s'appellent maintenant les structures temporelles. Différentes structures temporelles $(G, <)$ peuvent être intéressantes pour différents domaines d'application. On doit décider si le temps a un instant initial ou s'il doit s'étendre indéfiniment dans le passé, si les instants se suivent dans un ordre discret ou bien forment un espace continu dense, s'il n'y a qu'un futur ou bien si les branchements sont autorisés. Nous suivons ici la classification habituelle en logiques temporelles linéaires et logiques temporelles branchantes ("branching time temporal logics").

[1] Il s'agit d'un ensemble G, dit de mondes, et d'une relation binaire entre mondes, dite d'accessibilité (NdT).

[2] Soient les formules $A ::= X|A \vee A|A \wedge A|\neg A|\Diamond A|\Box A$. La sémantique de Kripke de ces formules est donnée par $w \models A$ (A est vraie au monde w, sous-entendu dans l'univers (G, R) et considérant la valuation v qui à chaque formule atomique X associe l'ensemble des mondes où elle est vraie), définie comme suit : $w \models X$ si et seulement si $w \in v(X)$; $w \models A \vee A'$ si et seulement si $w \models A$ ou $w \models A'$; $w \models A \wedge A'$ si et seulement si $w \models A$ et $w \models A'$; $w \models \neg A$ si et seulement si $w \not\models A$; $w \models \Diamond A$ si et seulement s'il existe w' R w tel que $w' \models A$; $w \models \Box A$ si et seulement si, pour tout w' R w, on a $w' \models A$. P.H. Schmitt note $R(s, t)$ au lieu de s R t, et si $\mathcal{K} = (G, R, v)$, il écrit $(\mathcal{K}, w) \models A$ au lieu de $w \models A$, ce qui est plus précis (NdT).

1.1.1 Introduction

Nous voulons d'abord ici exiger juste que $(G, <)$ soit un ordre partiel strict.

Lorsque l'on regarde de plus près cette logique temporelle simple, on constate cependant un manque décisif : on ne peut pas y exprimer toutes les relations temporelles dont on souhaiterait parler. Par exemple il n'y a aucune possibilité de parler du passé. Ceci se corrige en introduisant de nouveaux opérateurs temporels sip et aip avec le sens de "sometime in the past" et "always in the past".

Définition 1 (Opérateurs du passé) *Pour une structure temporelle $\mathcal{K} = (G, <, v)$ et $t \in G$ on pose :*

$(\mathcal{K}, t) \models \mathsf{sip}\ A$	*ssi*	*il existe $s \in G$ avec $s < t$ et $(\mathcal{K}, s) \models A$*
$(\mathcal{K}, t) \models \mathsf{aip}\ A$	*ssi*	*pour tout $s \in G$ avec $s < t$ on a $(\mathcal{K}, s) \models A$*

D'autres opérateurs temporels souvent utilisés sont les opérateurs à deux places until et since, où $A\ \mathsf{until}\ B$ signifie qu'à un moment quelconque du futur B sera vrai et que jusque là A est vrai. De façon duale, $A\ \mathsf{since}\ B$ est vrai quand il y a un instant dans le passé où B était vrai et que depuis et jusqu'à maintenant A était vrai.

Définition 2 (until et since) *Soit $\mathcal{K} = (G, <, v)$ une structure temporelle et $t \in G$.*

$(\mathcal{K}, t) \models A\ \mathsf{until}\ B$	*ssi il existe $s \in G$ tel que $t < s$, $(\mathcal{K}, s) \models B$, et pour tout r avec $t < r < s$, $(\mathcal{K}, r) \models A$*
$(\mathcal{K}, t) \models A\ \mathsf{since}\ B$	*ssi il existe $s \in G$ tel que $s < t$, $(\mathcal{K}, s) \models B$, et pour tout r avec $s < r < t$, $(\mathcal{K}, r) \models A$*

À côté de l'écriture infixe ($p\ \mathsf{until}\ q$) que nous avons choisie, on trouve aussi souvent l'écriture préfixe $\mathsf{until}\ (q, p)$. On constante que l'ordre des arguments s'en trouve changé, pour s'accorder avec la sémantique anglophone.

Définition 3 (Formules LTL) *Les formules de la logique du temps linéaire, LTL (linear temporal logic), sont définies récursivement par :*

1. *Les constantes logiques* true *et* false *sont des formules LTL,*
2. *toute variable propositionnelle est une formule LTL,*
3. *si F, G sont des formules LTL, alors leurs combinaisons propositionnelles $\neg F$, $F \vee G$, $F \wedge G$, etc., sont des formules LTL,*
4. *si F et G sont des formules LTL, alors $F\ \mathsf{since}\ G$ et $F\ \mathsf{until}\ G$ sont des formules LTL.*

Les opérateurs temporels sif, aif, sip, aip que nous venions d'introduire ne sont pas explicitement présents en LTL, car ils y sont définissables.

Lemme 4 *Les équivalences suivantes sont vraies dans toute structure temporelle :*

1. $\mathsf{sif}\ A \longleftrightarrow \mathsf{true\ until}\ A$
2. $\mathsf{aif}\ A \longleftrightarrow \neg(\mathsf{true\ until}\ \neg A)$
3. $\mathsf{sip}\ A \longleftrightarrow \mathsf{true\ since}\ A$
4. $\mathsf{aip}\ A \longleftrightarrow \neg(\mathsf{true\ since}\ \neg A)$

Démonstration. Facile. □

Lemme 5 *Les équivalences suivantes sont vraies dans toutes les structures temporelles linéaires :*

1. $C\ \mathsf{until}\ (A \vee B) \longleftrightarrow (C\ \mathsf{until}\ A) \vee (C\ \mathsf{until}\ B)$
2. $C\ \mathsf{since}\ (A \vee B) \leftrightarrow (C\ \mathsf{since}\ A) \vee (C\ \mathsf{since}\ B)$
3. $(B \wedge C)\ \mathsf{until}\ A \longleftrightarrow (B\ \mathsf{until}\ A) \wedge (C\ \mathsf{until}\ A)$
4. $(B \wedge C)\ \mathsf{since}\ A \leftrightarrow (B\ \mathsf{since}\ A) \wedge (C\ \mathsf{since}\ A)$

Démonstration. Facile. □

On peut ajouter une multitude de variations des opérateurs temporels considérés jusqu'ici, et tous seraient vraisemblablement déjà définis d'une façon ou d'une autre. Nous prenons juste quelques exemples. Dans la définition de $\mathsf{aif}\ A$, l'instant de référence était exclu. L'opérateur $\mathsf{fno}\ A$ ("from now on A") est défini par la définition sémantique

$$(\mathcal{K}, t) \models \mathsf{fno}\ A \text{ ssi pour tout } s \in G \text{ avec } t \leq s \text{ on a } (\mathcal{K}, s) \models A.$$

On peut modifier de même les opérateurs sif, aip et sip de sorte que l'instant de référence soit inclus dans le domaine de validité du quantificateur universel, resp. existentiel. On considérera aussi souvent les affaiblissements until_0 et since_0, avec la sémantique :

$$\begin{array}{ll}(\mathcal{K}, t) \models A\ \mathsf{until}_0\ B & \text{ssi pour tout } s \in G \\ & \text{avec } t < s \\ & \text{on a } (\mathcal{K}, s) \models \neg B, \\ & \quad \text{ou bien} \\ & \text{il existe } s \in G \\ & \text{tel que } t < s,\ (\mathcal{K}, s) \models B, \\ & \text{et pour tout } r \text{ avec } t < r < s, \\ & (\mathcal{K}, r) \models A \\ (\mathcal{K}, t) \models A\ \mathsf{since}\ B & \text{ssi pour tout } s \in G \\ & \text{avec } s < t \\ & \text{on a } (\mathcal{K}, s) \models \neg B, \\ & \quad \text{ou bien} \\ & \text{il existe } s \in G \\ & \text{tel que } s < t,\ (\mathcal{K}, s) \models B, \\ & \text{et pour tout } r \text{ avec } s < r < t, \\ & (\mathcal{K}, r) \models A\end{array}$$

Il est presque superflu de dire que toutes ces variantes sont définissables en LTL.

