Spécification logique de réseaux de Petri

Guillaume Feuillade

IRISA, S4

Soutenance de Thèse

Encadrement : Sophie Pinchinat

Contexte

- Systèmes réactifs
  - séquentiels
  - concurrents
- Formalismes de spécifications
  - logiques temporelles,
  - autre système,
  - autres : HMSC, ...

Contexte

- Systèmes réactifs
  - séquentiels
  - concurrents
- Formalismes de spécifications
  - logiques temporelles,
  - autre système,
  - autres : HMSC, ...

Ici : Problème de synthèse de réseaux de Petri à partir du Mu-calcul

Synthèse de systèmes

- À partir de formules du Mu-Calcul
  - propriété du modèle fini
    $\Rightarrow$ synthèse de modèles séquentiels
  - problème de la synthèse distribuée :
    généralement indécidable [PR90]
- Synthèse de réseau de Petri
  - méthode des régions [BD99,Dar04]
  - à partir de spécifications (*automatiques*) [BD04]

Plan

Intervalle de langages

Définition (Intervalle de langages)

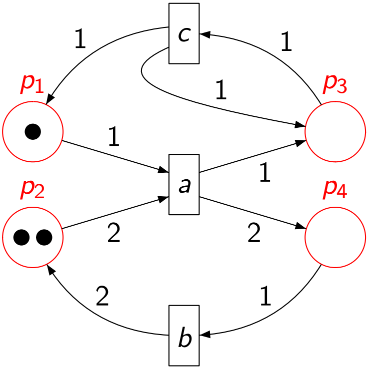
$[L_1, L_2]$ où L_1 et L_2 sont rationnels et clos par préfixe.

Modèles de $[L_1, L_2]$: langages L vérifiant $L_1 \subseteq L \subseteq L_2$.

Réseaux de Petri

$N = \langle P, \Sigma, F, W \rangle$ avec :

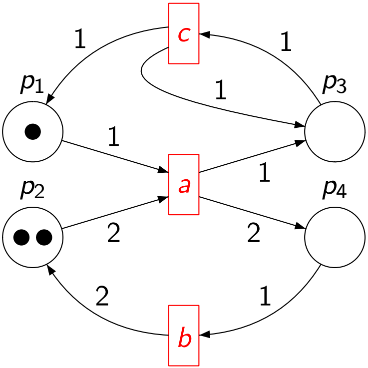
- P un ensemble de places,



Réseaux de Petri

$N = \langle P, \Sigma, F, W \rangle$ avec :

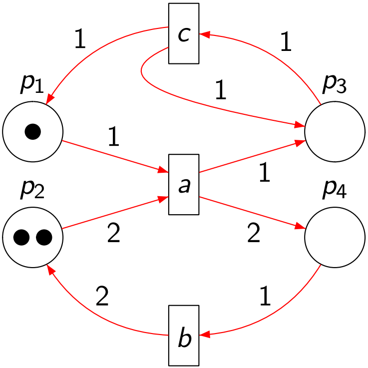
- P un ensemble de places,
- Σ un ensemble d'actions,



Réseaux de Petri

$N = \langle P, \Sigma, F, W \rangle$ avec :

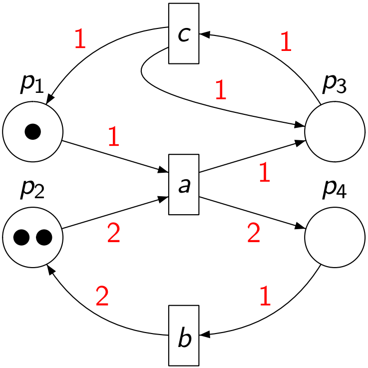
- P un ensemble de places,
- Σ un ensemble d'actions,
- $F \subseteq (\Sigma \times P) \cup (P \times \Sigma)$ un ensemble d'arcs,



Réseaux de Petri

$N = \langle P, \Sigma, F, W \rangle$ avec :

- P un ensemble de places,
- Σ un ensemble d'actions,
- $F \subseteq (\Sigma \times P) \cup (P \times \Sigma)$ un ensemble d'arcs,
- $W : F \rightarrow \mathbb{N}$ les poids des arcs.

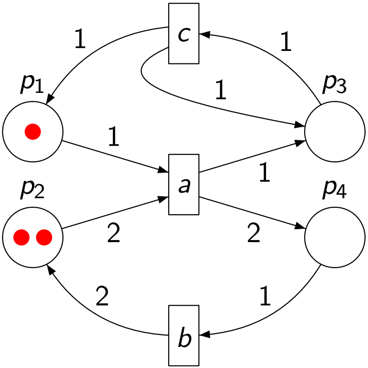


Réseaux de Petri

$N = \langle P, \Sigma, F, W \rangle$ avec :

- P un ensemble de places,
- Σ un ensemble d'actions,
- $F \subseteq (\Sigma \times P) \cup (P \times \Sigma)$ un ensemble d'arcs,
- $W : F \to \mathbb{N}$ les poids des arcs.

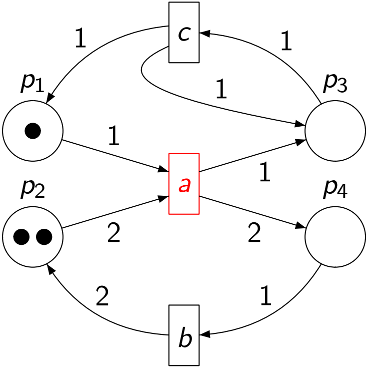
Un marquage est une application $P \to \mathbb{N}$



Réseaux de Petri 2 : règle de tir

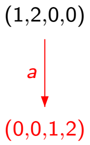
(1,2,0,0)

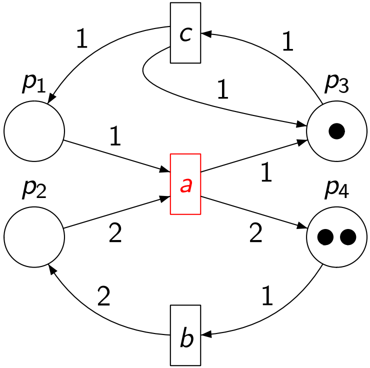
$(1,2,0,0)[a\rangle$?



Réseaux de Petri 2 : règle de tir

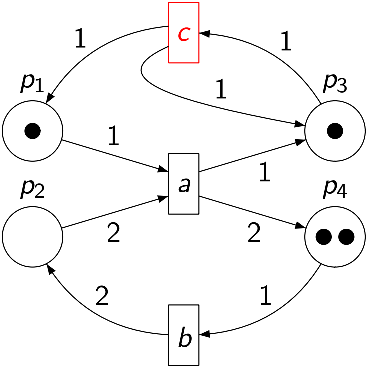
$(1,2,0,0)[a\rangle\,(0,0,1,2)$

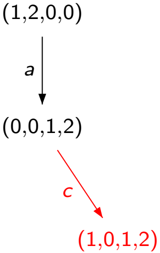




Réseaux de Petri 2 : règle de tir

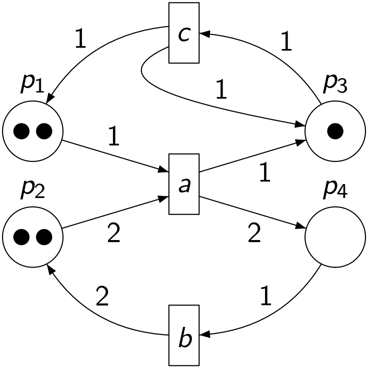
$(1,2,0,0)[a.c\rangle\,(1,0,1,2)$

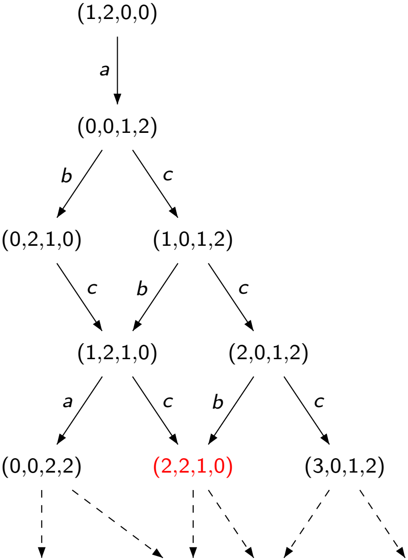




Réseaux de Petri 2 : règle de tir

$$(1,2,0,0)[a.c.c.b\rangle\,(2,2,1,0)$$





Réseaux de Petri 3

Définition (Langage d'un réseau de Petri)

