

Spécification et synthèse de réseaux de fonctions séquentielles

David Janin

LaBRI, Bordeaux, novembre 2002

Spécification et synthèse de réseaux de fonctions séquentielles

David Janin

LaBRI, Bordeaux, novembre 2002

Plan

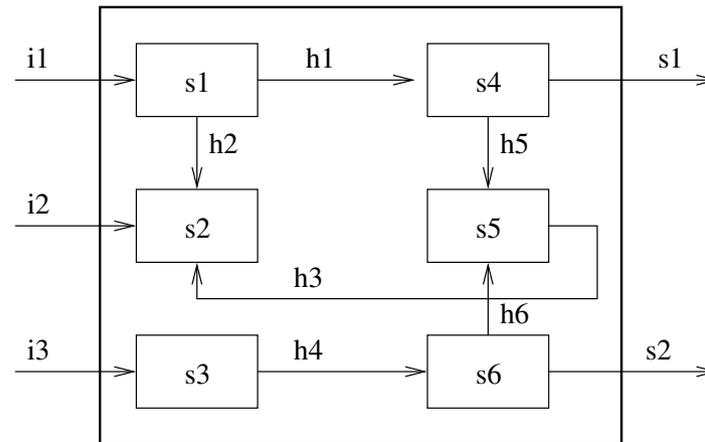
- Architectures, spécifications et synthèse,
- Résultats connus,
- Fonctions séquentielles et arbres infinis,
- Quelques opérateurs de combinaison,
- Synthèse distribuée (parallèle ou séquentielle),
- Preuve du cas séquentielle,
- Preuve revisitée,

Le problème de la synthèse distribuée

Données : (1) Une architecture composée de plusieurs modules interconnectés et (2) une spécification globale décrivant les liens souhaités entre entrées, sorties, signaux internes.

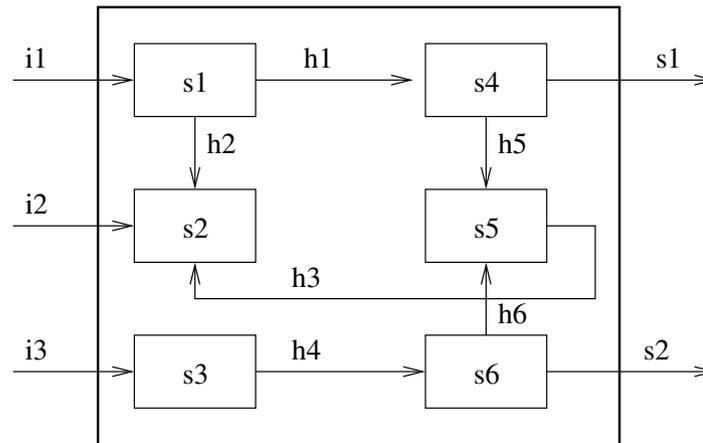
Le problème de la synthèse distribuée

Données : (1) Une architecture composée de plusieurs modules interconnectés et (2) une spécification globale décrivant les liens souhaités entre entrées, sorties, signaux internes.



Le problème de la synthèse distribuée

Données : (1) Une architecture composée de plusieurs modules interconnectés et (2) une spécification globale décrivant les liens souhaités entre entrées, sorties, signaux internes.



Problème : synthétiser, s'ils en existent, des programmes locaux, un par module, qui, en interagissant ensemble, satisfont la spécification globale.

Déclinaisons du problème

Calculabilité et complexité du problème de la synthèse selon :

Déclinaisons du problème

Calculabilité et complexité du problème de la synthèse selon :

- Le modèle de programme (synchrone vs asynchrone) et l'architecture ?

Déclinaisons du problème

Calculabilité et complexité du problème de la synthèse selon :

- Le modèle de programme (synchrone vs asynchrone) et l'architecture ?
- Le langage de spécifications :
 - FO/LTL,
 - mu-calcul.

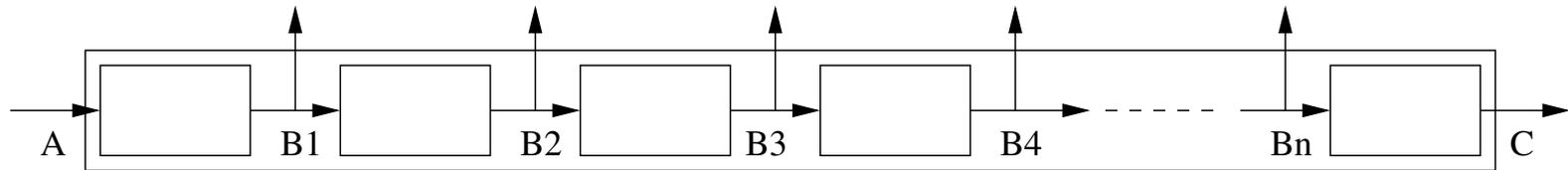
Déclinaisons du problème

Calculabilité et complexité du problème de la synthèse selon :