Une autre variante de l'opérateur until consiste en

$$(\mathcal{K}, t) \models A\ \mathsf{until}^1\ B \text{ ssi il existe } s \in G \text{ avec } t \leq s, (\mathcal{K}, s) \models B, \text{ et pour tout } r \text{ avec } t \leq r < s \text{ on a } (\mathcal{K}, r) \models A.$$

Un autre opérateur important est l'opérateur "next", souvent écrit $\bigcirc$. Il suppose que pour chaque instant t de la structure temporelle $\mathcal{K}$ il existe exactement un successeur. Nous écrivons alors $t + 1$ cet instant suivant.

$$(\mathcal{K}, t) \models \bigcirc A \text{ ssi } (\mathcal{K}, t + 1) \models A$$

On a l'équivalence

$$\bigcirc A \longleftrightarrow \mathsf{false\ until}\ A$$

Attention, cette définition n'est plus valable pour until^1 au lieu de until.

Définition 6 *Une structure temporelle $\mathcal{K}$ dans laquelle tout instant t a un unique successeur s'appelle une structure temporelle* discrète.

Références : le chapitre de Wolper [Wol95] dans le livre [BS95].

1.1.2 Pouvoir d'expression

Bien que l'on puisse exprimer la plupart des relations temporelles en LTL, il existe des propositions relativement simples qui ne sont pas formalisables en LTL. Nous présentons ici un résultat de [Wol83][3]

Lemme 7 *Soit p une variable propositionnelle. On ne peut décrire la formule*

pour chaque instant i avec $i = km$, $k \geq 0$, on a p

en LTL pour aucun $m \geq 2$.

[3] La démonstration est valable uniquement en l'absence d'opérateurs du passé... nous laissons le soin au lecteur d'adapter la démonstration aux opérateurs du passé ; une indication : un résultat classique énonce que pour toute formule F de LTL, il existe une formule G sans opérateur du passé telle que $(\mathcal{K}, 0) \models F$ ssi $(\mathcal{K}, 0) \models G$ pour tout structure temporelle discrète $\mathcal{K}$ ayant un instant initial 0 (NdT).

Démonstration. Nous avons besoin pour cette démonstration du résultat du lemme 8 qui suit.

Nous choisissons un k fixé, $k \geq 2$. Supposons qu'il existe une formule A qui, dans toute structure temporelle $\mathcal{K} = (\mathbb{N}, <, v)$, n'est vraie que quand $(\mathcal{K}, i) \models p$ pour tout instant i de la forme $i = km$. Soit n le nombre d'opérateurs until dans A. Par le lemme 8 il existe k avec $km - 1 > n$ tel que

$$(\mathcal{K}_{km}, 0) \models A \Leftrightarrow (\mathcal{K}_{km-1}, 0) \models A$$

Dans $\mathcal{K}_{km}$ cependant, la condition définissant A est satisfaite mais pas dans $\mathcal{K}_{km-1}$. Contradiction ! $\square$

Lemme 8 *Par* $\mathcal{K}_i = (\mathbb{N}, <, v_i)$ *nous désignons la structure temporelle ayant* $(\mathbb{N}, <)$ *comme ensemble d'instants et pour* $t \in \mathbb{N}$ *:*

$$v_i(p, t) = \begin{cases} 1 & si\ t \leq i \\ 0 & si\ t = i + 1 \\ 1 & si\ t > i + 1 \end{cases}$$

Soit A une formule LTL arbitraire et n le nombre d'occurrences de l'opérateur until *dans A. Alors soit*

$$(\mathcal{K}_i, 0) \models A \textit{ pour tout } i \geq n + 1$$

soit

$$(\mathcal{K}_i, 0) \models \neg A \textit{ pour tout } i \geq n + 1.$$

En d'autres termes pour $i > n$ la valeur de vérité de $(\mathcal{K}_i, 0) \models A$ ne dépend pas de i.

Démonstration. La démonstration se déroule par récurrence structurelle sur la complexité de la formule A[4]. Nous distinguons les cas suivants.

Cas de base : $A = p$. Pour tout i on a $(\mathcal{K}_i, 0) \models p$ ssi $v_i(p, 0) = 1$. Mais c'est le cas pour tout i.

[4]... par récurrence structurelle sur A (NdT).

Premier cas de récurrence : A est une combinaison booléenne de A_1, A_2. Conséquence facile de l'hypothèse de récurrence.

Second cas de récurrence : $A = B \ \mathsf{until}\ C$. Nous remarquons d'abord que pour toute structure temporelle discrète $\mathcal{K}$, et pour tout instant z de $\mathcal{K}$, on a $(\mathcal{K}, z) \models B \ \mathsf{until}\ C$ ssi

$$(\mathcal{K}, z + 1) \models C \text{ ou } (\mathcal{K}, z + 1) \models B \wedge (B \ \mathsf{until}\ C).$$

Nous avons ensuite besoin de l'observation que pour les structures temporelles spéciales $\mathcal{K}_i$, $0 \leq t \leq n$, $i > n$ et des formules F parfaitement arbitraires F on a

$$(\mathcal{K}_i, t) \models F \text{ ssi } (\mathcal{K}_{i-t}, 0) \models F.$$

Maintenant nous pouvons commencer la démonstration de l'étape de récurrence. Pour $i \geq n + 1$, nous obtenons par l'application répétée de la première observation : $(\mathcal{K}_i, 0) \models B \ \mathsf{until}\ C$ ssi

$$\begin{array}{lc} & (\mathcal{K}_i, 1) \models C \\ \text{ou} & (\mathcal{K}_i, 1) \models B \text{ et } (\mathcal{K}_i, 2) \models C \\ \text{ou} & (\mathcal{K}_i, 1) \models B \text{ et } (\mathcal{K}_i, 2) \models B \text{ et } (\mathcal{K}_i, 3) \models C \\ \text{ou} & \dots \\ \text{ou} & (\mathcal{K}_i, 1) \models B \text{ et} \\ & (\mathcal{K}_i, 2) \models B \text{ et} \\ & \dots \\ & (\mathcal{K}_i, i - n) \models B \text{ et} \\ & (\mathcal{K}_i, i - n) \models B \ \mathsf{until}\ C \end{array}$$