$L(N, m_0) = \{$ séquences tirables à partir de $m_0\}$

Remarque

Les réseaux de Petri non étiquetés n'engendrent pas tous les langages rationnels

Réseaux de Petri 3

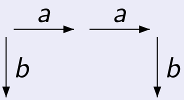
Définition (Langage d'un réseau de Petri)

$L(N, m_0) = \{$ séquences tirables à partir de $m_0\}$

Remarque

Les réseaux de Petri non étiquetés n'engendrent pas tous les langages rationnels

Contre Exemple



$\{$ 1,b,a,a.a,a.a.b $\}$ n'est pas langage de réseau.

Problématique de la synthèse de réseau

Problèmes : synthèse de réseau de Petri

- Donnée : L un langage rationnel
  existe-t-il un réseau tel que $L(N, m_0) = L$?

Problématique de la synthèse de réseau

Problèmes : synthèse de réseau de Petri

- Donnée : L un langage rationnel
  existe-t-il un réseau tel que $L(N, m_0) = L$?
- Donnée : S une spécification
  existe-t-il un réseau tel que $L(N, m_0) \in \mathrm{modèles}(S)$?

Plan

Synthèse : méthode des régions

$\Sigma = \{a_1 \dots a_n\}$ un alphabet fixé.

Définition (Régions d'un langage)

Une région de $L \subseteq \Sigma^*$ est un réseau à une place N tel que $L \subseteq L(N, m_0)$

Synthèse : méthode des régions 2

Proposition (Darondeau 04)

Les régions de L sont finiment engendrées

Théorème (Darondeau 04)

L'ensemble des générateurs des régions de L est calculable
Il forme le réseau N_L tel que

$$L(N_L, m_0) = min\{R \mid L \subseteq R\}$$

Synthèse : méthode des régions 2

Proposition (Darondeau 04)

Les régions de L sont finiment engendrées

Théorème (Darondeau 04)

L'ensemble des générateurs des régions de L est calculable
Il forme le réseau N_L tel que

$$L(N_L, m_0) = min\{R \mid L \subseteq R\}$$

Conséquences

- La synthèse d'un réseau à partir de L est décidable

Synthèse : méthode des régions 2

Proposition (Darondeau 04)

Les régions de L sont finiment engendrées

Théorème (Darondeau 04)

L'ensemble des générateurs des régions de L est calculable
Il forme le réseau N_L tel que

$$L(N_L, m_0) = min\{R \mid L \subseteq R\}$$

Conséquences

- La synthèse d'un réseau à partir de L est décidable
- La synthèse d'un réseau à partir de $[L_1, L_2]$ est décidable

Plan

Logiques considérées

Interprétées sur un langage L

- Intervalles de langages

Logiques considérées

Interprétées sur un langage L

- Intervalles de langages

- Mu-Calcul (sans alternation)

Logiques considérées

Interprétées sur un langage L

- Intervalles de langages
- Nu-Calcul Conjonctif
- Mu-Calcul (sans alternation)

Interprétation sur les réseaux de Petri

Définition

N satisfait Φ si et seulement si L_N est modèle de Φ.

Remarque (Modèle fini)

Le Mu-Calcul interprété sur les réseaux de Petri ne possède pas la propriété du modèle fini.

Contre Exemple

Φ : *"P peut effectuer a^* et $a.b$ mais pas b"*

Interprétation sur les réseaux de Petri

Définition

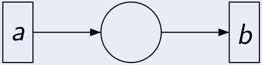
N satisfait Φ si et seulement si L_N est modèle de Φ.

Remarque (Modèle fini)

Le Mu-Calcul interprété sur les réseaux de Petri ne possède pas la propriété du modèle fini.

Contre Exemple

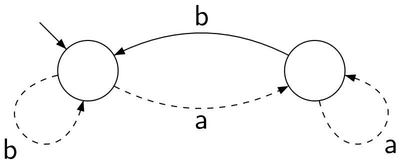
Φ : *"P peut effectuer a^* et $a.b$ mais pas b"*



Plan

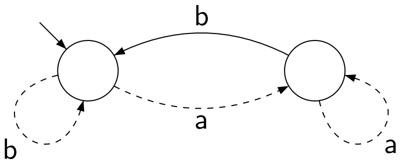
Un reconnaisseur : les spécifications modales

Une spécification S :



Un reconnaisseur : les spécifications modales

Une spécification S :

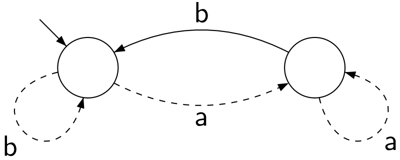


Définition (Spécification modale sur Σ)

$S = \langle \{C_a\}_{a \in \Sigma}, I \rangle$ où C_a et I sont des langages rationnels sur Σ

Un reconnaisseur : les spécifications modales

Une spécification S :



Définition (Spécification modale sur Σ)

$S = \langle \{C_a\}_{a \in \Sigma}, I \rangle$ où C_a et I sont des langages rationnels sur Σ

Ici : $C_a = \emptyset$, $C_b = (b^*.a)^+$, $C_c = \emptyset$ et $I = (a+b)^*.c$

Spécifications modales : sémantique

Sémantique

S-Complétion de L :

$$C_S(L) = \bigcup_{a \in \Sigma} (L \cap C_a).a$$

Modèles de S :

$$mod(S) = \{L \subseteq \Sigma^* \mid C_S(L) \subseteq L \text{ et } L \cap I = \emptyset\}$$

Spécifications modales : sémantique

Sémantique

S-Complétion de L :

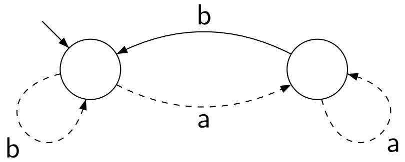
$$C_S(L) = \bigcup_{a \in \Sigma} (L \cap C_a).a$$

Modèles de S :

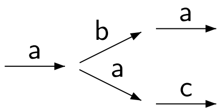
$$mod(S) = \{L \subseteq \Sigma^* \mid C_S(L) \subseteq L \text{ et } L \cap I = \emptyset\}$$

Spécifications modales : reconnaissance

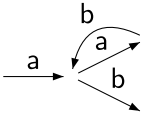
$S = \langle \{C_a\}_{a \in \Sigma}, I \rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a + b)^*.c$:



$L_1 \notin mod(S)$

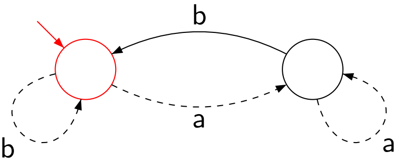


$L_2 \in mod(S)$

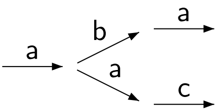


Spécifications modales : reconnaissance

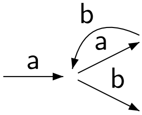
$S = \langle \{C_a\}_{a \in \Sigma}, I \rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a + b)^*.c$:



