Système de transition

Système de transition (ST)

Un système de transition est un triplet $\langle S, I, R\rangle$.

- S : ensemble d'états. Peut être fini ou infini.
- $I \subseteq S$: ensemble des états initiaux.
- $R \subseteq S \times S$: relation (de transitions) entre paires d'états. $(s, s') \in R$ signifie qu'il existe une transition faisant passer le système de l'état s à l'état s'.

Séquence

Séquence

Soit S un ensemble.
$S^\star \triangleq$ l'ensemble des séquences finies sur S.
$S^\omega \triangleq$ l'ensemble des séquences infinies sur S.
$\sigma_i \triangleq$ le $i^{\text{ème}}$ (à partir de 0) élément d'une séquence σ.

Conventions de représentation :

- Une séquence s est notée sous la forme : $\langle s_1 \to s_2 \to \ldots\rangle$.
- $\langle\rangle$: la séquence vide.

Pour une séquence finie σ :

- $\sigma^\star \triangleq$ l'ensemble des séquences finies produites par la répétition arbitraire de σ.
- $\sigma^+ \triangleq \sigma^\star \setminus \{\langle\rangle\}$
- $\sigma^\omega \triangleq$ la séquence infinie produite par la répétition infinie de σ.

Trace

Traces finies

Soit $\langle S, I, R\rangle$ un système de transition.
On appelle trace finie une séquence finie $\sigma \in S^\star$ telle que :

- $\sigma = \langle s_0 \to s_1 \to \ldots \to s_{n-1} \to s_n\rangle$
- $\forall i \in [0..n[: (s_i, s_{i+1}) \in R$

Traces finies maximales

Soit $\langle S, I, R\rangle$ un système de transition.
Une trace finie $\langle s_0 \to s_1 \to \ldots \to s_{n-1} \to s_n\rangle \in S^\star$ est maximale $\triangleq$ il n'existe pas d'état successeur à s_n, i.e. $\forall s \in S : (s_n, s) \notin R$.

Une trace maximale va le plus loin possible.

Traces infinies

Soit $\langle S, I, R\rangle$ un système de transition, et $s_0 \in S$.
On appelle trace infinie à partir de s_0 un élément $tr \in S^\omega$ tel que :

- $tr = \langle s_0 \to s_1 \to s_2 \ldots\rangle$
- $\forall i \in \mathbb{N} : (s_i, s_{i+1}) \in R$

Traces issues d'un état

Soit $\langle S, I, R\rangle$ un système de transition, et $s \in S$.
$Traces(s) \triangleq$ l'ensemble des traces infinies ou finies maximales commençant à l'état s.

Exécution

Exécutions

Soit $\mathcal{S} = \langle S, I, R\rangle$ un système de transitions.
Une exécution $\sigma = \langle s_0 \to \ldots\rangle$ est une trace infinie ou finie maximale telle que $s_0 \in I$.
$Exec(\mathcal{S}) \triangleq$ l'ensemble des exécutions de $\mathcal{S} = \bigcup_{s_0 \in I} Traces(s_0)$.

Etats accessibles

État accessible

Soit $\mathcal{S} = \langle S, I, R\rangle$ un système de transition.
$s \in S$ est un état accessible $\triangleq$ il existe une exécution qui passe par s (ou équivalent, il existe un préfixe d'exécution qui aboutit à s).
$Acc(\mathcal{S}) \triangleq$ l'ensemble des états accessibles de $\mathcal{S}$.

Graphe des exécutions

Graphe des exécutions

Soit $\mathcal{S} = \langle S, I, R\rangle$ un système de transition.
Le graphe des exécutions est le graphe orienté où :

- l'ensemble des sommets est $Acc(\mathcal{S})$;
- l'ensemble des arêtes orientées est R, restreint aux seuls états accessibles.

Il s'agit donc du graphe $\langle S \cap Acc(\mathcal{S}), R \cap (Acc(\mathcal{S}) \times Acc(\mathcal{S}))\rangle$.

