Projet VALMEM

Démonstration du prototype IMITATOR

Étienne André, Emmanuelle Encrenaz, Laurent Fribourg

Laboratoire Spécification et Vérification

Plan

1. L'algorithme
2. Démonstration sur l'exemple du latch
3. Application à SPSMALL
4. Travaux futurs

Plan

1 L'algorithme

2 Démonstration sur l'exemple du latch

3 Application à SPSMALL

4 Travaux futurs

Our Method

- Input
  - A PTA $\mathcal{A}$ with initial state s_{init}
  - An instantiation π of all the parameters of $\mathcal{A}$
    - ⋆ Exemplifying a good behaviour
- Output : generalisation
  - A constraint K on the parameters such that
    - ⋆ $\pi \models K$
    - ⋆ For all instantiation $\pi' \models K$, the set of traces under π' is the same as the set of traces under π

The Algorithm InverseMethod

Input $\mathcal{A}$: PTA
π_0 : Valuation of P
Output K_0 : Constraint on the parameters
Variables i : Current iteration
S : Current set of symbolic states ($S = Post^i_{\mathcal{A}(K)}$)
S' : Former set of symbolic states
K : Current constraint on the parameters

$i := 0$; $K := True$; $S := \{s_0\}$; $S' := \{s_0\}$
DO
  DO UNTIL S is π_0-compatible
    Select a π_0-incompatible state (q, C) of S
    Select an inequality J of $(\exists X : C)$ such that $\pi_0 \models \neg J$
    $S' := S$; $K := K \wedge \neg J$; $S := Post^i_{\mathcal{A}(K)}$
  OD %% S π_0-compatible
  $S' := S$; $S := Post_{\mathcal{A}(K)}(S)$; $i := i + 1$
  IF $S = S'$ %% S π_0-compatible and $S = Post^*_{\mathcal{A}(K)}$
    THEN RETURN $K_0 := \bigcap_{(q,C)\in S}(\exists X : C)$
  FI
OD

Le prototype IMITATOR

- IMITATOR
  - Inverse Method for Inferring Time AbstracT behaviOR
- Programme de 1500 lignes en Python
  - Appels à HyTech pour le calcul du *Post*
  - Appels à Prolog (non essentiels)
- En entrée : un fichier HyTech contenant une instantiation des paramètres
- En sortie : une contrainte sur les paramètres assurant le même comportement
- Temps de mise au point : environ 2 « Étienne-mois »