$L_1 \notin mod(S)$

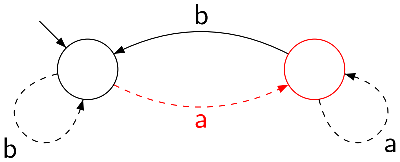


$L_2 \in mod(S)$

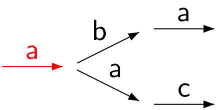


Spécifications modales : reconnaissance

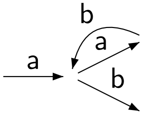
$S = \langle \{C_a\}_{a \in \Sigma}, I \rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a+b)^*.c$:



$L_1 \notin mod(S)$

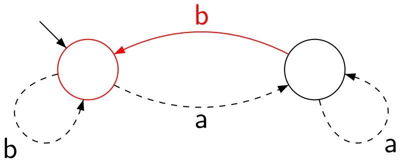


$L_2 \in mod(S)$

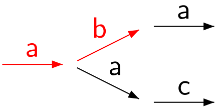


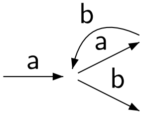
Spécifications modales : reconnaissance

$S = \langle \{C_a\}_{a \in \Sigma}, I \rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a+b)^*.c$:



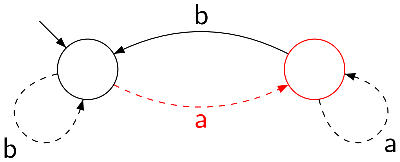
$L_1 \notin mod(S)$ $L_2 \in mod(S)$



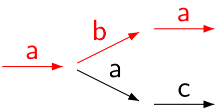


Spécifications modales : reconnaissance

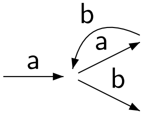
$S = \langle \{C_a\}_{a\in\Sigma}, I\rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a+b)^*.c$:



$L_1 \notin mod(S)$

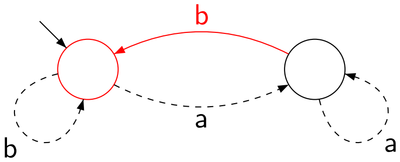


$L_2 \in mod(S)$

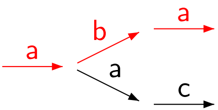


Spécifications modales : reconnaissance

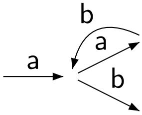
$S = \langle \{C_a\}_{a \in \Sigma}, I \rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a+b)^*.c$:



$L_1 \notin mod(S)$

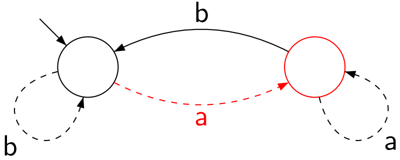


$L_2 \in mod(S)$

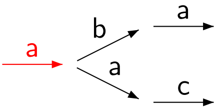


Spécifications modales : reconnaissance

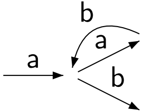
$S = \langle \{C_a\}_{a\in\Sigma}, I \rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a+b)^*.c$:



$L_1 \notin mod(S)$

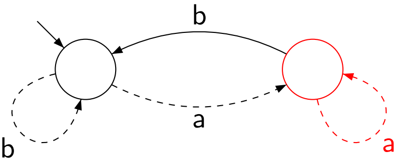


$L_2 \in mod(S)$

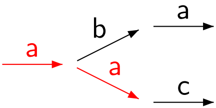


Spécifications modales : reconnaissance

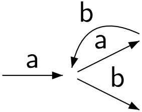
$S = \langle \{C_a\}_{a\in\Sigma}, I\rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a+b)^*.c$:



$L_1 \notin mod(S)$

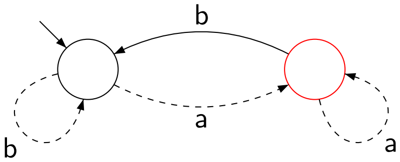


$L_2 \in mod(S)$

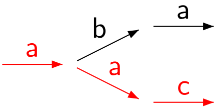


Spécifications modales : reconnaissance

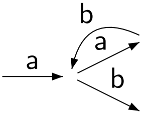
$S = \langle \{C_a\}_{a \in \Sigma}, I \rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a + b)^*.c$:



$L_1 \notin mod(S)$

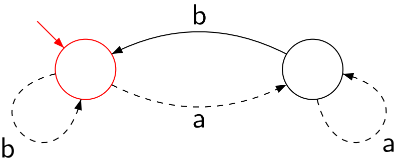


$L_2 \in mod(S)$

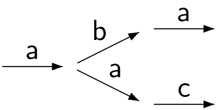


Spécifications modales : reconnaissance

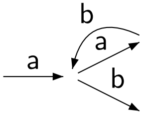
$S = \langle \{C_a\}_{a\in\Sigma}, I \rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a+b)^*.c$:



$L_1 \notin mod(S)$

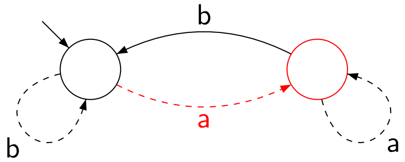


$L_2 \in mod(S)$

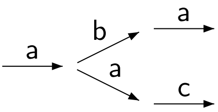


Spécifications modales : reconnaissance

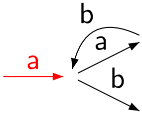
$S = \langle \{C_a\}_{a \in \Sigma}, I \rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a+b)^*.c$:



$L_1 \notin mod(S)$

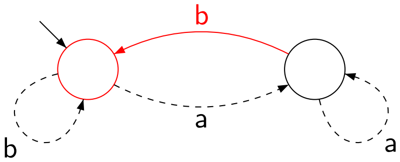


$L_2 \in mod(S)$

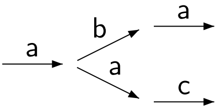


Spécifications modales : reconnaissance

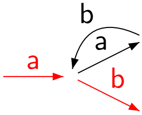
$S = \langle \{C_a\}_{a \in \Sigma}, I \rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a+b)^*.c$:



$L_1 \notin mod(S)$

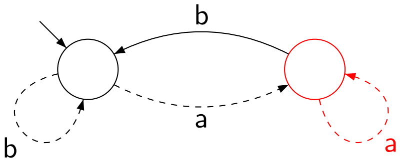


$L_2 \in mod(S)$

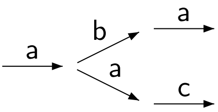


Spécifications modales : reconnaissance

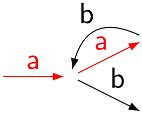
$S = \langle \{C_a\}_{a\in\Sigma}, I\rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a+b)^*.c$:



$L_1 \notin mod(S)$

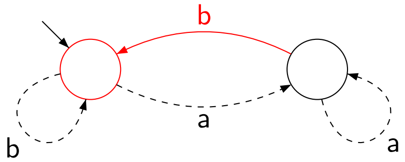


$L_2 \in mod(S)$

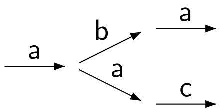


Spécifications modales : reconnaissance

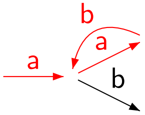
$S = \langle \{C_a\}_{a\in\Sigma}, I\rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a+b)^*.c$:



$L_1 \notin mod(S)$

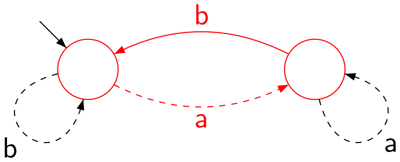


$L_2 \in mod(S)$

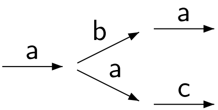


Spécifications modales : reconnaissance

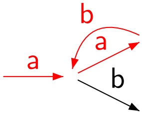
$S = \langle \{C_a\}_{a \in \Sigma}, I \rangle$ sur $\Sigma = \{a, b, c\}$, avec $C_b = (b^*.a)^+$ et $I = (a+b)^*.c$:



$L_1 \notin mod(S)$



$L_2 \in mod(S)$



Structure des modèles d'une spécification

On suppose $mod(S) \neq \emptyset$.