Représentation en intention

Système de transition à base de variables

Un triplet $\langle V, Init, Trans\rangle$ où

- $V = \{v_1, \ldots, v_n\}$: ensemble fini de variables.
- $Init(v_1, \ldots, v_n)$: prédicat définissant les états initiaux et portant sur les variables v_i.
- $Trans(v_1, \ldots, v_n, v_1', \ldots, v_n')$: prédicat définissant les transitions, portant sur les variables v_i représentant l'état courant et les variables v_i' représentant l'état suivant.

```
i = 0;
while (i < N) {
  i = i+1;
}
```

En extension pour $N = 5$:
$\langle\{0, 1, 2, 3, 4, 5\}, \{0\}, \{(0, 1), (1, 2), (2, 3), (3, 4), (4, 5)\}\rangle$

Graphe d'exécution pour $N = 5$:

$$\longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 4 \longrightarrow 5$$

Symboliquement (en intention) :
$V \triangleq i \in \mathbb{N}$
$I \triangleq i = 0$
$T \triangleq i < N \land i' = i + 1$ ou $T \triangleq i' \leq N \land i' - i = 1$

Interblocage

Interblocage

Un système possède un interblocage (deadlock) $\triangleq$ il existe un état accessible sans successeur par la relation R.
De manière équivalente un système possède un interblocage s'il existe des exécutions finies.

Système réinitialisable

Réinitialisable

Un système est réinitialisable $\triangleq$ depuis tout état accessible, il existe une trace finie menant à un état initial.

Bégaiement

Bégaiement

Un état s bégaie $\triangleq$ l'état possède une boucle : $(s, s) \in R$.
Un système de transition bégaie $\triangleq$ tout état possède une boucle vers lui-même : $Id \subseteq R$.

TLA+

Constantes

- Constantes explicites : 0, 1, TRUE, FALSE, "toto"
- Constantes nommées : CONSTANT N
  généralement accompagnées de propriétés :
  ASSUME $N \in Nat \land N \geq 2$

Expressions autorisées

Tout ce qui est axiomatisable :

- expressions logiques : $\neg$, $\land$, $\lor$, $\forall x \in S : p(x)$, $\exists x \in S : p(x) \ldots$
- expressions arithmétiques : $+$, $-$, $>$ ...
- expressions ensemblistes : $\in$, $\cup$, $\cap$, $\subset$, $\{e_1, e_2, \ldots, e_n\}$, $n..m$, $\{x \in S : p(x)\}$, $\{f(x) : x \in S\}$, UNION S, SUBSET S
- IF $pred$ THEN e_1 ELSE e_2
- fonctions de X dans Y
- tuples, séquences, ...

Opérateurs ensemblistes

$\{e_1, \ldots, e_n\}$	ensemble en extension
$n..m$	$\{i \in Nat : n \leq i \leq m\}$
$\{x \in S : p(x)\}$	l'ensemble des éléments de S vérifiant la propriété p $\{n \in 1..10 : n\%2 = 0\} = \{2, 4, 6, 8, 10\}$ $\{n \in Nat : n\%2 = 1\}$ = les entiers impairs
$\{f(x) : x \in S\}$	l'ensemble des valeurs de l'operateur f en S $\{2 * n : n \in 1..5\} = \{2, 4, 6, 8, 10\}$ $\{2 * n + 1 : n \in Nat\}$ = les entiers impairs
UNION S	l'union des éléments de S UNION $\{\{1, 2\}, \{2, 3\}, \{3, 4\}\} = \{1, 2, 3, 4\}$
SUBSET S	l'ensemble des sous-ensembles de S SUBSET $\{1, 2\} = \{\{\}, \{1\}, \{2\}, \{1, 2\}\}$

Actions

Action

Action = prédicat de transition = expression booléenne contenant des constantes, des variables et des variables primées.

Une action n'est pas une affectation.

$x' = x + 1$
$\equiv x' - x = 1$
$\equiv x = x' - 1$
$\equiv (x > 1 \land x'/x = 1 \land x'\%x = 1) \lor (1 = x \land 2 = x')$
$\lor (x = 0 \land x' \in \{y \in Nat : y + 1 = 2 * y\})$

Autres exemples d'actions :

- $x' > x$ ou $x' \in \{x + 1, x + 2, x + 3\}$ (non déterministe)
- $x' \in \{y \in \mathbb{N} : \exists z \in \mathbb{N} : z * y = x \land z\%2 = 0\}$ (non évaluable)
- $x' = y \land y' = x$ (plusieurs variables)

Bégaiement

$[\mathcal{A}]_f \triangleq \mathcal{A} \lor f' = f$, où f est un tuple de variables.

exemple : $[x' = x + 1]_{\langle x, y\rangle} = (x' = x + 1 \lor (\langle x, y\rangle' = \langle x, y\rangle))$
$= (x' = x + 1 \lor (x' = x \land y' = y))$

UNCHANGED

UNCHANGED $e \triangleq e' = e$

Fonctions

- $[X \to Y]$ = ensemble des fonctions de X dans Y.
- f fonction de X dans Y : $f \in [X \to Y]$
- $f[x] \triangleq$ la valeur de f en x.

