Logiques temporelles basées sur actions

Radu Mateescu

INRIA Rhône-Alpes / VASY

655, avenue de l'Europe

38330 Montbonnot Saint Martin

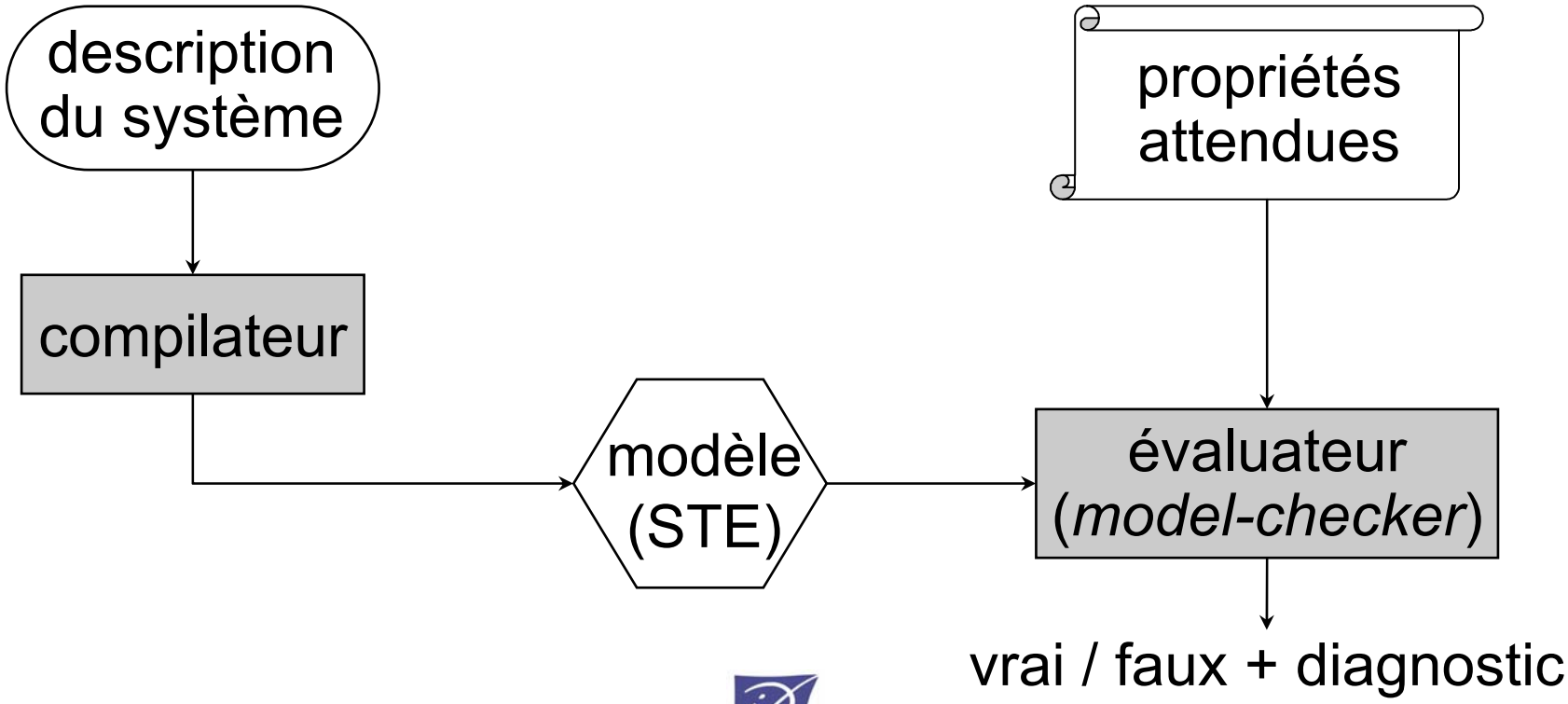
Plan

- Introduction
- Logiques modales
- Logiques arborescentes
- Logiques régulières
- Logiques de point fixe

Vérification basée sur les modèles

Objectif : vérifier qu'un système parallèle satisfait ses propriétés de bon fonctionnement.

Approche basée sur les modèles (*model-checking*) :



Spécification des propriétés

Les propriétés de bon fonctionnement du système (service attendu) peuvent être spécifiées au moyen de différents formalismes.

Deux approches sont principalement utilisées :

- logiques temporelles (approche « bi-langage »)

  $\Rightarrow$ vérification par *évaluation* de formules logiques sur un STE

- automates finis (approche « mono-langage »)

  $\Rightarrow$ vérification par *comparaison* de deux STEs (bisimulations, préordres)

Avantages des LT pour la spécification

- LT : formalismes permettant de décrire l'ordonnancement des actions (événements) au cours de l'exécution d'un programme parallèle
- Spécification en LT = liste de formules logiques, chacune exprimant une propriété du programme
- Avantages des LT versus automates [Pnueli-90] :

Abstraction

les propriétés exprimées en LT sont indépendantes de la description (ou implémentation) du système

Modularité

on peut rajouter ou enlever une propriété, sans remettre en cause les autres propriétés de la spécification

Classification des LT

Suivant le type des propriétés exprimées, on distingue généralement deux classes de LT :

- LT *linéaires* :

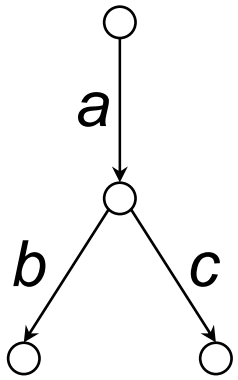
  propriétés sur les chemins d'exécution individuels du programme (les branchements des transitions sont ignorés)

- LT *arborescentes* :

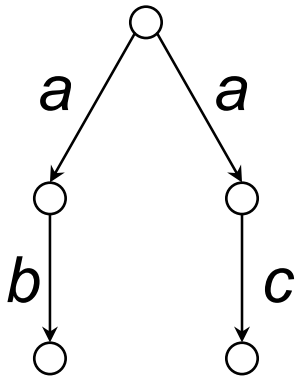
  propriétés sur les arbres d'exécution du programme (les branchements sont pris en compte)

  Pour les systèmes non-déterministes (dont on ne peut pas ignorer les branchements), les LT arborescentes sont plus appropriées.

Exemple



M_1



M_2

$L(M_1) = \{ a.b, a.c \}$

$L(M_2) = \{ a.b, a.c \}$

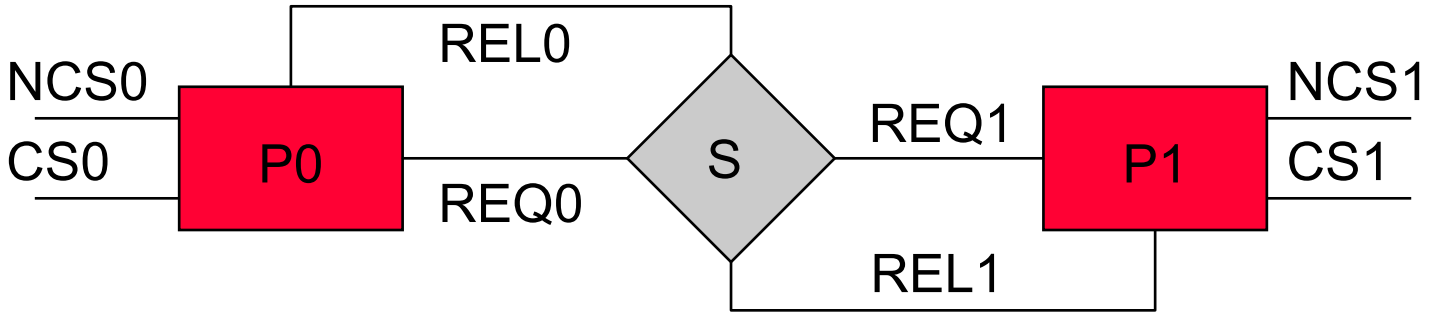
- Une LT *linéaire* ne permet pas de distinguer les deux STE M_1 et M_2, qui ont le même ensemble de séquences d'exécution, mais qui ne sont pas équivalents (modulo la bisimulation forte)
- Une LT *arborescente* permet de prendre en compte le non-déterminisme et donc de distinguer M_1 et M_2 (voir plus loin)

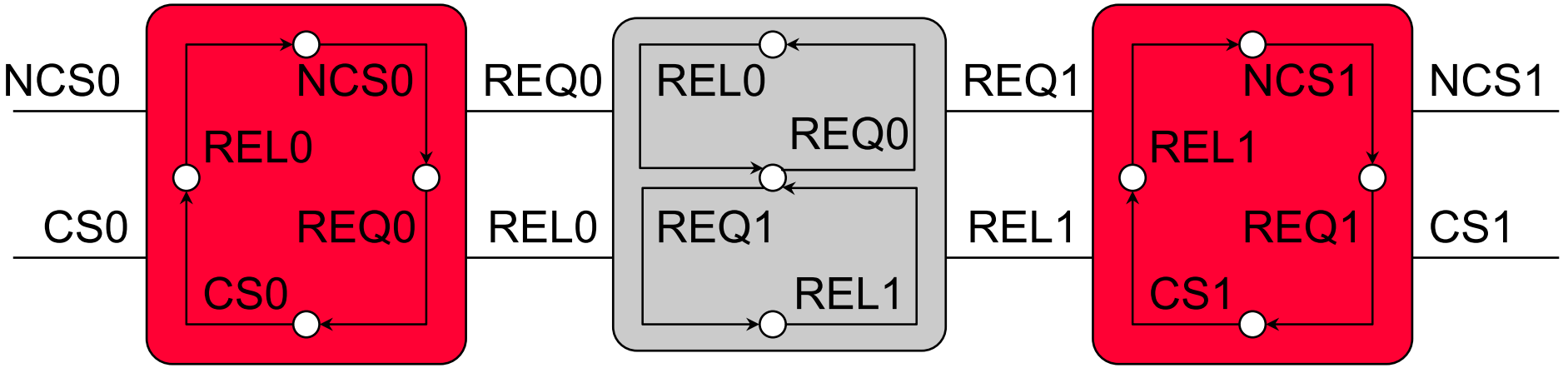
Interprétation des LT sur des STE

- Modèle STE $M = \langle S, A, T, s_0 \rangle$, où :
  - S : ensemble d'états
  - A : ensemble d'actions (événements)
  - $T \in S \times A \times S$: relation de transition
  - $s_0 \in S$: état initial
- Interprétation de φ sur M: $[[\varphi]] = \{ s \in S \mid s \models \varphi \}$
  ($[[\varphi]]$ définie inductivement sur la structure de φ)
- Un STE M satisfait une formule de LT φ ($M \models \varphi$)
  ssi chaque état de S satisfait φ ($s \models \varphi$) :

$$M \models \varphi \quad \Leftrightarrow \quad \forall s \in S \,.\, s \models \varphi$$
$$\Leftrightarrow \quad [[\varphi]] = S$$

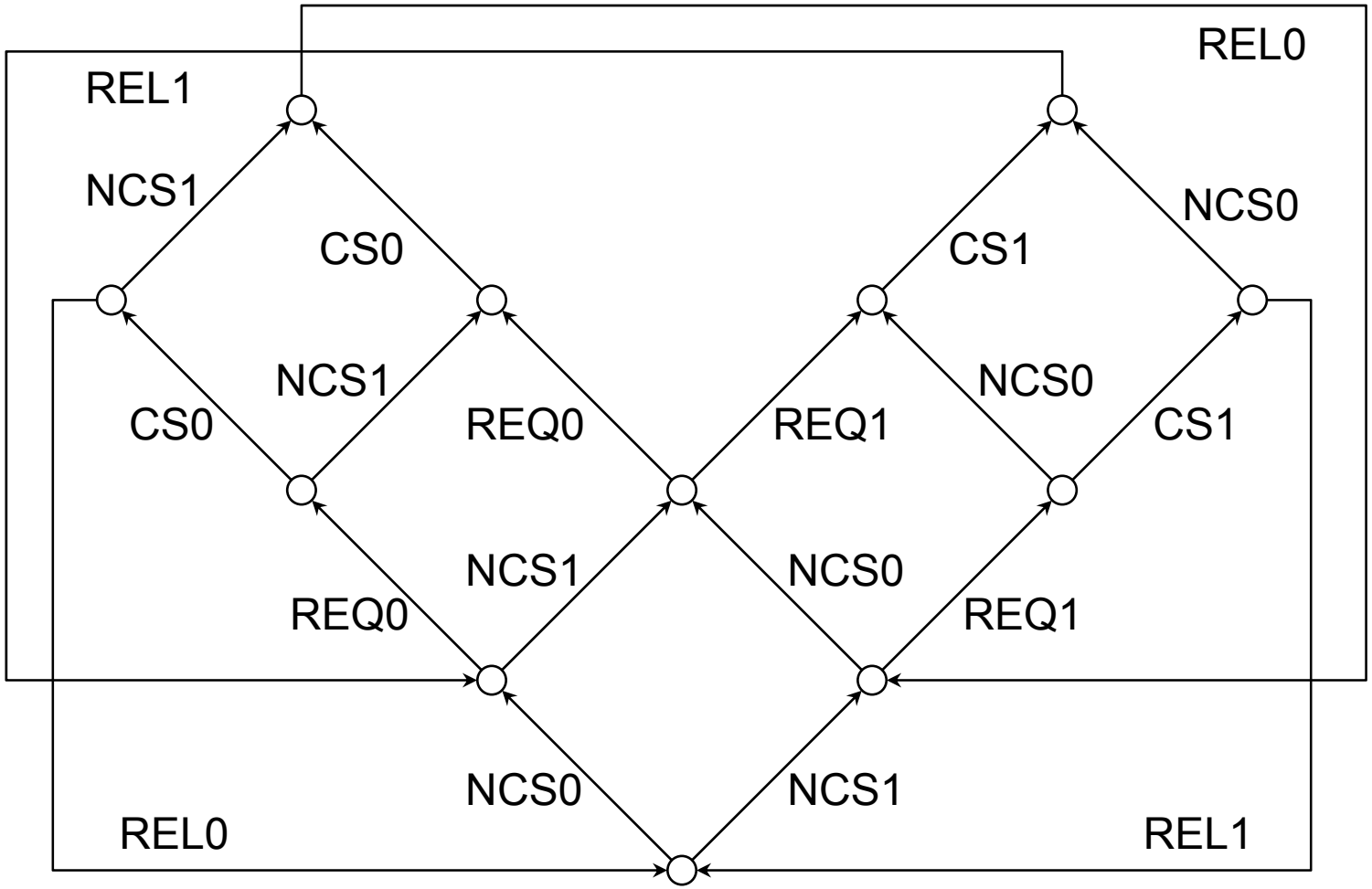
Exemple : exclusion mutuelle avec un sémaphore





Spécification avec des automates communicants

Modèle STE



Logiques modales

- Ce sont les logiques les plus simples permettant de raisonner sur le séquencement et le branchement des transitions dans un STE.
- Opérateurs modaux de base :

  Possibilité

  à partir d'un état, il existe une transition étiquetée par une certaine action qui mène à un certain état

  Nécessité

  à partir d'un état, toutes les transitions étiquetées par une certaine action mènent à certains états
- Hennessy-Milner Logic (HML) [Hennessy-Milner-85]

Prédicats sur actions : syntaxe

$\alpha ::= a$		proposition atomique ($a \in A$)
$\mid$	tt	constante « vrai »
$\mid$	ff	constante « faux »
$\mid$	$\alpha_1 \vee \alpha_2$	disjonction
$\mid$	$\alpha_1 \wedge \alpha_2$	conjonction
$\mid$	$\neg\alpha_1$	négation
$\mid$	$\alpha_1 \Rightarrow \alpha_2$	implication $\neg\alpha_1 \vee \alpha_2$
$\mid$	$\alpha_1 \Leftrightarrow \alpha_2$	équivalence $(\alpha_1 \Rightarrow \alpha_2) \wedge (\alpha_2 \Rightarrow \alpha_1)$