Théorème (Structure des modèles de S)

$(mod(S), \subseteq)$ *forme un treillis complet distributif.*

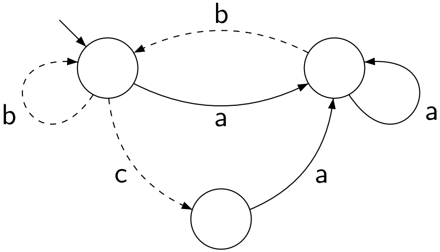
Proposition

$L_\top = \max(mod(S))$ *et* $L_\perp = \min(mod(S))$ *sont des langages rationnels*

Plus fort que la propriété du modèle fini du Mu-Calcul

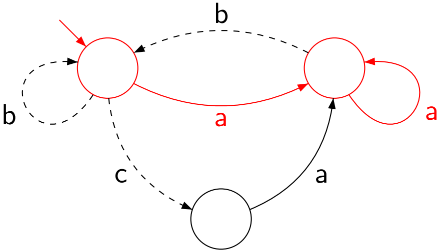
Structure des modèles d’une spécification 2

La spécification S :



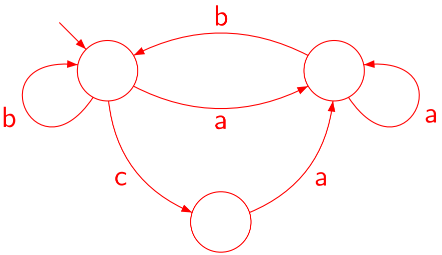
Structure des modèles d'une spécification 2

Le langage $L_\perp$:



Structure des modèles d'une spécification 2

Le langage $L_{\top}$:



Nu-Calcul conjonctif et spécifications modales [MSR05]

Théorème

Nu-Calcul conjonctif $=$ *spécifications modales.*

Λ_i-Spécifications modales et synthèse

Définition (Spécification Λ_i)

L est Λ_i si $L = \bigcup_{finie}\{u.(w_1 + \ldots + w_i)^*.v\}$

S est Λ_i si ses composantes sont Λ_i

Décidabilité de la synthèse

Λ_i-Spécifications modales et synthèse

Définition (Spécification Λ_i)

L est Λ_i si $L = \bigcup_{finie}\{u.(w_1 + \ldots + w_i)^*.v\}$
S est Λ_i si ses composantes sont Λ_i

Décidabilité de la synthèse

Spécification	Décidable ?

Λ_i-Spécifications modales et synthèse

Définition (Spécification Λ_i)

L est Λ_i si $L = \bigcup_{finie}\{u.(w_1 + \ldots + w_i)^*.v\}$
S est Λ_i si ses composantes sont Λ_i

Décidabilité de la synthèse

Spécification	Décidable ?	
Λ_0	oui	$\bigcup_{finie} [L_1, L_2]$

Λ_i-Spécifications modales et synthèse

Définition (Spécification Λ_i)

L est Λ_i si $L = \bigcup_{finie} \{u.(w_1 + \ldots + w_i)^*.v\}$
S est Λ_i si ses composantes sont Λ_i

Décidabilité de la synthèse

Spécification	Décidable ?	
Λ_0	oui	$\bigcup_{finie} [L_1, L_2]$
Λ_1	oui	

Λ_i-Spécifications modales et synthèse

Définition (Spécification Λ_i)

L est Λ_i si $L = \bigcup_{finie} \{u.(w_1 + \ldots + w_i)^*.v\}$
S est Λ_i si ses composantes sont Λ_i

Décidabilité de la synthèse

Spécification	Décidable ?	
Λ_0	oui	$\bigcup_{finie} [L_1, L_2]$
Λ_1	oui	
$R^*.\Lambda_1$	oui	

Λ_i-Spécifications modales et synthèse

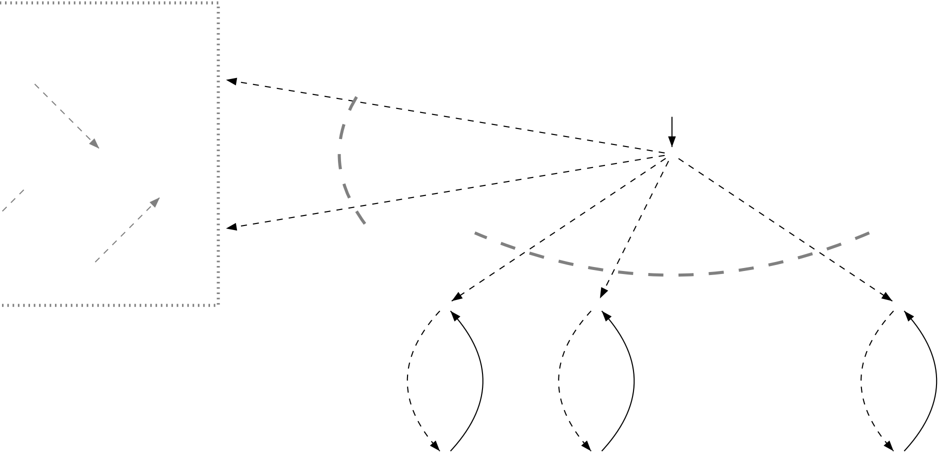
Définition (Spécification Λ_i)

L est Λ_i si $L = \bigcup_{finie} \{u.(w_1 + \ldots + w_i)^*.v\}$
S est Λ_i si ses composantes sont Λ_i

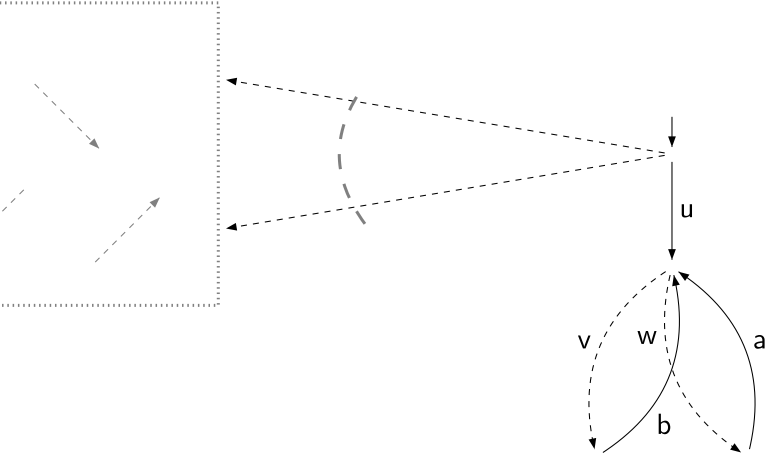
Décidabilité de la synthèse

Spécification	Décidable ?	
Λ_0	oui	$\bigcup_{finie} [L_1, L_2]$
Λ_1	oui	
$R^*.\Lambda_1$	oui	
Λ_2	?	