Avancement

1 L'algorithme

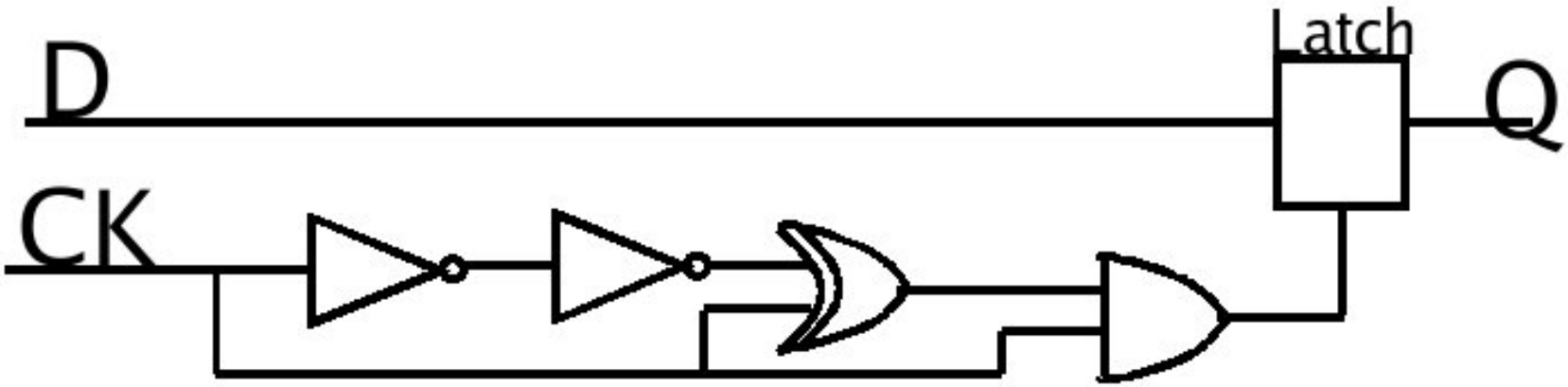
2 Démonstration sur l'exemple du latch

3 Application à SPSMALL

4 Travaux futurs

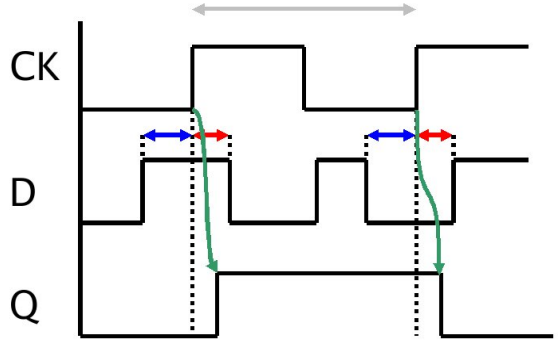
Rappel du circuit latch

- Cinq éléments :
  - Deux portes « non »
  - Une porte « ou exclusif »
  - Une porte « et »
  - Une bascule à niveau (« latch »)

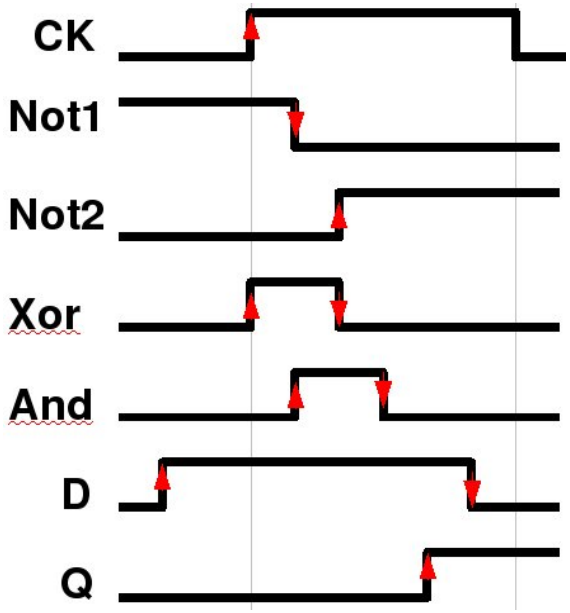


Signaux

- Signaux fournis par Rémy

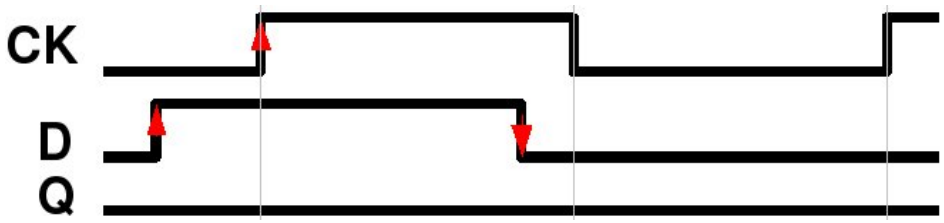


- Signaux aux différentes portes



Modélisation

- Modélisation des portes
  - Seules les transitions marquées en rouge sont modélisées
  - Ex : l'automate de la porte « not 1 » ne fait que produire un front descendant
- Définition d'un mauvais état
  - Un cycle d'horloge complet a eu lieu
  - Q n'a pas effectué de front montant



- Utilisation de paramètres
  - Exemple : durée de front montant du latch : $\delta_{Latch^{\uparrow}}$

Génération d'une contrainte

- Valeurs des paramètres (π_0) générées par Patricia

$$T_{HI} = 1000 \quad T_{LO} = 1000 \quad T_{Hold} = 350 \quad T_{Setup} = 0$$
$$\delta_{Not1\uparrow} = 219 \quad \delta_{Not1\downarrow} = 147 \quad \delta_{Not2\uparrow} = 155 \quad \delta_{Not2\downarrow} = 163$$
$$\delta_{Xor\uparrow} = 147 \quad \delta_{Xor\downarrow} = 416 \quad \delta_{And\uparrow} = 80 \quad \delta_{And\downarrow} = 155$$
$$\delta_{Latch\uparrow} = 240$$

- Appel à IMITATOR... (en live !)