Prédicats sur actions : sémantique

Soit $M = (S, A, T, s_0)$. Interprétation $[\![\alpha]\!] \subseteq A$:

- $[\![a]\!] = \{ a \}$
- $[\![\mathrm{tt}]\!] = A$
- $[\![\mathrm{ff}]\!] = \emptyset$
- $[\![\alpha_1 \vee \alpha_2]\!] = [\![\alpha_1]\!] \cup [\![\alpha_2]\!]$
- $[\![\alpha_1 \wedge \alpha_2]\!] = [\![\alpha_1]\!] \cap [\![\alpha_2]\!]$
- $[\![\neg\alpha_1]\!] = A \setminus [\![\alpha_1]\!]$
- $[\![\alpha_1 \Rightarrow \alpha_2]\!] = (A \setminus [\![\alpha_1]\!]) \cup [\![\alpha_2]\!]$
- $[\![\alpha_1 \Leftrightarrow \alpha_2]\!] = ((A \setminus [\![\alpha_1]\!]) \cup [\![\alpha_2]\!]) \cap ((A \setminus [\![\alpha_2]\!]) \cup [\![\alpha_1]\!])$

Exemples

$A = \{ NCS_0, NCS_1, CS_0, CS_1, REQ_0, REQ_1, REL_0, REL_1 \}$

- $[\![NCS_0]\!] = \{ NCS_0 \}$
- $[\![\neg NCS_0]\!] = \{ NCS_1, CS_0, CS_1, REQ_0, REQ_1, REL_0, REL_1 \}$
- $[\![NCS_0 \wedge \neg NCS_1]\!] = \{ NCS_0 \} = [\![NCS_0]\!]$
- $[\![NCS_0 \vee NCS_1]\!] = \{ NCS_0, NCS_1 \}$
- $[\![(NCS_0 \vee NCS_1) \wedge (NCS_0 \vee REQ_0)]\!] = \{ NCS_0 \}$
- $[\![NCS_0 \wedge NCS_1]\!] = \varnothing = [\![ff]\!]$
- $[\![NCS_0 \vee \neg NCS_0]\!] = \{ NCS_0, NCS_1, CS_0, CS_1, REQ_0, REQ_1, REL_0, REL_1 \} = [\![tt]\!]$

Logique HML : syntaxe

$\varphi ::= tt$	constante « vrai »
$\mid ff$	constante « faux »
$\mid \varphi_1 \vee \varphi_2$	disjonction
$\mid \varphi_1 \wedge \varphi_2$	conjonction
$\mid \neg\varphi_1$	négation
$\mid \langle\ \alpha\ \rangle\ \varphi_1$	possibilité
$\mid [\ \alpha\ ]\ \varphi_1$	nécessité

- Dualité : $[\ \alpha\ ]\ \varphi = \neg\langle\ \alpha\ \rangle\ \neg\varphi$

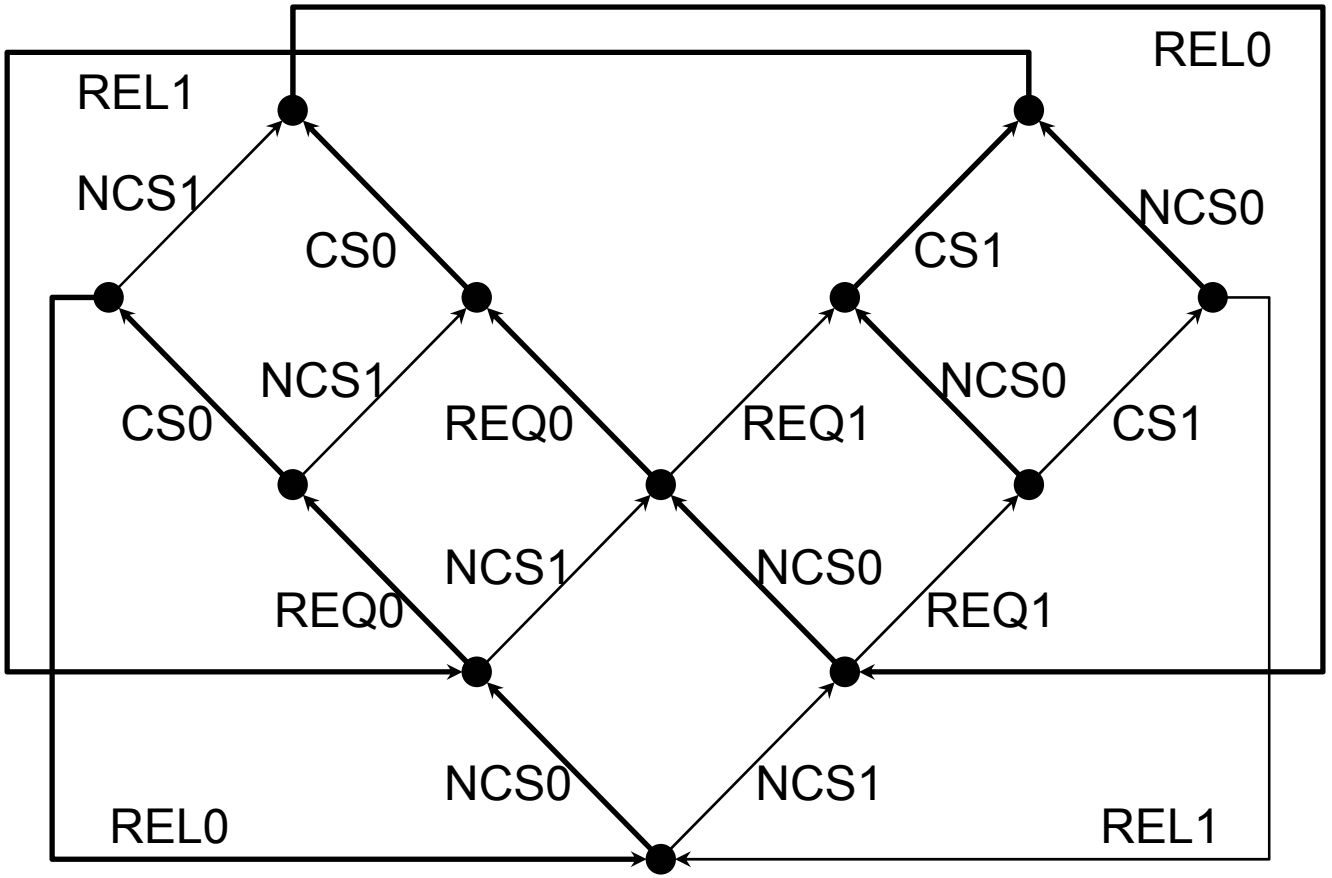
Logique HML : sémantique

Soit $M = (S, A, T, s_0)$. Interprétation $[\![\varphi]\!] \subseteq S$:

- $[\![\mathrm{tt}]\!] = S$
- $[\![\mathrm{ff}]\!] = \varnothing$
- $[\![\varphi_1 \vee \varphi_2]\!] = [\![\varphi_1]\!] \cup [\![\varphi_2]\!]$
- $[\![\varphi_1 \wedge \varphi_2]\!] = [\![\varphi_1]\!] \cap [\![\varphi_2]\!]$
- $[\![\neg\varphi_1]\!] = S \setminus [\![\varphi_1]\!]$
- $[\![\langle \alpha \rangle \varphi_1]\!] = \{ s \in S \mid \exists (s, a, s') \in T . a \in [\![\alpha]\!] \wedge s' \in [\![\varphi_1]\!] \}$
- $[\![[\alpha] \varphi_1]\!] = \{ s \in S \mid \forall (s, a, s') \in T . a \in [\![\alpha]\!] \Rightarrow s' \in [\![\varphi_1]\!] \}$

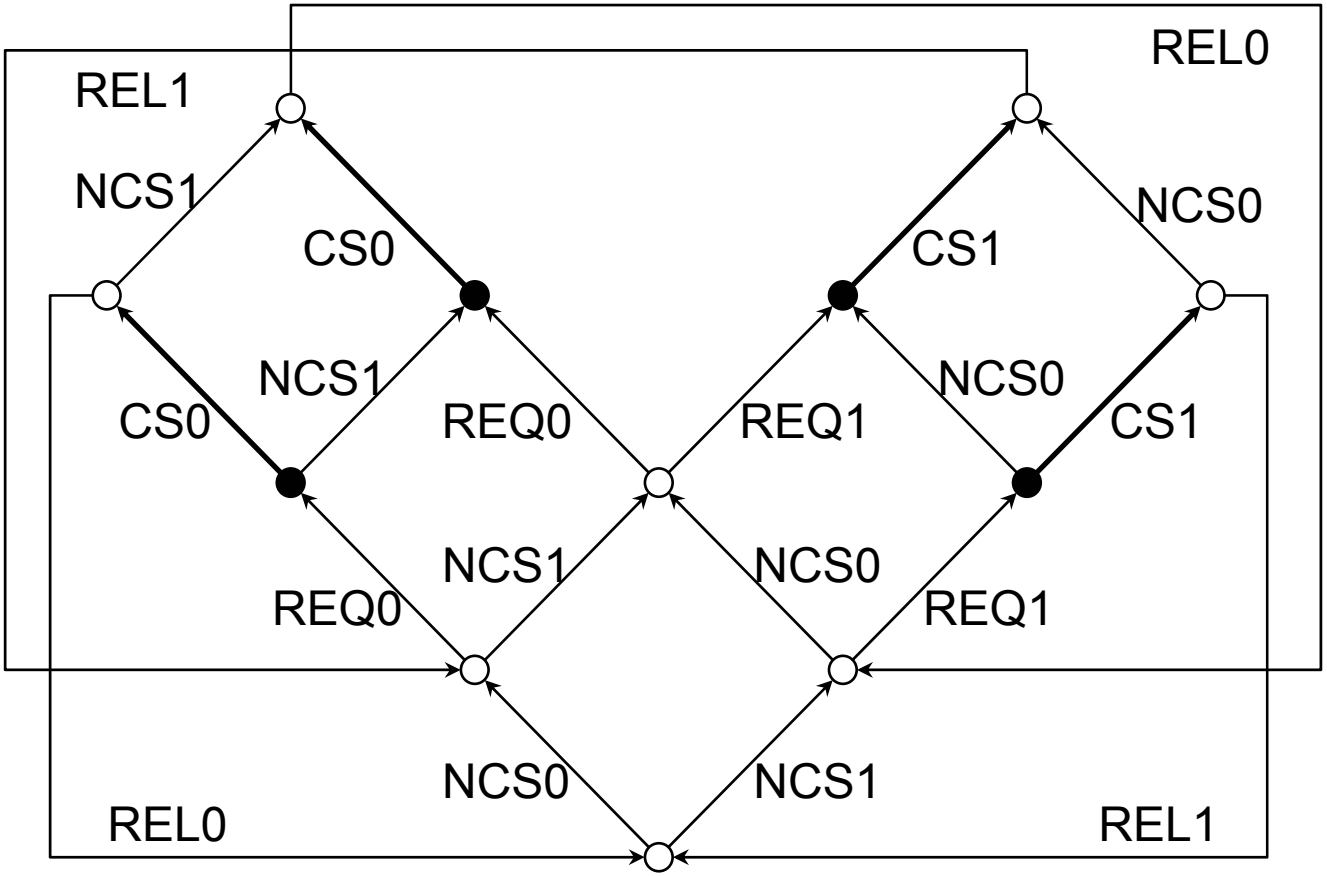
Exemple (1)

Absence de blocage (*deadlock freedom*) : $\langle \text{tt} \rangle \text{tt}$



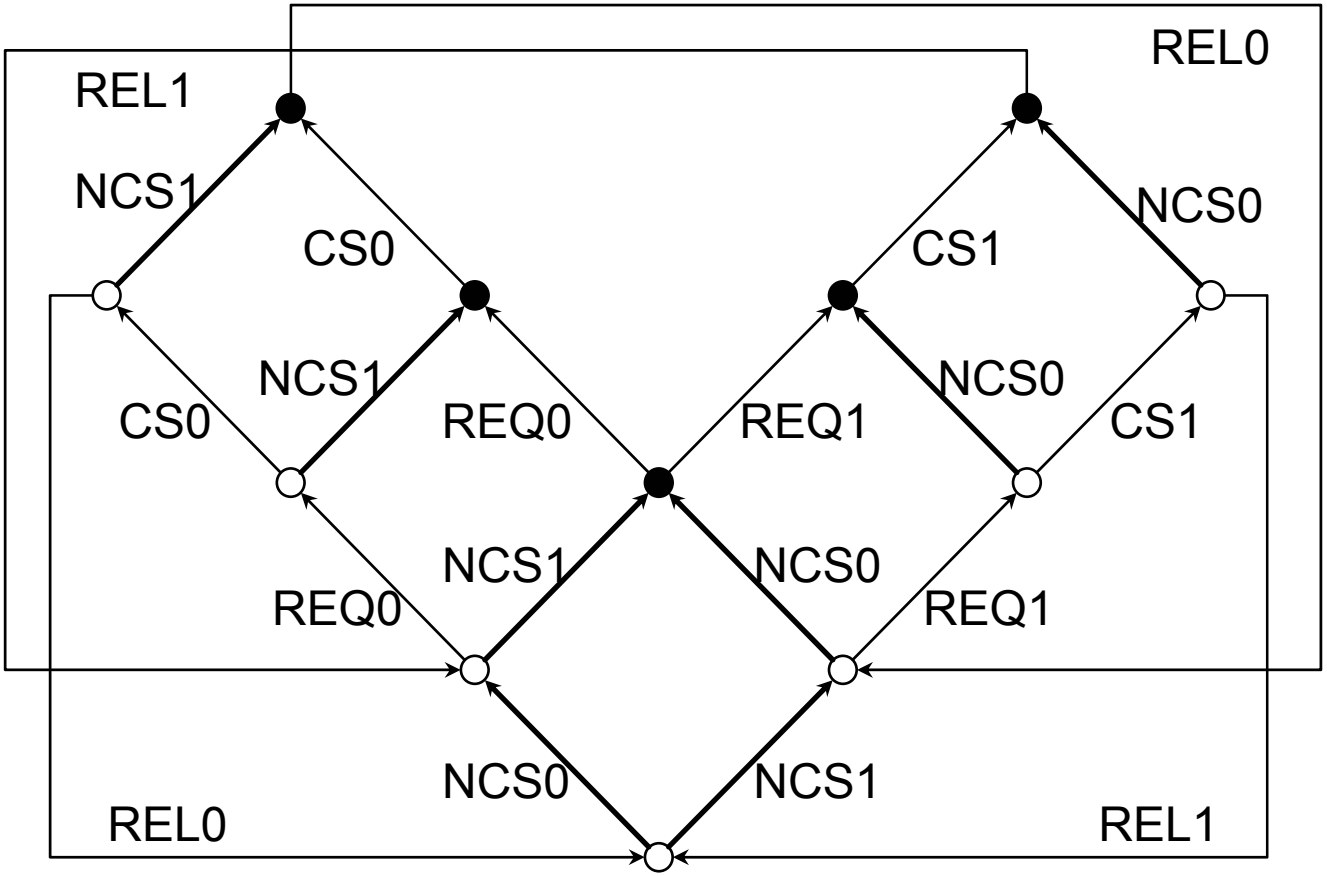
Exemple (2)