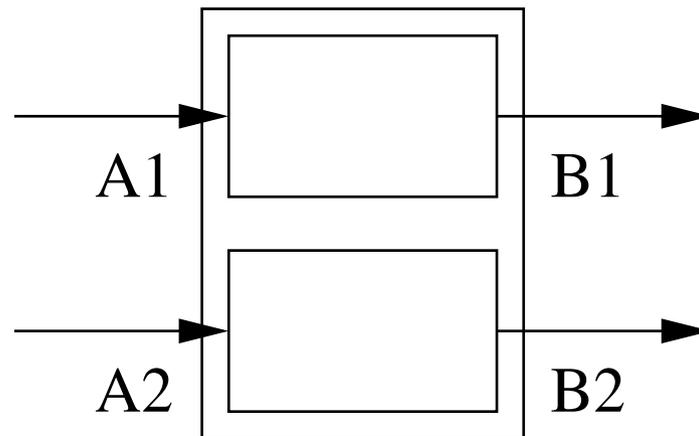
- Le modèle de programme (synchrone vs asynchrone) et l'architecture ?
- Le langage de spécifications :
 - FO/LTL,
 - mu-calcul.
- Les contraintes supplémentaires entre spécifications et architecture (e.g. spécifications complètes de certains modules) ?

Quelques résultats

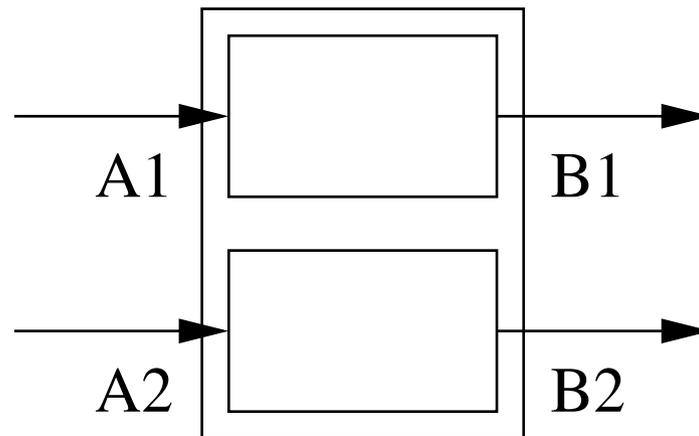
Théorème (PR90, KV00, Mad00) Le problème de la synthèse à partir de spécifications régulières (MSOL ou mu-calcul) sur les architectures «pipelines» est non élémentairement décidable.



Théorème (PR90) Le problème de la synthèse à partir de spécifications LTL sur l'architecture parallèle déconnectée est indécidable.



Théorème (PR90) Le problème de la synthèse à partir de spécifications LTL sur l'architecture parallèle déconnectée est indécidable.

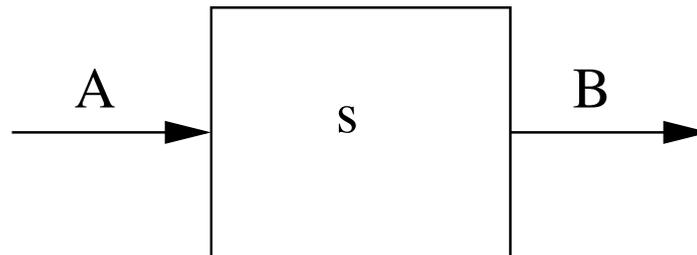


Même avec des conditions très simples (contraintes locales + accessibilité ou sûreté)

Les fonctions séquentielles

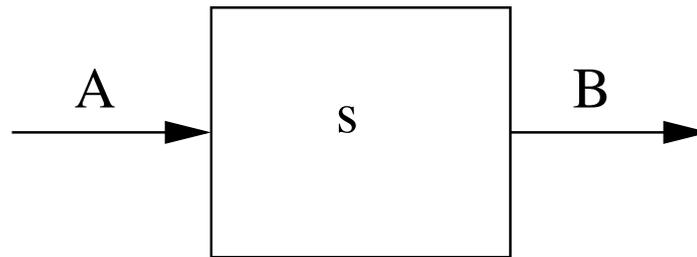
Fonctions séquentielles

Toute fonction $s : A^* \rightarrow B^*$ qui produit **séquentiellement** ses sorties en fonctions des entrées.



Fonctions séquentielles

Toute fonction $s : A^* \rightarrow B^*$ qui produit **séquentiellement** ses sorties en fonctions des entrées.