Définition d'un symbole constant

$f[x \in Nat] \triangleq$ expression utilisant x

Exemple : $Succ[x \in Nat] \triangleq x + 1$

Définition d'une valeur

$[x \in S \mapsto expr]$

Exemples : $[x \in 1..4 \mapsto 2 * x]$, $[x \in \{1, 2, 3, 5, 7, 11\} \mapsto 2 * x + 1]$

Domain

DOMAIN f = domaine de définition de f

Codomaine (range)

$Codomain(f) \triangleq \{f[x] : x \in$ DOMAIN $f\}$

Except

EXCEPT

$[a$ EXCEPT $![i] = v]$ équivalent à
$[j \in$ DOMAIN $a \mapsto$ IF $j = i$ THEN v ELSE $a[j]]$

Enregistrement

Enregistrement

Un enregistrement (record) est une fonction de $[X \to Y]$ où X est un ensemble de chaînes.

Écriture simplifiée :

$$[\text{"}toto\text{"} \mapsto 1, \text{"}titi\text{"} \mapsto 2] \;=\; [toto \mapsto 1, titi \mapsto 2]$$
$$rec[\text{"}toto\text{"}] \;=\; rec.toto$$

N-tuples

n-tuple

Notation : $\langle a, b, c\rangle$.
Un n-tuple est une fonction de domaine $= \{1, \ldots, n\}$:
$\langle a, b, c\rangle[3] = c$

Pratique pour représenter des relations :
$\{\langle x, y\rangle \in X \times Y : R(x, y)\}$.
Exemple : $\{\langle a, b\rangle \in Nat \times Nat : a = 2 * b\}$.

Séquences

Séquences

$Seq(T) \triangleq$ UNION $\{[1 \,..\, n \to T] : n \in Nat\}$
$\triangleq$ ensemble des séquences finies contenant des T.

Opérateurs $Len(s)$, $s \circ t$ (concaténation), $Append(s, e)$, $Head(s)$, $Tail(s)$.

Exemple de définition des opérateurs :

$$s \circ t \triangleq [i \in 1..(Len(s) + Len(t)) \mapsto \text{IF } i \leq Len(s) \text{ THEN } s[i] \text{ ELSE } t[i - Len(s)]]$$
$$Append(s, e) \triangleq s \circ \langle e\rangle$$

Récursivité

- Fonction :
  $fact[n \in Nat] \triangleq$ IF $n = 0$ THEN 1 ELSE $n * fact[n - 1]$
- Opérateur :
  RECURSIVE $fact(\_)$
  $fact(n) \triangleq$ IF $n = 0$ THEN 1 ELSE $n * fact(n - 1)$

Symbole local

LET

Expression : LET $v \triangleq e$ IN f

Équivalent à l'expression f où toutes les occurrences du symbole v sont remplacées par e.
Exemple : LET $i \triangleq g(x)$ IN $f(i)$
$\equiv f(g(x))$

$pythagore(x, y, z) \triangleq$ LET $carre(n) \triangleq n * n$ IN $carre(x) + carre(y) = carre(z)$

Choix déterministe (choix aléatoire)

Opérateur de choix

CHOOSE $x \in S : p \triangleq$ choix arbitraire *déterministe* d'un élément dans l'ensemble S et qui vérifie le prédicat p.

Maximum d'un ensemble

$max[S \in$ SUBSET $Nat] \triangleq$ CHOOSE $m \in S : (\forall p \in S : m \geq p)$

Somme des éléments d'un ensemble

$Somme[S \in$ SUBSET $Nat] \triangleq$
IF $S = \emptyset$ THEN 0
LET $e \triangleq$ CHOOSE $x \in S$: TRUE
IN $e + Somme[S \setminus \{e\}]$

CHOOSE $x \in S : p =$ CHOOSE $x \in S : p$ (aïe)
Pour un ensemble S et une propriété p, l'élément choisi est toujours le même, dans toutes les exécutions et tout au long de celles-ci. Ce n'est pas un sélecteur aléatoire qui donne un élément distinct à chaque appel.