Exécution possible d'un ensemble d'actions : $\langle CS_0 \vee CS_1 \rangle$ tt



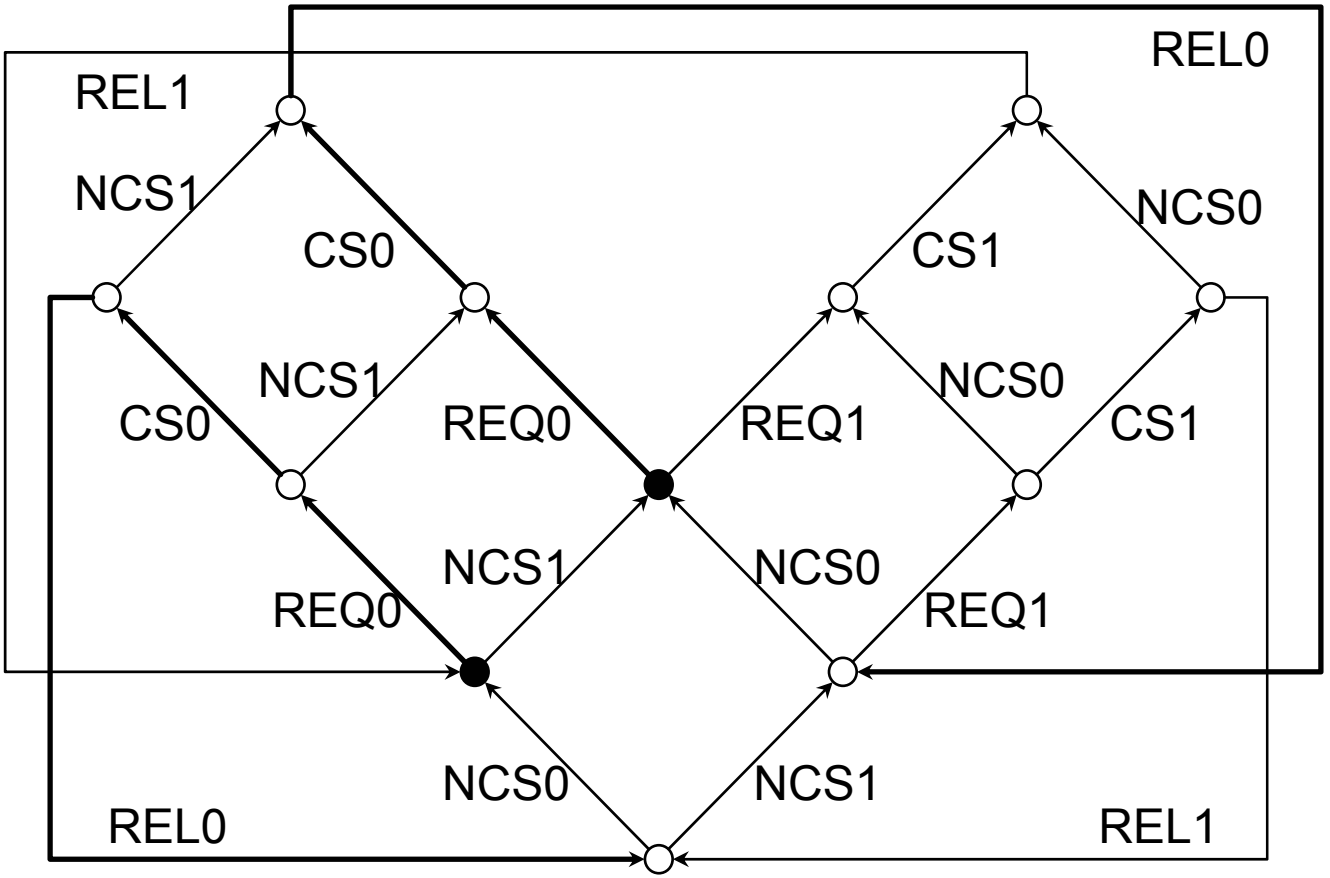
Exemple (3)

Interdiction d'un ensemble d'actions : [$NCS_0 \vee NCS_1$] ff



Exemple (4)

Exécution d'une séquence d'actions: $\langle REQ_0 \rangle \langle CS_0 \rangle \langle REL_0 \rangle$ tt



Quelques identités

Tautologies :

- $\langle\ \alpha\ \rangle\ ff = \langle\ ff\ \rangle\ \varphi = ff$
- $[\ \alpha\ ]\ tt = [\ ff\ ]\ \varphi = tt$

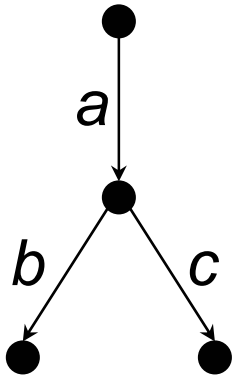
Distributivité des modalités sur $\vee$ et $\wedge$:

- $\langle\ \alpha\ \rangle\ \varphi_1 \vee \langle\ \alpha\ \rangle\ \varphi_2 = \langle\ \alpha\ \rangle\ (\varphi_1 \vee \varphi_2)$
- $\langle\ \alpha_1\ \rangle\ \varphi \vee \langle\ \alpha_2\ \rangle\ \varphi = \langle\ \alpha_1 \vee \alpha_2\ \rangle\ \varphi$
- $[\ \alpha\ ]\ \varphi_1 \wedge [\ \alpha\ ]\ \varphi_2 = [\ \alpha\ ]\ (\varphi_1 \wedge \varphi_2)$
- $[\ \alpha_1\ ]\ \varphi \wedge [\ \alpha_2\ ]\ \varphi = [\ \alpha_1 \vee \alpha_2\ ]\ \varphi$

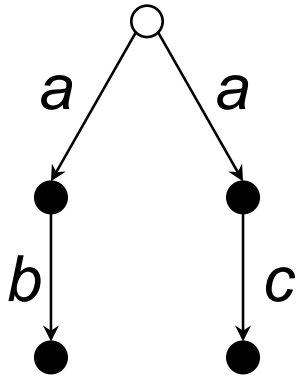
Monotonie des modalités sur φ et α :

- $(\varphi_1 \Rightarrow \varphi_2) \Rightarrow (\langle\ \alpha\ \rangle\ \varphi_1 \Rightarrow \langle\ \alpha\ \rangle\ \varphi_2) \wedge ([\ \alpha\ ]\ \varphi_1 \Rightarrow [\ \alpha\ ]\ \varphi_2)$
- $(\alpha_1 \Rightarrow \alpha_2) \Rightarrow (\langle\ \alpha_1\ \rangle\ \varphi \Rightarrow \langle\ \alpha_2\ \rangle\ \varphi) \wedge ([\ \alpha_2\ ]\ \varphi \Rightarrow [\ \alpha_1\ ]\ \varphi)$

Caractérisation du branchement



M_1



M_2

$L(M_1) = \{ a.b, a.c \}$

$L(M_2) = \{ a.b, a.c \}$

Formule modale (HML) qui distingue M_1 et M_2 :

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

$$M_1 \models \varphi \text{ et } \quad M_2 \not\models \varphi$$

Logiques modales (résumé)

- Permettent d'exprimer des propriétés arborescentes reliant les états $s \in S$ et les actions $a \in A$ d'un modèle STE
- Mais :
  - prennent en compte uniquement un voisinage borné d'un état (imbrication des modalités)
  - ne peuvent pas exprimer des propriétés sur un chemin de transitions de longueur quelconque
- Exemple : la propriété

  « à partir d'un état *s*, il existe un chemin menant à un état *s'* où l'action *a* est exécutable »

  n'est pas exprimable en HML

Logiques arborescentes

- Ce sont des logiques permettant de raisonner sur les arbres (infinis) de transitions dans un STE.
- Opérateurs temporels de base :

  Potentialité

  à partir d'un état, il existe une séquence de transitions qui mène à un certain état

  Inévitabilité

  à partir d'un état, toutes les séquences de transitions mènent à certains états
- Action Computation Tree Logic (ACTL) [DeNicola-Vaandrager-90]

Logique ACTL : syntaxe

$\varphi ::= tt \mid ff$	constantes
$\mid \varphi_1 \vee \varphi_2 \mid \neg\varphi_1$	opér. booléens
$\mid \langle \alpha \rangle \varphi_1 \mid [\alpha] \varphi_1$	modalités
$\mid EF_\alpha \varphi_1$	potentialité
$\mid AF_\alpha \varphi_1$	inévitabilité
$\mid AG_\alpha \varphi_1$	invariance
$\mid EG_\alpha \varphi_1$	trajectoire

- Dualité : $AG_\alpha \varphi = \neg EF_\alpha \neg\varphi$

  $EG_\alpha \varphi = \neg AF_\alpha \neg\varphi$

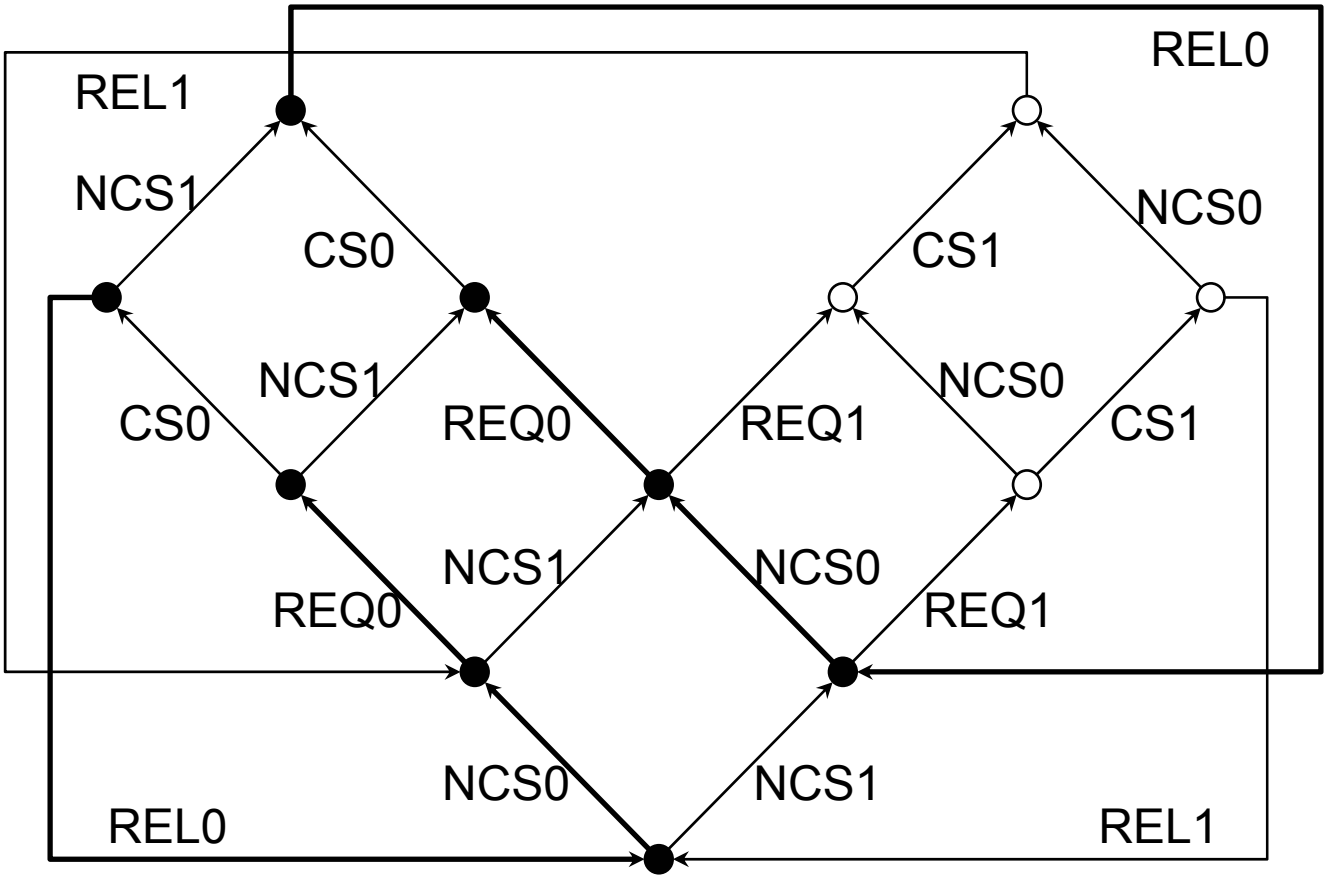
Logique ACTL : sémantique

Soit $M = (S, A, T, s_0)$. Interprétation $[\![\varphi]\!] \subseteq S$:

- $[\![EF_\alpha \varphi_1]\!] = \{ s \in S \mid \exists\, s \rightarrow^{a1} s_1 \rightarrow^{a2} s_2 \rightarrow \ldots \;.\; \exists k \geq 0.$
  $\forall 1 \leq i \leq k \;.\; a_i \in [\![\alpha]\!] \wedge s_k \in [\![\varphi_1]\!] \}$
- $[\![AF_\alpha \varphi_1]\!] = \{ s \in S \mid \forall\, s \rightarrow^{a1} s_1 \rightarrow^{a2} s_2 \rightarrow \ldots \;.\; \exists k \geq 0.$
  $\forall 1 \leq i \leq k \;.\; a_i \in [\![\alpha]\!] \wedge s_k \in [\![\varphi_1]\!] \}$
- $[\![AG_\alpha \varphi_1]\!] = \{ s \in S \mid \forall s \rightarrow^{a1} s_1 \rightarrow^{a2} s_2 \rightarrow \ldots \;.\; \forall k \geq 0.$
  $\forall 1 \leq i \leq k \;.\; a_i \in [\![\alpha]\!] \Rightarrow s_k \in [\![\varphi_1]\!] \}$
- $[\![EG_\alpha \varphi_1]\!] = \{ s \in S \mid \exists\, s \rightarrow^{a1} s_1 \rightarrow^{a2} s_2 \rightarrow \ldots \;.\; \forall k \geq 0.$
  $\forall 1 \leq i \leq k \;.\; a_i \in [\![\alpha]\!] \Rightarrow s_k \in [\![\varphi_1]\!] \}$

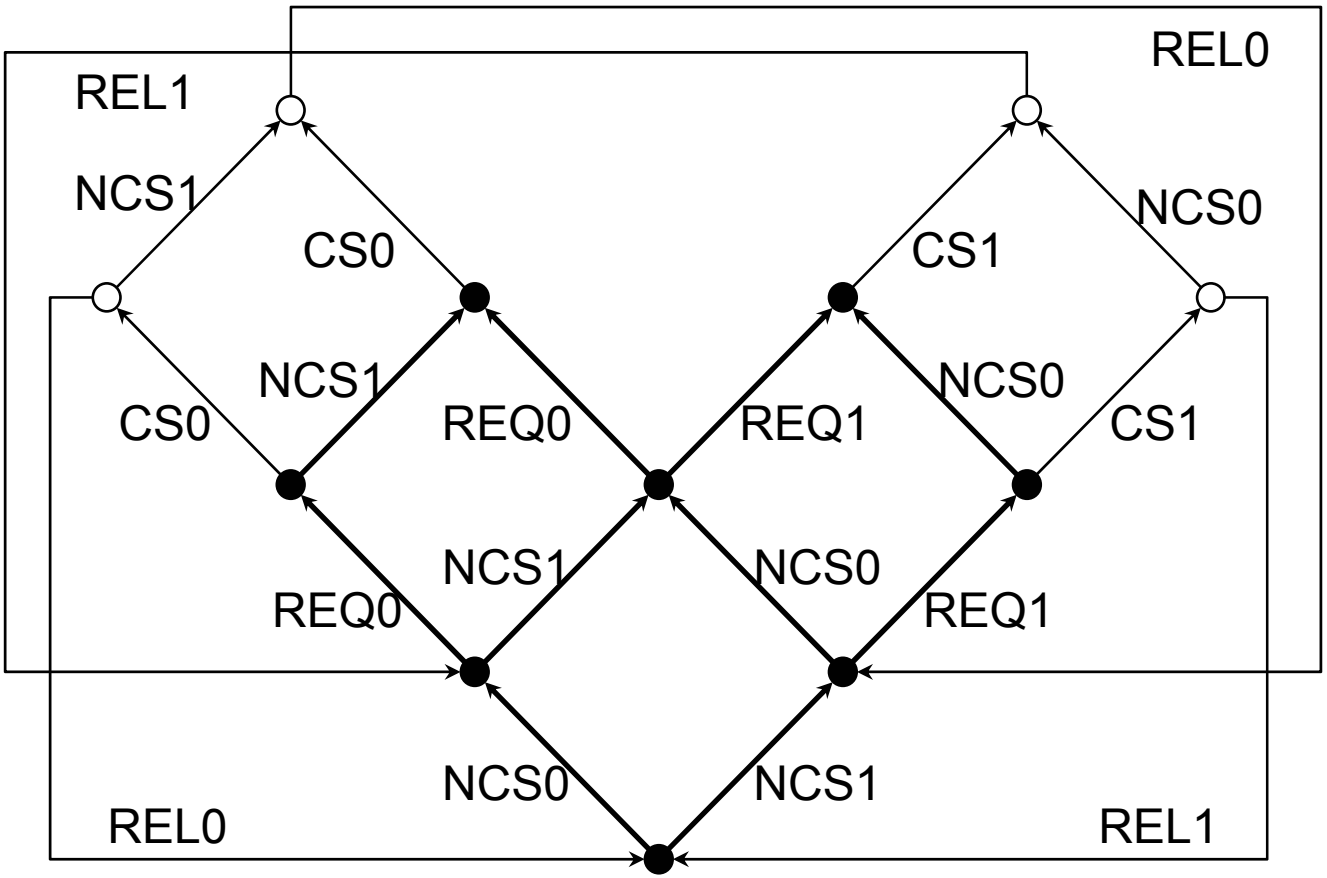
Exemple (1)

