

UPMC Paris Universitاس
Paradigmes de Programmation Concurrente MU5IN553



Cours 4 - Hennessy-Milner Logic

Carlos Agon

24 octobre 2020

Plan du cours

- Logiques modales
- Logique de Hennessy-Milner
- Logique de Hennessy-Milner avec récursion

Introduction

- Voir un terme de CCS comme une spécification
- On peut faire de *queries* au système pour voir si elles sont vérifiées
- Pour vérifier les propriétés d'un processus donné, de fois il est plus facile de le faire en explorant l'espace des états du processus que en cherchant des équivalences.
- On a besoin d'un langage pour exprimer les propriétés (syntaxe, sémantique et un algorithme de vérification)

Logiques modales

- Connue par les grecques, mais formalisée fin XIX siècle
- Introduction des opérateurs
 - L (il est nécessaire) \square
 - M (il est possible) \diamond
 - liées par les formules : $Lp = \neg M\neg p$ et $Mp = \neg L\neg p$

Sémantique



Saul Kripke

- La sémantique est donnée par un modèle $I = (W, R, s)$ avec W un ensemble non vide de mondes. R une relation sur $W \times W$ et $s : (w, p) \rightarrow \{tt, ff\}$ avec p une proposition et $w \in W$
 - On note $I_{w,s}f$ le fait que s satisfait la formule f dans le monde w du modèle I
- 1 $I_{w,s} f \iff s(w, f)$ est vrai
 - 2 $I_{w,s} A \vee B \iff I_{w,s} A \vee I_{w,s} B$
 - 3 $I_{w,s} \neg A \iff \neg I_{w,s} A$
 - 4 $I_{w,s} \Box A \iff I_{w',s} A \forall w' \text{ tq. } wRw'$
 - 5 $I_{w,s} \Diamond A \iff \exists w' \text{ tq. } wRw' \text{ et } I_{w',s} A$

Logiques temporelles

Dans le cas des logiques temporelles L et M sont remplacés par P et F

Fp : p sera vrai à tout instant future

Pp : p a toujours été vrai dans le passé

W représente les points du temps

R exprime la notion de précédence temporelle

- $I_{w,s} FA \iff I_{w',s} A \forall w' \text{ tq. } wRw'$
- $I_{w,s} PA \iff I_{w',s} A \forall w' \text{ tq. } w'Rw$
- Sp : p sera vrai à un instant future : $\neg F\neg p$
- Ep : p a été vrai à un instant dans le passé : $\neg P\neg p$

Systèmes classiques par rapport à R

- K : pas de restriction $F(p \Rightarrow q) \Rightarrow (Fp \Rightarrow Fq)$
- T : réflexive $Fp \Rightarrow p$
- B : symétrique $p \Rightarrow FPp$
- S4 : transitive $Fp \Rightarrow FFp$

Hennessy-Milner logic (HML)

Définition

L'ensemble \mathcal{M} des formules Hennessy-Milner sur un ensemble d'actions Act est donné par :

$F, G ::= tt \mid ff \mid F \wedge G \mid F \vee G \mid \langle a \rangle F \mid [a]F$

avec $a \in Act$, $tt = \text{vrai}$, $ff = \text{faux}$

- Un processus P satisfait $F \vee G$ ssi P satisfait F ou P satisfait G
- Un processus P satisfait $F \wedge G$ ssi P satisfait F et P satisfait G
- Un processus P satisfait $\langle a \rangle F$ ssi il existe une transition a à partir de P menant à un processus satisfaisant F
- Un processus P satisfait $[a]F$ ssi toute transition a partant de P mène à un processus satisfaisant F
- Tous les processus satisfont tt
- Aucun processus satisfait ff

Hennessy-Milner logic (notation)

En fait c'est une logique modale avec :

◇ *possiblement*

$\langle a \rangle F$: Il est *possible* de s'engager dans une action a et satisfaire F