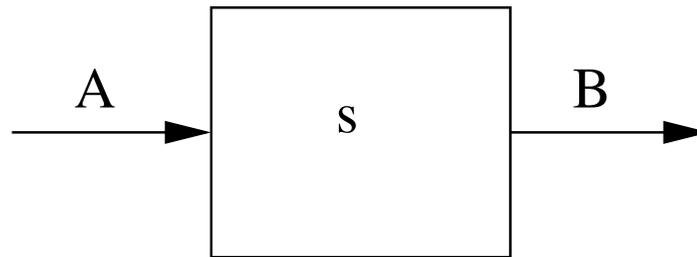


i.e. toute fonction $s : A^* \rightarrow B^*$ telle qu'il existe $f : A^+ \rightarrow B$ telle que

- $s(\epsilon) = \epsilon$,
- $s(u.a) = f(u).s(u.a)$.

Fonctions séquentielles

Toute fonction $s : A^* \rightarrow B^*$ qui produit **séquentiellement** ses sorties en fonctions des entrées.



i.e. toute fonction $s : A^* \rightarrow B^*$ telle qu'il existe $f : A^+ \rightarrow B$ telle que

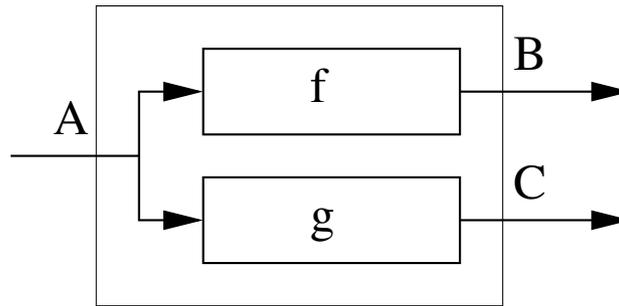
- $s(\epsilon) = \epsilon$,
- $s(u.a) = f(u).s(u.a)$.

On dit aussi que s , aussi notée s_f , est **engendrée** par f .

Compositions de fonctions séquentielles

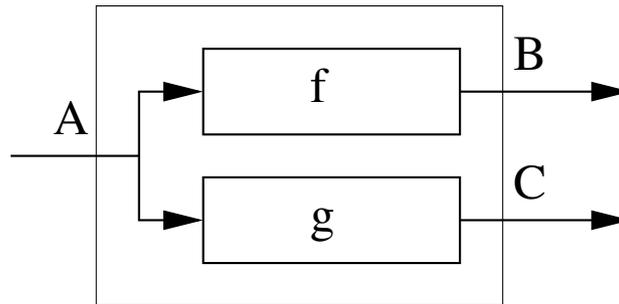
Compositions de fonctions séquentielles

- Composition parallèle:

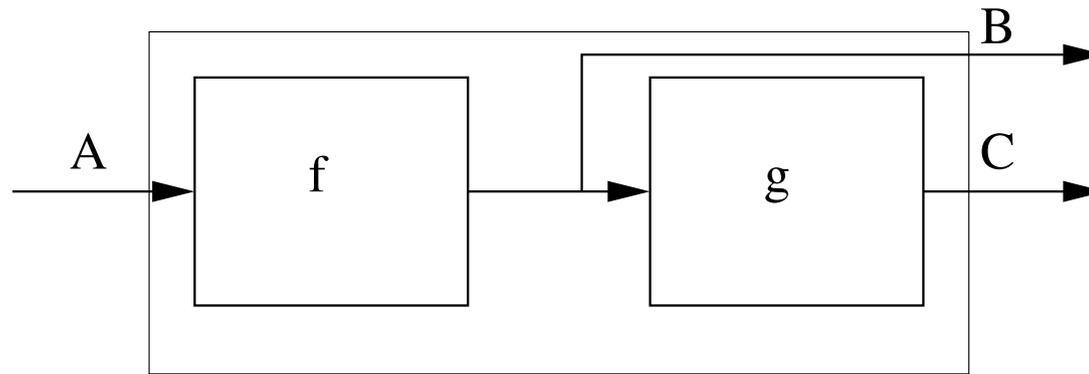


Compositions de fonctions séquentielles

- Composition parallèle:

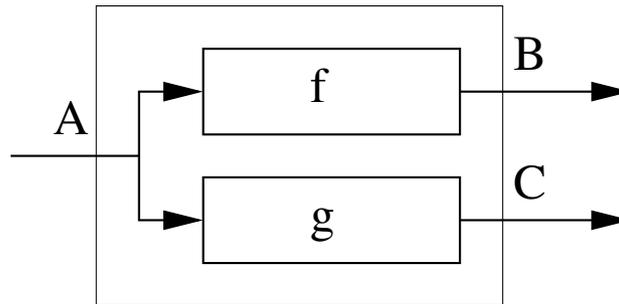


- Composition séquentielle:

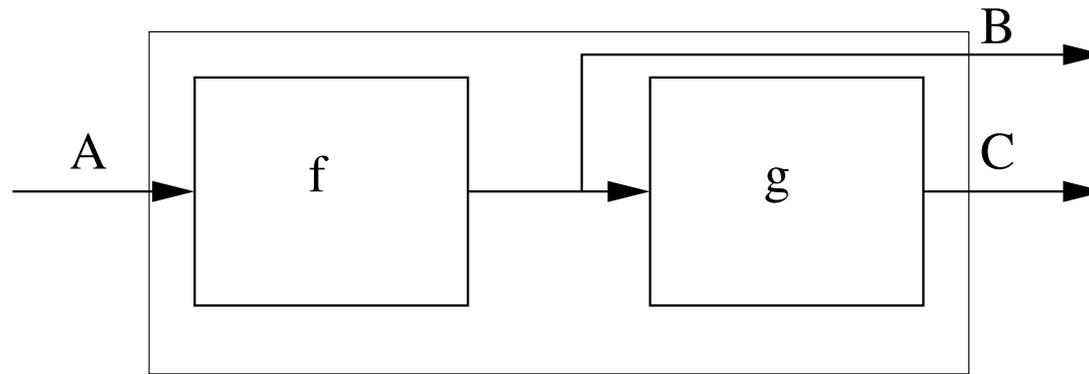


Compositions de fonctions séquentielles

- Composition parallèle:



- Composition séquentielle:

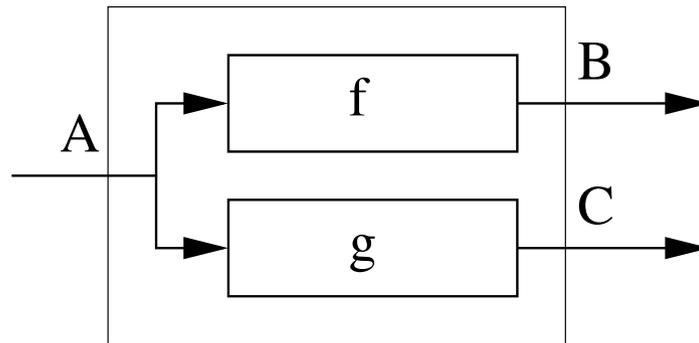


- Autres opérateurs : délais, effacement, retro-action...

Composition parallèle

Avec $s_f : A^* \rightarrow B_1^*$ et $s_g : A^* \rightarrow B_2^*$, on définit

$$\begin{aligned} s_f || s_g : A^* &\rightarrow (B_1 \times B_2)^* \\ u &\mapsto \text{colle}(s_f(u), s_g(u)) \end{aligned}$$



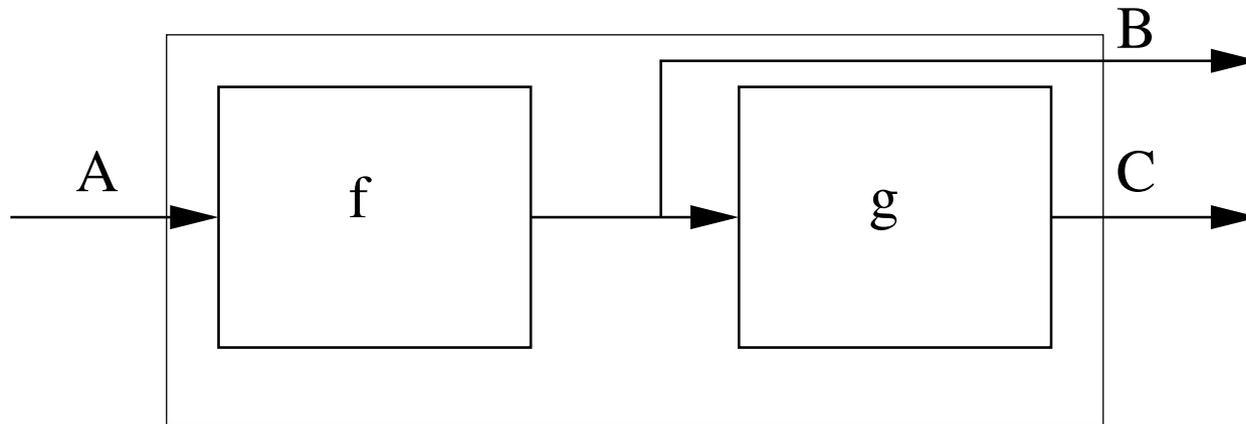
C'est la fonction engendrée par

$$\begin{aligned} f || g : A^+ &\rightarrow (B_1 \times B_2) \\ u &\mapsto (f(u), g(u)) \end{aligned}$$