Par la deuxième observation nous pouvons transformer ceci en la formulation équivalente : $(\mathcal{K}_i, 0) \models B \ \mathsf{until}\ C$ ssi

$$\begin{array}{lc} & (\mathcal{K}_{i-1}, 0) \models C \\ \text{ou} & (\mathcal{K}_{i-1}, 0) \models B \text{ et } (\mathcal{K}_{i-2}, 0) \models C \\ \text{ou} & (\mathcal{K}_{i-1}, 0) \models B \text{ et } (\mathcal{K}_{i-2}, 0) \models B \text{ et } (\mathcal{K}_{i-3}, 0) \models C \\ \text{ou} & \dots \end{array}$$

$$
\begin{array}{lc}
\text{ou} & (\mathcal{K}_{i-1}, 0) \models B \text{ et} \\
& (\mathcal{K}_{i-2}, 0) \models B \text{ et} \\
& \dots \\
& (\mathcal{K}_n, 0) \models B \text{ et} \\
& (\mathcal{K}_n, 0) \models B \text{ until } C
\end{array}
$$

Par hypothèse de récurrence il s'ensuit : $(\mathcal{K}_i, 0) \models B$ until C ssi

$$
\begin{array}{lc}
& (\mathcal{K}_n, 0) \models C \\
\text{ou} & (\mathcal{K}_n, 0) \models B \text{ et } (\mathcal{K}_n, 0) \models C \\
\text{ou} & (\mathcal{K}_n, 0) \models B \text{ et } (\mathcal{K}_n, 0) \models B \text{ et } (\mathcal{K}_n, 0) \models C \\
\text{ou} & \dots \\
\text{ou} & (\mathcal{K}_n, 0) \models B \text{ et} \\
& (\mathcal{K}_n, 0) \models B \text{ et} \\
& \dots \\
& (\mathcal{K}_n, 0) \models B \text{ et} \\
& (\mathcal{K}_n, 0) \models B \text{ until } C
\end{array}
$$

Ainsi on a montré que $(\mathcal{K}_i, 0) \models B$ until C ne dépend pas de i pour $i \geq n + 1$. □

1.1.3 Définissabilité

Il est plus intéressant de se demander s'il existe ne serait-ce qu'un opérateur temporel qui ne soit pas définissable en LTL. Ce n'est pas encore une question précise car nous n'avons pas défini ce que doit être en général un opérateur temporel. On peut concrétiser la question de la complétude définitionnelle de LTL en faisant le détour par la réduction de formules LTL en formules de la logique des prédicats. Le vocabulaire V_{temp} dans lequel les formules traduites vont se trouver contient, pour chaque variable propositionnelle p, un symbole de relation unaire que nous écrirons avec la même lettre, donc $p(x)$, et à part cela un symbole de relation binaire $<$ pour la relation d'ordre.

Définition 9 (Traduction en PL1) *La fonction de traduction μ des formules LTL en formules de logique des prédicats dans $Fml(V_{temp})$ est définie récursivement par :*

1. $\mu(\mathsf{true}) = A(x) \vee \neg A(x)$ *et* $\mu(\mathsf{false}) = A(x) \wedge \neg A(x)$ *pour un prédicat unaire arbitraire $A(x)$;*
2. $\mu(p) = p(x)$ *pour tout variable propositionnelle p,*
3. $\mu(\neg F) = \neg\mu(F)$, $\mu(F \vee G) = \mu(F) \vee \mu(G)$, $\mu(F \wedge G) = \mu(F) \wedge \mu(G)$, *etc.,*
4. $\mu(F \text{ since } G) = \exists y \cdot y < x \wedge \mu(G)(y/x) \wedge \forall z \cdot y < z < x \Rightarrow \mu(F)(z/x)$[5]
5. $\mu(F \text{ until } G) = \exists y \cdot x < y \wedge \mu(G)(y/x) \wedge \forall z \cdot x < z < y \Rightarrow \mu(F)(z/x)$.

On observe que pour chaque formule LTL F, la traduction $\mu(F)$ contient exactement x comme variable libre.

Lemme 10 *Pour une structure temporelle $\mathcal{K} = (G, <, v)$, soit $\mathcal{K}_{temp}$ la structure de logique des prédicats $(G, <)$ telle quelle, et dans laquelle la relation unaire $A(s)$ est vraie si et seulement si $v(s, A) = 1$*[6]*. Alors, pour toute formule LTL F et tout $t \in G$:*

$$(\mathcal{K}, t) \models F \iff \mathcal{K}_{temp} \models \mu(F)[t]$$

La notation $\mu(F)[t]$ signale que t est utilisé comme dénotation de l'(unique) variable libre x dans $\mu(F)$.

Démonstration. Calcul facile. □

Nous pouvons maintenant concrétiser la notion de complétude définitionnelle de LTL comme suit :

[5] On note ici $\mu(G)(y/x)$ pour $\mu(G)[x := y]$, et $y < z < x$ pour $y < z \wedge z < x$ (NdT).

[6] Ce que j'avais noté $s \in v(A)$ dans ma première note en bas de page (NdT).

Définition 11 (Complétude définitionnelle) *Soit $\mathcal{M}$ une classe de structures temporelles. On dit que LTL est définitionnellement complète par rapport à $\mathcal{M}$ si pour toute formule $F(x)$ de logique des prédicats sur la signature V_{temp} avec exactement une variable libre x, il existe une formule LTL A telle que pour tout $(G, <) \in \mathcal{M}$ et toute valuation v on a :*

$$(G, <, v)_{temp} \models \forall x \cdot \mu(A) \Leftrightarrow F(x).$$

Théorème 12 *1. LTL est définitionnellement complète pour la structure temporelle $(\mathbb{Z}, <)$ des entiers relatifs.*

2. LTL est définitionnellement complète pour la structure temporelle $(\mathbb{R}, <)$ des nombres réels.

3. LTL n'est pas définitionnellement complète pour la structure temporelle $(\mathbb{Q}, <)$ des nombres rationnels.

4. LTL n'est pas définitionnellement complète pour la class $\mathcal{M}_{lin}$ de toutes les structures linéaires.

Démonstration. Les démonstrations sont pour certaines très compliquées. La meilleure référence est [GHR94]. Le pionnier du domaine de la définissabilité d'opérateurs temporels est [Kam68]. Pour convoyer une impression de la sorte de plan de démonstration et des sous-problèmes qu'il faudra alors résoudre, nous voulons au moins traiter plus précisément du plus simple des cas positifs de définissabilité sur les nombres entiers. Nous suivons la présentation de [GHR94]. Nous rassemblons d'abord quelques équivalences qui nous serons utiles plus tard.

Lemme 13 *Les équivalences suivantes sont valides dans la structure temporelle des entiers relatifs :*

1. $\neg(B \ \mathsf{until}\ A) \longleftrightarrow (\mathsf{aif}\ \neg A) \vee (\neg A \ \mathsf{until}\ (\neg A \wedge \neg B))$
2. $\neg(B \ \mathsf{since}\ A) \longleftrightarrow (\mathsf{aip}\ \neg A) \vee (\neg A \ \mathsf{since}\ (\neg A \wedge \neg B))$
3. $\neg(B \ \mathsf{until}\ A) \longleftrightarrow (\mathsf{aif}\ \neg A) \vee ((\neg A \wedge B) \ \mathsf{until}\ (\neg A \wedge \neg B))$
4. $\neg(B \ \mathsf{since}\ A) \longleftrightarrow (\mathsf{aip}\ \neg A) \vee ((\neg A \wedge B) \ \mathsf{since}\ (\neg A \wedge \neg B))$.