□ *nécessairement*

$[a]F$: Peut importe comme on s'engage dans une action a , F sera *nécessairement* satisfaite

- Si $A = \{a_1, \dots, a_n\} \subseteq \text{Act}$, on utilise

$\langle A \rangle F$ pour $\langle a_1 \rangle F \vee, \dots, \vee \langle a_n \rangle F$

$\langle \bar{A} \rangle F$ pour $\langle \text{Act} \setminus A \rangle F$

$[A]F$ pour $[a_1]F \wedge, \dots, \wedge [a_n]F$

$[\bar{A}]F$ pour $[\text{Act} \setminus A]F$

$\langle . \rangle F$ pour $\langle \text{Act} \rangle F$

$[.]F$ pour $[\text{Act}]F$

Sémantique d'une formule

La sémantique est paramétrisée par un LTS ($Proc, Act, \{ \xrightarrow{a} \mid a \in Act \}$)

On dénote $\llbracket F \rrbracket$ l'ensemble de processus dans $Proc$ qui satisfont F

Définition

On définit $\llbracket F \rrbracket \subseteq Proc$ pour $F \in \mathcal{M}$ par :

$$\llbracket tt \rrbracket = Proc$$

$$\llbracket ff \rrbracket = \emptyset$$

$$\llbracket F \wedge G \rrbracket = \llbracket F \rrbracket \cap \llbracket G \rrbracket$$

$$\llbracket F \vee G \rrbracket = \llbracket F \rrbracket \cup \llbracket G \rrbracket$$

$$\llbracket \langle a \rangle F \rrbracket = \langle .a. \rangle \llbracket F \rrbracket$$

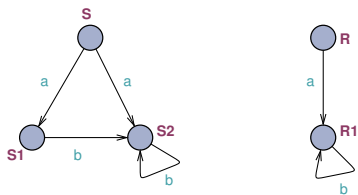
$$\llbracket [a] F \rrbracket = [.a.] \llbracket F \rrbracket$$

avec

$$\langle .a. \rangle S = \{ p \in Proc \mid \exists p' \in S \text{ tq. } p \xrightarrow{a} p' \}$$

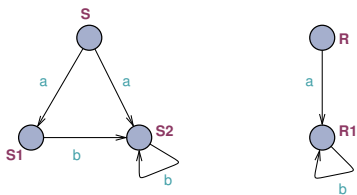
$$[.a.] S = \{ p \in Proc \mid \forall p' \text{ tq. } p \xrightarrow{a} p' \Rightarrow p' \in S \}$$

Example 1



$$\langle .a. \rangle \{S1, R1\} =$$

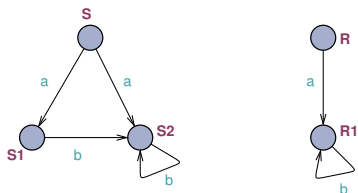
Example 1



$$\langle .a. \rangle \{S1, R1\} = \{S, R\}$$

$$[.a.] \{S1, R1\} =$$

Exemple 1



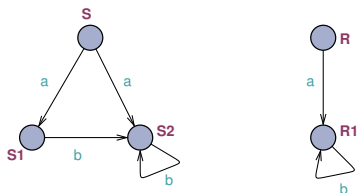
$$\langle .a. \rangle \{S1, R1\} = \{S, R\}$$

$$[.a.] \{S1, R1\} = \{S1, S2, R, R1\}$$

Pour S1 il est vrai que $\forall p' tq. S1 \xrightarrow{a} p' \Rightarrow p' \in \{S1, R1\}$

$$\langle .b. \rangle \{S1, R1\} =$$

Exemple 1



$$\langle .a. \rangle \{S1, R1\} = \{S, R\}$$

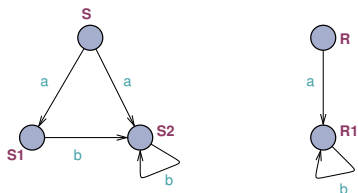
$$[.a.] \{S1, R1\} = \{S1, S2, R, R1\}$$