Accessibilité potentielle : $EF_{\neg\, REL1} \langle CS_0 \rangle tt$



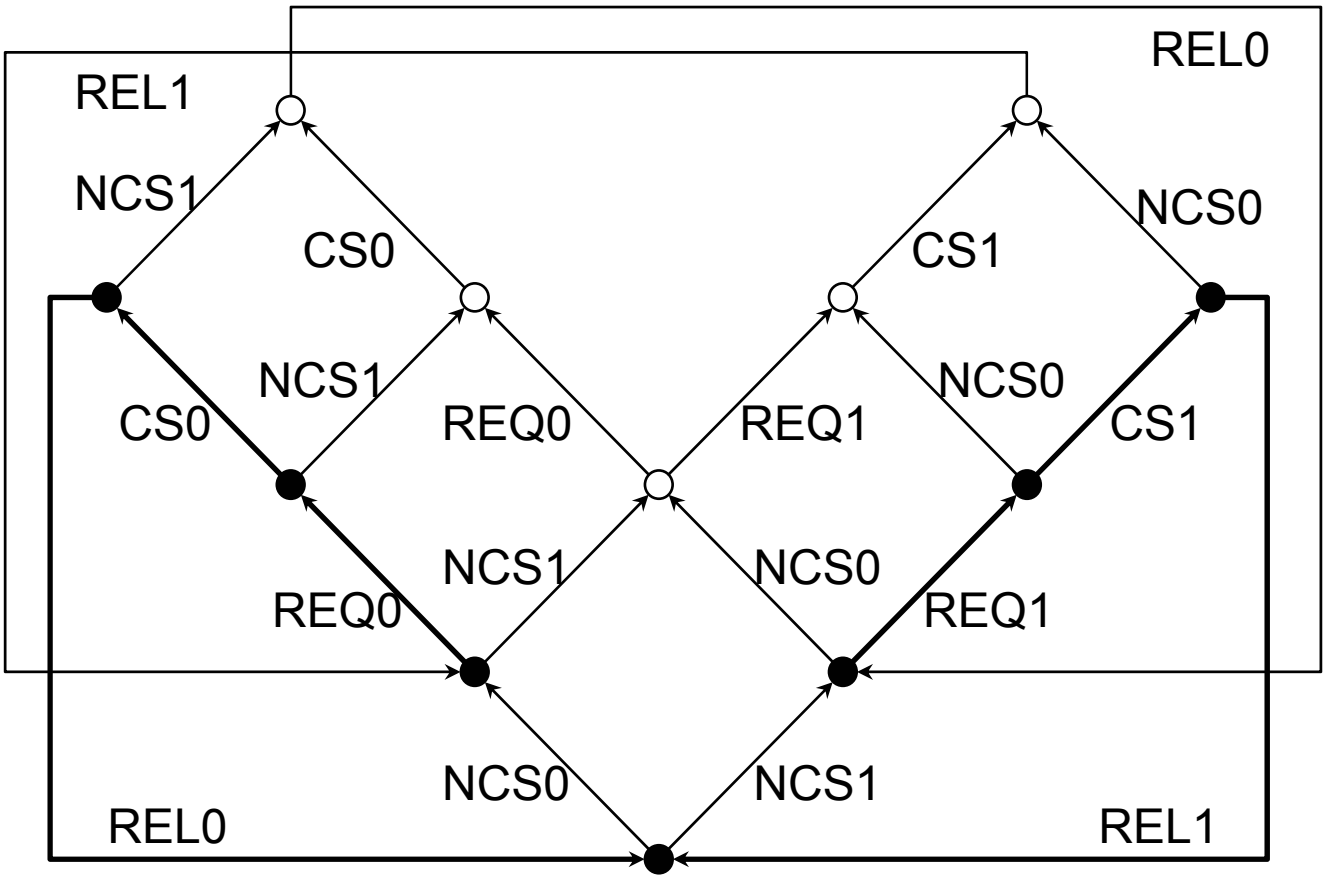
Exemple (2)

Accessibilité inévitable : $AF_{\neg\,(REL0 \vee REL1)} \langle\, CS_0 \vee CS_1 \,\rangle\, tt$



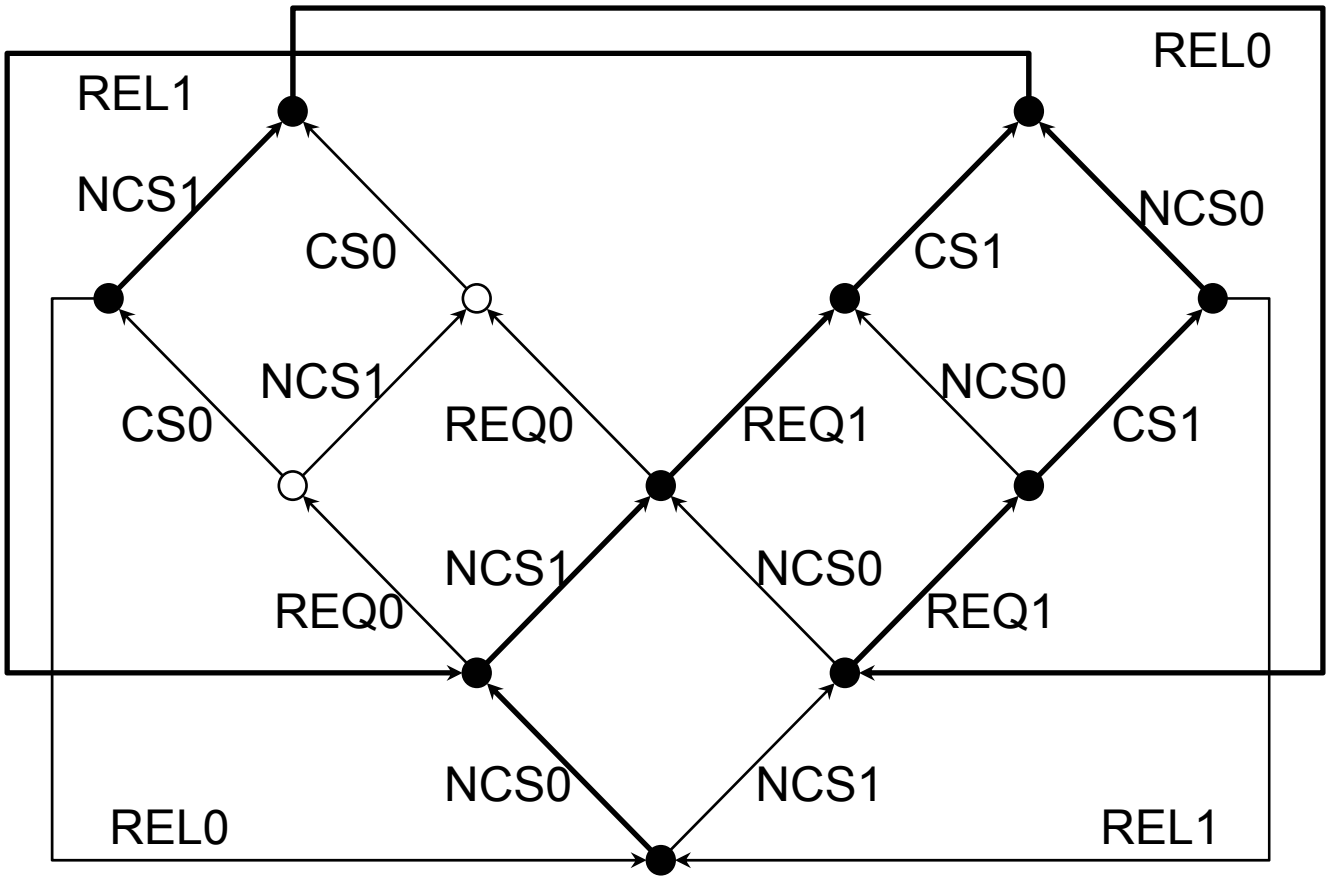
Exemple (3)

Invariance : $AG_{\neg (NCS0 \vee NCS1)} \langle NCS_0 \vee NCS_1 \rangle tt$



Exemple (4)

Trajectoire : $EG_{\neg CS0}\ [\ CS_0\ ]\ ff$



Propriétés de sûreté

- Informellement : « rien de mal n'arrivera »
- Une manière de spécifier les propriétés de sûreté : interdire les séquences d'actions indésirables
  - Exclusion mutuelle :

    $\neg \langle CS_0 \rangle EF_{\neg REL0} \langle CS_1 \rangle tt$

    $= [CS_0] AG_{\neg REL0} [CS_1] ff$
  - Opérateur « not-to-unless » :

    not *a* **to** *b* **unless** *c* $= [a] \neg EF_{\neg c} \langle b \rangle tt$

    $= [a] AG_{\neg c} [b] ff$
- L'interdiction d'une séquence s'exprime en combinant les opérateurs $[\alpha] \varphi$ et $AG_{\alpha} \varphi$

Propriétés de vivacité

- Informellement: « quelque chose de bien arrivera »
- Une manière de spécifier les propriétés de vivacité: imposer des séquences (ou des arbres) d'actions désirables
  - Libération potentielle de la section critique :

    $\langle NCS_0 \rangle EF_{tt} \langle REQ_0 \rangle EF_{tt} \langle REL_0 \rangle tt$
  - Inevitabilité de l'accès à la section critique :

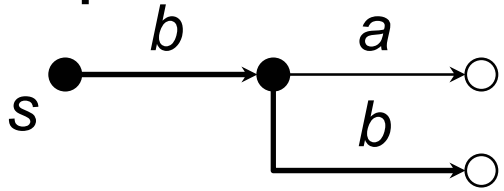
    $AF_{tt} \langle CS_0 \rangle tt$
- L'existence d'une séquence s'exprime en combinant les opérateurs $\langle \alpha \rangle \varphi$ et $EF_\alpha \varphi$

Remarques sur l'inévitabilité

- *Accessibilité inévitable* : tous les chemins issus d'un état mènent à des états où l'action *a* est exécutable

  $AF_{tt} \langle a \rangle tt$

- *Exécution inévitable* : tous les chemins issus d'un état contiennent l'action *a*
- Exécution inévitable $\Rightarrow$ accessibilité inévitable mais la réciproque est fausse :



$s \models AF_{tt} \langle a \rangle tt$

- Pour exprimer l'exécution inévitable, il faut un autre opérateur inev(*a*) défini par un point fixe

Logiques arborescentes (résumé)

- Opérateurs $EF_\alpha\,\varphi$, $AF_\alpha\,\varphi$, $EG_\alpha\,\varphi$, $AG_\alpha\,\varphi$: strictement plus puissants que les modalités $\langle\,\alpha\,\rangle\,\varphi$ et $[\,\alpha\,]\,\varphi$
- Permettent d'exprimer des propriétés arborescentes sur une profondeur quelconque dans un STE
- Mais :
  - Ne permettent pas d'exprimer la répétition non bornée d'une sous-séquence d'actions
- Exemple : la propriété

  « à partir d'un état, il existe un chemin *a.b.a.b* … *a.b* menant à un état où l'action *c* est exécutable »

  n'est pas exprimable en ACTL

Logiques régulières

- Ce sont des logiques permettant de raisonner sur les séquences d'exécution régulières d'un STE.
- Opérateurs de base :

  Expressions régulières

  deux états sont reliés par une séquence régulière de transitions

  Modalités sur les séquences

  à partir d'un état, possibilité (nécessité) d'effectuer une séquence régulière de transitions
- Propositional Dynamic Logic (PDL) [Fischer-Ladner-79]

Formules régulières : syntaxe

$\beta ::= \alpha$	transition
$\mid$ nil	séquence vide
$\mid \beta_1 \,.\, \beta_2$	concaténation
$\mid \beta_1 \mid \beta_2$	alternative
$\mid \beta_1^*$	itération (≥ 0)
$\mid \beta_1^+$	itération (≥ 1)

- Identités :

$$\text{nil} = \text{ff}^* \qquad \beta^+ = \beta \,.\, \beta^*$$

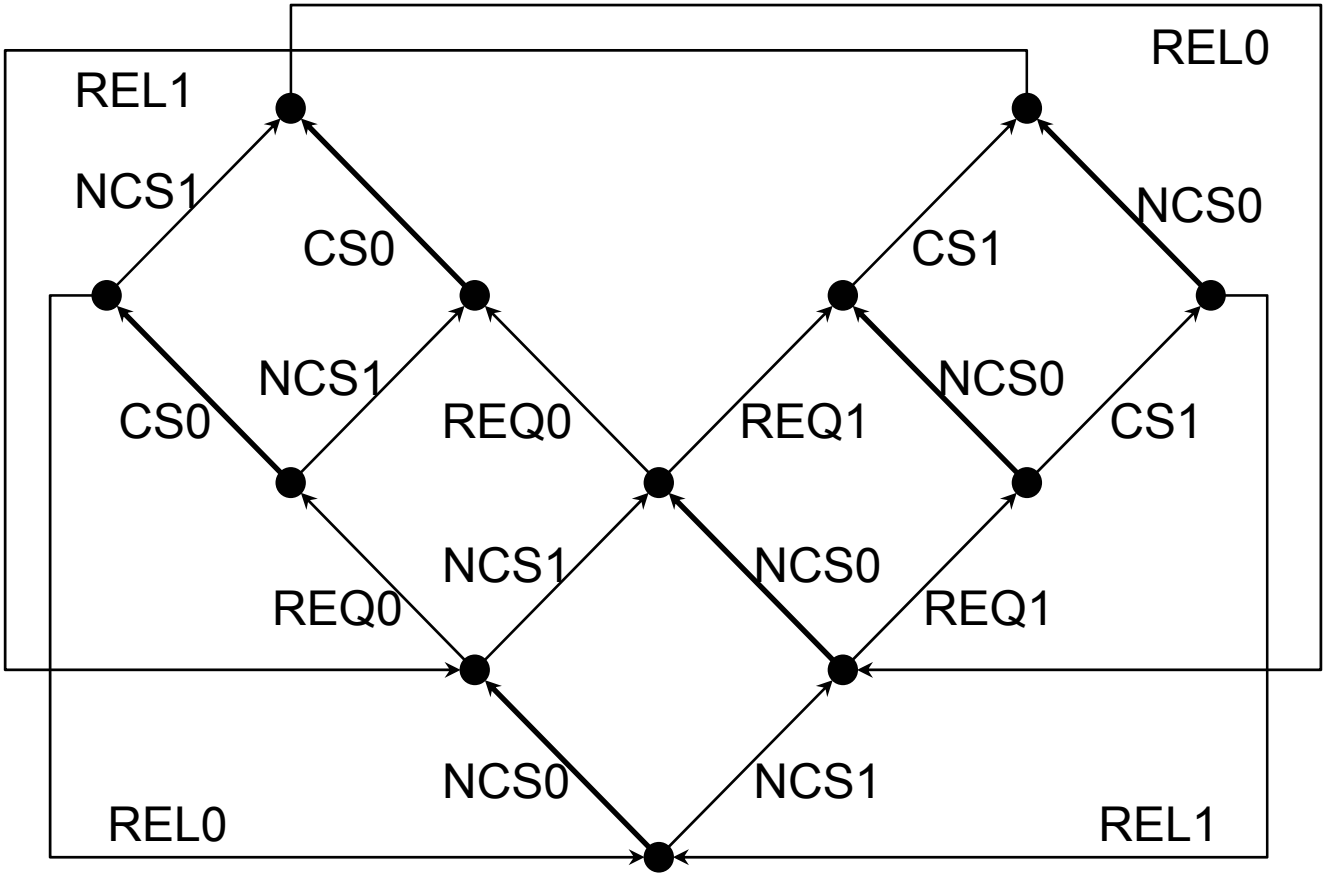
Formules régulières : sémantique

Soit $M = (S, A, T, s_0)$. Interprétation $[[\beta]] \subseteq S \times S$:

- $[[\alpha]] = \{ (s, s') \mid \exists a \in [[\alpha]] . (s, a, s') \in T \}$
- $[[\text{nil}]] = \{ (s, s) \mid s \in S \}$ (identité)
- $[[\beta_1 . \beta_2]] = [[\beta_1]] \hat{\imath} [[\beta_2]]$ (composition)
- $[[\beta_1 \mid \beta_2]] = [[\beta_1]] \cup [[\beta_2]]$ (union)
- $[[\beta_1^*]] = [[\beta_1]] ^*$ (fermeture transitive et réflexive)
- $[[\beta_1^+]] = [[\beta_1]] ^+$ (fermeture transitive)

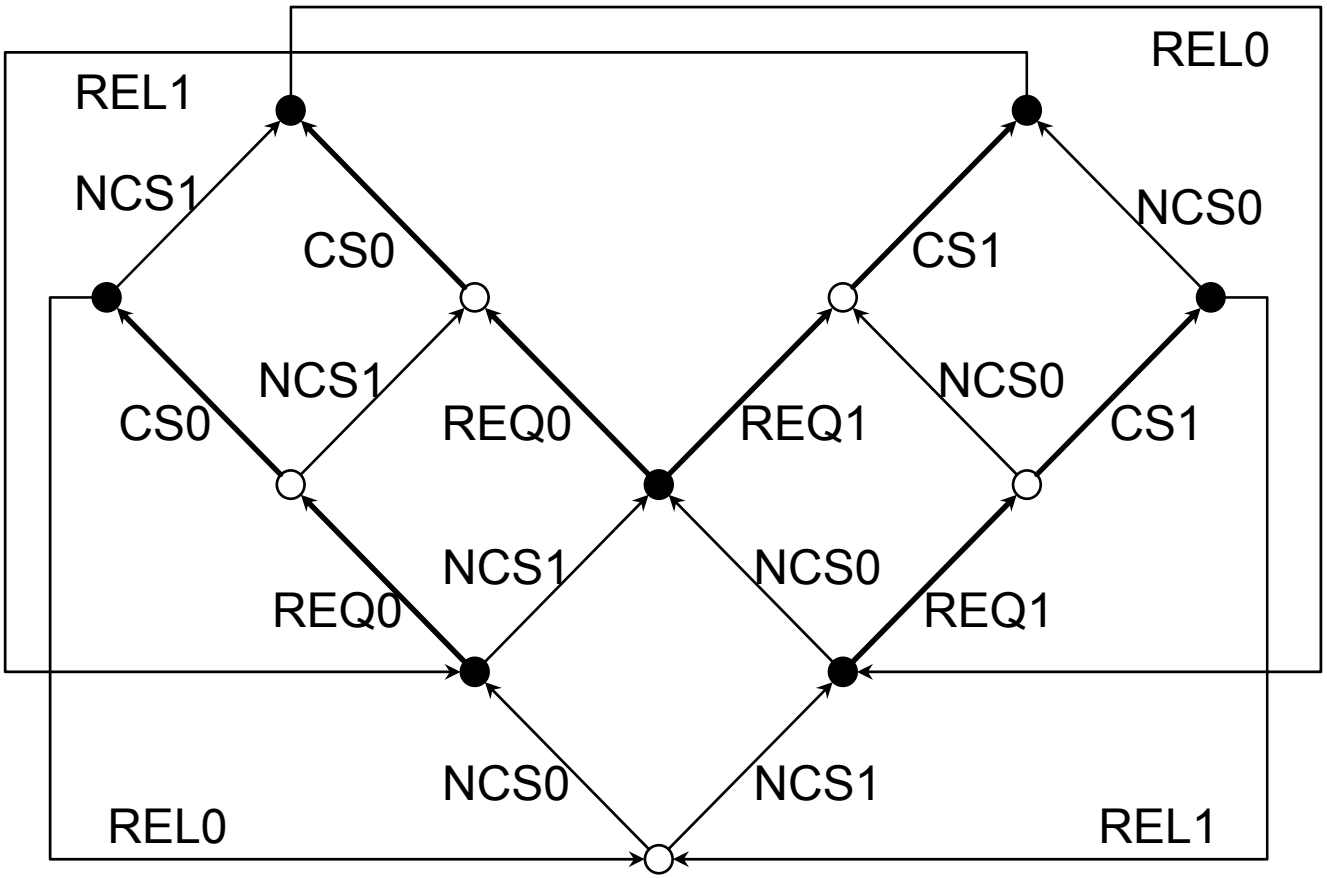
Exemple (1)

Séquences contenant une seule transition : $NCS_0 \vee CS_0$



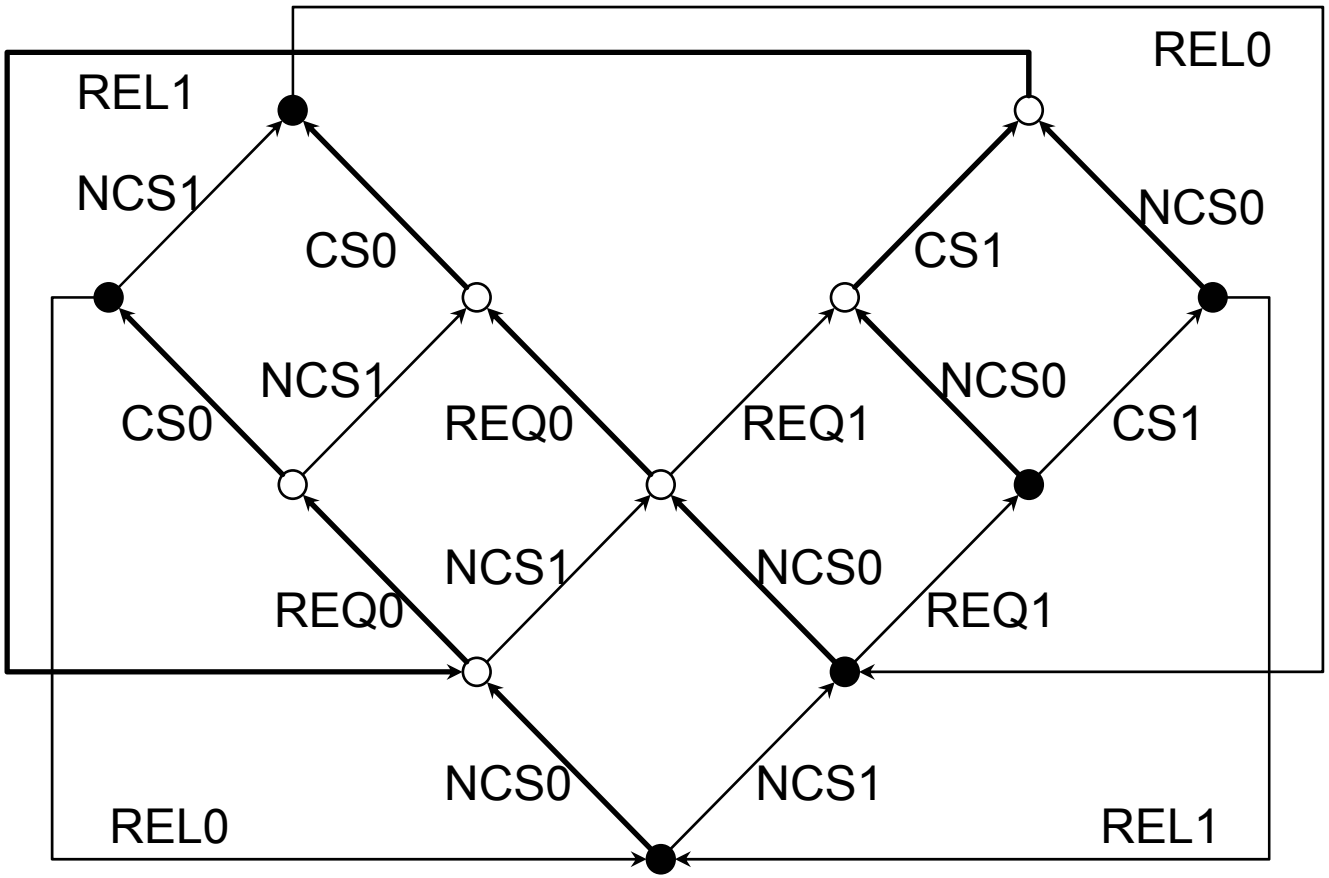
Exemple (2)

Séquences alternatives : $(REQ_0 . CS_0) \mid (REQ_1 . CS_1)$



Exemple (3)

Séquences avec répétition : $NCS_0 . (\neg NCS_1)^* . CS_0$



Logique PDL : syntaxe

$\varphi ::= tt$	constante « vrai »
$\mid ff$	constante « faux »
$\mid \varphi_1 \vee \varphi_2$	disjonction
$\mid \varphi_1 \wedge \varphi_2$	conjonction
$\mid \neg\varphi_1$	négation
$\mid \langle \beta \rangle \varphi_1$	possibilité
$\mid [\beta] \varphi_1$	nécessité

- Dualité : $[\beta] \varphi = \neg\langle \beta \rangle \neg\varphi$

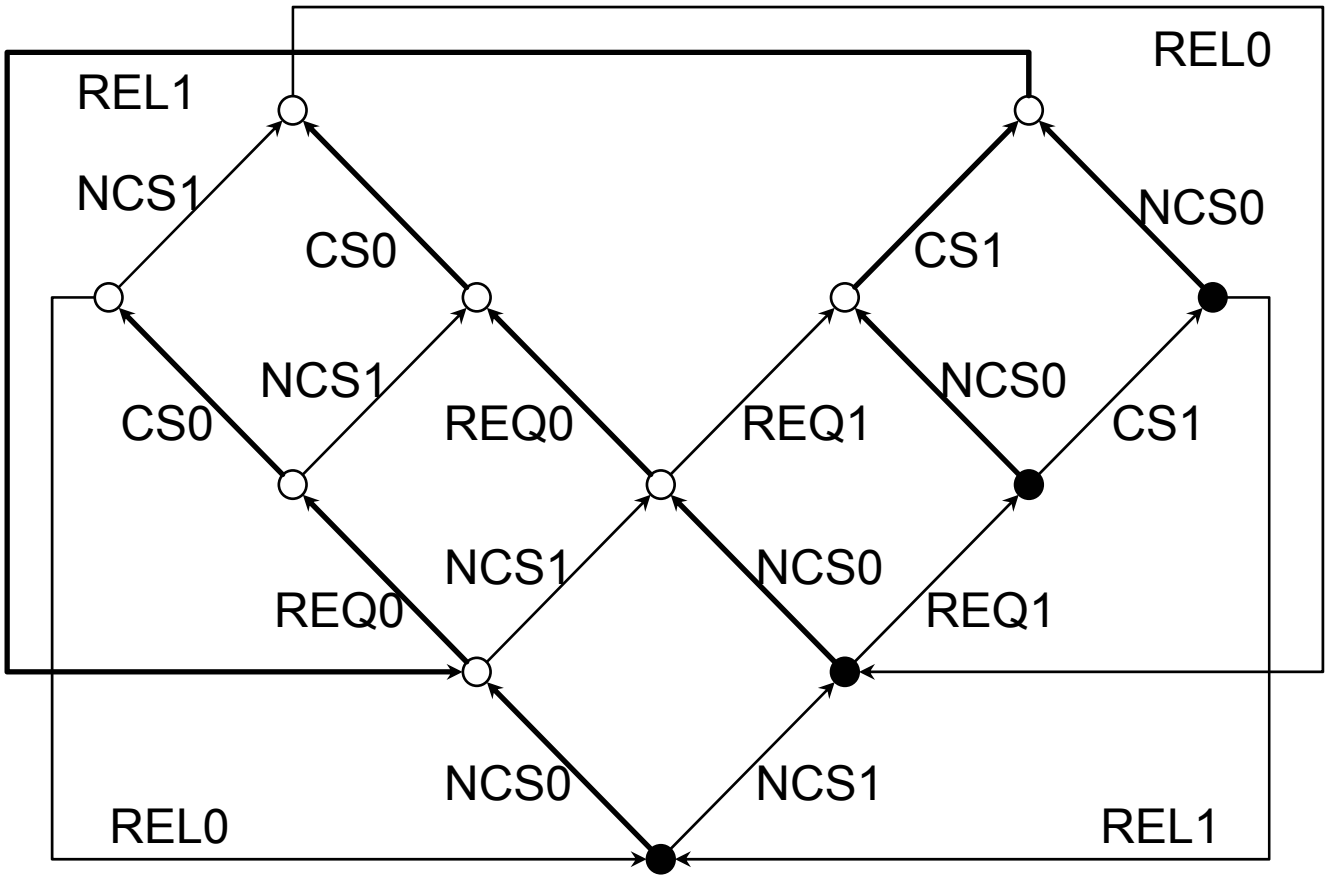
Logique PDL : sémantique

Soit $M = (S, A, T, s_0)$. Interprétation $[\![\varphi]\!] \subseteq S$:

- $[\![\mathrm{tt}]\!] = S$
- $[\![\mathrm{ff}]\!] = \varnothing$
- $[\![\varphi_1 \vee \varphi_2]\!] = [\![\varphi_1]\!] \cup [\![\varphi_2]\!]$
- $[\![\varphi_1 \wedge \varphi_2]\!] = [\![\varphi_1]\!] \cap [\![\varphi_2]\!]$
- $[\![\neg\varphi_1]\!] = S \setminus [\![\varphi_1]\!]$
- $[\![\langle \beta \rangle \varphi_1]\!] = \{ s \in S \mid \exists s' \in S . (s, s') \in [\![\beta]\!] \wedge s' \in [\![\varphi_1]\!] \}$
- $[\![[\beta] \varphi_1]\!] = \{ s \in S \mid \forall s' \in S . (s, s') \in [\![\beta]\!] \Rightarrow s' \in [\![\varphi_1]\!] \}$