Démonstration. Il suffit de démontrer 3 et 4, puisque 1 et 2 en sont des cas particuliers. Nous montrons 3. Par définition, $t \models \neg(B \ \mathsf{until}\ A)$ est équivalent à

Pour tout $s > t$ tel que $s \models A$ il existe s', $s > s' > t$ avec $s' \models \neg B$.

Nous voulons nous convaincre que le côté droit de 3 s'ensuit. Si aif $\neg A$ est vrai, c'est fini. Sinon il existe $s > t$ avec $s \models A$. Nous choisissons le plus petit de ces s, autrement dit pour tout s' avec $s > s' > t$ on a $s' \models \neg A$. Par hypothèse il existe un s' avec $s > s' > t$ tel que $s' \models \neg B$. Nous choisissons de nouveau le plus petit s'. Ainsi on a $s' \models \neg A \wedge \neg B$ et pour tout t' avec $s' > t' > t$ on a $t' \models B \wedge \neg A$. Mais ceci veut dire $t \models (\mathsf{aif}\ \neg A) \vee ((\neg A \wedge B) \ \mathsf{until}\ (\neg A \wedge \neg B))$.

Supposons réciproquement que le côté droit de 3 soit vrai à l'instant t. Si $t \models \mathsf{aif}\ \neg A$, alors visiblement on a aussi $t \models \neg(B \ \mathsf{until}\ A)$. Sinon il existe $s > t$ avec $s \models (\neg A \wedge \neg B)$ et pour tout s' avec $s > s' > t$ on a $s' \models (\neg A \wedge B)$. Nous montrons que le côté gauche de 3 s'en déduit. Soit donc un $u > t$ donné tel que $u \models A$, alors on doit avoir $u > s$ et donc il existe $u > s' > t$ tel que $s' \models \neg B$.

La partie 4 du lemme se montre de façon analogue. □

L'idée essentielle dans la démonstration de complétude définitionnelle que nous allons considérer dans la suite est l'introduction de la notion d'une logique séparable. Cette notion n'est pas superficielle, elle n'apparaît pas immédiatement, mais à travers une analyse profonde du problème, et l'on a découvert qu'une logique temporelle est définitionnellement complète exactement quand elle possède la propriété de séparabilité. La démonstration de cette connexion forme la première étape de notre démonstration. La deuxième étape consiste maintenant, pour une logique propositionnelle donnée TL et une classe de structures temporelles $\mathcal{M}$, à établir la séparabilité. Maintenant, les détails !

Définition 14 (Formules du futur, du passé) *Soit TL une logique temporelle propositionnelle quelconque, $\mathcal{M}$ une classe de structures temporelles.*

Une formule A de TL s'appelle une formule du futur *par rapport à $\mathcal{M}$ si pour toute structure temporelle $(G, <)$ dans $\mathcal{M}$, pour tout instant $t \in G$ et pour toutes fonctions d'interprétation v_1, v_2 avec $v_1(s) = v_2(s)$ pour tout $s > t$ on a*

$$(G, <, v_1, t) \models A \textit{ ssi } (G, <, v_2, t) \models A.$$

Une formule A de TL s'appelle une formule du passé *par rapport à $\mathcal{M}$ si pour toute structure temporelle $(G, <)$ dans $\mathcal{M}$, pour tout instant $t \in G$ et pour toutes fonctions d'interprétation v_1, v_2 avec $v_1(s) = v_2(s)$ pour tout $s < t$ on a*

$$(G, <, v_1, t) \models A \textit{ ssi } (G, <, v_2, t) \models A.$$

En bref, A est une formule du futur quand sa validité au temps t ne dépend que du futur, autrement dit ne dépend que des $s > t$.

Exemple 15 *Les formules*

$$\mathsf{sif}\ p, \mathsf{aif}\ p, p\ \mathsf{until}\ q, p\ \mathsf{until}\ (q\ \mathsf{until}\ r)$$

sont des formules du futur,

$$\mathsf{sip}\ p, \mathsf{aip}\ p, p\ \mathsf{since}\ q, p\ \mathsf{since}\ (q\ \mathsf{since}\ r)$$

sont des formules du passé, et

$$p\ \mathsf{until}\ (q\ \mathsf{since}\ r), p\ \mathsf{since}\ (q\ \mathsf{until}\ r)$$

ne sont ni des formules du futur ni des formules du passé.

Définition 16 (Séparabilité) *Une logique temporelle propositionnelle TL est* séparable *par rapport à $\mathcal{M}$ lorsque toute formule A sur $\mathcal{M}$ est équivalent à une combinaison booléenne de formules du futur, de formules du passé, et de formules atomiques.*

Théorème 17 *Soit TL une logique temporelle propositionnelle, $\mathcal{M}$ une classe de structures temporelles linéaires, telle qu'au moins les opérateurs* sif *et* sip *sont définissables en TL sur $\mathcal{M}$, et telle que TL est séparable par rapport à $\mathcal{M}$. Alors TL est définitionnellement complète pour $\mathcal{M}$.*

Démonstration. Récapitulons encore ce que l'on doit montrer. Pour chaque formule $F(x)$ de la logique des prédicats du premier ordre sur le vocabulaire V_{temp} avec au plus x comme variable libre, il existe une formule A de la logique temporelle TL de sorte que pour chaque structure temporelle $\mathcal{K} = (G, <, v)$ avec $(G, <) \in \mathcal{M}$ et pour tout instant $t \in G$ on a

$$(\mathcal{K}, t) \models A \text{ ssi } \mathcal{K}_{temp} \models F(t).$$

Cette correspondance sera montrée par récurrence sur le nombre de quantificateurs imbriqués dans $F(x)$. Si $F(x)$ ne contient pas de quantificateur, alors x est l'unique variable qui apparaît dans F. $F(x)$ est aussi une combinaison booléenne de formules atomiques $Q(x)$ à une place et des formules de la forme $x = x$, $x < x$. On peut éliminer ces deux dernières, en les remplaçant par les constantes true, resp. false. On obtient maintenant la formule temporelle A en remplaçant le prédicat $Q(x)$ à une place par la variable propositionnelle Q. Jusqu'ici, c'est simple. Nous arrivons maintenant à l'**étape de récurrence** : $F(x) = \exists y \cdot H(x, y)$. Nous supposons que la variable x n'apparaît pas quantifiée dans $F(x)$, et que les sous-formules $x = x$ et $x < x$ ont été éliminées comme ci-dessus. Nous ne pouvons pas établir l'équivalence souhaitée en une seule étape, et nous montrerons d'abord un résultat un peu plus faible. Pour s'y préparer, nous étendons temporairement le vocabulaire V_{temp} en V^0_{temp} en ajoutant les symboles de relations à une place $R_=$, $R_<$, et $R_>$. Soit $H^0(x, y)$ la formule provenant de $H(x, y)$ dans laquelle pour toutes les variables z on a remplacé toutes les sous-formules $x = z$ par $R_=(z)$, toutes les sous-formules $x < z$ par $R_<(z)$ et tous les $x > z$ par $R_>(z)$. Pour une V_{temp}-structure $\mathcal{K}_{temp}$ de domaine G et $t \in G$ on définit une V^0_{temp}-structure $\mathcal{K}^t_{temp}$ par la définition[7]

$$\begin{array}{lll} \mathcal{K}^t_{temp} \models R_=(s) & \text{ssi} & s = t \\ \mathcal{K}^t_{temp} \models R_<(s) & \text{ssi} & t < s \\ \mathcal{K}^t_{temp} \models R_>(s) & \text{ssi} & t > s \end{array}$$