Composition séquentielle

Avec $s_f : A^* \rightarrow B^*$ et $s_g : B^* \rightarrow C^*$, on définit

$$s_f \otimes s_g : A^* \rightarrow (B \times C)^*$$
$$u \mapsto \mathbf{colle}(s_f(u), s_g \circ s_f(u))$$



C'est la fonction engendrée par

$$f \otimes g : A^+ \rightarrow (B \times C)$$
$$u \mapsto (f(u), g \circ s_f(u))$$

Autres compositions

Autres compositions

- Délais: pour $s_f : A^* \rightarrow B^*$,

$$\text{delay}_b(s_f)(u.a) = b.s_f(u)$$

pour tout $a \in A, b \in B, u \in A^*$.

Autres compositions

- Délais: pour $s_f : A^* \rightarrow B^*$,

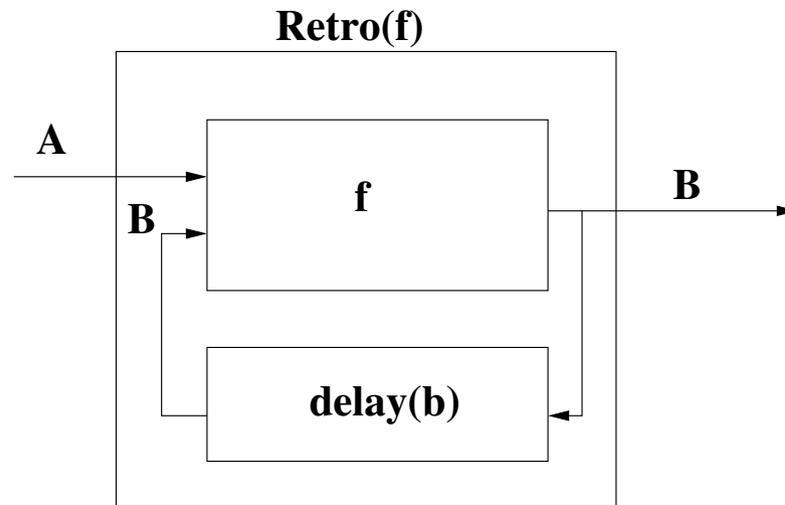
$$\text{delay}_b(s_f)(u.a) = b.s_f(u)$$

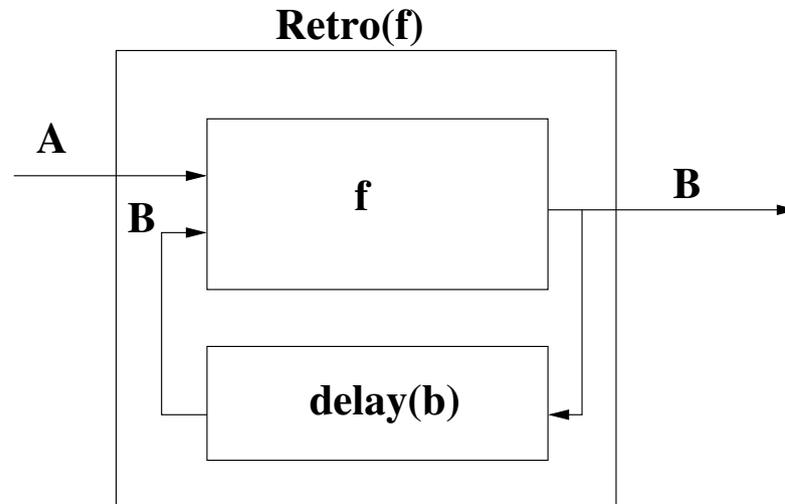
pour tout $a \in A$, $b \in B$, $u \in A^*$.

- Effacement (desynchronisation): pour $s_f : A^* \rightarrow B^*$,

$$\text{delete}(s_f, a)(u) = s_f(\pi_{A-\{a\}}(u))$$

pour tout $a \in A$, $u \in A^*$. Attention, dans ce dernier cas, on sort du cadre des fonctions séquentielles...