Le problème Λ_2



Le problème Λ_2



Cycles : $(v.b$ et $w.a)$, $v.b$, $w.a$, $(k_1 \times v.b + k_2 \times w.a), \ldots$

Plan

Théorie d'un réseau : intuition

Une formule $\Rightarrow$ une disjonction (infinie) de contraintes sur les places

Théorie d'un réseau : intuition

Une formule $\Rightarrow$ une disjonction (infinie) de contraintes sur les places

- universelles :

"N peut effectuer a^*"

Théorie d'un réseau : intuition

Une formule $\Rightarrow$ une disjonction (infinie) de contraintes sur les places

- universelles :

$$\text{“}N \text{ peut effectuer } a^*\text{”}$$

$\Rightarrow$ toute place de N est uniquement en sortie de la transition a.

Théorie d'un réseau : intuition

Une formule $\Rightarrow$ une disjonction (infinie) de contraintes sur les places

- universelles :

  "N peut effectuer a^*"

  $\Rightarrow$ toute place de N est uniquement en sortie de la transition a.

- existentielles :

  "N ne peut effectuer b"

Théorie d'un réseau : intuition

Une formule $\Rightarrow$ une disjonction (infinie) de contraintes sur les places

- universelles :

  "N peut effectuer a^*"

  $\Rightarrow$ toute place de N est uniquement en sortie de la transition a.

- existentielles :

  "N ne peut effectuer b"

  $\Rightarrow$ il existe une place de N dont la transition b est en entrée.

Théorie d'un réseau 2

Objectif

Énoncé logique de la structure d'un réseau.

Méthode

Donnée : Un ensemble K de contraintes linéaires sur une ou toute place d'un réseau.

1. extension de l'alphabet $\Sigma_e = \Sigma \cup \Sigma_u$
2. Σ_e-énoncé pour K
3. Projection des réseaux modèles sur Σ

$\Rightarrow$ les réseaux obtenus vérifient les contraintes K.

Théorie d'un réseau 3

Définition (Théorie d'un réseau)

Σ_e-énoncé Φ_N telle que $N' \in mod(\Phi_N) \Rightarrow N'|_{\Sigma}$ contient N

Théorie d'un réseau 3

Définition (Théorie d'un réseau)

Σ_e-énoncé Φ_N telle que $N' \in mod(\Phi_N) \Rightarrow N'|_\Sigma$ contient N

Théorème (Expression de la théorie d'un réseau)

Φ_N est de la forme $[L_1, L_2]$.

Plan

Machines à compteurs

Définition (Machine à deux compteurs)

$M = \langle Q, C, \Delta \rangle$ avec

- $Q = \{q_0, \ldots, q_n\}$ un ensemble d'états
  - q_0 état initial
  - q_n état final

Machines à compteurs

Définition (Machine à deux compteurs)

$M = \langle Q, C, \Delta \rangle$ avec

- $Q = \{q_0, \ldots, q_n\}$ un ensemble d'états
  - q_0 état initial
  - q_n état final
- $C = \{c_0, c_1\}$ deux compteurs

Machines à compteurs

Définition (Machine à deux compteurs)

$M = \langle Q, C, \Delta \rangle$ avec

- $Q = \{q_0, \ldots, q_n\}$ un ensemble d'états
  - q_0 état initial
  - q_n état final
- $C = \{c_0, c_1\}$ deux compteurs
- $\Delta = \{\delta_0, \ldots, \delta_{n-1}\}$ un ensemble de règles
  - Type I :
    $c_i := c_i + 1$; aller en q_k
  - Type II :
    si $c_i \neq 0$ alors $\{c_i := c_i - 1$; aller en $q_k\}$ sinon $\{$aller en $q_h\}$

Machines à compteurs 2

Définition (Configurations d'une machine à deux compteurs)

Une configuration J est un triplet (q, j_0, j_1) avec

- q un état
- $j_0, j_1 \geq 0$ les valeurs des compteurs

Définition (Exécutions d'une machine)

Une exécution de M pour $J^0 = (q_0, j_0, j_1)$

$$J^0, J^1, \dots J^i, \dots$$

s'obtient en appliquant les règles de transition.

L'exécution est finie si q_n est atteint (M termine pour J^0), infinie sinon

Machines à compteurs 3

Problèmes indécidables

Problème (VL : Vacuité du langage)

Existe-t-il une configuration initiale J^0 telle que l'exécution de M est finie ?

Problème (EB : Exécution bornée)

Les valeurs des compteurs de M restent-elles bornées lors de l'exécution de M pour $J^0 = (q_0, 0, 0)$?

Indécidabilité : idée générale

Idée générale

- place = compteur $\Rightarrow$ réseau de compteurs R
- énoncé logique de VL ou EB

Principe

Donnée : une machine M

Construire : une formule Φ_M telle que

- $R \models \Phi_M \Leftrightarrow$

Indécidabilité : idée générale

Idée générale

- place = compteur $\Rightarrow$ réseau de compteurs R
- énoncé logique de VL ou EB

Principe

Donnée : une machine M

Construire : une formule Φ_M telle que

- $R \models \Phi_M \Leftrightarrow$
  - $\exists\ J^0$ telle que M termine (Mu-Calcul)

Indécidabilité : idée générale

Idée générale

- place = compteur $\Rightarrow$ réseau de compteurs R
- énoncé logique de VL ou EB

Principe

Donnée : une machine M

Construire : une formule Φ_M telle que

- $R \models \Phi_M \Leftrightarrow$
  - $\exists\ J^0$ telle que M termine (Mu-Calcul)
  - l'exécution de M pour J^0 est bornée (Nu-Calcul Conjonctif)

Indécidabilité : idée générale

Idée générale

- place = compteur $\Rightarrow$ réseau de compteurs R
- énoncé logique de VL ou EB

Principe

Donnée : une machine M

Construire : une formule Φ_M telle que

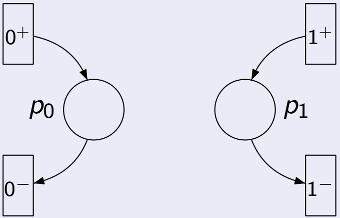
- $R \models \Phi_M \Leftrightarrow$
  - $\exists\ J^0$ telle que M termine (Mu-Calcul)
  - l'exécution de M pour J^0 est bornée (Nu-Calcul Conjonctif)
- $\exists N$ réseau avec $N \models \Phi_M \Leftrightarrow R \models \Phi_M$

Indécidabilité : réseaux de compteurs

Alphabet : $\Sigma = \{0^+, 0^-, 1^+, 1^-\}$

Réseau de compteurs

Pour le Mu-Calcul



Indécidabilité : Mu-Calcul

- Formule Θ_i de test à 0 du compteur c_i :
  satisfaite uniquement si la place p_i est vide

Indécidabilité : Mu-Calcul

- Formule Θ_i de test à 0 du compteur c_i :
  satisfaite uniquement si la place p_i est vide
- Une formule Φ_l par état q_l de M en fonction de la règle δ_l.
  Aller en q_l = satisfaire Φ_l où

  - Type I :

$$\begin{array}{lccc} \delta_l = & \text{“}c_i := c_i + 1 & ; & \text{aller en } q_k\text{”} \\ \Rightarrow \Phi_l = & \text{“ tirer } i^+ & ; & \text{satisfaire } \Phi_k\text{”} \end{array}$$

Indécidabilité : Mu-Calcul

- Formule Θ_i de test à 0 du compteur c_i :
  satisfaite uniquement si la place p_i est vide
- Une formule Φ_l par état q_l de M en fonction de la règle δ_l.
  Aller en q_l = satisfaire Φ_l où

  - Type I :
    $$\begin{array}{lccc} \delta_l = & \text{“}c_i := c_i + 1 & ; & \text{aller en } q_k\text{”} \\ \Rightarrow \Phi_l = & \text{“ tirer } i^+ & ; & \text{satisfaire } \Phi_k\text{”} \end{array}$$