Alors pour tout $s \in G$ on a

$$\mathcal{K}_{temp} \models H(t, s) \text{ ssi } \mathcal{K}^t_{temp} \models H^0(t, s).$$

La variable x n'apparaît plus dans H^0 que comme argument des prédicats à une place de V_{temp}. Ainsi on peut transformer H^0 via des réécritures élémentaires en une formule logiquement équivalente de la forme

$$\bigvee_j [H_j(x) \wedge G^0_j(y)]$$

où

- chaque $H_j(x)$ est une combinaison booléenne de symboles de prédicats à une place sur le vocabulaire non étendu V_{temp},
- chaque $G^0_j(y)$ est une formule avec strictement moins de quantificateurs imbriqués que $F(x)$ et qui contient au plus y comme variable libre.

Par hypothèse de récurrence il existe des formules C_j de TL telles que pour toute structure temporelle $\mathcal{H} = (G, <, v)$ avec $(G, <) \in \mathcal{M}$ et tout instant $s \in G$ on ait

$$(\mathcal{H}, s) \models C_j \text{ ssi } \mathcal{H}_{temp} \models G^0_j(s).$$

[7]Les deux cas pour $R_<$ et $R_>$ étaient inversés dans le manuscrit original (NdT).

Clairement on a donc pour tout $t \in G$

$$\begin{gathered} (\mathcal{H}, t) \models \mathsf{sif}\ C_j \vee C_j \vee \mathsf{sip}\ C_j \text{ ssi} \\ \mathcal{H}_{temp} \models \exists y \cdot G^0_j(y). \end{gathered}$$

Comme l'on peut aussi exprimer les formules atomiques $H_j(x)$ par des formules de TL, et que l'on peut dans le cas présent tirer le quantificateur existentiel vers l'intérieur à travers les disjonctions et les conjonctions, nous obtenons au total une formule C de TL telle que pour toute structure temporelle $\mathcal{H} = (G, <, v)$ avec $(G, <) \in \mathcal{M}$ et pour tout instant $t \in G$ on a

$$(\mathcal{H}, t) \models C \text{ ssi } \mathcal{H}_{temp} \models \exists y \cdot H^0(t, y).$$

Ce n'est pas encore le résultat souhaité, car les prédicats $R_=$, $R_>$ et $R_<$ apparaissent dans le vocabulaire V^0_{temp} de $H^0(x, y)$ et $\mathcal{H}_{temp}$, et de même les variables propositionnelles $r_=$, $r_>$ et $r_<$ dans le vocabulaire V^0 de C et $\mathcal{H}$. Dans la deuxième partie de la démonstration nous allons montrer comment l'on peut se débarrasser de ces symboles supplémentaires, et pour cela nous avons besoin de la propriété de séparabilité de TL par rapport à $\mathcal{M}$. D'après cette propriété, il existe une formule de TL équivalent à C par rapport à $\mathcal{M}$ de la forme

$$\bigvee_j [C^p_j \wedge C^f_j \wedge C^=_j]$$

où les C^p_j sont des formules purement du passé, les C^f_j des formules purement du futur et les $C^=_j$ sont des variables propositionnelles niées ou non[8]. On obtient les formules B^p_j de TL en remplaçant dans C^p_j la variable propositionnelle temporelle $r_>$ par la constante logique true et les variables $r_<$ et $r_=$ par false. De façon analogue on obtient B^f_j à partir de

[8]ou plutôt des conjonctions de littéraux : le total doit fournir une forme normale disjonctive de la combinaison booléenne obtenue par séparabilité à partie de C (NdT).

C_j^f en remplaçant $r_<$ par true et $r_>$ et $r_=$ par false. Finalement $B_j^=$ provient de $C_j^=$ via le remplacement de $r_=$ par true et de $r_>$ et $r_<$ par false. La formule

$$B = \bigvee_j [B_j^p \wedge B_j^f \wedge B_j^=]$$

est ensuite une formule de TL dans le vocabulaire V de départ. Nous montrons que pour ce qui nous occupe on peut remplacer C par B. Soit $v : V \times G \to \{0,1\}$ une interprétation sur le vocabulaire non étendu et soit l'extension $v^t : V^0 \times G \to \{0,1\}$ de v définie par

$$\begin{aligned} v^t(r_>, s) = 1 &\quad\longleftrightarrow\quad t > s \\ v^t(r_<, s) = 1 &\quad\longleftrightarrow\quad t < s \\ v^t(r_=, s) = 1 &\quad\longleftrightarrow\quad t = s \end{aligned}$$

Nous considérons une deuxième fonction d'interprétation $v^p : V^0 \times G \to \{0,1\}$ définie par

$$\begin{aligned} v^t(r_>, s) = 1 &\qquad \text{pour tout } s \\ v^t(r_<, s) = 0 &\qquad \text{pour tout } s \\ v^t(r_=, s) = 0 &\qquad \text{pour tout } s \end{aligned}$$

Clairement v^p et v^t coincident sur le passé de t, d'où :

$$(G, v^t, t) \models C_j^p \text{ ssi } (G, v^p, t) \models C_j^p.$$

Par définition de v^p cependant on a aussi

$$(G, v^p, t) \models C_j^p \text{ ssi } (G, v^p, t) \models B_j^p.$$

Comme seules les variables temporelles de V apparaissent dans B_j^p nous avons au total l'équivalence

$$(G, v^t, t) \models C_j^p \text{ ssi } (G, v, t) \models B_j^p.$$

De façon analogue on montre

$$(G, v^t, t) \models C_j^f \text{ ssi } (G, v, t) \models B_j^f$$

et

$$(G, v^t, t) \models C_j^= \text{ ssi } (G, v, t) \models B_j^=$$

et donc finalement

$$(G, v^t, t) \models C \text{ ssi } (G, v, t) \models B.$$

En incluant aussi la première partie de la démonstration dans le résumé, nous obtenons :

$$\begin{aligned} \mathcal{K}_{temp} \models F(t) \quad &\text{ssi} \quad \mathcal{K}_{temp} \models \exists y \cdot H(t, y) \\ &\text{ssi} \quad \mathcal{K}_{temp}^t \models \exists y \cdot H^0(t, y) \\ &\text{ssi} \quad (\mathcal{K}^t, t) \models C \\ &\text{ssi} \quad (\mathcal{K}, t) \models B \end{aligned}$$

□

Le théorème 17 énonce que sous les hypothèses précisées, la séparabilité d'une logique temporelle est une condition suffisante pour sa complétude définitionnelle. La séparabilité est en fait aussi une condition nécessaire.

Théorème 18 *Soit TL une logique propositionnelle temporelle, qui est définitionnellement complète pour la classe $\mathcal{M}$ de structures temporelles.*

Alors TL possède la propriété de séparabilité par rapport à $\mathcal{M}$.