- **Rétroactions (récursion):** pour $s_f : (A \times B)^* \rightarrow B^*$,

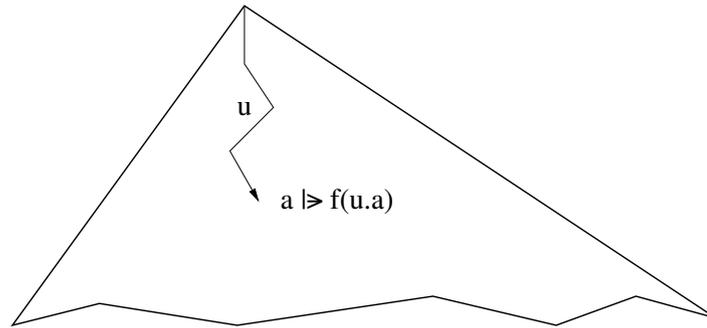
$$\text{retro}_b(s_f)(u.a) = s_f(\text{colle}(u.a, b.\text{retro}_b(s_f)(u)))$$

pour tout $a \in A, b \in B, u \in A^*$.

Spécification et synthèse de fonctions séquentielles

Des fonctions aux arbres

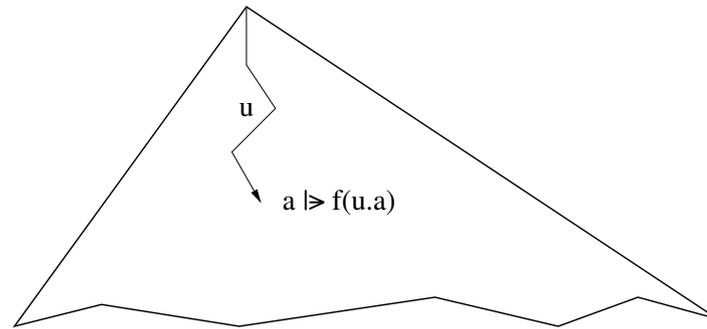
Tout générateur $f : A^+ \rightarrow B$ est un arbre t_f , A -aire, étiqueté sur l'alphabet $A \rightarrow B$,



i.e. $t_f : A^* \rightarrow (A \rightarrow B)$ définit, pour tout $u \in A^*$, tout $a \in A$, par $(t_f(u))(a) = f(u.a)$.

Des fonctions aux arbres

Tout générateur $f : A^+ \rightarrow B$ est un arbre t_f , A -aire, étiqueté sur l'alphabet $A \rightarrow B$,



i.e. $t_f : A^* \rightarrow (A \rightarrow B)$ définit, pour tout $u \in A^*$, tout $a \in A$, par $(t_f(u))(a) = f(u.a)$.

⇒ Retour dans un domaine connu: les langages d'arbres

Le cas non distribué (connu)

Théorème (Rabin et al.) Décider du vide est, selon le formalisme de spécification choisie :

- dans $NP \cap co-NP$ avec les automates non déterministes à parité,
- NP-complet avec les automates non déterministes à cond. de Rabin,
- EXPTIME-complet avec les automates alternants,
- non élémentaire avec la logique MSO.

Le cas distribué (moins connu)

Trouver des fonction séquentielles f_1, f_2, \dots, f_n telles que

$$C(f_1, \dots, f_n) \models \varphi$$

avec :

- une spécification d'architecture C construite à partir d'opérateurs élémentaires de combinaison de fonctions séquentielles,
- une spécification globale φ de la fonction séquentielle induite par cette combinaison.

Plus généralement, à partir d'une spécification globale φ et de spécifications supplémentaires $\varphi_1, \varphi_2, \dots, \varphi_n$, vérifier que

$$C(\varphi_1, \varphi_2, \dots, \varphi_n) \wedge \varphi$$

est satisfiable, où

$$f \models C(\varphi_1, \varphi_2, \dots, \varphi_n)$$

lorsqu'il existe $f_1 \models \varphi_1, f_2 \models \varphi_2, \dots, f_n \models \varphi_n$, telle que $f = C(f_1, \dots, f_n)$.

Le cas parallèle

La composition $\varphi_1 || \varphi_2$ est exprimable en MSO !

Le cas parallèle

La composition $\varphi_1 || \varphi_2$ est exprimable en MSO !

Proposition Pour tout langage régulier $L_1 \subseteq A^+ \rightarrow B_1$ et $L_2 \subseteq A^+ \rightarrow B_2$ le langage $L_1 || L_2 = \{f || g : f \in L_1, g \in L_2\}$ est régulier.

Le cas parallèle

La composition $\varphi_1 || \varphi_2$ est exprimable en MSO !

Proposition Pour tout langage régulier $L_1 \subseteq A^+ \rightarrow B_1$ et $L_2 \subseteq A^+ \rightarrow B_2$ le langage $L_1 || L_2 = \{f || g : f \in L_1, g \in L_2\}$ est régulier.