  - Type II :
    $$\begin{array}{ll} \delta_l = & \text{“si } c_i \neq 0 \text{ alors } \{c_i := c_i - 1\,;\ \text{aller en } q_k\} \\ & \qquad\qquad \text{sinon } \{\text{aller en } q_h\}\text{”} \\ \Rightarrow \Phi_l = & \text{“(non } \Theta_i \text{ et (tirer } i-\,;\ \text{satisfaire } \Phi_k)) \\ & \qquad\qquad \text{ou (} \Theta_i\,;\ \text{satisfaire } \Phi_h)\text{”} \end{array}$$

Indécidabilité : Mu-Calcul

- Formule Θ_i de test à 0 du compteur c_i :
  satisfaite uniquement si la place p_i est vide
- Une formule Φ_l par état q_l de M en fonction de la règle δ_l.
  Aller en q_l = satisfaire Φ_l où

  - Type I :
    $\delta_l =$ "$c_i := c_i + 1$; aller en q_k"
    $\Rightarrow \Phi_l =$ " tirer i^+ ; satisfaire Φ_k"
  - Type II :
    $\delta_l =$ "si $c_i \neq 0$ alors $\{c_i := c_i - 1\,;$ aller en $q_k\}$
    sinon $\{$aller en $q_h\}$"
    $\Rightarrow \Phi_l =$ "(non Θ_i et (tirer $i-$; satisfaire Φ_k))
    ou (Θ_i ; satisfaire Φ_h)"
  - $\Phi_n =$ "*vrai*"

Indécidabilité : Mu-Calcul

- Formule Θ_i de test à 0 du compteur c_i :
  satisfaite uniquement si la place p_i est vide
- Une formule Φ_l par état q_l de M en fonction de la règle δ_l.
  Aller en q_l = satisfaire Φ_l où
  - Type I :
    $\delta_l =$ "$c_i := c_i + 1$; aller en q_k"
    $\Rightarrow \Phi_l =$ " tirer i^+ ; satisfaire Φ_k"
  - Type II :
    $\delta_l =$ "si $c_i \neq 0$ alors $\{c_i := c_i - 1$; aller en $q_k\}$
    sinon $\{$aller en $q_h\}$"
    $\Rightarrow \Phi_l =$ "(non Θ_i et (tirer $i-$; satisfaire Φ_k))
    ou (Θ_i ; satisfaire Φ_h)"
  - $\Phi_n =$ "*vrai*"
- Poser $\Phi_M = \Phi_0$

Indécidabilité : Mu-Calcul 2

Théorème

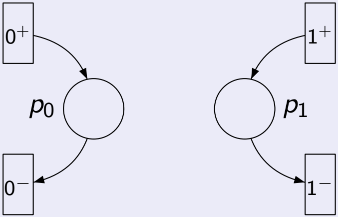
Le problème de synthèse de réseaux de Petri est indécidable pour le Mu-Calcul.

Indécidabilité : réseaux de compteurs bornés

Alphabet : $\Sigma = \{0^+, 0^-, 1^+, 1^-\}$

Réseau de compteurs

Pour le Mu-Calcul

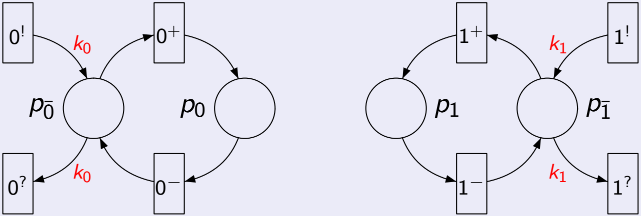


Indécidabilité : réseaux de compteurs bornés

Alphabet : $\Sigma = \{0^+, 0^-, 1^+, 1^-\}$

Réseau de compteurs

Pour le Nu-Calcul Conjonctif, une famille de réseaux de compteurs bornés



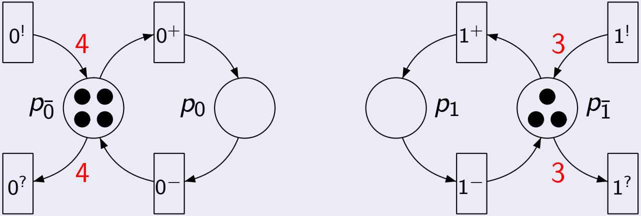
Marquage initial m_0 : $m_0(p_0) = 0$, $m_0(p_1) = 1$, $m_0(p_{\bar{0}}) = k_0$, $m_0(p_1) = k_1$

