

Quatrième partie

LTL – logique temporelle linéaire



Plan

- 1 Logiques temporelles
- 2 LTL
 - Syntaxe
 - Sémantique
 - Réduction
- 3 Expressivité



Logiques temporelles

Objectif

Exprimer des **propriétés** portant sur les **exécutions** des systèmes.

Spécification non opérationnelle : pas de relation de transition explicite, pas de notion d'états initiaux.

Une logique est définie par :

- une syntaxe : opérateurs de logique classique plus des opérateurs temporels pour parler du futur et du passé.
- une sémantique : domaine des objets (appelés modèles) sur lesquels on va tester la validité des formules, plus l'interprétation des opérateurs.



Plan

- 1 Logiques temporelles
- 2 LTL
 - Syntaxe
 - Sémantique
 - Réduction
- 3 Expressivité



Linear Temporal Logic

Modèles

Une formule LTL se rapporte toujours à une **trace** donnée σ d'un système.

Les traces constituent les modèles de cette logique.

Note : plutôt que d'*état*, on parle souvent d'*instant*.



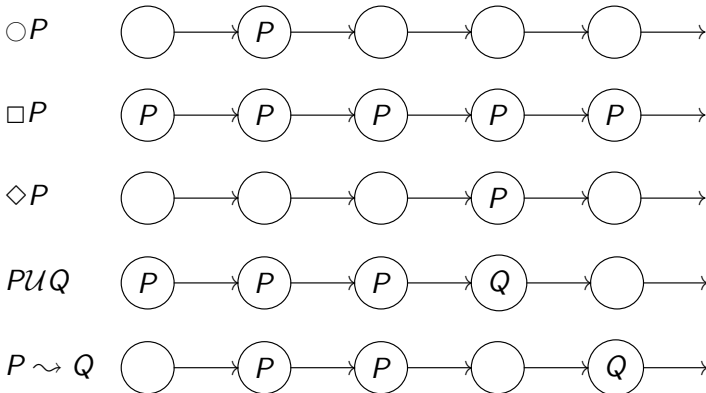
Syntaxe de la LTL

formule	nom	interprétation
$\bigcirc P$	next	P est vrai à l'instant suivant
$\square P$	always	P est toujours vrai i.e. à tout instant à partir de l'instant courant
$\diamond P$	eventually	P finit par être vrai (dans le futur)
PUQ	until	Q finit par être vrai, et en attendant P reste vrai
$P \leadsto Q$	leadsto	quand P est vrai, alors Q l'est plus tard
$\neg P$		
$P \vee Q$		
$P \wedge Q$		
s		l'état courant de l'exécution est s

Dans les approches symboliques, l'opérateur \bigcirc représentant l'instant suivant peut être remplacé par des variables primées qui représentent la valeur des variables du système dans l'état suivant.



Intuition sémantique



Opérateurs minimaux

Les opérateurs minimaux sont $\bigcirc P$ et $P \mathcal{U} Q$:

- $\bigcirc P \stackrel{\Delta}{=} \text{True } \mathcal{U} P$
- $\square P \stackrel{\Delta}{=} \neg \bigcirc \neg P$
- $P \rightsquigarrow Q \stackrel{\Delta}{=} \square (P \Rightarrow \bigcirc Q)$



Syntaxe alternative

On trouve fréquemment une autre syntaxe :

- ↔ G (globally)
- ◇ ↔ F (finally)
- ↔ X (next)

Opérateurs complémentaires

- Opérateur waiting-for (ou unless ou weak-until)
 $PWQ \triangleq \square P \vee PUQ$
Q finit peut-être par être vrai et en attendant P reste vrai
- Opérateur release
 $PRQ \triangleq \neg(\neg PU\neg Q)$
Q est toujours vrai, sauf à partir du moment où P est vrai.

Opérateurs du passé

formule	nom	interprétation
$\ominus P$	previously	P est vrai dans l'instant précédent
$\boxminus P$	has-always-been	P a toujours été vrai jusqu'à l'instant courant
$\diamond P$	once	P a été vrai dans le passé
PSQ	since	Q a été vrai dans le passé et P est resté vrai depuis la dernière occurrence de Q
PBQ	back-to	P est vrai depuis la dernière occurrence de Q , ou depuis l'instant initial si Q n'a jamais été vrai

Guère d'utilité en pratique...



Sémantique (système)

On note (σ, i) pour le suffixe $\langle s_i \rightarrow s_{i+1} \rightarrow \dots \rangle$ d'une exécution $\sigma = \langle s_0 \rightarrow s_1 \rightarrow \dots \rangle$.