Pour S1 il est vrai que $\forall p' tq. S1 \xrightarrow{a} p' \Rightarrow p' \in \{S1, R1\}$

$$\langle .b. \rangle \{S1, R1\} = \{R1\}$$

$$[.b.] \{S1, R1\} =$$

Exemple 1



$$\langle .a. \rangle \{S1, R1\} = \{S, R\}$$

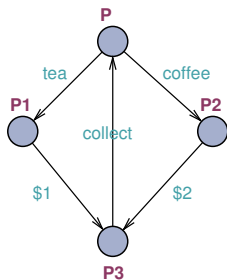
$$[.a.] \{S1, R1\} = \{S1, S2, R, R1\}$$

Pour S1 il est vrai que $\forall p' tq. S1 \xrightarrow{a} p' \Rightarrow p' \in \{S1, R1\}$

$$\langle .b. \rangle \{S1, R1\} = \{R1\}$$

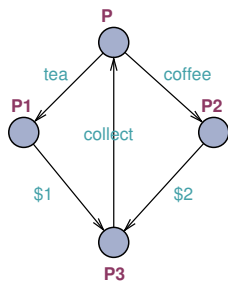
$$[.b.] \{S1, R1\} = \{S, R, R1\}$$

Exemple 2



Comment formuler "Le client peut choisir un café" ?

Exemple 2

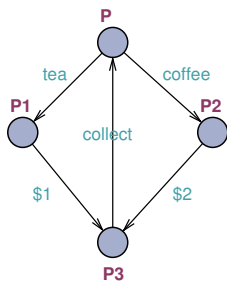


Comment formuler "Le client peut choisir un café" ?

Il peut faire une transition $\langle \text{coffee} \rangle$

donc la formule est $\langle \text{coffee} \rangle F$, mais c'est qui F ?

Exemple 2



Comment formuler "Le client peut choisir un café" ?

Il peut faire une transition $\langle \text{coffee} \rangle$

donc la formule est $\langle \text{coffee} \rangle F$, mais c'est qui F ?

($F = tt$) Il est possible de s'engager dans une action *coffee* et satisfaire tt

$\llbracket \langle \text{coffee} \rangle tt \rrbracket = \langle .\text{coffee}. \rrbracket \llbracket tt \rrbracket = \langle .\text{coffee}. \rrbracket \text{Proc} =$

$\{p \mid \exists p' \in \text{Proc} \text{ tq. } p \xrightarrow{\text{coffee}} p'\} = \{P\}$

Exemple 2

Le client ne peut pas choisir du tea.

Exemple 2

Le client ne peut pas choisir du tea.

$[tea]ff$

Aucun processus satisfait ff donc les seuls processus qui marchent sont ceux qui n'ont pas de transition tea.

$$\begin{aligned} \llbracket [tea]ff \rrbracket &= [.tea.]\llbracket ff \rrbracket = [.tea.]\emptyset = \{p \mid \text{si } p \xrightarrow{tea} p' \text{ alors } p \in \emptyset\} = \\ &= \{p \mid p \not\xrightarrow{tea}\} = \{P1, P2, P3\} \end{aligned}$$

Exemple 2

Que veut dire `[coffee]<biscuit>tt` ?

Comment écrire :

Le client est prêt pour choisir un tea ou un coffee

Exemple 2

Que veut dire $[coffee]\langle biscuit \rangle tt$?

Comment écrire :

Le client est prêt pour choisir un tea ou un coffee

$\langle coffee \rangle tt \vee \langle tea \rangle tt$

Le client est prêt pour choisir un tea, mais pas un coffee

Exemple 2

Que veut dire $[coffee]\langle biscuit \rangle tt$?