Preuve. Pour tout $f : A^+ \rightarrow B_1 \times B_2$, les deux projections $\pi_1(f) : A^+ \rightarrow B_1$ et $\pi_2(f) : A^+ \rightarrow B_2$ de f sont définissables (sur f) en MSO (par transduction MSO). Ainsi, le langage

$$L_1 || L_2 = \{f \in A^+ \rightarrow B_1 \times B_2 : \pi_1(f) \in L_1 \wedge \pi_2(f) \in L_2\}$$

est lui aussi définissable.

Le cas séquentiel

La composition $\varphi_1 \otimes \varphi_2$ n'est pas exprimable en MSO !

Le cas séquentiel

La composition $\varphi_1 \otimes \varphi_2$ n'est pas exprimable en MSO !

Proposition. Le langage $(A^+ \rightarrow B) \otimes (B^+ \rightarrow C)$ des fonctions séquentiellement décomposables n'est pas reconnaissable.

Mais,

Mais,

Théorème Pour tout $L \subseteq A^+ \rightarrow B \times C$, $L_1 \subseteq A^+ \rightarrow B$, réguliers, le langage «quotient»

$$L / \otimes L_1 = \{g \in B^+ \rightarrow C : \exists f \in L_1, f \otimes g \in L\}$$

est régulier.

Mais,

Théorème Pour tout $L \subseteq A^+ \rightarrow B \times C$, $L_1 \subseteq A^+ \rightarrow B$, réguliers, le langage «quotient»

$$L/\otimes L_1 = \{g \in B^+ \rightarrow C : \exists f \in L_1, f \otimes g \in L\}$$

est régulier.

Corollaire. Pour tout langage régulier additionnel

$$L_2 \subseteq B^+ \rightarrow C,$$

$$(L_1 \otimes L_2) \cap L \neq \emptyset \text{ iff } (L/\otimes L_1) \cap L_2 \neq \emptyset$$

Idée de preuve

Idée de preuve

- Etant donné $L \subseteq A^+ \rightarrow B \times C$, posons $\tilde{L} = L / \otimes A^+ \rightarrow B$, i.e.

$$\tilde{L} = \{g : A^+ \rightarrow C \mid \exists f : A^+ \rightarrow B, f \otimes g \in L\}$$

Idée de preuve

- Etant donné $L \subseteq A^+ \rightarrow B \times C$, posons $\tilde{L} = L / \otimes A^+ \rightarrow B$, i.e.

$$\tilde{L} = \{g : A^+ \rightarrow C \mid \exists f : A^+ \rightarrow B, f \otimes g \in L\}$$

- Avec $L' = L \cap (L_1 \parallel A^+ \rightarrow C)$, on a $L / \otimes L_1 = \tilde{L}'$.

Idée de preuve

- Etant donné $L \subseteq A^+ \rightarrow B \times C$, posons $\tilde{L} = L / \otimes A^+ \rightarrow B$, i.e.

$$\tilde{L} = \{g : A^+ \rightarrow C \mid \exists f : A^+ \rightarrow B, f \otimes g \in L\}$$

- Avec $L' = L \cap (L_1 \parallel A^+ \rightarrow C)$, on a $L / \otimes L_1 = \tilde{L}'$.
- Il suffit donc de démontrer :

Lemme. Pour tout langage régulier $L \subseteq A^+ \rightarrow B \times C$, le langage \tilde{L} est régulier.

Idée de preuve (suite)

Idée de preuve (suite)

- Soit $L \subseteq A^* \rightarrow B \times C$ reconnu par un **automate non déterministe** à parité

$$\mathcal{A} = \langle A, A \rightarrow B \times C, Q, Q_0, \delta, \Omega \rangle$$

avec $\delta : Q \times (A \rightarrow B \times C) \rightarrow \mathcal{P}(A \rightarrow Q)$, i.e. pour tout $q \in Q$, tout $F : A \rightarrow B \times C$,

$$\delta(q, F) \simeq \bigvee_{T \in \delta(q, F)} \bigwedge_{a \in A} \diamond_a T(a)$$

- On définit l'**automate alternant**

$$\tilde{A} = \langle B, B \rightarrow C, \tilde{Q}, \tilde{Q}_0, \tilde{\delta}, \tilde{\Omega} \rangle$$

avec

- On définit l'**automate alternant**

$$\tilde{A} = \langle B, B \rightarrow C, \tilde{Q}, \tilde{Q}_0, \tilde{\delta}, \tilde{\Omega} \rangle$$

avec

- $\tilde{Q} = Q \times (A \rightarrow B),$

- On définit l'**automate alternant**

$$\tilde{A} = \langle B, B \rightarrow C, \tilde{Q}, \tilde{Q}_0, \tilde{\delta}, \tilde{\Omega} \rangle$$

avec

- $\tilde{Q} = Q \times (A \rightarrow B),$
- $\tilde{Q}_0 = Q_0 \times (A \rightarrow B),$