Indécidabilité : réseaux de compteurs bornés

Alphabet : $\Sigma = \{0^+, 0^-, 1^+, 1^-\}$

Réseau de compteurs

Pour le Nu-Calcul Conjonctif, une famille de réseaux de compteurs bornés



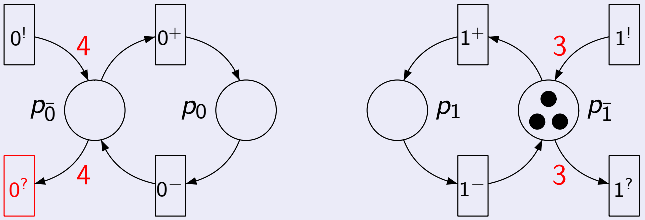
Exemple : $k_0 = 4$, $k_1 = 3$

Indécidabilité : réseaux de compteurs bornés

Alphabet : $\Sigma = \{0^+, 0^-, 1^+, 1^-\}$

Réseau de compteurs

Pour le Nu-Calcul Conjonctif, une famille de réseaux de compteurs bornés



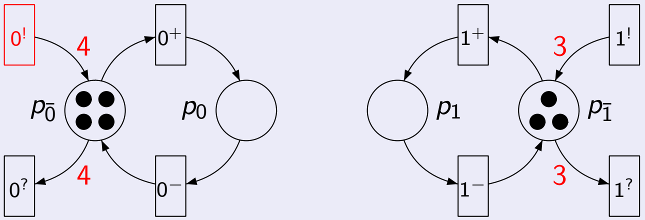
Exemple : $k_0 = 4$, $k_1 = 3$

Indécidabilité : réseaux de compteurs bornés

Alphabet : $\Sigma = \{0^+, 0^-, 1^+, 1^-\}$

Réseau de compteurs

Pour le Nu-Calcul Conjonctif, une famille de réseaux de compteurs bornés



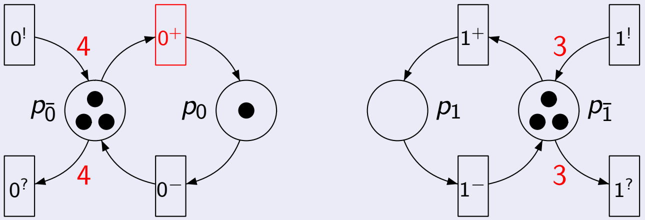
Exemple : $k_0 = 4$, $k_1 = 3$

Indécidabilité : réseaux de compteurs bornés

Alphabet : $\Sigma = \{0^+, 0^-, 1^+, 1^-\}$

Réseau de compteurs

Pour le Nu-Calcul Conjonctif, une famille de réseaux de compteurs bornés



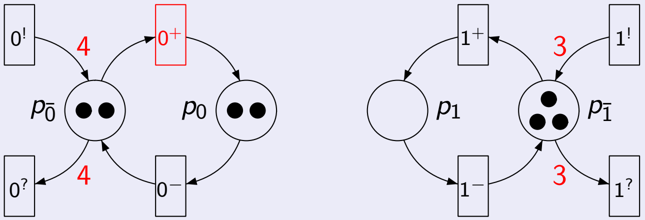
Exemple : $k_0 = 4$, $k_1 = 3$

Indécidabilité : réseaux de compteurs bornés

Alphabet : $\Sigma = \{0^+, 0^-, 1^+, 1^-\}$

Réseau de compteurs

Pour le Nu-Calcul Conjonctif, une famille de réseaux de compteurs bornés



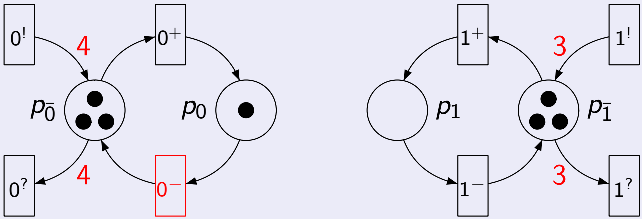
Exemple : $k_0 = 4$, $k_1 = 3$

Indécidabilité : réseaux de compteurs bornés

Alphabet : $\Sigma = \{0^+, 0^-, 1^+, 1^-\}$

Réseau de compteurs

Pour le Nu-Calcul Conjonctif, une famille de réseaux de compteurs bornés



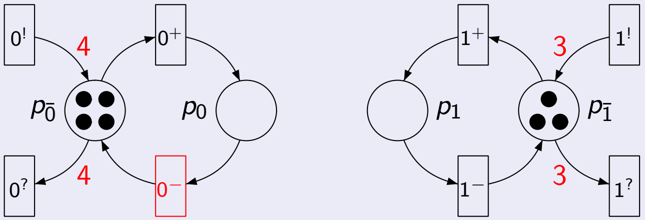
Exemple : $k_0 = 4$, $k_1 = 3$

Indécidabilité : réseaux de compteurs bornés

Alphabet : $\Sigma = \{0^+, 0^-, 1^+, 1^-\}$

Réseau de compteurs

Pour le Nu-Calcul Conjonctif, une famille de réseaux de compteurs bornés



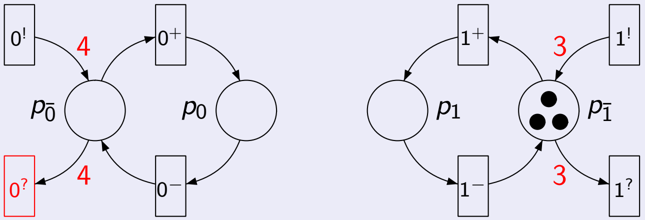
Exemple : $k_0 = 4$, $k_1 = 3$

Indécidabilité : réseaux de compteurs bornés

Alphabet : $\Sigma = \{0^+, 0^-, 1^+, 1^-\}$

Réseau de compteurs

Pour le Nu-Calcul Conjonctif, une famille de réseaux de compteurs bornés



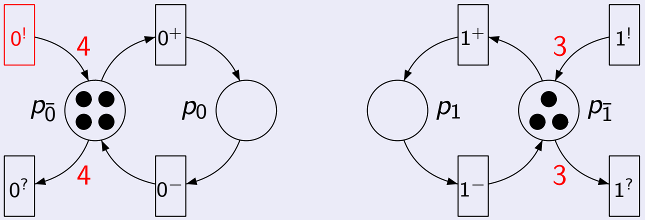
Exemple : $k_0 = 4$, $k_1 = 3$

Indécidabilité : réseaux de compteurs bornés

Alphabet : $\Sigma = \{0^+, 0^-, 1^+, 1^-\}$

Réseau de compteurs

Pour le Nu-Calcul Conjonctif, une famille de réseaux de compteurs bornés



Exemple : $k_0 = 4$, $k_1 = 3$

Indécidabilité : Nu-Calcul Conjonctif

$$\color{red}{\Phi_M} = \Phi_c \text{ et } \Phi_{m_0} \text{ et } \Psi_M \text{ et } \Phi_I$$

Avec :

Indécidabilité : Nu-Calcul Conjonctif

$$\Phi_M = \Phi_c \text{ et } \Phi_{m_0} \text{ et } \Psi_M \text{ et } \Phi_I$$

Avec :

- Φ_c : théorie de la famille de réseaux de compteurs bornés

Indécidabilité : Nu-Calcul Conjonctif

$$\Phi_M = \Phi_c \text{ et } \Phi_{m_0} \text{ et } \Psi_M \text{ et } \Phi_I$$

Avec :

- Φ_c : théorie de la famille de réseaux de compteurs bornés
- Φ_{m_0} : formule qui énonce le marquage initial

Indécidabilité : Nu-Calcul Conjonctif

$$\Phi_M = \Phi_c \text{ et } \Phi_{m_0} \text{ et } \Psi_M \text{ et } \Phi_I$$

Avec :

- Φ_c : théorie de la famille de réseaux de compteurs bornés
- Φ_{m_0} : formule qui énonce le marquage initial
- Ψ_M : formule qui énonce que l'exécution de M est bornée

Indécidabilité : Nu-Calcul Conjonctif

$$\Phi_M = \Phi_c \text{ et } \Phi_{m_0} \text{ et } \Psi_M \text{ et } \Phi_I$$

Avec :

- Φ_c : théorie de la famille de réseaux de compteurs bornés
- Φ_{m_0} : formule qui énonce le marquage initial
- Ψ_M : formule qui énonce que l'exécution de M est bornée
- Φ_I : formule qui interdit un ensemble de comportements

Indécidabilité : Nu-Calcul Conjonctif

Construction de Ψ_M = une spécification modale

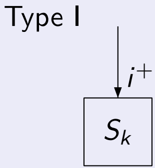
Une spécification modale S_l par état q_l de M en fonction de δ_l

Indécidabilité : Nu-Calcul Conjonctif

Construction de Ψ_M = une spécification modale

Une spécification modale S_l par état q_l de M en fonction de δ_l

- Type I :
  "$c_i := c_i + 1$, aller en q_k"



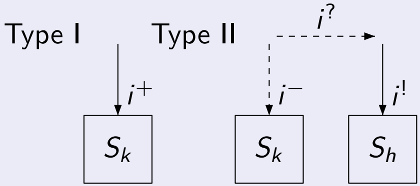
Indécidabilité : Nu-Calcul Conjonctif

Construction de Ψ_M = une spécification modale

Une spécification modale S_l par état q_l de M en fonction de δ_l

- Type I :
  "$c_i := c_i + 1$, aller en q_k"

- Type II :
  "si $c \neq 0$, alors c := c - 1 ; aller en q_k sinon aller en q_h"



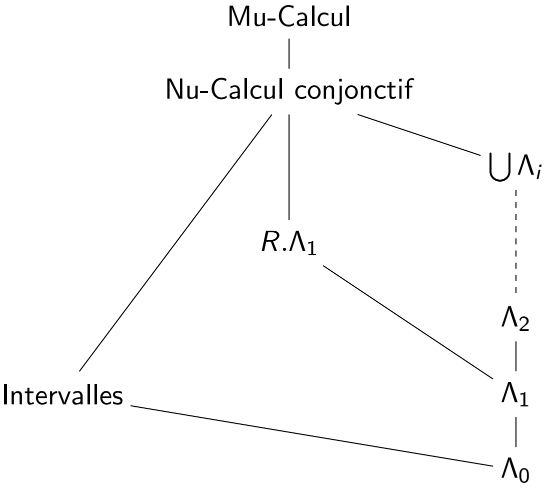
Indécidabilité : Mu-Calcul 2

Théorème

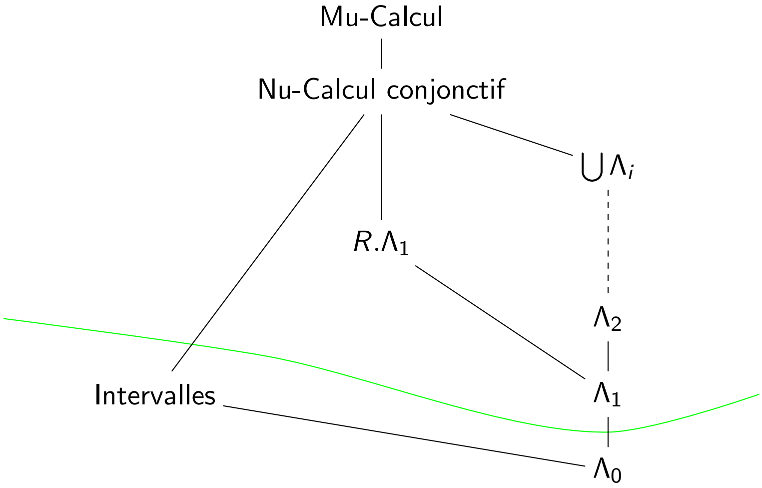
Le problème de synthèse de réseaux de Petri purs est indécidable pour le Nu-Calcul-Conjonctif.