Comment écrire :

Le client est prêt pour choisir un tea ou un coffee

$\langle coffee \rangle tt \vee \langle tea \rangle tt$

Le client est prêt pour choisir un tea, mais pas un coffee

$[coffee]ff \wedge \langle tea \rangle tt$

Le client peut choisir un tea après avoir choisit deux coffee

Exemple 2

Que veut dire $[coffee]\langle biscuit \rangle tt$?

Comment écrire :

Le client est prêt pour choisir un tea ou un coffee

$\langle coffee \rangle tt \vee \langle tea \rangle tt$

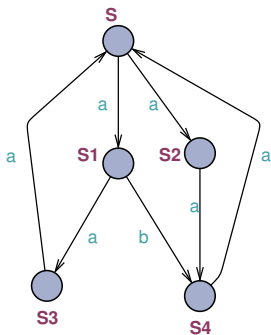
Le client est prêt pour choisir un tea, mais pas un coffee

$[coffee]ff \wedge \langle tea \rangle tt$

Le client peut choisir un tea après avoir choisit deux coffee

$[coffee][coffee]\langle tea \rangle tt$

Exemple 3



On va écrire $p \models F \Leftrightarrow p \in \llbracket F \rrbracket$.

Est-ce que $s \models \langle a \rangle (\langle a \rangle tt \wedge \langle b \rangle tt)$?

$s \models [a][a]\langle b \rangle tt$?

La negation

Pour chaque formule $F \in \mathcal{M}$ il y a une formule équivalente à la négation de F , F^C définie par :

- $tt^C = ff$, $ff^C = tt$
- $(F \wedge G)^C = F^C \vee G^C$
- $(F \vee G)^C = F^C \wedge G^C$
- $(\langle a \rangle F)^C = [a]F^C$
- $([a]F)^C = \langle a \rangle F^C$

En particulier :

$$(\langle a \rangle tt)^C = [a]ff$$

$$([a]ff)^C = \langle a \rangle tt$$

Exemple 3

Trouver une formule qui soit satisfaite par $a.(b.0 + c.0)$, mais pas pour $a.b.0 + a.c.0$?

Exemple 3

Trouver une formule qui soit satisfaite par $a.(b.0 + c.0)$, mais pas pour $a.b.0 + a.c.0$?

$[a](\langle b \rangle tt \wedge \langle c \rangle tt)$

Trouver une formule qui soit satisfaite par $a.(b.c.0 + b.c.0)$, mais pas pour $a.b.c.0 + a.b.c.0$?

Théorème de Hennessy-Milner

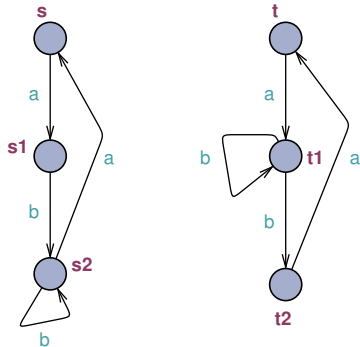
Définition : LTS à image finie

Un état est à image finie si pour toute a l'ensemble $\{P' \mid P \xrightarrow{a} P'\}$ est fini.
Un LTS est à image finie si tous ses états le sont.

Théorème : Hennessy-Milner

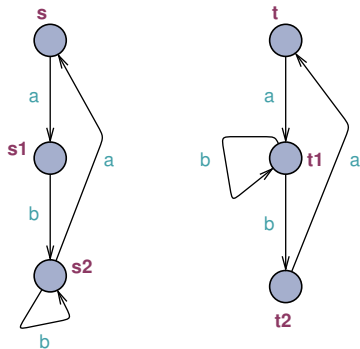
Soit $(\text{Proc}, \text{Act}, \{\xrightarrow{a} \mid a \in \text{Act}\})$ un LTS à image finie et $P, Q \in \text{Proc}$. Alors $P \sim Q$ ssi P et Q satisfont exactement les mêmes formules dans la logique de Hennessy-Milner.