L'équité

Etats récurrents

Ensemble récurrent d'états

Soit $\mathcal{S} = \langle S, I, R\rangle$ un système de transition et $\sigma = \langle s_0 \to \ldots\rangle$ une exécution.
Un ensemble d'états P est récurrent dans σ si :

- cas σ infinie : $\forall i \in \mathbb{N} : \exists j \geq i : s_j \in P$
  (P apparaît une infinité de fois dans σ).
- cas σ finie : l'état final de σ est dans P.

$Inf_S(P, \sigma) \triangleq P$ est un ensemble récurrent d'états dans σ.

Note : on dit aussi *infiniment souvent présent* dans σ.

Transitions récurrentes

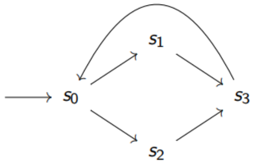
Ensemble récurrent de transitions

Soit $\mathcal{S} = \langle S, I, R\rangle$ un système de transition et $\sigma = \langle s_0 \to \ldots\rangle$ une exécution.
Un ensemble de transitions Q est récurrent dans σ si :

- cas σ infinie : $\forall i \in \mathbb{N} : \exists j \geq i : s_j \to s_{j+1} \in Q$
  (des transitions de Q apparaissent une infinité de fois dans σ).
- cas σ finie : la transition finale de σ est dans Q
  $(\sigma = \langle s_0 \to \ldots \to s \to s'\rangle \wedge s \to s' \in Q)$.

$Inf_T(Q, \sigma) \triangleq Q$ est un ensemble récurrent de transitions dans σ.

Exemple



s_1	récurrent dans	$\langle(s_0 \to s_1 \to s_3)^\omega\rangle$
s_1	récurrent dans	$\langle(s_0 \to s_1 \to s_3 \to s_0 \to s_2 \to s_3)^\omega\rangle$
s_1	pas récurrent dans	$\langle(s_0 \to s_1 \to s_3)^* \to (s_0 \to s_2 \to s_3)^\omega\rangle$
$s_1 \to s_3$	récurrente dans	$\langle(s_0 \to s_1 \to s_3 \to s_0 \to s_2 \to s_3)^\omega\rangle$
$s_1 \to s_3$	pas récurrente dans	$\langle(s_0 \to s_1 \to s_3)^* \to (s_0 \to s_2 \to s_3)^\omega\rangle$

Equité simple sur les états

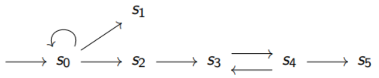
Équité simple

Soit un système de transition $\langle S, I, R\rangle$.
On se donne $F \subseteq S$ un ensemble d'états équitables.
Alors toute exécution σ doit être telle que $Inf_S(F, \sigma)$.

F est récurrent dans σ, i.e. σ contient une infinité d'états dans F (cas σ infini), ou le dernier état de σ est dans F (cas σ fini).

Remarque : l'ensemble F est récurrent, pas nécessairement chaque élément de F. Pour $\mathcal{S} \triangleq i = 0 \wedge \Box((i' = i + 1) \vee (i' = i))$, si on se donne $F \triangleq \{i\%2 = 0\}$, les exécutions $0 \to 1 \to 2^\omega$ et $0 \to 1 \to 2 \to 3 \to 4 \ldots$ sont valides.

Exemple



$$Exec(S) = \langle s_0{}^\omega\rangle, \langle s_0{}^+ \to s_2 \to (s_3 \to s_4)^\omega\rangle, \langle s_0{}^+ \to s_1\rangle, \langle s_0{}^+ \to s_2 \to (s_3 \to s_4)^+ \to s_5\rangle$$

Équité simple	Exécutions
$\{s_0\}$	$\langle s_0{}^\omega\rangle$
$\{s_1, s_4\}$	$\langle s_0{}^+ \to s_2 \to (s_3 \to s_4)^\omega\rangle, \langle s_0{}^+ \to s_1\rangle$
$\{s_1, s_5\}$	$\langle s_0{}^+ \to s_1\rangle, \langle s_0{}^+ \to s_2 \to (s_3 \to s_4)^+ \to s_5\rangle$

Equité multiple

Équité multiple

Soit un système de transition $\langle S, I, R\rangle$.
On se donne un ensemble dénombrable, indexable par un ensemble d'entiers $J = \{0, 1, 2, \ldots\}$, d'ensembles équitables $\{F_i\}_{i \in J}$.
Toute exécution σ doit être telle que $\forall i \in J : Inf_S(F_i, \sigma)$.