Génération d'une contrainte

- Valeurs des paramètres (π_0) générées par Patricia

$$T_{HI} = 1000 \quad T_{LO} = 1000 \quad T_{Hold} = 350 \quad T_{Setup} = 0$$
$$\delta_{Not1\uparrow} = 219 \quad \delta_{Not1\downarrow} = 147 \quad \delta_{Not2\uparrow} = 155 \quad \delta_{Not2\downarrow} = 163$$
$$\delta_{Xor\uparrow} = 147 \quad \delta_{Xor\downarrow} = 416 \quad \delta_{And\uparrow} = 80 \quad \delta_{And\downarrow} = 155$$
$$\delta_{Latch\uparrow} = 240$$

- Appel à IMITATOR... (en live !)
- Contrainte K_0 (obtenue après simplification par HyTech) :

$$\begin{aligned}
& 0 < \delta_{And\downarrow} \\
\wedge\ & \delta_{Xor\uparrow} = \delta_{Not1\downarrow} \\
\wedge\ & \delta_{And\uparrow} + \delta_{Latch\uparrow} < T_{Hold} \\
\wedge\ & \delta_{Not1\downarrow} + \delta_{Not2\uparrow} < \delta_{And\uparrow} + \delta_{Latch\uparrow} \\
\wedge\ & T_{Setup} < T_{LO} \\
\wedge\ & T_{Hold} < \delta_{Not1\downarrow} + \delta_{Not2\uparrow} + \delta_{Xor\downarrow} \\
\wedge\ & \delta_{And\uparrow} < \delta_{Not1\downarrow} \\
\wedge\ & \delta_{Not1\downarrow} + \delta_{Not2\uparrow} + \delta_{Xor\downarrow} + \delta_{And\downarrow} \leq T_{HI}
\end{aligned}$$

Interprétation des contraintes obtenues

- Interprétation de $\delta_{And^{\uparrow}} + \delta_{Latch^{\uparrow}} < T_{Hold}$
  - Le temps de maintien de D doit être supérieur à la somme des temps maxima du front montant du « and » et de franchissement du latch

Interprétation des contraintes obtenues

- Interprétation de $\delta_{And^{\uparrow}} + \delta_{Latch^{\uparrow}} < T_{Hold}$
  - Le temps de maintien de D doit être supérieur à la somme des temps maxima du front montant du « and » et de franchissement du latch

- Minimisation du T_{Hold}
  - Après instanciation de K_0 pour tous les paramètres sauf T_{Hold}

$$320 < T_{Hold} < 718$$

Avancement

1 L'algorithme

2 Démonstration sur l'exemple du latch

3 Application à SPSMALL

4 Travaux futurs

Application à une portion de SPSMALL

- Portion de SPSMALL
- Modèle généré automatiquement depuis la description des transistors
- Automates temporisés paramétrés et leurs valeurs temporelles fournis d'après simulation par Remy

Application à une portion de SPSMALL

- Portion de SPSMALL
- Modèle généré automatiquement depuis la description des transistors
- Automates temporisés paramétrés et leurs valeurs temporelles fournis d'après simulation par Remy
- Application de IMITATOR (après semi-instanciation)
  - Valeurs initiales des setup

$$T_{setupwen} = 48$$
$$T_{setupd} = 108$$

Application à une portion de SPSMALL

- Portion de SPSMALL
- Modèle généré automatiquement depuis la description des transistors
- Automates temporisés paramétrés et leurs valeurs temporelles fournis d'après simulation par Remy
- Application de IMITATOR (après semi-instanciation)
  - Valeurs initiales des setup

$$T_{setupwen} = 48$$
$$T_{setupd} = 108$$

  - Contraintes obtenues

$$46 < T_{setupwen} < 54$$
$$\wedge \quad 99 < T_{setupd} < 110$$
$$\wedge \quad T_{setupd} < T_{setupwen} + 61$$

Avancement

1 L'algorithme

2 Démonstration sur l'exemple du latch

3 Application à SPSMALL

4 Travaux futurs

Travaux futurs

- Méthode générant une contrainte imitant exactement le même comportement que les valeurs initiales
  - Contrainte trop forte pour notre cas : un circuit peut avoir d'autres bons fonctionnements
  - Idée : élargir la zone progressivement par application de la méthode, tant que le mauvais comportement n'est pas atteint
- Passage à l'échelle
  - L'usage d'HyTech empêche de considérer plus de 10 automates (peu !)
  - Temps bien trop élevé du simple fait d'une spécificité de HyTech (composition *a priori* des automates)
  - Idée : utilisation d'un autre outil, ou création d'un outil *ad hoc*
  - Objectif : contrainte sur la mémoire SPSMALL complète