Vérification par un système

Un système \mathcal{S} vérifie (valide) la formule F ssi toutes les exécutions de \mathcal{S} la valident à partir de l'instant initial :

$$\frac{\forall \sigma \in Exec(\mathcal{S}) : (\sigma, 0) \models F}{\mathcal{S} \models F}$$



Sémantique (opérateurs logiques)

$$\overline{\neg (\langle \rangle, i) \models P}$$

$$\frac{(\sigma, i) \models P \quad (\sigma, i) \models Q}{(\sigma, i) \models P \wedge Q}$$

$$\frac{\neg (\sigma, i) \models P}{(\sigma, i) \models \neg P}$$

Sémantique (opérateurs temporels)

$$\frac{\sigma_i = s}{(\sigma, i) \models s}$$

$$\frac{(\sigma, i + 1) \models P}{(\sigma, i) \models \bigcirc P}$$

$$\frac{\exists k \geq 0 : (\sigma, i + k) \models Q \wedge \forall k' < k : (\sigma, i + k') \models P}{(\sigma, i) \models PUQ}$$

Sémantique (opérateurs temporels dérivés)

$$\frac{\exists k \geq 0 : (\sigma, i + k) \models P}{(\sigma, i) \models \diamond P}$$

$$\frac{\forall k \geq 0 : (\sigma, i + k) \models P}{(\sigma, i) \models \square P}$$

$$\frac{\forall k \geq 0 : ((\sigma, i + k) \models P \Rightarrow \exists k' \geq k : (\sigma, i + k') \models Q)}{(\sigma, i) \models P \rightsquigarrow Q}$$

Réduction à la logique pure

- La logique temporelle linéaire possède une expressivité telle qu'elle peut représenter exactement n'importe quelle spécification opérationnelle décrite en termes de système de transitions, d'où :
- vérifier qu'un système de transitions \mathcal{M} possède la propriété temporelle F_{Spec} :

$$\mathcal{M} \models F_{Spec}$$

- revient à déterminer la validité de :

$$F_{\mathcal{M}} \Rightarrow F_{Spec}$$

où $F_{\mathcal{M}}$ est une formule représentant exactement les exécutions du modèle \mathcal{M} (i.e. ses états initiaux, ses transitions, ses contraintes d'équité).

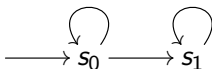


Plan

- 1 Logiques temporelles
- 2 LTL
 - Syntaxe
 - Sémantique
 - Réduction
- 3 Expressivité

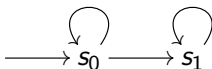


Exemple 1



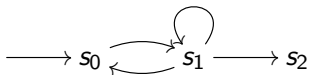
	pas d'équité	équité faible (s_0, s_1)
$s_0 \wedge \bigcirc s_0$		
$s_0 \wedge \bigcirc (s_0 \vee s_1)$		
$\square (s_0 \Rightarrow \bigcirc s_0)$		
$\square (s_0 \Rightarrow \bigcirc (s_0 \vee s_1))$		
$\square (s_1 \Rightarrow \bigcirc s_1)$		
$\diamond (s_0 \wedge \bigcirc s_1)$		
$\square s_0$		
$\diamond \neg s_0$		
$\diamond \square s_1$		
$s_0 \mathcal{W} s_1$		
$s_0 \mathcal{U} s_1$		

Exemple 1



	pas d'équité	équité faible (s_0, s_1)
$s_0 \wedge \bigcirc s_0$	$n (s_0 \rightarrow s_1^\omega)$	n
$s_0 \wedge \bigcirc (s_0 \vee s_1)$	o	o
$\square (s_0 \Rightarrow \bigcirc s_0)$	$n (s_0 \rightarrow s_1^\omega)$	n
$\square (s_0 \Rightarrow \bigcirc (s_0 \vee s_1))$	o	o
$\square (s_1 \Rightarrow \bigcirc s_1)$	o	o
$\diamond (s_0 \wedge \bigcirc s_1)$	$n (s_0^\omega)$	o
$\square s_0$	$n (s_0 \rightarrow s_1^\omega)$	n
$\diamond \neg s_0$	$n (s_0^\omega)$	o
$\diamond \square s_1$	$n (s_0^\omega)$	o
$s_0 \mathcal{W} s_1$	o	o
$s_0 \mathcal{U} s_1$	$n (s_0^\omega)$	o