Exécutions vérifiant l'équité multiple = intersection des exécutions vérifiant l'équité simple sur chacun des F_i.
$\Rightarrow$ l'équité simple est un cas particulier de l'équité multiple.

Equité conditionnelle

Équité conditionnelle

Soit un système de transition $\langle S, I, R\rangle$.
On se donne deux ensembles F et G.
Toute exécution σ doit être telle que $Inf_S(F, \sigma) \Rightarrow Inf_S(G, \sigma)$.

Si F est récurrent dans σ, alors G doit être récurrent dans σ.

On garde les exécutions ou F n'est pas récurrent et les exécutions ou G est récurrents !!!

Equité simple sur les transitions

L'équité sur les transitions est plus précise que l'équité sur les états. Informellement, une exécution infinie est non équitable vis-à-vis d'une transition si :

- la transition n'apparaît qu'un nombre fini de fois,
- et la transition est continûment faisable (équité faible) ou infiniment souvent faisable (équité forte).

Equité faible

Équité faible

Soit un ST $\langle S, I, R\rangle$ et $F \subseteq R$ un sous-ensemble des transitions.
F est faiblement équitable ssi dans toute exécution σ :

$$Inf_S(S \setminus dom(F), \sigma) \vee Inf_T(F, \sigma)$$

(l'ensemble d'états $S \setminus dom(F)$ est récurrent,
ou l'ensemble de transitions F est récurrent)

Ou, de manière équivalente :

$$\neg Inf_S(S \setminus dom(F), \sigma) \Rightarrow Inf_T(F, \sigma)$$

(si l'ensemble d'états $S \setminus dom(F)$ n'est pas récurrent,
alors l'ensemble de transitions F est récurrent)

L'équité faible exprime que l'on n'a pas le droit de rester indéfiniment dans un ensemble spécifié d'états alors qu'il existe toujours une transition en équité faible qui est exécutable.

Equité forte

Équité forte

Soit un ST $\langle S, I, R\rangle$ et $F \subseteq R$ un sous-ensemble des transitions.
F est fortement équitable ssi dans toute exécution σ :

$$\neg Inf_S(dom(F), \sigma) \vee Inf_T(F, \sigma)$$

l'ensemble d'états $dom(F)$ n'est pas récurrent,
ou l'ensemble de transitions F est récurrent.

Ou, de manière équivalente :

$$Inf_S(dom(F), \sigma) \Rightarrow Inf_T(F, \sigma)$$

si l'ensemble d'états $dom(F)$ est récurrent,
alors l'ensemble de transitions F est récurrent.

L'équité forte exprime que si l'on passe infiniment souvent dans un ensemble d'états où des transitions de r sont exécutables, alors une transition de r finit par être exécutée.

Logique Temporelle Linéaire (LTL)

Syntaxe

formule	nom	interprétation
s		le premier état de la trace est s
$\neg P$		
$P \vee Q$		
$P \wedge Q$		
$\bigcirc P$	*next*	P est vrai à l'instant suivant
$\Box P$	*always*	P est toujours vrai i.e. à tout instant à partir de l'instant courant
$\Diamond P$	*eventually*	P sera vrai (dans le futur)
$P \mathcal{U} Q$	*until*	Q sera vrai, et en attendant P reste vrai
$P \leadsto Q$	*leadsto*	quand P est vrai, alors Q est vrai plus tard

- Losange = il existe un moment (futur)
- Carré = en tout point ou état ou moment
- s0Ws1 (Waiting for) = s0 reste vrai jusqu'à ce que s1 le devient

Opérateurs minimaux

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

- $\Diamond P \triangleq True\ \mathcal{U} P$
- $\Box P \triangleq \neg\Diamond\neg P$
- $P \leadsto Q \triangleq \Box(P \Rightarrow \Diamond Q)$

Autres

Opérateurs complémentaires

- Opérateur *waiting-for* (ou *unless* ou *weak-until*)
  $P \mathcal{W} Q \triangleq \Box P \vee P \mathcal{U} Q$
  Q finit peut-être par être vrai et en attendant P reste vrai
- Opérateur *release*
  $P \mathcal{R} Q \triangleq Q \mathcal{U}(P \wedge Q)$
  Q reste vrai jusqu'à ce que P le devienne.