Exercice



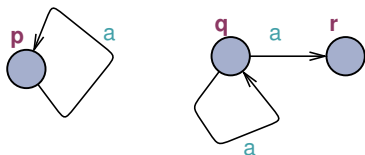
Prouver que $s \not\sim t$

Exercice



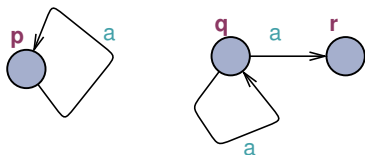
Prouver que $s \not\sim t$
 $[a][b](\langle a \rangle tt \wedge \langle b \rangle)$

Formules récursives



Après avoir fait a p peut toujours faire une autre a (ce n'est pas vrai pour q)
 $p \models [a]\langle a \rangle tt$, mais $q \not\models [a]\langle a \rangle tt$

Formules récursives



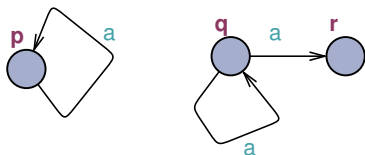
Après avoir fait a p peut toujours faire une autre a (ce n'est pas vrai pour q)
 $p \models [a]\langle a \rangle tt$, mais $q \not\models [a]\langle a \rangle tt$

On aimerait écrire les formules :

$$Inv(\langle a \rangle tt) = \langle a \rangle tt \wedge [a]\langle a \rangle tt \dots \wedge [a]^n \langle a \rangle tt \wedge \dots$$

Toujours on peut faire une a transition (Invariante)

Formules récursives



Après avoir fait a p peut toujours faire une autre a (ce n'est pas vrai pour q)
 $p \models [a]\langle a \rangle tt$, mais $q \not\models [a]\langle a \rangle tt$

On aimerait écrire les formules :

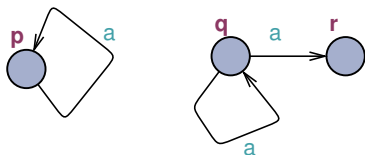
$$Inv(\langle a \rangle tt) = \langle a \rangle tt \wedge [a]\langle a \rangle tt \dots \wedge [a]^n \langle a \rangle tt \wedge \dots$$

Toujours on peut faire une a transition (Invariante)

$$Pos([a]ff) = [a]ff \vee \langle a \rangle [a]ff \dots \vee \langle a \rangle^n [a]ff \vee \dots$$

On peut refuser a tout suite ou après un certain nombre de a (Propriété)

Formules récursives



Après avoir fait a p peut toujours faire une autre a (ce n'est pas vrai pour q)
 $p \models [a]\langle a \rangle tt$, mais $q \not\models [a]\langle a \rangle tt$

On aimerait écrire les formules :

$$Inv(\langle a \rangle tt) = \langle a \rangle tt \wedge [a]\langle a \rangle tt \dots \wedge [a]^n \langle a \rangle tt \wedge \dots$$

Toujours on peut faire une a transition (Invariante)

$$Pos([a]ff) = [a]ff \vee \langle a \rangle [a]ff \dots \vee \langle a \rangle^n [a]ff \vee \dots$$

On peut refuser a tout suite ou après un certain nombre de a (Propriété)

$$X \equiv \langle a \rangle tt \wedge [a]X$$

$$Y \equiv [a]ff \vee \langle a \rangle Y$$

HML avec récursion

Le sense S d'une formule X par rapport à un LTS est donné par l'ensemble de processus dans P qui satisfont X .