Exemple 2



	pas d'équité	faible (s_1, s_2)	forte (s_1, s_2)
$\Box \Diamond \neg s_1$			
$\Box (s_1 \Rightarrow \Diamond s_2)$			
$\Diamond \Box (s_1 \vee s_2)$			
$\Box (s_1 \mathcal{U} s_2)$			
$\Box (s_0 \Rightarrow s_0 \mathcal{U} s_1)$			
$\Box (s_0 \mathcal{U} (s_1 \vee s_2))$			
$\Box (s_1 \Rightarrow s_1 \mathcal{U} s_2)$			
$\Diamond (s_1 \mathcal{U} s_2)$			
$\Diamond (s_1 \mathcal{W} s_2)$			
$\Box \Diamond (s_1 \mathcal{U} (s_0 \vee s_2))$			

Exemple 2



	pas d'équité	faible (s_1, s_2)	forte (s_1, s_2)
$\Box \Diamond \neg s_1$	$n (s_0 \rightarrow s_1^\omega)$	o	o
$\Box (s_1 \Rightarrow \Diamond s_2)$	$n (s_0 \rightarrow s_1^\omega)$	n	o
$\Diamond \Box (s_1 \vee s_2)$	$n ((s_0 \rightarrow s_1)^\omega)$	n	o
$\Box (s_1 \mathcal{U} s_2)$	$n (s_0 \rightarrow \dots)$	n	n
$\Box (s_0 \Rightarrow s_0 \mathcal{U} s_1)$	o	o	o
$\Box (s_0 \mathcal{U} (s_1 \vee s_2))$	o	o	o
$\Box (s_1 \Rightarrow s_1 \mathcal{U} s_2)$	$n (s_0 \rightarrow s_1^\omega)$	$n ((s_0 \rightarrow s_1)^\omega)$	$n (s_0 \rightarrow s_1 \rightarrow s_0 \rightarrow s_1 \rightarrow s_2)$
$\Diamond (s_1 \mathcal{U} s_2)$	$n (s_0 \rightarrow s_1^\omega)$	$n ((s_0 \rightarrow s_1)^\omega)$	o
$\Diamond (s_1 \mathcal{W} s_2)$	$n ((s_0 \rightarrow s_1)^\omega)$	n	o
$\Box \Diamond (s_1 \mathcal{U} (s_0 \vee s_2))$	$n (s_0 \rightarrow s_1^\omega)$	o	o

Invariance, stabilité

Invariance

Spécifier un sur-ensemble des états accessibles d'un système :

$$\mathcal{S} \models \Box P$$

Stabilité

Spécifier la stabilité d'une situation si elle survient :

$$\mathcal{S} \models \Box(P \Rightarrow \Box P)$$



Possibilité

Possibilité

Spécifier qu'il est possible d'atteindre un certain état vérifiant P dans une certaine exécution :

Impossible pour P arbitraire, mais pour P un prédicat d'état :

$$S \not\models \Box \neg P$$

Attention à la négation : $\neg \Box P = \Diamond \neg P$ mais $S \not\models \Box P \not\Rightarrow S \models \Diamond \neg P$



Négation

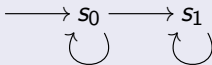
Négation : danger !

$$\sigma \models \neg P \equiv \sigma \not\models P$$

$$S \models \neg P \Rightarrow S \not\models P \text{ mais pas l'inverse !}$$

$S \not\models Q$ signifie qu'il existe **au moins une** exécution qui invalide Q (= qui valide $\neg Q$), mais pas que toutes les exécutions le font.

En LTL, on peut avoir $S \not\models Q \wedge S \not\models \neg Q$:



$$\frac{s_0^+ \rightarrow s_1^\omega \not\models \Box s_0}{S \not\models \Box s_0}$$

$$\frac{s_0^\omega \not\models \Diamond \neg s_0}{S \not\models \Diamond \neg s_0}$$