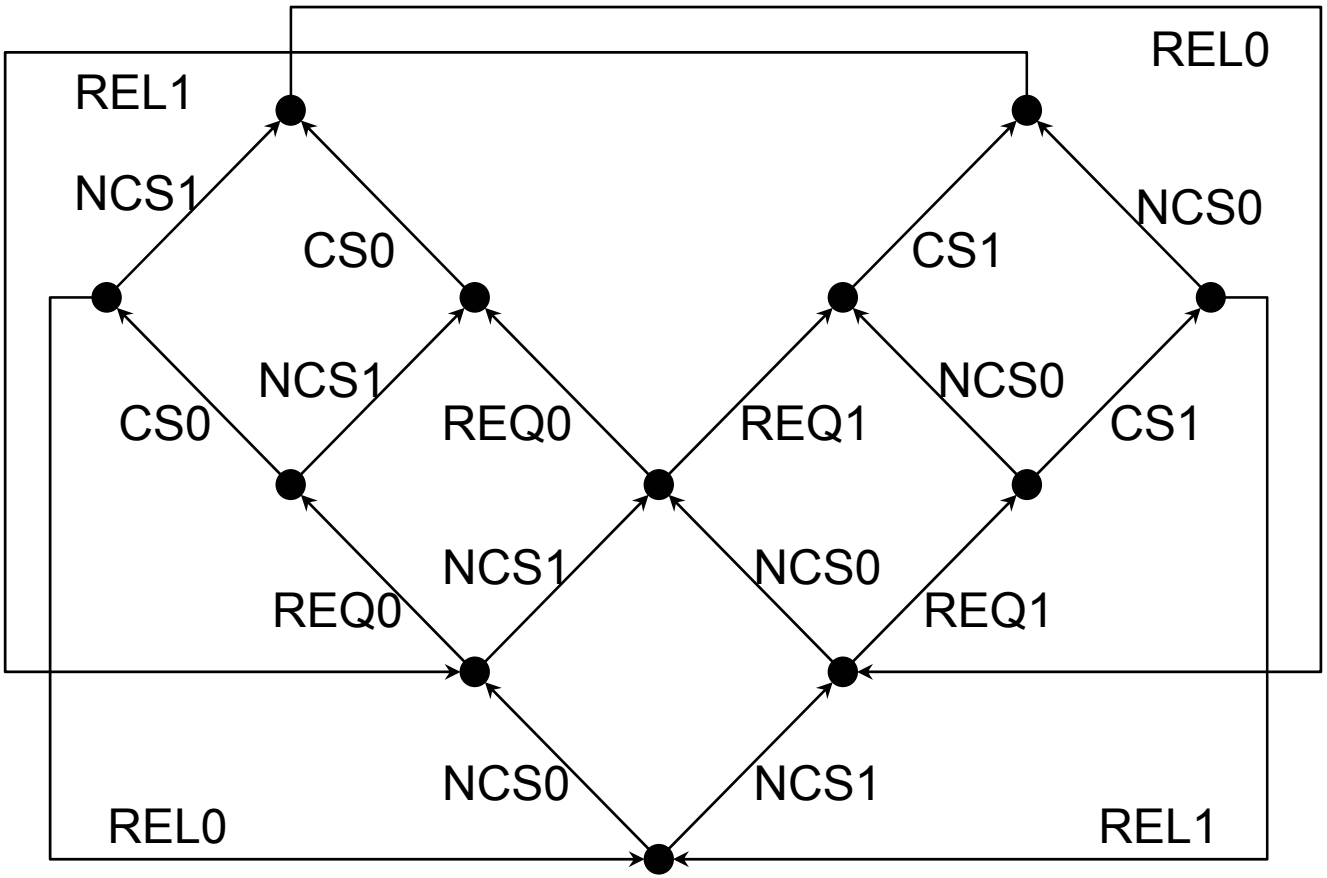
Exemple (1)

Accessibilité de la section critique : $\langle NCS_0 . tt^* . CS_0 \rangle tt$



Exemple (2)

Exclusion mutuelle : [CS_0 . $(\neg REL_0)^*$. CS_1] ff



Quelques identités

Distributivité des opérateurs réguliers sur $\langle\ \rangle$ et $[\ ]$:

- $\langle\ \beta_1 \cdot \beta_2\ \rangle\ \varphi = \langle\ \beta_1\ \rangle \langle\ \beta_2\ \rangle\ \varphi$
- $\langle\ \beta_1 \mid \beta_2\ \rangle\ \varphi = \langle\ \beta_1\ \rangle\ \varphi \vee \langle\ \beta_2\ \rangle\ \varphi$
- $\langle\ \beta^*\ \rangle\ \varphi = \varphi \vee \langle\ \beta\ \rangle \langle\ \beta^*\ \rangle\ \varphi$
- $[\ \beta_1 \cdot \beta_2\ ]\ \varphi = [\ \beta_1\ ]\ [\ \beta_2\ ]\ \varphi$
- $[\ \beta_1 \mid \beta_2\ ]\ \varphi = [\ \beta_1\ ]\ \varphi \wedge [\ \beta_2\ ]\ \varphi$
- $[\ \beta^*\ ]\ \varphi = \varphi \wedge [\ \beta\ ]\ [\ \beta^*\ ]\ \varphi$

Opérateurs de potentialité et d'invariance d'ACTL :

- $EF_\alpha\ \varphi = \langle\ \alpha^*\ \rangle\ \varphi$
- $AG_\alpha\ \varphi = [\ \alpha^*\ ]\ \varphi$

Propriétés de sûreté

- Interdire les séquences d'actions indésirables
  - Exclusion mutuelle :

    $\neg \langle CS_0 \,.\, (\neg REL_0)^* \,.\, CS_1 \rangle tt$

    $= [\, CS_0 \,.\, (\neg REL_0)^* \,.\, CS_1 \,]\, ff$
  - Opérateur « not-to-unless » :

    not *a* **to** *b* **unless** *c* $= [\, a \,]\, AG_{\neg c}\, [\, b \,]\, ff$ (ACTL)

    $= [\, a \,.\, (\neg c)^* .\, b \,]\, ff$ (PDL)
- L'interdiction d'une séquence régulière β s'exprime avec la modalité $[\, \beta \,]\, ff$

Propriétés de vivacité

- Imposer des séquences (ou des arbres) d'actions désirables
  - Libération potentielle de la section critique :

    $\langle NCS_0 . tt^* . REQ_0 . tt^* . REL_0 \rangle tt$
  - Inevitabilité de l'accès à la section critique :

    inev (CS_0) : inexprimable en PDL
- L'existence d'une séquence régulière β s'exprime avec la modalité $\langle \beta \rangle tt$

Logiques régulières (résumé)

- Proposées initialement pour analyser les programmes séquentiels
- Permettent la description directe d'arborescences d'exécution (non bornées) d'un programme
- Description plus intuitive des propriétés de sûreté (interdiction des séquences indésirables [β] ff)
- Mais :
  - Ne permettent pas d'exprimer les propriétés d'inévitabilité
- Exemple : la propriété

  « il est inévitable d'exécuter une action *a* »

  est inexprimable en PDL

Logiques de point fixe

- Logiques très expressives pour caractériser les arbres d'exécution (infinis) contenus dans un STE.
- Opérateurs temporels de base :

  Point fixe minimal (μ)

  « fonction récursive » définie sur le STE : arbres d'exécution *finis* issus d'un état

  Point fixe maximal (ν)

  dual de l'opérateur de point fixe minimal : arbres d'exécution *infinis* issus d'un état

- Mu-calcul modal ($L\mu$) [Kozen-83]

Mu-calcul modal : syntaxe

$\varphi ::= tt \mid ff$	constantes
$\mid \varphi_1 \vee \varphi_2 \mid \neg\varphi_1$	opérateurs booléens
$\mid \langle \alpha \rangle \varphi_1$	possibilité
$\mid [\alpha] \varphi_1$	nécessité
$\mid X$	variable propositionnelle
$\mid \mu X . \varphi_1$	point fixe minimal
$\mid \nu X . \varphi_1$	point fixe maximal

- Dualité : $\nu X . \varphi = \neg \mu X . \neg \varphi [\neg X / X]$

Monotonie syntaxique

- Pour définir la sémantique des points fixes, les formules φ doivent être *syntaxiquement monotones* [Kozen-83] :

  Dans chaque formule $\sigma X \,.\, \varphi$ (où $\sigma \in \{ \mu, \nu \}$), toutes les occurrences libres de X dans φ doivent être précédées d'un nombre pair de négations

- Exemples :

  $\mu X \,.\, \langle\, \mathrm{tt} \,\rangle\, \mathrm{tt} \wedge [\, \neg \mathrm{a} \,]\, X$ — synt. monotone

  $\mu X \,.\, \langle\, \mathrm{tt} \,\rangle\, \mathrm{tt} \wedge \neg X$ — synt. non monotone

- Formules φ en *forme normale positive (FNP)* : toutes les négations de φ ont été éliminées (par propagation « vers le bas » au moyen des dualités)

Mu-calcul modal : sémantique

Soit $M = (S, A, T, s_0)$, φ une formule en FNP et $\rho : \boldsymbol{X} \rightarrow 2^S$ un contexte qui associe des ensembles d'états aux variables propositionnelles libres de φ. Interprétation $[[\varphi]] \rho \subseteq S$:

- $[[ff]] \rho = \varnothing$
- $[[tt]] \rho = S$
- $[[\varphi_1 \vee \varphi_2]] \rho = [[\varphi_1]] \rho \cup [[\varphi_2]] \rho$
- $[[\varphi_1 \wedge \varphi_2]] \rho = [[\varphi_1]] \rho \cap [[\varphi_2]] \rho$

Sémantique (suite)

- $[[\langle \alpha \rangle \varphi]] \rho = \{ s \in S \mid \exists s \rightarrow_a s' . a \in [[\alpha]] \wedge s' \in [[\varphi]] \rho \}$
- $[[[\alpha] \varphi]] \rho = \{ s \in S \mid \forall s \rightarrow_a s' . a \in [[\alpha]] \Rightarrow s' \in [[\varphi]] \rho \}$
- $[[X]] \rho = \rho (X)$
- $[[\mu X . \varphi]] \rho = \cup_{k \geq 0} \Phi_\rho^k (\varnothing)$
- $[[\nu X . \varphi]] \rho = \cap_{k \geq 0} \Phi_\rho^k (S)$

  où $\Phi_\rho : 2^S \rightarrow 2^S, \quad \Phi_\rho (U) = [[\varphi]] \rho [U / X]$

  est la fonctionnelle associée à φ et ρ

  Φ_ρ est monotone : $U_1 \subseteq U_2 \Rightarrow \Phi_\rho (U_1) \subseteq \Phi_\rho (U_2)$

Point fixe minimal

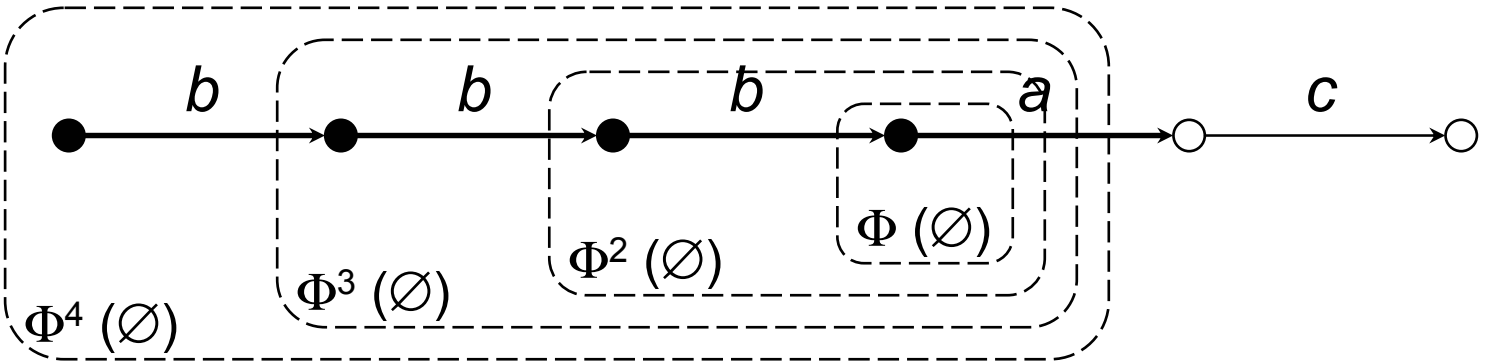
- Exécution potentielle d'une action *a* (existence d'un chemin qui mène à une *a*-transition) :

$$\mu X \,.\, \langle\, a \,\rangle\, \mathrm{tt} \vee \langle\, \mathrm{tt} \,\rangle\, X$$

- Fonctionnelle associée :

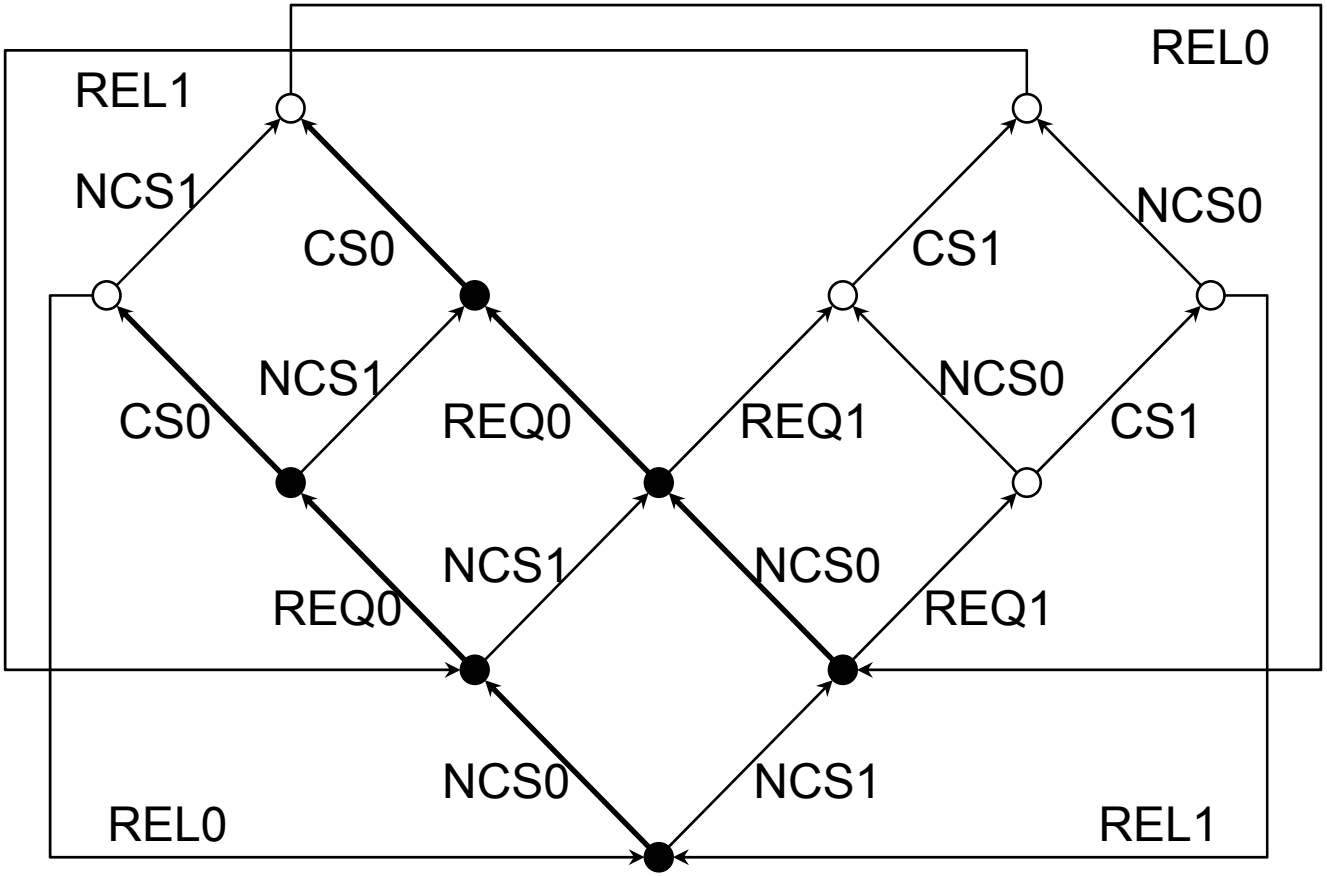
$$\Phi\,(U\,) = [[\, \langle\, a \,\rangle\, \mathrm{tt} \vee \langle\, \mathrm{tt} \,\rangle\, X \,]]\ \ [\, U \,/\, X \,]$$