Démonstration. Voir [GHR94], théorème 9.3.4, p.361.[9] □

Dans la démonstration suivante de la propriété de séparabilité de LTL sur les entiers relatifs, nous utilisons une définition syntaxique de la séparabilité.

Définition 19 (Formules syntaxiquement séparées)

1. *Une formule de la forme A until B telle que l'opérateur since n'apparaît ni dans A ni dans B s'appelle une* **formule syntaxiquement du futur**.

[9] Le manuscrit donne "p.36.1" ; page non vérifiée (NdT).

2. *Une formule de la forme A since B telle que l'opérateur until n'apparaît ni dans A ni dans B s'appelle une* **formule syntaxiquement du passé**.
3. *Une formule s'appelle* **syntaxiquement séparée** *si elle est une combinaison booléenne de formules atomiques et de formules syntaxiquement du passé et du futur.*

Lemme 20 *1. Une formule syntaxiquement du futur est une formule du futur.*

2. *Une formule syntaxiquement du passé est une formule du passé.*
3. *Au cas où toute formule d'une logique temporelle propositionnelle TL est équivalent à une formule syntaxiquement séparée par rapport à une classe $\mathcal{M}$ de structures temporelles, alors TL possède la propriété de séparabilité.*

Démonstration. On montre les assertions 1 et 2 par récurrence sur la complexité des formules A et B. La partie 3 est une conséquence immédiate de 1 et 2. □

On va finalement montrer la séparabilité de LTL par un argument de récurrence. Les étapes critiques de la récurrence sont données par les trois lemmes suivants.

Lemme 21 *Pour toutes formules A, B, C, D on a les équivalences suivantes sur la structures des entiers relatifs :*

1. $D \text{ since } (C \wedge (B \text{ until } A)) \quad \Leftrightarrow$
$(D \text{ since } C) \wedge (B \text{ since } C) \wedge B \wedge (B \text{ until } A)$
$\vee$
$(D \text{ since } C) \wedge (B \text{ since } C) \wedge A$
$\vee$
$D \text{ since } (A \wedge D \wedge (D \text{ since } C) \wedge (B \text{ since } C))$
2. $(C \vee (B \text{ until } A)) \text{ since } D \quad \Leftrightarrow$
$C \text{ since } D$
$\vee$
$(\Phi \text{ since } D) \wedge B \wedge (B \text{ until } A)$
$\vee$
$(\Phi \text{ since } D) \wedge A$
$\vee$
$C \text{ since } (A \wedge (\Phi \text{ since } D))$
où
$\Phi = B \vee (C \text{ since } D) \vee (C \text{ since } (A \wedge C)) \vee A.$
3. $D \text{ until } (C \wedge (B \text{ since } A)) \quad \Leftrightarrow$
$(D \text{ until } C) \wedge (B \text{ until } C) \wedge B \wedge (B \text{ since } A)$
$\vee$
$(D \text{ until } C) \wedge (B \text{ until } C) \wedge A$
$\vee$
$D \text{ until } (A \wedge D \wedge (D \text{ until } C) \wedge (B \text{ until } C))$
4. $(C \vee (B \text{ since } A)) \text{ until } D \quad \Leftrightarrow$
$C \text{ until } D$
$\vee$
$(\Phi \text{ until } D) \wedge B \wedge (B \text{ since } A)$
$\vee$
$(\Phi \text{ until } D) \wedge A$
$\vee$
$C \text{ until } (A \wedge (\Phi \text{ until } D))$
où
$\Phi = B \vee (C \text{ until } D) \vee (C \text{ until } (A \wedge C)) \vee A.$

Démonstration. Pour 1 :
Si la formule D since $(C \wedge (B \text{ until } A))$ est vraie à l'instant t, alors il existe un instant $s < t$ et un instant $u > s$ avec $s \models C$, $u \models A$, pour tout $s < s' < t$ on a $s' \models D$ et pour tout $s < u' < u$ on a $u' \models B$. On peut distinguer trois cas, selon la situation du point u : $u > t$, $u = t$, et $u < t$, voir la figure 1. On voit tout de suite que dans le premier cas le premier membre de la disjonction de la formule à droite du $\Leftrightarrow$ est vrai, dans le second cas le second membre, et dans le troisième cas le troisième.

Supposons maintenant le côté droit de l'équivalence. Nous considérons le cas où le premier membre de la disjonction est vrai :

$$t \models (D \text{ since } C) \wedge (B \text{ since } C) \wedge B \wedge (B \text{ until } A)$$

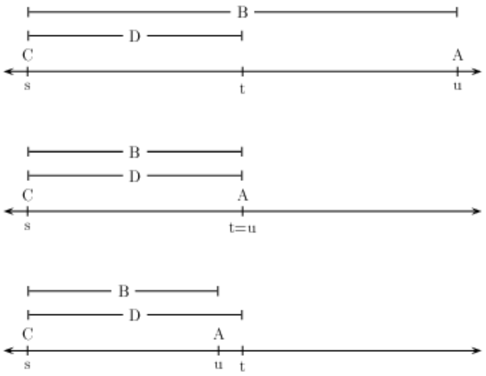

FIG. 1 – Trois possibilités pour $t \models D$ since $(C \wedge (B$ until $A))$.

Alors il existe $s_1 < t$ et $s_2 < t$ avec $s_1 \models C$, $s_2 \models C$, pour tout $s_1 < s' < t$ on a $s' \models D$ et pour tout $s_2 < s' < t$ on a $s' \models B$. À cause de la linéarité de la structure temporelle considérée il existe aussi un $s < t$ (par exemple $s = \max(s_1, s_2)$) avec $s \models C$ et pour tout $s < s' < t$, $s' \models D \wedge B$. Comme on a aussi $t \models B \wedge (B$ until $A)$, il existe u avec $t < u$, $u \models A$ et pour tout $t \leq u' < u$, $u' \models B$. Il s'ensuit au total que $s \models B$ until A, et donc $t \models D$ since $(C \wedge (B$ until $A))$. Les démonstrations des cas où les deuxième ou troisième membres de la disjonction sont vrais se déroulent de façon analogue.

Pour 2 :

Supposons $(C \vee (B$ until $A))$ since D. Alors il existe $s < t$ avec $s \models D$ et pour tout $s < s' < t$ on a $s' \models C \vee (B$ until $A)$. Si pour tout $s < s' < t$ on a $s' \models C$, alors $t \models C$ since D et on a fini. Sinon il existe un instant s_1 avec $s < s_1$ où $\neg C$ est devenu vrai pour la première fois. Maintenant on doit avoir $s_1 \models B$ until A. Il existe aussi un instant u_1 tel que $s_1 < u_1$, $u_1 \models A$ et pour tout $s_1 < s' < u_1$ on a $s' \models B \wedge \neg A$. Ici u_1 est choisi comme étant le premier instant au-dessus de s_1 où A est vrai. Au cas où $u_1 < t$ tout recommence : soit on a $s' \models C$ pour tout $u_1 < s' < t$ soit il existe un s_2 après u_1 où de nouveau $\neg C$ est vrai pour la première fois, etc. Il se peut ici aussi que $s_2 = u_1$. Pour résumer, il existe donc des instants $s_1, \ldots, s_k$ et $u_1, \ldots, u_k$ avec[10]

$$s < s_1 < u_1 \leq s_2 < u_2 \leq \ldots \leq s_k < u_k$$

et

$$\begin{array}{rrcl} \text{pour tout } i & s_i & \models & \neg C \\ \text{pour tout } s < s' < s_1 & s' & \models & C \\ \text{pour tout } s_i < s' < u_i & s' & \models & B \wedge \neg A \\ \text{pour tout } i & u_i & \models & A \\ \text{pour tout } u_i \leq s' < s_{i+1} & s' & \models & C \end{array}$$

Nous distinguons de nouveau trois cas, selon que $t < u_k$, $t = u_k$ ou $u_k < t$. Ces cas correspondent aussi dans le même ordre aux membres 2 à 4 de la disjonction sur le côté droit de l'équivalence. Nous traitons complètement le troisième cas et laissons au lecteur le soin de faire les adaptations faciles pour les démonstrations des cas restants.