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
$\diamondminus P$	*once*	P a été vrai dans le passé
$P \mathcal{S} Q$	*since*	Q a été vrai dans le passé et P est resté vrai depuis la dernière occurrence de Q
$P \mathcal{B} Q$	*back-to*	P est vrai depuis la dernière occurrence de Q, ou depuis l'instant initial si Q n'a jamais été vrai

LTL en TLA+

Expressions logiques

Expressions de LTL avec $\Box$, $\Diamond$, $\leadsto$ (leads-to) et variables primées + quantificateurs $\forall, \exists$.

Pas de $\mathcal{U}$, ni de $\mathcal{W}$, mais :

$$\Box(p \Rightarrow (p\mathcal{W}q)) \quad = \quad \Box(p \Rightarrow (p' \vee q))$$
$$\Box(p \Rightarrow (p\mathcal{U}q)) \quad = \quad \Box(p \Rightarrow (p' \vee q)) \wedge \Box(p \Rightarrow \Diamond q)$$

Logique temporelle arborescente (CTL)

Modèles

Une formule CTL se rapporte toujours à un état donné s d'un système, duquel partent des traces *Traces*(s).
Les états de S constituent les modèles de cette logique.

La différence (syntaxiquement parlant) avec LTL réside dans l'apparition dans les opérateurs temporels de quantificateurs de traces.

Syntaxe

Quantification universelle

formule	interprétation (pour s un état) pour toute trace partant de s
$\forall\bigcirc P$	P est vrai à l'instant suivant
$\forall\Box P$	P est toujours vrai à chaque état
$\forall\Diamond P$	P finit par être vrai (dans le futur)
$P \forall\mathcal{U} Q$	Q finit par être vrai, et en attendant P reste vrai

Quantification existentielle

formule	interprétation (pour s un état) pour au moins une trace partant de s
$\exists\bigcirc P$	P est vrai à l'instant suivant
$\exists\Box P$	P est toujours vrai à chaque état
$\exists\Diamond P$	P finit par être vrai (dans le futur)
$P \exists\mathcal{U} Q$	Q finit par être vrai, et en attendant P reste vrai

Opérateur complémentaire waiting-for

$P \exists\mathcal{W} Q \triangleq \exists\Box P \vee P \exists\mathcal{U} Q$
$P \forall\mathcal{W} Q \not\triangleq \forall\Box P \vee P \forall\mathcal{U} Q$ – *trop fort*
$\triangleq \neg(\neg Q \exists\mathcal{U}(\neg P \wedge \neg Q))$

$\exists\Diamond P$ - Il existe une trace ou on peut atteindre l'état P

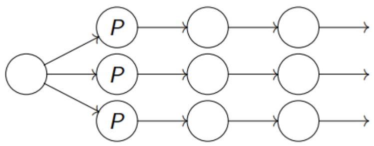
$\forall\Box\forall\Diamond s_2$ - Quel que soit le point où on est, on passera toujours par s2 (tous les traces),

$\forall\Diamond\exists\Diamond s_1$ - Pour chaque trace, il existe un point, où de ce point il existe une trace qui passe par s1

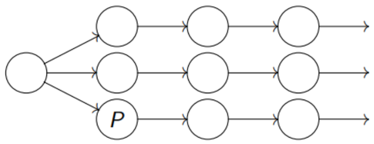
$\forall\Box\exists\Diamond s_1$ - Quel que soit le point où on est, il existe une trace qui passera par s1,