Combinaisons

Infiniment souvent – Réponse

Spécifier que P est infiniment souvent vrai dans toute exécution :

$$\mathcal{S} \models \Box \Diamond P$$

Finalement toujours – Persistance

Spécifier que P finit par rester définitivement vrai :

$$\mathcal{S} \models \Diamond \Box P$$

Note : $\Box \Box P = \Box P$ et $\Diamond \Diamond P = \Diamond P$



Client/serveur

Réponse

Spécifier qu'un système (jouant le rôle d'un serveur) répond toujours (par Q) à un requête donnée (par P) :

$$S \models \Box(P \Rightarrow \Diamond Q)$$

Souvent nommé leads-to :

$$S \models P \rightsquigarrow Q$$

Stabilité d'une requête

Spécifier que la requête P d'un système (jouant le rôle d'un client) est stable tant qu'il n'y a pas de réponse favorable Q :

$$S \models \Box(P \Rightarrow P\mathcal{W}Q)$$

Équité – *Fairness*

Équité faible des transitions

Soit $r \subseteq R$. Les transitions r sont en équité faible dans \mathcal{S} :

$$\mathcal{S} \models \diamond \square \text{dom}(r) \Rightarrow \square \diamond \text{codom}(r)$$

$$\mathcal{S} \models \square \diamond \neg \text{dom}(r) \vee \square \diamond \text{codom}(r)$$

Équité forte des transitions

Soit $r \subseteq R$. Les transitions r sont en équité forte dans \mathcal{S} :

$$\mathcal{S} \models \square \diamond \text{dom}(r) \Rightarrow \square \diamond \text{codom}(r)$$

$$\mathcal{S} \models \diamond \square \neg \text{dom}(r) \vee \square \diamond \text{codom}(r)$$

Limites de l'expressivité

Tout n'est pas exprimable en LTL :

- Possibilité arbitraire : si P devient vrai, il est toujours possible (mais pas nécessaire) que Q le devienne après.
- Réinitialisabilité : quelque soit l'état, il est possible de revenir dans un des états initiaux.



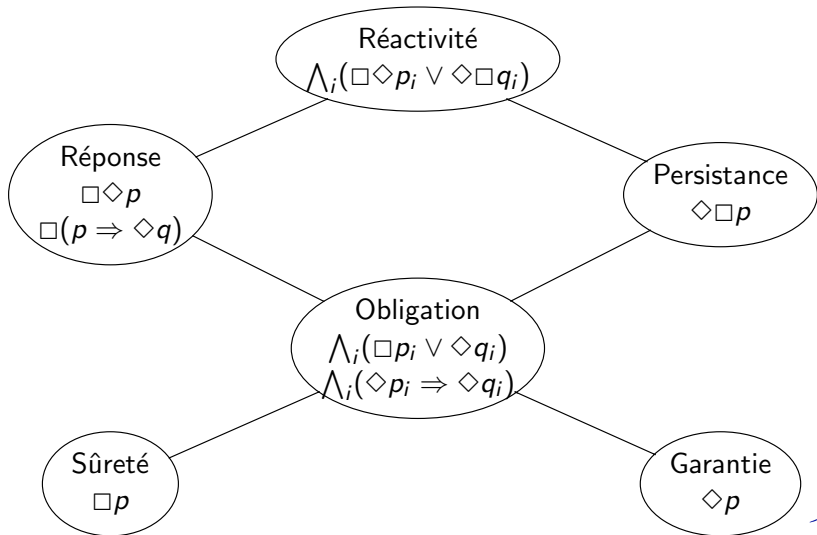
Sûreté/vivacité – *Safety/Liveness*

On qualifie de

- Sûreté : rien de mauvais ne se produit
= une propriété qui s'invalide sur un préfixe fini :
 $\Box P, \Box(P \Rightarrow \Box P), PWQ\dots$
- Vivacité : quelque chose de bon finit par se produire
= une propriété qui peut toujours être validée en étendant le préfixe d'une exécution :
 $\Diamond P, P \rightsquigarrow Q\dots$
- Certaines propriétés combinent vivacité et sûreté :
 $PUQ, \Box P \wedge \Diamond Q\dots$
 - Réponse : $\Box \Diamond P$
 - Persistance : $\Diamond \Box P$



Classification



Spécification d'un ST

Si on utilise une description en intention, et si l'on remplace l'utilisation de l'opérateur \bigcirc par les variables primées, alors on peut spécifier toutes les exécutions permises par un système $\langle S, I, R \rangle$:

$$\mathcal{S} \models I \wedge \square R$$

L'utilisation de variables primées n'est pas nécessaire mais simplifie les formules.

Par exemple $P(x, x')$ est équivalent à la formule :

$$\forall v : x = v \Rightarrow \bigcirc P(v, x)$$

qui nécessite une quantification sur une variable.



Logiques modales

La LTL est un cas particulier de logique modale.

Autres interprétations :

- \Box = nécessité, \Diamond = possibilité
- logique de la croyance : « je crois que P est vrai »
- logique épistémique : « X sait que P »
- logique déontique : « P est obligatoire/interdit/permis »
- ...