- On définit l'**automate alternant**

$$\tilde{A} = \langle B, B \rightarrow C, \tilde{Q}, \tilde{Q}_0, \tilde{\delta}, \tilde{\Omega} \rangle$$

avec

- $\tilde{Q} = Q \times (A \rightarrow B),$
- $\tilde{Q}_0 = Q_0 \times (A \rightarrow B),$
- $\tilde{\Omega} = \Omega \circ \pi_1,$

- On définit l'**automate alternant**

$$\tilde{A} = \langle B, B \rightarrow C, \tilde{Q}, \tilde{Q}_0, \tilde{\delta}, \tilde{\Omega} \rangle$$

avec

- $\tilde{Q} = Q \times (A \rightarrow B),$
- $\tilde{Q}_0 = Q_0 \times (A \rightarrow B),$
- $\tilde{\Omega} = \Omega \circ \pi_1,$
- $\tilde{d} : \tilde{Q} \times (B \rightarrow C) \rightarrow \mathcal{P}(B \rightarrow \mathcal{P}(\tilde{Q}))$

- On définit l'**automate alternant**

$$\tilde{A} = \langle B, B \rightarrow C, \tilde{Q}, \tilde{Q}_0, \tilde{\delta}, \tilde{\Omega} \rangle$$

avec

- $\tilde{Q} = Q \times (A \rightarrow B),$
- $\tilde{Q}_0 = Q_0 \times (A \rightarrow B),$
- $\tilde{\Omega} = \Omega \circ \pi_1,$
- $\tilde{d} : \tilde{Q} \times (B \rightarrow C) \rightarrow \mathcal{P}(B \rightarrow \mathcal{P}(\tilde{Q}))$ avec, pour tout $(q, F) \in \tilde{Q}$, tout $G \in B \rightarrow C$, $\tilde{\delta}((q, F), G)$ équivalente à

$$\bigvee_{\substack{T \in \delta(q, F \times G \circ F) \\ F' \in A \rightarrow B}} \bigwedge_{b \in B} \diamond_b \bigwedge (T(F^{-1}(b)) \times \{F'\})$$

- et on vérifie (facilement) que:
Proposition. $L(\tilde{\mathcal{A}}) = \tilde{L}$.

- et on vérifie (facilement) que:

Proposition. $L(\tilde{\mathcal{A}}) = \tilde{L}$.

- Attention: L'automate $\tilde{\mathcal{A}}$ est un automate alternant qu'il faut «non déterminer» pour réappliquer la procédure. Et la «non détermination» est exponentielle !

- et on vérifie (facilement) que:

Proposition. $L(\tilde{\mathcal{A}}) = \tilde{L}$.

- Attention: L'automate $\tilde{\mathcal{A}}$ est un automate alternant qu'il faut «non déterminer» pour réappliquer la procédure. Et la «non détermination» est exponentielle !

Corollaire (KV00) Le problème de la synthèse sur architecture «pipeline» à partir de spéc. régulières est **non élémentairement** décidable.

Peut-on faire mieux ?

Peut-on faire mieux ?

Proposition (?) Le problème de la résolution du produit séquentielle n'est pas symétrique: à partir de $L \subseteq A^+ \rightarrow B \times C$ et $L_2 \subseteq B^+ \rightarrow C$ réguliers, le langage

$$L /_d \otimes L_2 = \{f : A^+ \rightarrow B : \exists g : B^+ \rightarrow C, f \otimes g \in L\}$$

n'est pas, en général, régulier.

Peut-on faire mieux ?

Proposition (?) Le problème de la résolution du produit séquentielle n'est pas symétrique: à partir de $L \subseteq A^+ \rightarrow B \times C$ et $L_2 \subseteq B^+ \rightarrow C$ réguliers, le langage

$$L /_d \otimes L_2 = \{f : A^+ \rightarrow B : \exists g : B^+ \rightarrow C, f \otimes g \in L\}$$

n'est pas, en général, régulier.

Et, le problème du pipeline est intrinséquement non élémentaire.

Et maintenant ?

Perspectives

Perspectives

- Expliciter le cas du délais

Perspectives

- Expliciter le cas du délais (*facile*)
- Etudier le cas de la retro-action

Perspectives

- Expliciter le cas du délais (*facile*)
- Etudier le cas de la retro-action (*classique*)
- Etudier le cas de l'effacement

Perspectives

- Expliciter le cas du délais (*facile*)
- Etudier le cas de la retro-action (*classique*)
- Etudier le cas de l'effacement (*un espoir*)
- Voir comment tous celà se combine... et relier à l'approche « théorie du contrôle »