Sémantique

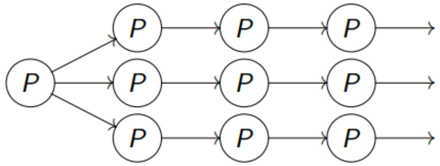
$\forall\bigcirc P$



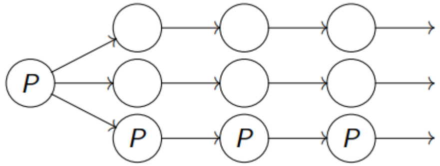
$\exists\bigcirc P$



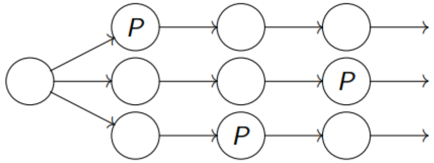
$\forall\Box P$



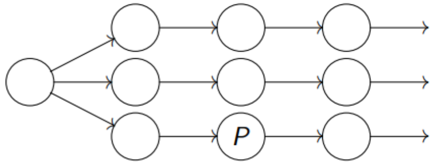
$\exists\Box P$



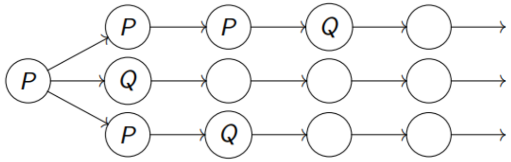
$\forall\Diamond P$



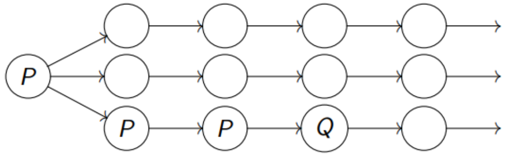
$\exists\Diamond P$



$P \forall\mathcal{U} Q$



$P \exists\mathcal{U} Q$



Mes simplifications

simple

QlqS carré = depuis n'importe et tous les états
Existe carré = depuis n'importe et tous les états d'une trace
QlqS losange = existe un état dans toutes les traces
Existe losange = existe un état dans une trace

Composé

QlqS carré QlqS losange **s** = depuis n'importe quel état, toutes les traces mènent vers s
QlqS carré Existe losange **s** = depuis n'importe quel état, il existe une trace qui mène vers s
QlqS losange QlqS carré **s** = il existe un état dans toutes les traces qui boucle toujours sur s
Existe losange QlqS carré **s** = il existe un état dans une trace qui boucle sur s
QlqS losange Existe losange **s** = il existe un état s' dans toutes les traces dont une trace mène vers s
Existe losange QlqS losange **s** = il existe un état s' dans une trace, dont toutes les traces mènent vers s
QlqS losange Existe carré **s** = il existe un état s dans toutes les traces dont la trace boucle sur soi même

Questions de cours

Changer la valeur de w

Donner une action qui change w en une valeur quelconque de y.

$w' \in y$ ou $\exists v \in y : w' = v$

$(\wedge \text{ UNCHANGED } \langle x, y, z \rangle)$

Domaine d'une fonction

Donner une propriété temporelle qui dit que le domaine de x est toujours inclus dans y.

$\Box((\text{DOMAIN } x) \subseteq y)$

Appartenance à deux ensembles différents

Donner un prédicat qui dit que w est dans y et dans z.

$w \in y \wedge w \in z$ ou $w \in y \cap z$

Changer la valeur d'une fonction

Donner une action qui change la valeur de x en 0 pour prendre la valeur de w.

$x' = [x \text{ EXCEPT } ![0] = w]$ $(\wedge \text{ UNCHANGED } \langle w, y, z \rangle)$

Ensemble d'entier

Donner une propriété temporelle qui dit que y est toujours un ensemble d'entiers.

$\Box(y \in \text{SUBSET } Nat)$ ou $\Box(y \subseteq Nat)$ ou moins élégant $\Box(\forall i \in y : i \in Nat)$

Ajouter un élément à un ensemble

Donner une action qui ajoute w à l'ensemble y

$y' = y \cup \{w\}$ $(\wedge \text{ UNCHANGED } \langle w, x \rangle)$

Domaine d'une fonction d'une autre manière (Prédicat)

Donner un prédicat qui dit que toutes les valeurs de x sont supérieures ou égales à 2.

$\forall i \in \text{DOMAIN } x : x[i] \geq 2$

($\forall i \in Nat : x[i] \geq 2$ *est approximatif : une fonction dans* $[X \to Y]$ *a un domaine inclus dans* X *mais pas nécessairement tout* X*, et pareil pour le codomaine)*

Echange de valeur entre deux variables

Donner une action qui échange le contenu des variables x et y.

$x' = y \wedge y' = x$ $(\wedge \text{ UNCHANGED } z)$

Partition d'ensemble

Donner un prédicat qui dit que y et z forment une partition de x (pas d'éléments en commun et l'union forme x).