- $X \equiv \langle a \rangle tt \wedge [a]X$
 $S = \langle .a \rangle Proc \cap [.a.]S$
 $S = \emptyset$ est un solution, mais il y a d'autres ...
 $S = \{p\}$ c'est celle qu'on cherche, alors on va dire que :
 $X \equiv^{max} \langle a \rangle tt \wedge [a]X$

HML avec récursion

- $Y \equiv [a]ff \vee \langle a \rangle Y$

$$S = [.a.]\emptyset \cup \langle .a. \rangle S$$

il y a $\{p, q, r\}$ comme solution, mais nous on ne veut pas p

$S = \{q, r\}$ c'est celle qu'on cherche, alors on va dire :

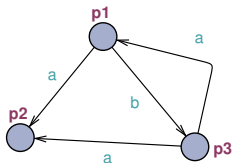
$$Y \equiv^{min} [a]ff \vee \langle a \rangle Y$$

Syntaxe et sémantique de $M_{\{X\}}$

Logique de Hennessy-Milner avec une variable X

- $F, G ::= X | tt | ff | F \wedge G | F \vee G | \langle a \rangle F | [a] F$
- Sémantiquement, une formule F (qui peut contenir une variable X) est interprétée comme une fonction $\mathcal{O}_F : \mathcal{P}(Proc) \rightarrow \mathcal{P}(Proc)$ que pour un ensemble de processus satisfaisant X retourne un ensemble de processus que satisfont F

Exemple

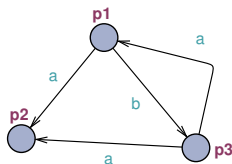


Soit $F = \langle a \rangle X$

$\mathcal{O}_{\langle a \rangle X}(\{p1\}) = \langle .a. \rangle \{p1\} = \{p3\}$

si X est satisfaite par p_1 alors $\langle a \rangle X$ sera satisfaite par p_3

Exemple



Soit $F = \langle a \rangle X$

$$\mathcal{O}_{\langle a \rangle X}(\{p1\}) = \langle .a. \rangle \{p1\} = \{p3\}$$

si X est satisfaite par $p1$ alors $\langle a \rangle X$ sera satisfaite par $p3$

$$\mathcal{O}_{\langle a \rangle X}(\{p1, p2\}) = \langle .a. \rangle \{p1, p2\} = \{p1, p3\}$$

si X est satisfaite par $p1$ et $p2$ alors $\langle a \rangle X$ sera satisfaite par $p1$ et $p3$

Sémantique de $M_{\{X\}}$

- $\mathcal{O}_X(S) = S$
- $\mathcal{O}_{tt}(S) = \text{proc}$
- $\mathcal{O}_{ff}(S) = \emptyset$
- $\mathcal{O}_{F_1 \wedge F_2}(S) = \mathcal{O}_{F_1}(S) \cap \mathcal{O}_{F_2}(S)$
- $\mathcal{O}_{F_1 \vee F_2}(S) = \mathcal{O}_{F_1}(S) \cup \mathcal{O}_{F_2}(S)$
- $\mathcal{O}_{\langle a \rangle F}(S) = \langle .a. \rangle \mathcal{O}_F(S)$
- $\mathcal{O}_{[a]F}(S) = [.a.] \mathcal{O}_F(S)$

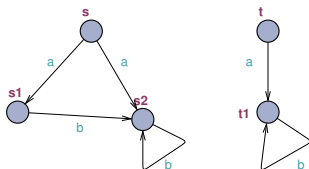
Points fixes

\mathcal{O}_{F_X} est monotone et continue à gauche (ou à droite) et on est sur un treille complet ($\top = Proc$ et $\perp = \emptyset$)

$\text{FIX } \mathcal{O}_{F_X} = (\mathcal{O}_{F_X})^M Proc$ et

$\text{fix } \mathcal{O}_{F_X} = (\mathcal{O}_{F_X})^m \emptyset$

Points fixes



$$X \equiv^{max} \langle b \rangle tt \wedge [b]X$$

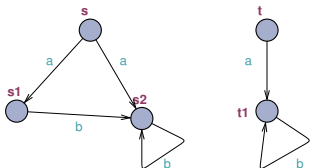
L'ensemble de processus qui satisfait cette équation est la plus grande solution à l'équation :

$$[[X]] = \langle b \rangle tt \wedge [b][[X]]$$

$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(S) = \mathcal{O}_{\langle b \rangle tt}(S) \cap \mathcal{O}_{[b]X}(S)$$