Plan

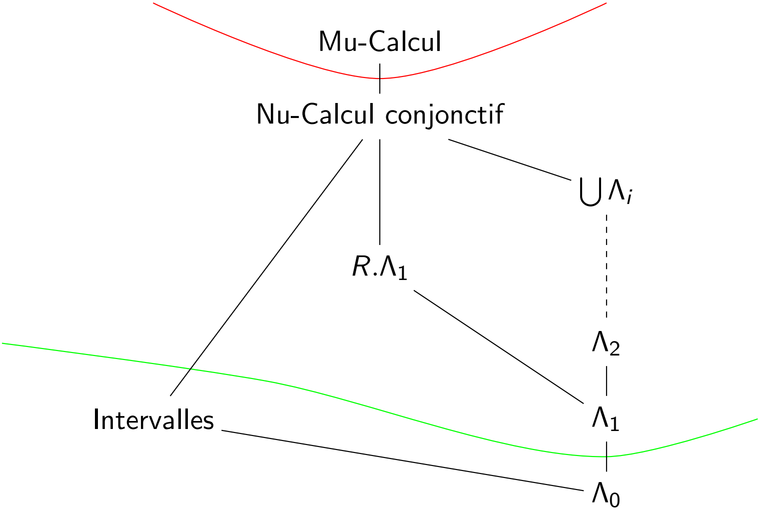
Conclusion


Mu-Calcul
Nu-Calcul conjonctif
$\bigcup \Lambda_i$
$R.\Lambda_1$
Λ_2
Intervalles
Λ_1
Λ_0

Conclusion

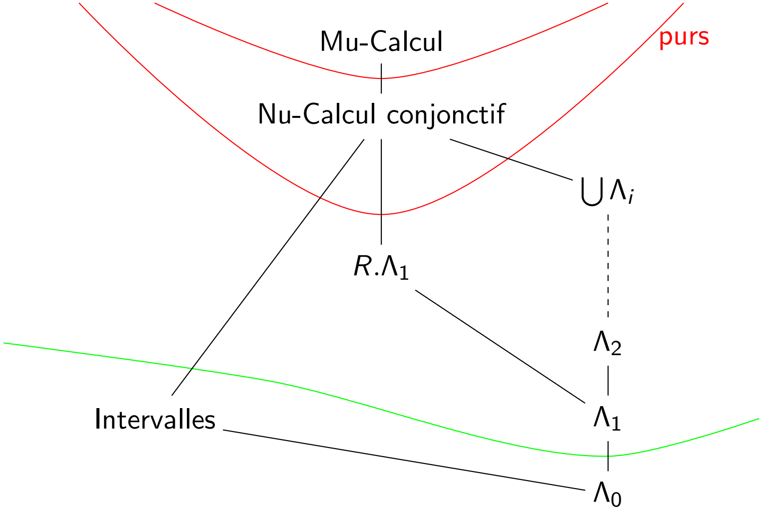

Mu-Calcul
Nu-Calcul conjonctif
$\bigcup \Lambda_i$
$R.\Lambda_1$
Λ_2
Intervalles
Λ_1
Λ_0

Conclusion



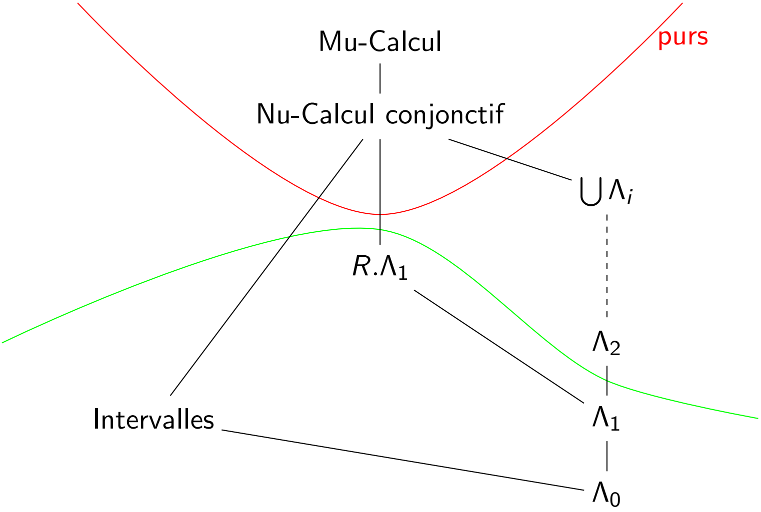
Mu-Calcul
Nu-Calcul conjonctif
$\bigcup \Lambda_i$
$R.\Lambda_1$
Λ_2
Intervalles
Λ_1
Λ_0

Conclusion



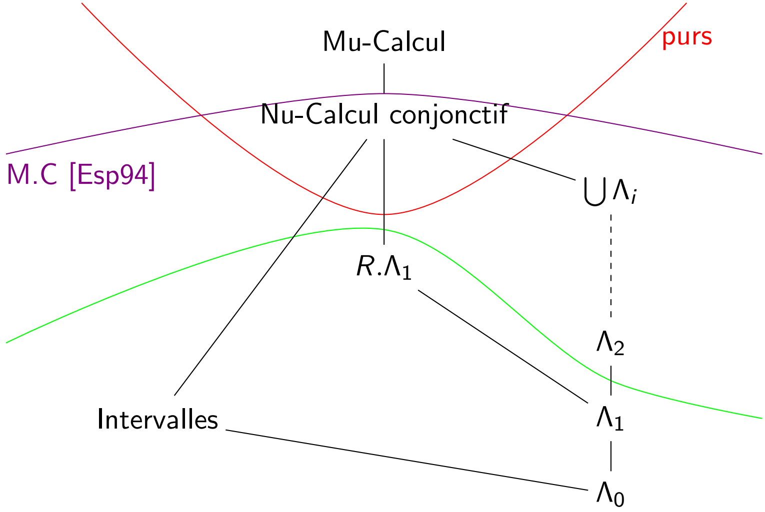
purs
Mu-Calcul
Nu-Calcul conjonctif
$\bigcup \Lambda_i$
$R.\Lambda_1$
Λ_2
Intervalles
Λ_1
Λ_0

Conclusion



Mu-Calcul
Nu-Calcul conjonctif
purs
$\bigcup \Lambda_i$
$R.\Lambda_1$
Λ_2
Intervalles
Λ_1
Λ_0

Conclusion



purs
Mu-Calcul
Nu-Calcul conjonctif
M.C [Esp94]
$\bigcup \Lambda_i$
$R.\Lambda_1$
Λ_2
Intervalles
Λ_1
Λ_0

Perspectives

- Réseaux bornés
- Application au contrôle
- Synthèse conjointe d'un réseau et d'un automate

C'est terminé

Merci de votre attention.