- Evaluation sur un modèle STE :



Exemple

Exécution potentielle : $\mu X \,.\, \langle\, CS_0 \,\rangle\, tt \vee \langle\, \neg(REL_1 \vee REL_0) \,\rangle\, X$



Point fixe maximal

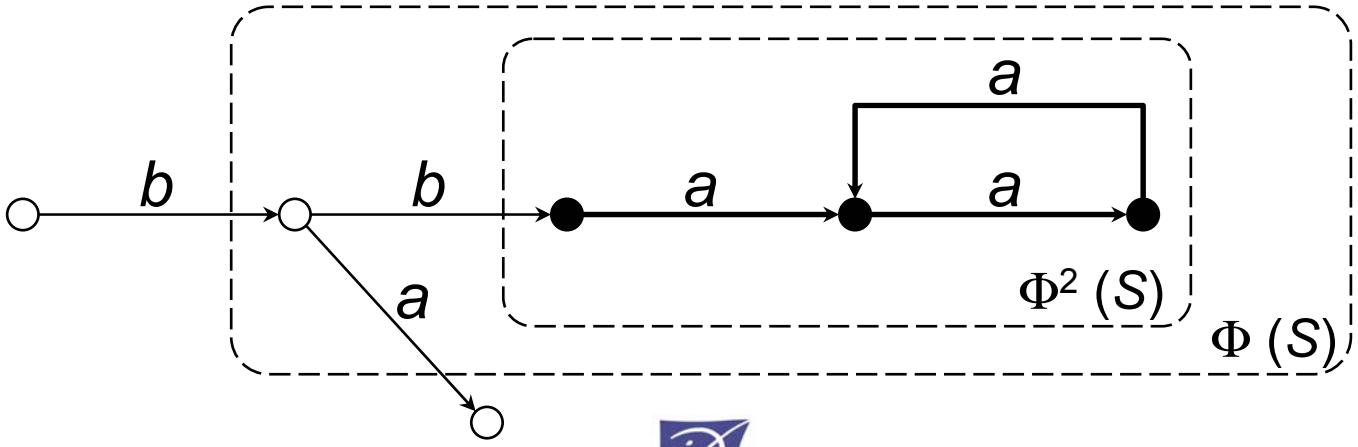
- Répétition infinie d'une action *a* (existence d'un circuit composé de *a*-transitions) :

$$\nu X \,.\, \langle\, a \,\rangle\, X$$

- Fonctionnelle associée :

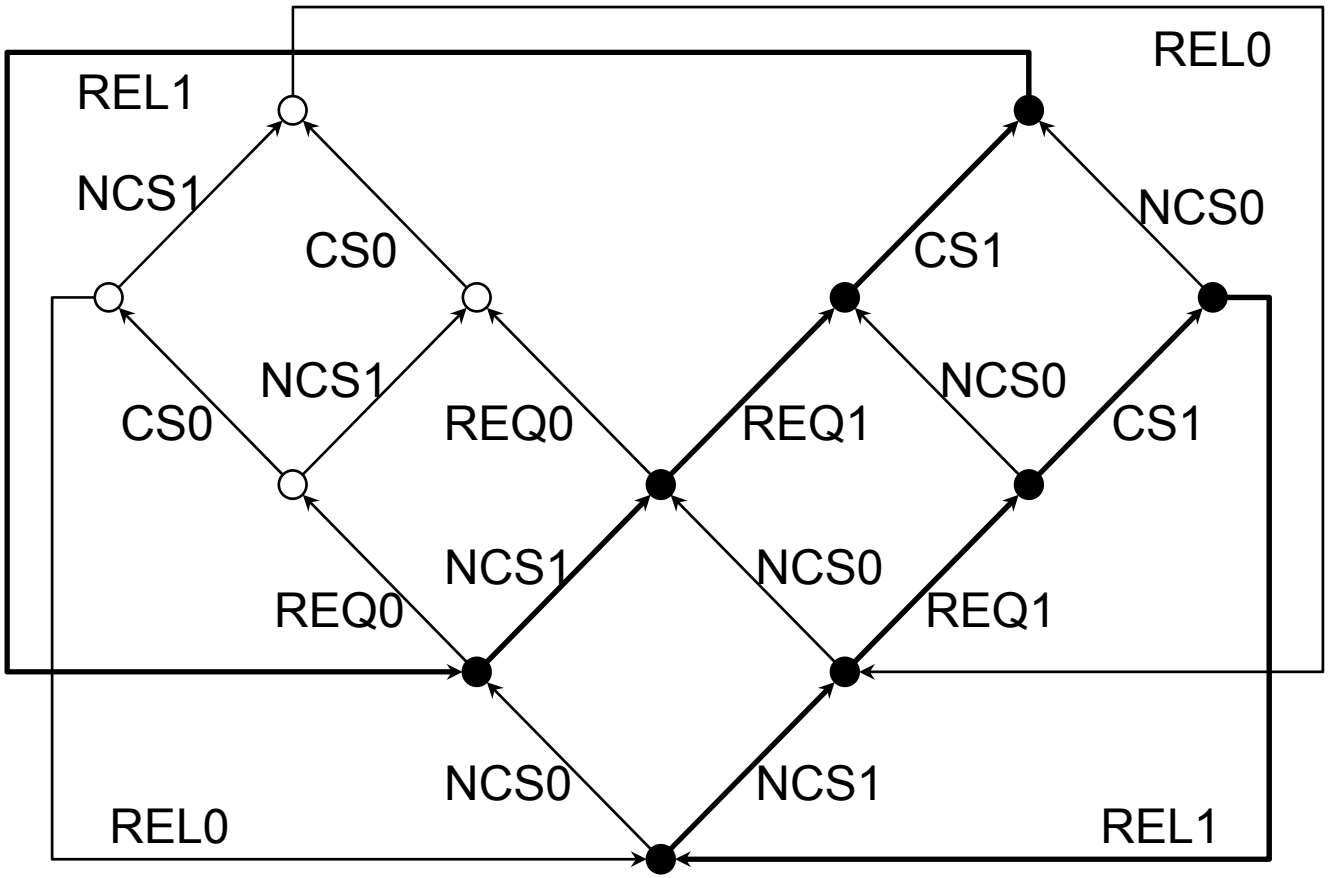
$$\Phi\,(U\,) = [[\,\langle\, a \,\rangle\, X\,]]\;\; [\,U\,/\,X\,]$$

- Evaluation sur un modèle STE :



Exemple

Répétition infinie : $\nu X \,.\, \langle\, NCS_1 \vee REQ_1 \vee CS_1 \vee REL_1 \,\rangle\, X$



Quelques identités

Théorème de Tarski [Tarski-55] :

- $U \subseteq [[\ \varphi\ ]]\ \rho\ [\ U\ /\ X\ ] \Rightarrow U \subseteq [[\ \nu X\ .\ \varphi\ ]]\ \rho$
- $[[\ \varphi\ ]]\ \rho\ [\ U\ /\ X\ ] \subseteq U \Rightarrow [[\ \mu X\ .\ \varphi\ ]]\ \rho \subseteq U$

Description des opérateurs d'ACTL :

- $EF_\alpha\ \varphi = \mu X\ .\ \varphi \vee \langle\ \alpha\ \rangle\ X$
- $AF_\alpha\ \varphi = \mu X\ .\ \varphi \vee (\langle\ tt\ \rangle\ tt \wedge [\ \alpha\ ]\ X\ )$
- $AG_\alpha\ \varphi = \nu X\ .\ \varphi \wedge [\ \alpha\ ]\ X$
- $EG_\alpha\ \varphi = \nu X\ .\ \varphi \wedge ([\ tt\ ]\ ff \vee \langle\ \alpha\ \rangle\ X\ )$

Description des opérateurs de PDL :

- $\langle\ \beta^*\ \rangle\ \varphi = \mu X\ .\ \varphi \vee \langle\ \beta\ \rangle\ X$
- $[\ \beta^*\ ]\ \varphi = \nu X\ .\ \varphi \wedge [\ \beta\ ]\ X$

Accessibilité inévitable

- Accessibilité inévitable d'une action *a* :

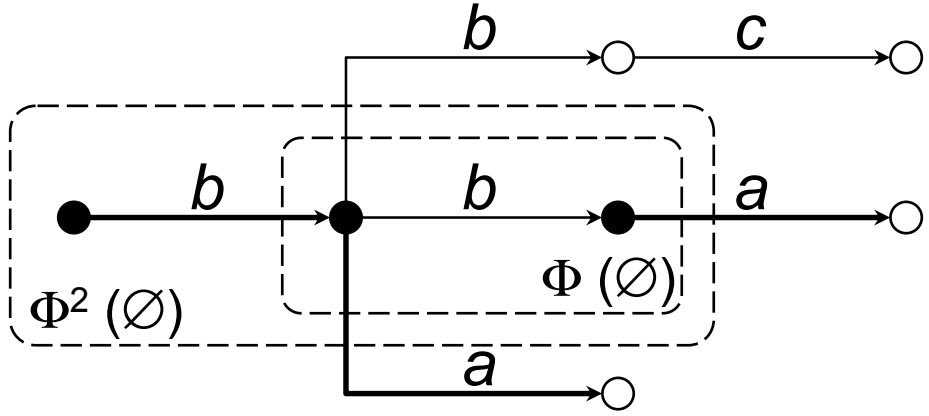
  access (*a*) = $AF_{tt} \langle a \rangle tt =$

  $\mu X . \langle a \rangle tt \vee (\langle tt \rangle tt \wedge [tt] X)$

- Fonctionnelle associée :

  $\Phi (U) = [[\langle a \rangle tt \vee (\langle tt \rangle tt \wedge [tt] X)]] [U / X]$

- Evaluation sur un modèle STE :



Exécution inévitable

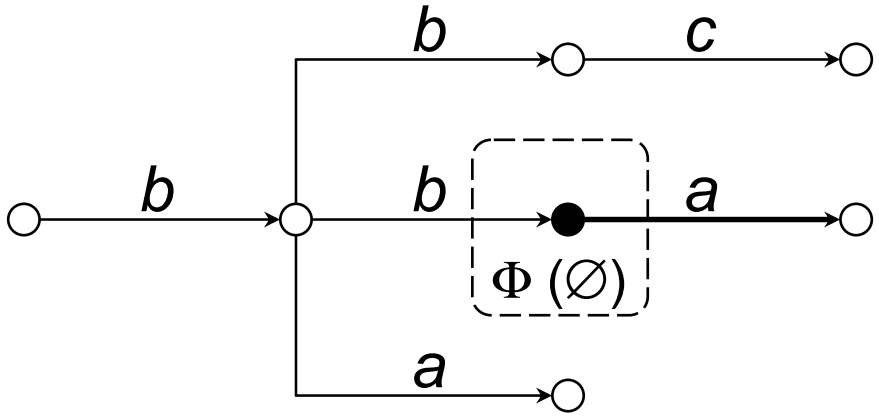
- Exécution inévitable d'une action *a* :

  $$\text{inev}\,(a) = \mu X\,.\,\langle\,\text{tt}\,\rangle\,\text{tt} \wedge [\,\neg a\,]\,X$$

- Fonctionnelle associée :

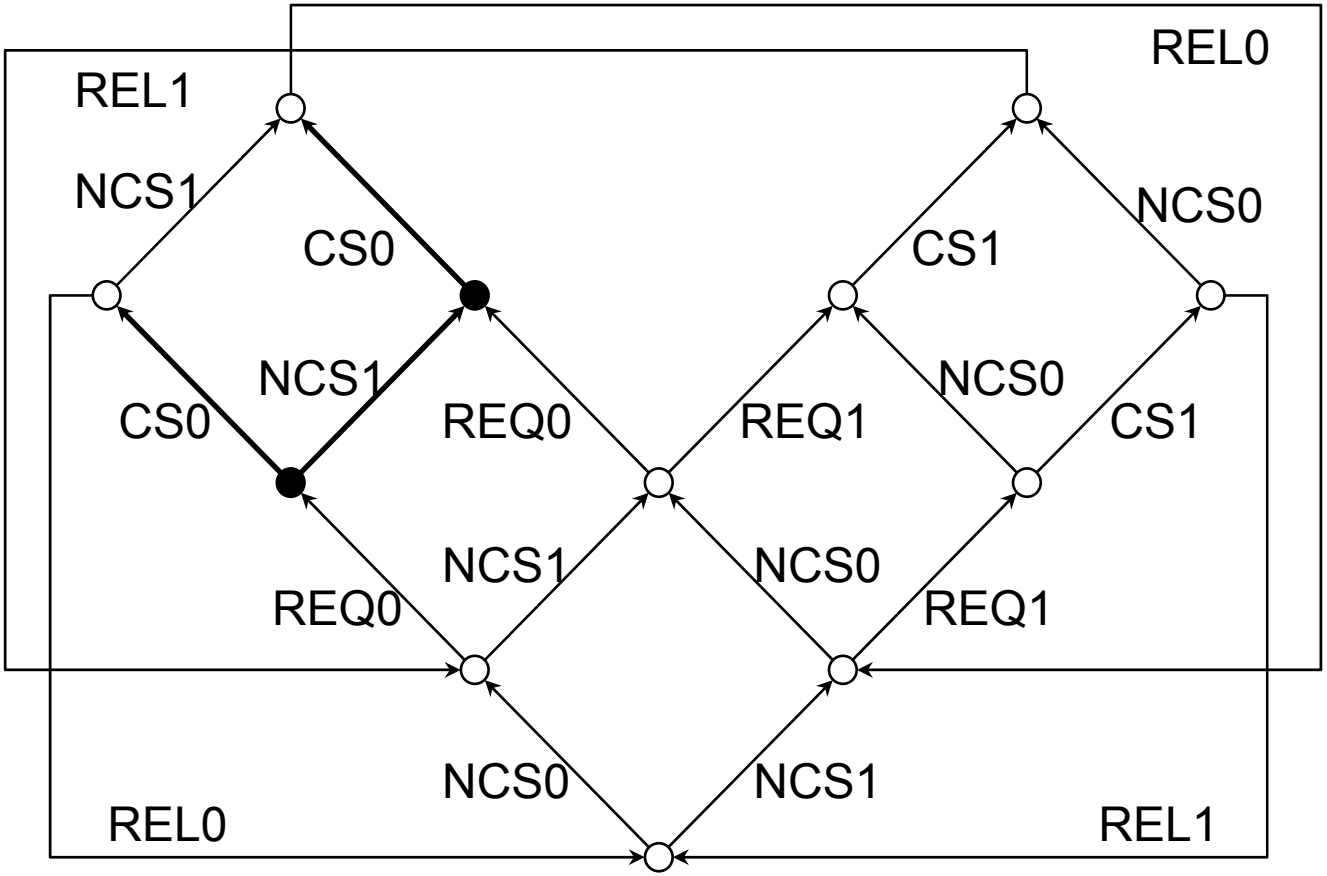
  $$\Phi\,(U\,) = [[\,\langle\,\text{tt}\,\rangle\,\text{tt} \wedge [\,\neg a\,]\,X\,]]\;\;[\,U\,/\,X\,]$$

- Evaluation sur un modèle STE :