$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(S) = (\langle .b. \rangle \{s, s1, s2, t, t1\}) \cap ([.b.]S)$$

Points fixes



$$X \equiv^{max} \langle b \rangle tt \wedge [b]X$$

L'ensemble de processus qui satisfait cette équation est la plus grande solution à l'équation :

$$[[X]] = \langle b \rangle tt \wedge [b][[X]]$$

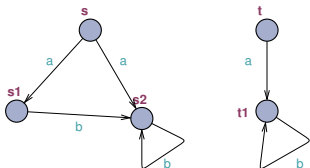
$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(S) = \mathcal{O}_{\langle b \rangle tt}(S) \cap \mathcal{O}_{[b]X}(S)$$

$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(S) = (\langle .b. \rangle \{s, s1, s2, t, t1\}) \cap ([.b.]S)$$

Pour trouver la solution on itère en partant de \top

$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(\{s, s1, s2, t, t1\}) = (\langle .b. \rangle \{s, s1, s2, t, t1\}) \cap ([.b.]\{s, s1, s2, t, t1\})$$

Points fixes



$$X \equiv^{\max} \langle b \rangle tt \wedge [b]X$$

L'ensemble de processus qui satisfait cette équation est la plus grande solution à l'équation :

$$[[X]] = \langle b \rangle tt \wedge [b][[X]]$$

$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(S) = \mathcal{O}_{\langle b \rangle tt}(S) \cap \mathcal{O}_{[b]X}(S)$$

$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(S) = (\langle .b. \rangle \{s, s1, s2, t, t1\}) \cap ([.b.]S)$$

Pour trouver la solution on itère en partant de \top

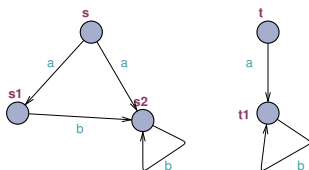
$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(\{s, s1, s2, t, t1\}) = (\langle .b. \rangle \{s, s1, s2, t, t1\}) \cap ([.b.] \{s, s1, s2, t, t1\})$$

$$= \{s2, s1, t1\} \cap \{s, s1, s2, t, t1\}$$

$$= \{s2, s1, t1\}$$

$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(\{s2, s1, t1\}) = (\langle .b. \rangle \{s, s1, s2, t, t1\}) \cap ([.b.] \{s2, s1, t1\})$$

Points fixes



$$X \equiv^{\max} \langle b \rangle tt \wedge [b]X$$

L'ensemble de processus qui satisfait cette équation est la plus grande solution à l'équation :

$$[[X]] = \langle b \rangle tt \wedge [b][[X]]$$

$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(S) = \mathcal{O}_{\langle b \rangle tt}(S) \cap \mathcal{O}_{[b]X}(S)$$

$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(S) = (\langle .b. \rangle \{s, s1, s2, t, t1\}) \cap ([.b.]S)$$

Pour trouver la solution on itère en partant de \top

$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(\{s, s1, s2, t, t1\}) = (\langle .b. \rangle \{s, s1, s2, t, t1\}) \cap ([.b.] \{s, s1, s2, t, t1\})$$

$$= \{s2, s1, t1\} \cap \{s, s1, s2, t, t1\}$$

$$= \{s2, s1, t1\}$$

$$\mathcal{O}_{\langle b \rangle tt \wedge [b]X}(\{s2, s1, t1\}) = (\langle .b. \rangle \{s, s1, s2, t, t1\}) \cap ([.b.] \{s2, s1, t1\})$$

$$= \{s2, s1, t1\} \cap \{s, s1, s2, t, t1\}$$

$$= \{s2, s1, t1\}$$