Dans le troisième cas on a $s_k < t < u_k$ et de là, par le choix des instants s_i et u_i on a immédiatement

[10] Le fait que le processus s'arrête, ici après k itérations, est dû au fait que l'ordre sur $\mathbb{Z}$ n'est pas dense, et que tous les s_i et u_i sont à l'intérieur de l'intervalle s, t (NdT), sauf éventuellement u_k (NdT).

$t \models B$ et $t \models B$ until A. Il reste à montrer que pour chaque instant $s < t' < t$ on a la proposition $t' \models \Phi$, dont nous répétons la définition pour le confort du lecteur :

$$\Phi = B \vee (C \text{ since } D) \vee (C \text{ since } (A \wedge C)) \vee A$$

Pour $s < t' \leq s_1$ on a $t' \models C$ since D et donc aussi $t' \models \Phi$. Pour $s_i < t' < u_i$ on a $t' \models B$ donc le premier membre de la disjonction Φ. Pour tout $i \geq 1$ on a $u_i \models A$, également un membre de la disjonction Φ. Si $u_i < s_{i+1}$, pour tout $u_i < t' < s_{i+1}$ on a $t' \models C$ since $(A \wedge C)$. Ici on remarque qu'au cas où $u_i \models \neg C$ on doit avoir $u_i = s_{i+1}$ par le choix des instants.

Réciproquement supposons que $t \models (\Phi \text{ since } D) \wedge B \wedge (B \text{ until } A)$. Nous voulons montrer qu'il s'ensuit que $t \models (C \vee (B \text{ until } A)) \text{ since } D$. Il existe donc des instants $s < t$ et u tel que $t < u$ avec $s \models D$, $u \models A$, pour tout $t \leq u' < u$ on a $u' \models B$ et pour tout $s < s' < t$ on a $s' \models \Phi$. Nous pouvons de plus choisir s de sorte que pour tout $s < s' < t$ on ait $s' \models \neg D$[11]. Nous voulons montrer que pour tout $s < s' < t$, on a $s' \models C \vee (B \text{ until } A)$. Supposons que cela ne soit pas le cas et qu'il existe un instant s' dans l'intervalle donné tel que $s' \models \neg C \wedge \neg(B \text{ until } A)$. Nous savons déjà qu'il existe un instant $u > t > s'$ avec $u \models A$, donc B until A ne peut pas échouer à partir de t. De $s' \models \neg(B \text{ until } A)$ on déduit l'existence d'un instant $s' < t' < t$ tel que $t' \models \neg B \wedge \neg A$ et pour tout $s' < r < t'$ on a $r \models B \wedge \neg A$. Nous argumentons que $t' \models \Phi$ ne peut pas être vrai, ce qui fournira la contradiction souhaitée. Très clairement, le premier et le dernier membre de la disjonction Φ sont faux à l'instant t'. Mais $t' \models C$ since D n'est pas non plus possible, car par notre choix de s chaque instant $s_0 < t'$ qui satisfait la formule D doit aussi satisfaire $s_0 \leq s$. De $s_0 < s' < t'$ et $s' \models \neg C$ on déduit $t' \models \neg(C \text{ since } D)$. Il reste à investiguer la dernière possibilité $t' \models C \text{ since } (A \wedge C)$. Mais le premier instant avant t' où $A \wedge C$ peut être satisfaite est dans tous les cas strictement avant s', ce qui exclut aussi cette possibilité. On a donc abouti à une contradiction à partir de l'hypothèse $s' \models \neg C \wedge \neg(B \text{ until } A)$.

Pour 3 et 4 : analogues à 1 et 2. □

[11] Il suffit ici de choisir s maximal tel que $s \models D$ (NdT).

Lemme 22 *Pour toutes formules A, B, C, D on a les équivalences suivantes sur la structures des entiers relatifs*[12] *:*

1. $D \text{ since } (C \wedge \neg(B \text{ until } A)) \quad \Leftrightarrow$ *?*
2. $(C \vee \neg(B \text{ until } A)) \text{ since } D \quad \Leftrightarrow$ *?*
3. $D \text{ until } (C \wedge \neg(B \text{ since } A)) \quad \Leftrightarrow$ *?*
4. $(C \vee \neg(B \text{ since } A)) \text{ until } D \quad \Leftrightarrow$ *?*

Lemme 23 *Pour toutes formules A, B, C, D on a les équivalences suivantes sur la structures des entiers relatifs*[13]

1. $(D \vee (B \text{ until } A)) \text{ since } (C \wedge \neg(B \text{ until } A))$ $\Leftrightarrow$ *?*
2. $(C \vee \neg(B \text{ until } A)) \text{ since } (D \wedge (B \text{ until } A))$ $\Leftrightarrow$ *?*

[12] Le manuscrit de l'auteur ne donne pas les côtés droits des équivalences. On peut voir ce lemme comme un exercice. Pour la première équivalence, on peut utiliser le lemme 13.3, d'où $D \text{ since } (C \wedge \neg(B \text{ until } A))$ est équivalent à $D \text{ since } (C \wedge ((\text{aif } \neg A) \vee ((\neg A \wedge B) \text{ until } (\neg A \wedge \neg B))))$, donc en distribuant la conjonction sur la disjonction, à $D \text{ since } ((C \wedge \text{aif } \neg A) \vee (C \wedge ((\neg A \wedge B) \text{ until } (\neg A \wedge \neg B))))$. On peut sortir le ou de la droite du since , par l'équivalence évidente entre $F \text{ since } (G \vee H)$ et $(F \text{ since } G) \vee (F \text{ since } H)$. On peut ensuite réécrire $D \text{ since } (C \wedge ((\neg A \wedge B) \text{ until } (\neg A \wedge \neg B)))$ en utilisant le lemme 21.1, et $D \text{ since } (C \wedge \text{aif } \neg A)$ est clairement équivalent à $((D \wedge \neg A) \text{ since } C) \wedge \neg A \wedge \text{aif } \neg A$ (NdT).

[13] Le manuscrit de l'auteur ne donne pas les côtés droits des équivalences. Exercice. On doit bien admettre qu'on verse dans le cauchemar si on veut vraiment le faire ! (NdT)

3. $(D \lor (B \text{ since } A))$ until $(C \land \neg(B \text{ since } A))$ $\Leftrightarrow$?
4. $(C \lor \neg(B \text{ since } A))$ until $(D \land (B \text{ since } A))$ $\Leftrightarrow$?

Pour montrer que l'on peut transformer une formule en une formule syntaxiquement séparée équivalente par application répétée des lemmes 21, 22 et 23, nous avons besoin d'une mesure de complexité des formules LTL.