$x = y \cup z \wedge y \cap z = \emptyset$

Deux variables ne sont jamais égales

Donner une propriété temporelle qui dit que x et y ne sont jamais égaux

$\Box(x \neq y)$

Définir une fonction

Donner une expression correspondant à une fonction définie sur x vers BOOLEAN, et qui vaut TRUE au point où la valeur est dans y ou z, et FALSE ailleurs.

$[i \in x \mapsto i \in (y \cup z)]$

Eléments d'un tableau S appartient à un ensemble T

Donner un prédicat qui exprime que tous les éléments de S sont dans T.

$\{S[i] : i \in 1..Len(S)\} \subseteq T$

$\forall i \in 1..Len(S) : S[i] \in T$

Restriction d'un ensemble

Donner une action qui enlève de T les éléments plus petits ou égaux à x.

$T' = T \setminus 0..x$

$T' = \{i \in T : i > x\}$

Taille d'une séquence (Tableau)

Donner une propriété temporelle qui dit que S contient toujours au moins deux éléments.

$\Box(Len(S) >= 2)$

Appartenance à une séquence et un ensemble à la fois

Donner une propriété temporelle qui dit que x n'est jamais à la fois dans S et T.

$\Box(\neg(x \in T \wedge x \in Range(S)))$

$\Box(\neg(x \in T \wedge \exists i \in 1..Len(S) : S[i] = x))$

Sous-ensemble conditionnel

Donner une expression qui représente le sous-ensemble des éléments de S qui sont plus grands que x et plus petits que y.

$\{a \in S : x \leq a \wedge a \leq y\}$ ou $S \cap x..y$

Ajout conditionnel d'un élément à un ensemble

Donner une action qui ajoute x à S à condition que y soit déjà dans S.

$y \in S \wedge S' = S \cup \{x\} \wedge \text{UNCHANGED } \langle x, y, f \rangle$

Réalisation d'une action dans le futur (condition serait vrai)

Donner une propriété temporelle qui exprime qu'une valeur de la fonction f sera un jour supérieure à x.

$\Diamond(\exists i \in S : f[i] > x)$

Sous-ensemble d'un ensemble

Donner une action qui change A en un sous-ensemble de B, à condition que A et B soient disjoints.

$A \cap B = \emptyset \wedge A' \in \text{SUBSET } B$ (ou $\wedge A' \subseteq B$)

Ensemble des images par une fonction

Donner une expression représentant l'ensemble des x de A tels que $f[x]$ est dans B.

$\{x \in A \mid f[x] \in B\}$ ou $f^{-1}(B) \cap A$

Ensemble des images est paires

Donner une propriété temporelle qui dit que la fonction f ne prend que des valeurs paires.

$\Box(\forall x \in A : f[x]\%2 = 0)$

$\Box(\{x \in A : f[x]\%2 \neq 0\} = \emptyset)$

Ensemble ne décroit pas (inclusion)

Donner une propriété temporelle qui dit que l'ensemble A ne décroît pas.

selon l'interprétation :

$\Box(A \subseteq A')$

$\Box(Cardinality(A) \leq Cardinality(A'))$

$\forall k \in Nat : \Box(Cardinality(A) = k \Rightarrow \Box(Cardinality(A) \geq k))$

Ensemble de solution f(x) = 0 non vide

Donner un prédicat testant si au plus un entier x de S est tel que $f[x] = 0$.

$Cardinality(\{x \in S : f[x] = 0\}) \leq 1$

Ensemble maximum d'une fonction

Donner une expression représentant l'ensemble des entiers x de S tels que $f[x]$ est l'élément maximum de l'ensemble $f(D)$.

$\{x \in S : (\forall y \in D : f[x] \geq f[y])\}$

$\{x \in S : f[x] = max(\{f[y] : y \in D\})\}$

Action d'ajout à un ensemble

Donner une action ajoutant à S les entiers x de D tels que $f[x] = 0$.

$S' = S \cup \{x \in D : f[x] = 0\}$

$S' = S \cup f^{-1}[0]$

Ensemble non vide

Donner une propriété temporelle exprimant que l'ensemble $f(S)$ n'est jamais vide.

$\Box(\{f[x] : x \in S\} \neq \emptyset)$ ou $\Box(S \neq \emptyset)$