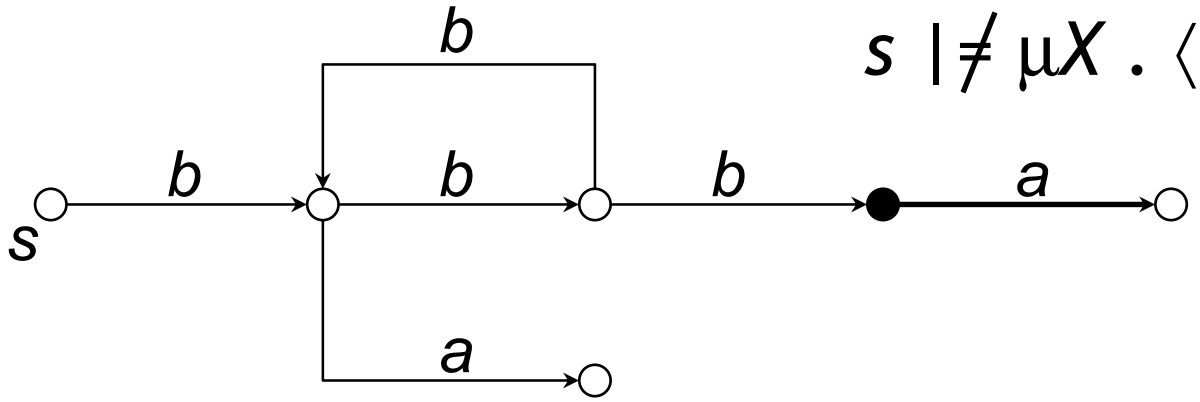
Exemple

Exécution inévitable : $\mu X \,.\, \langle\, tt \,\rangle\, tt \wedge [\, \neg CS_0 \,]\, X$



Propriétés d'équité

- Problème : pas d'exécution inévitable de l'action CS_0 à partir de l'état initial du STE $\Rightarrow$ le processus P_1 peut monopoliser indéfiniment la section critique



$$s \not\models \mu X \,.\, \langle\, \mathrm{tt} \,\rangle\, \mathrm{tt} \wedge [\, \neg a \,]\, X$$

- *Exécution équitable* :

  à partir d'un état, tous les chemins sans circuits contiennent l'action *a*

Exécution équitable

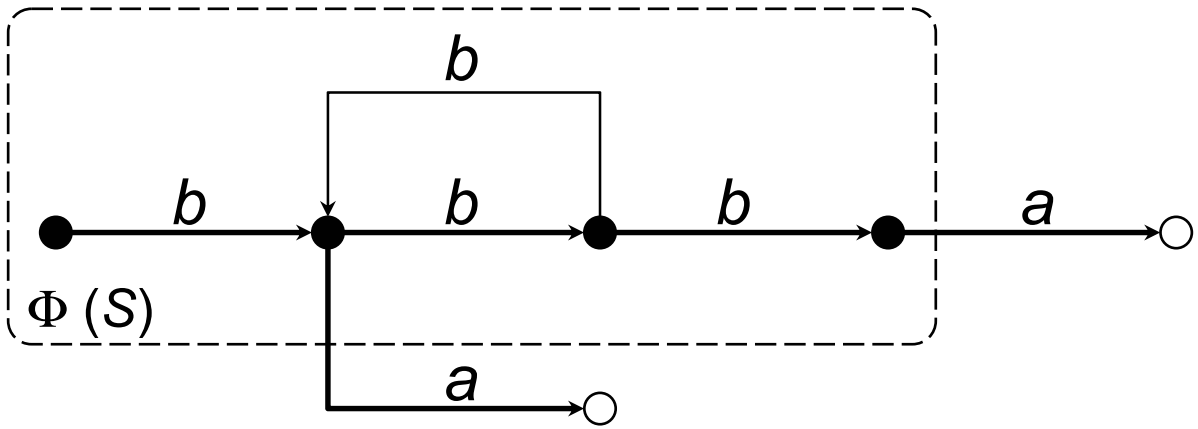
- Exécution équitable d'une action *a* :

$$\text{fair}\ (a) = [\ (\neg a)^*\ ]\ \langle\ \text{tt}^*.\ a\ \rangle\ \text{tt}$$

$$= \nu X\ .\ \langle\ \text{tt}^*.\ a\ \rangle\ \text{tt} \wedge [\ \neg a\ ]\ X$$

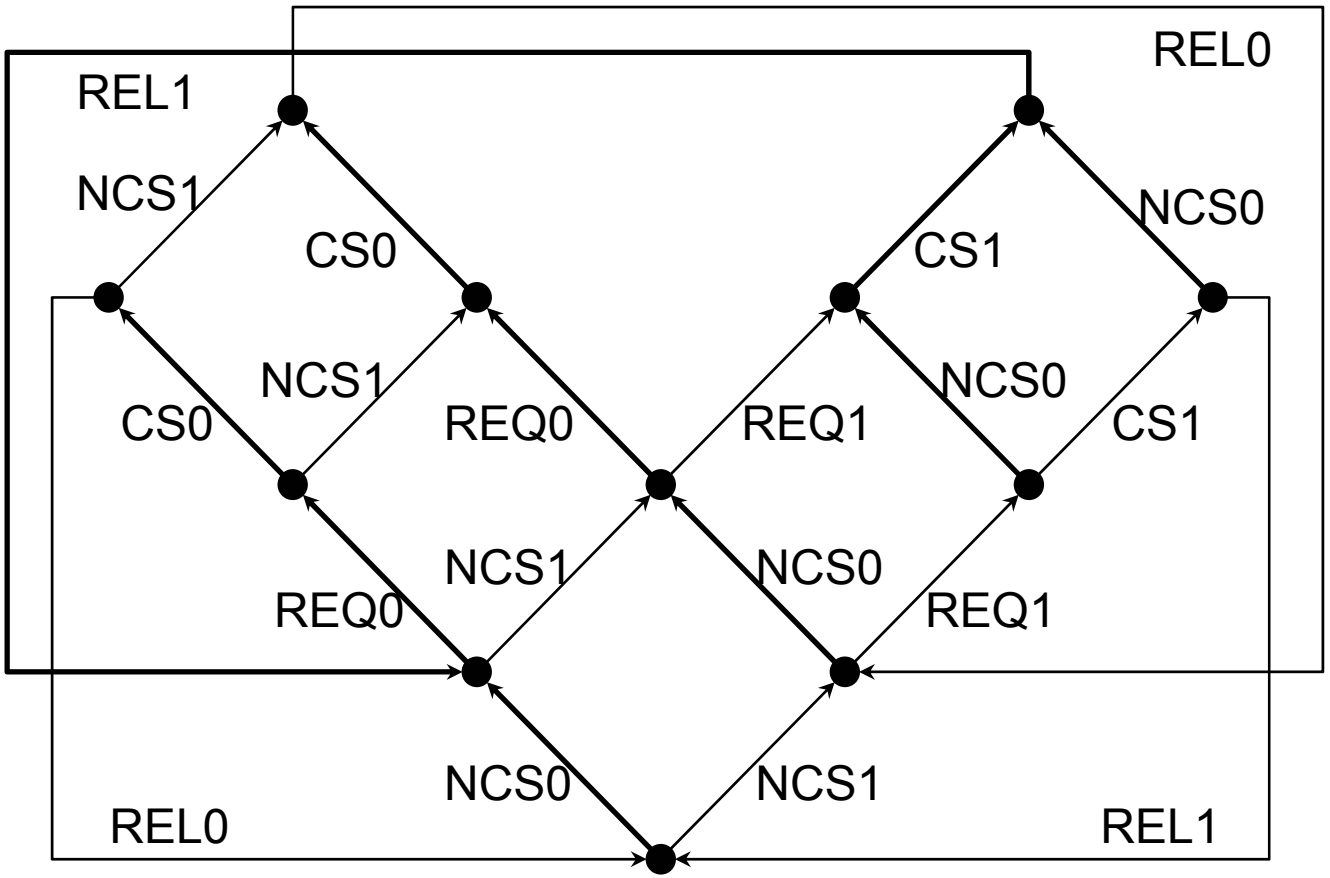
- Fonctionnelle associée :

$$\Phi\ (U\ ) = [[\ \langle\ \text{tt}^*.\ a\ \rangle\ \text{tt} \wedge [\ \neg a\ ]\ X\ ]]\ \ [\ U\ /\ X\ ]$$



Exemple

Exécution équitable : $[\ \neg\ CS_0\ ]\ \langle\ tt^*.\ CS_0\ \rangle\ tt$



Profondeur d'alternance

- Profondeur d'alternance (*alternation depth*) d'une formule φ [Emerson-Lei-86] : degré de récursivité mutuelle des points fixes μ et ν dans φ
- Exemples :

  formule d'alternance 1 :

  $$\mu X \,.\, \langle\, \mathrm{tt} \,\rangle\, \mathrm{tt} \vee \langle\, a \,\rangle\, X$$

  formule d'alternance 2 :

  $$\nu X \,.\, \langle\, a^* .\, b \,\rangle\, X = \nu X \,.\, \mu Y \,.\, \langle\, b \,\rangle\, X \vee \langle\, a \,\rangle\, Y$$

- Les formules d'alternance ≥ 3 n'apparaissent quasiment jamais en pratique

Hiérarchie des fragments du mu-calcul

- Les fragments du mu-calcul d'alternance *n* (notés $L\mu_n$) forment une hiérarchie en termes d'expressivité et de complexité du model-checking:

  $L\mu_1 < L\mu_2 < \ldots < L\mu_n < \ldots L\mu$

- Model-checking de $L\mu_n$ sur un STE $M = (S, A, T, s_0)$:

  $O((|\varphi| \times (|S| + |T|))^n)$

  où $|\varphi|$ = nombre d'opérateurs de φ

  $|S|$, $|T|$ = nombre d'états et de transitions de *M*

- En pratique, le fragment intéressant est $L\mu_1$ (*alternation-free*) : pas de récursivité mutuelle des points fixes μ et ν, mais model-checking linéaire

Mu-calcul d'alternance 1

- Récursivité mutuelle autorisée uniquement entre des points fixes de même signe :

  $\langle$ (Send . tt* . Error)* . Recv $\rangle$ tt =

  μX . ($\langle$ Recv $\rangle$ tt $\vee$ $\langle$ Send $\rangle$ μY . ($\langle$ Error $\rangle$ X $\vee$ $\langle$ tt $\rangle$ Y))
- Subsume différentes logiques utiles :
  - logiques arborescentes (ACTL)
  - logiques régulières (PDL)
- Model-checking linéaire en taille de φ et M :
  - Algorithmes globaux [Cleaveland-Steffen-93]
  - Algorithmes locaux [Andersen-94,Mateescu-Sighireanu-00]

Mu-calcul et bisimulation

- Soient $M_1 = (S_1, A, T_1, s_{01})$ et $M_2 = (S_2, A, T_2, s_{02})$ deux STE. La *bisimulation forte* $\approx \subseteq S_1 \times S_2$ est la plus grande relation telle que

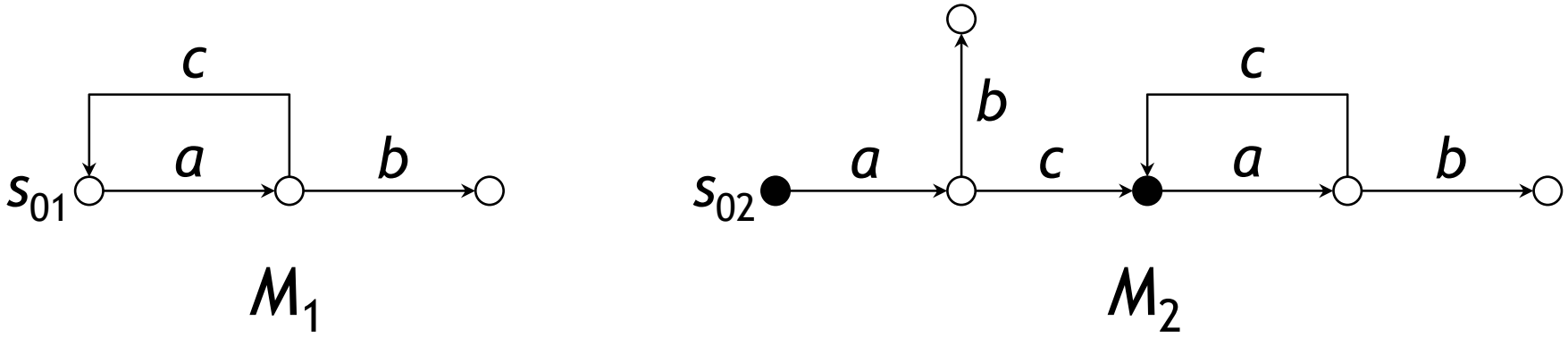
$$s_1 \approx s_2 \Leftrightarrow \forall (s_1,a,s_1') \in T_1 \,.\, \exists (s_2,a,s_2') \in T_2 \,.\, s_1' \approx s_2'$$
$$\wedge\ \forall (s_2,a,s_2') \in T_2 \,.\, \exists (s_1,a,s_1') \in T_1 \,.\, s_1' \approx s_2'$$

- M_1 et M_2 sont *bisimilaires* ssi $s_{01} \approx s_{02}$
- Description de la bisimulation au moyen de *formules caractéristiques* [Ingolfsdottir-Steffen-94]:

  si φ_{M1} est la formule caractéristique de M_1

  alors $s_{01} \approx s_{02} \Leftrightarrow s_{02} \models \varphi_{M1}$

Exemple



- Formule caractéristique de M_1 :

  $\varphi_{M1} = \nu X \,.\, [\neg a]\ \mathrm{ff} \wedge \langle a \rangle\ ([\neg(b \vee c)]\ \mathrm{ff} \wedge \langle b \rangle\ \mathrm{tt} \wedge \langle c \rangle\ X)$

$$s_{02} \models \varphi_{M1} \quad \Rightarrow \quad s_{01} \approx s_{02}$$

Mu-calcul et bisimulations faibles

- En présence d'actions internes (τ), des bisimulations *faibles* sont nécessaires pour comparer deux STEs :
  - Équiv. observationnelle [Milner-89]
  - Équiv. de branchement [Glabbeek-Weijland-89]
- Il existe des procédures de construction de formules caractéristiques pour les bisimulations faibles [Ingolfsdottir-Steffen-94]
- Ces formules sont d'alternance 2 (points fixes maximaux contenant des modalités $\langle \tau^* \rangle$ tt)

Logiques de point fixe (résumé)

- Très expressives : subsument virtuellement toutes les LT proposées dans la littérature
- Permettent de coder les relations de bisimulation (équivalence forte et équivalences faibles)
- Puissance d'expression obtenue en *imbriquant* les opérateurs de point fixe :

  $\langle\ (a\ .\ b^*)^*.c\ \rangle\ \mathrm{tt} = \mu X.\langle\ c\ \rangle\ \mathrm{tt} \vee \langle\ a\ \rangle\ \mu Y\ .\ (X \vee \langle\ b\ \rangle\ Y\ )$
- Pour le μ-calcul modal complet (alternance quelconque) : model-checking exponentiel en taille de la formule et du STE
- En pratique : restriction aux fragments utiles (alternance 1 $\Rightarrow$ model-checking linéaire)

Model-checking

- Problème : étant donné un STE $M = (S, A, T, s_0)$ et une formule fermée φ, vérifier si $M \models \varphi$
- On distingue deux types de model-checking :

  Model-checking global : $S = [[\varphi]]$

  le STE doit être préalablement construit

  Model-checking local : $s_0 \in [[[tt^*] \varphi]]$

  le STE peut être construit à la volée
- Les algorithmes locaux permettent de détecter des erreurs même si le STE est trop grand pour être stocké dans la mémoire d'une machine

Outils de vérification

CADP / Evaluator 3.0 (INRIA Rhône-Alpes / VASY)

- Logique temporelle : μ-calcul régulier d'alternance 1 (modalités de PDL + points fixes)
- Vérification à la volée (construction du STE pendant la vérification d'une formule)
- Génération de diagnostics (exemples et contre-exemples) pour les formules
- Facilités de macro-définition et inclusion de bibliothèques d'opérateurs personnalisés
- Utilisation pour 12 études de cas

  http://www.inrialpes.fr/vasy/cadp