Définition 24 *Pour une formule LTL arbitraire A nous définissons la* **complexité** until $uk(A)$*, resp. la* **complexité** since $sk(A)$ *comme suit :*

1. *$uk(A)$ est le cardinal de $um(A)$, ensemble des sous-formules de A de la forme B until C ;*
2. *$sk(A)$ est le cardinal de $sm(A)$, ensemble des sous-formules de A de la forme B since C.*

Il est important de remarquer qu'une formule B until C qui apparaît plusieurs fois dans A n'est comptée qu'une fois dans $uk(A)$.

Lemme 25 *Soient A, B des formules syntaxiquement séparées, alors il existe des formules syntaxiquement séparées R_1, resp. R_2, respectivement équivalentes à*

1. *A since B ;*
2. *A until B.*

De plus on peut choisir R_1 et R_2 de sorte que

1. *$uk(R_1) = uk(A \text{ since } B)$;*
2. *$sk(R_2) = sk(A \text{ until } B)$.*

Démonstration. On montre 1 par récurrence sur $uk = uk(A) + uk(B)$. Lorsque $uk = 0$, A since B est déjà syntaxiquement séparé. Soit donc $uk > 0$. Soit $B = \bigvee_j B_j$ une forme normale conjonctive de B et $A = \bigwedge_k A_k$ une forme normale conjonctive de A. D'après le lemme 5 il suffit de montrer de montrer que pour chaque j et chaque k, on peut transformer A_k since B_j en une formule syntaxiquement séparée équivalente. Chaque A_k et chaque B_j est de la forme

$$\begin{aligned} C_0 \quad \land \quad & (D_1 \text{ until } E_1) \land \ldots \land (D_n \text{ until } E_n) \\ \land \quad & \neg(D_{n+1} \text{ until } E_{n+1}) \land \ldots \land (D_m \text{ until } E_m) \end{aligned}$$

Ici C_0 est une conjonction de littéraux propositionnels. Nous choisissons un D_i until E_i qui n'apparaît pas comme sous-formule stricte d'un D_r until E_r. Comme la relation de sous-formule stricte ne peut pas avoir de cycle, il doit exister une telle D_i until E_i. Nous pouvons de plus supposer qu'à l'intérieur de A_k, resp. de B_j, les sous-formules D_i until E_i n'apparaissent qu'une fois, car on peut condenser les doubles occurrences avec le même signe de façon équivalente en une seule occurrence. Si D_i until E_i apparaît une fois niée et une fois non niée dans A_k, alors on peut remplacer A_k tout entier par false, et si cette situation se produit dans B_j, nous remplaçons B_j par false. Il reste en tout six cas à explorer qui correspondent selon le cas aux deux premières parties des lemmes 21, 22 et 23.

Nous traitons complètement le cas où D_i until E_i apparaît non nié dans un B_j. Indépendamment de tous les indices, ce que l'on doit faire est de trouver une formule syntaxiquement séparée équivalente à $F = A \text{ since } (C \land (D \text{ until } E))$, où D until E n'apparaît pas dans A ni C et où D et E sont déjà syntaxiquement séparées. Nous pouvons de plus faire appel à l'hypothèse de récurrence, selon laquelle toutes les formules de complexité until strictement inférieure à $uk(F)$ peuvent être transformées en des formules syntaxiquement séparées équivalentes. D'après le lemme 21 partie 1 nous pouvons réécrire F en une combinaison booléenne des formules suivantes :

1. A since C
2. D since C
3. D

4. D until E
5. E
6. A since $(E \wedge (A \text{ since } C) \wedge (D \text{ since } C))$.

Par le choix de D until E la complexité until de toutes ces formules est strictement plus petite que $uk(F)$. Une exception posible est la formule 4, mais elle est de toute façon déjà syntaxiquement séparée. Les formules 1–5 peuvent donc être transformées en des formules syntaxiquement séparées équivalentes. La formule 6 a certes une complexité until plus petite que $uk(F)$, mais le côté droit des opérateurs since ci-dessus n'est plus nécessairement syntaxiquement séparé — dans E et dans D pourraient encore se trouver d'autres opérateurs until . L'hypothèse de récurrence nous permet cependant de changer le côté droit

$$E \wedge (A \text{ since } C) \wedge (D \text{ since } C)$$

en une formule syntaxiquement transformée équivalente R. Par une utilisation supplémentaire de l'hypothèse de récurrence nous pouvons maintenant séparer syntaxiquement A since R. À ce point nous devons cependant faire appel à l'information additionnelle que

$$uk(R) = uk(E \wedge (A \text{ since } C) \wedge (D \text{ since } C)).$$

Comme les formules 1–6 ne sont construites qu'à partir de sous-formules de F et qu'aucune nouvelle formule G until H n'est introduite, il est de nouveau vrai de la formule R syntaxiquement séparée équivalente à F que $uk(R) = uk(F)$. □

Théorème 26 *Pour chaque formule LTL F, il existe une formule syntaxiquement séparée R qui est équivalente à F sur la structure temporelle des entiers relatifs.*

Démonstration. Par les préparatifs ci-dessus, une récurrence simple sur la structure de la formule suffit. Soient A, B des formules syntaxiquement séparées, alors trivialement $\neg A$, $A \wedge B$ et $A \vee B$, et par le lemme 25 A since B et A until B le sont aussi[14]. □

1.1.4 Exercices

Exercice 27 *Parmi les hypothèses du théorème 17 on demande que les structures de $\mathcal{M}$ soient linéaires. Où cette hypothèse est-elle utilisée dans la démonstration ?*

Exercice 28 *Nous réexaminons les assertions du lemme 5. Montrez que :*

1. *Les assertions 1 et 2 sont vraies pour des structures temporelles arbitraires.*
2. *Donner des contre-exemples montrant que 3 et 4 peuvent devenir fausses pour des structures temporelles non linéaires.*

Exercice 29 *Trouvez une formule LTL A_{alt} telle que pour tout structure temporelle $\mathcal{K} = (\mathbb{N}, <, v)$ on ait*

$$(\mathcal{K}, 0) \models A_{alt} \textit{ ssi } (\mathcal{K}, i) \models p \textit{ est équivalent au fait pour } i \textit{ d'être pair.}$$

Remarquez la différence avec la propriété non définissable du lemme 7.

Références

[BS95] Leonard Bolc and Andrzej Szałas, editors. *Time and Logic, a Computational Approach.* University College London Press, 1995.

[GHR94] Dov M. Gabbay, Ian Hodkinson, and Mark Reynolds. *Temporal Logic, Mathematical Foundations and Computational Aspects*, volume 1 of *Oxford Logic*

[14]. . . ou plutôt peuvent être transformées en des formules syntaxiquement séparées équivalentes (NdT).

Guides, vol. 28. Clarendon Press, Oxford, 1994.

[Kam68] Hans Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, Dept. Philosophy, UCLA, 1968.

[Wol83] Pierre Wolper. Temporal logic can be more expressive. *Information and Control*, 56(1–2) :72–99, 1983.

[Wol95] Pierre Wolper. *On the Relation of Programs and Computations to Models of Temporal Logic*, chapter 3, pages 131–178. In Bolc and Szałas [BS95], 1995.