

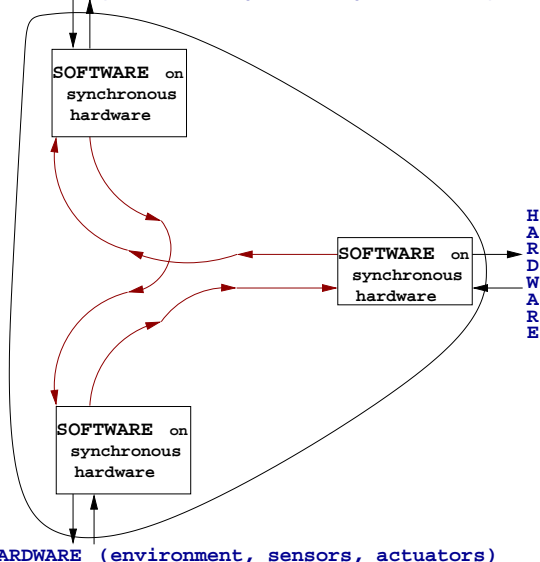
Certification de la résistance des réseaux de programmes synchrones aux erreurs matérielles

Julien Bertrane
bertrane@di.ens.fr

25 janvier 2007

Système type à analyser : systèmes embarqués

HARDWARE (environment, sensors, actuators)



HARDWARE (environment, sensors, actuators)

Objectifs

- Analyser des Systèmes composés de **plusieurs programmes synchrones communicants** (chacun avec sa propre horloge). Pourquoi ?
 - ▶ certains systèmes embarqués sont **trop grands** pour une unique horloge : la transmission de l'information serait **trop lente**
 - ▶ les systèmes critiques sont **redondants** pour résister à la panne d'une unité (\Rightarrow plusieurs horloges)

Objectifs

- Analyser des Systèmes composés de **plusieurs programmes synchrones communicants** (chacun avec sa propre horloge). Pourquoi ?
 - ▶ certains systèmes embarqués sont **trop grands** pour une unique horloge : la transmission de l'information serait **trop lente**
 - ▶ les systèmes critiques sont **redondants** pour résister à la panne d'une unité (\Rightarrow plusieurs horloges)
- la preuve doit être automatique, rapide, robuste aux changements mineurs dans le code. Basé sur la théorie de l'interprétation abstraite

Spécifications à vérifier

- Spécifications de **sécurité**
 - ▶ **Pour tout comportement s , à tout instant t , $s(t) \neq false$**

Spécifications à vérifier

- Spécifications de **sécurité**
 - ▶ **Pour tout comportement s , à tout instant t , $s(t) \neq false$**
- Spécifications **temporelles**
 - ▶ **Pour tout comportement s , à aucun instant t on n'a :**

pour tout $t' \in [t, t + \alpha], s(t') = true$

Spécifications à vérifier

- Spécifications de **sécurité**
 - ▶ **Pour tout comportement s , à tout instant t , $s(t) \neq false$**
- Spécifications **temporelles**
 - ▶ **Pour tout comportement s , à aucun instant t on n'a :**

pour tout $t' \in [t, t + \alpha]$, $s(t') = true$

- Spécifications **quantitatives**
 - ▶ les **sorties** de 2 systèmes redondants sont **égales au moins 50% du temps** durant chaque intervalle de temps de largeur minimale δ .

Programme synchrone

- ▶ Initialize(S)
 - ▶ while true do
 - ★ $(O, S) := \text{Compute}(S, I)$
 - ★ wait for clock
 - ▶ od

où I : entrées, S : variables d'état, O : sortie

Contraintes sur le système étudié

- Imperfections matérielles inévitables :
 - ▶ Les horloges des unités de commande se **désynchronisent**

Contraintes sur le système étudié

- Imperfections matérielles inévitables :
 - ▶ Les horloges des unités de commande se **désynchronisent**
 - ▶ Les communications ne sont **pas instantanées**

Contraintes sur le système étudié

- Imperfections matérielles inévitables :
 - ▶ Les horloges des unités de commande se **désynchronisent**
 - ▶ Les communications ne sont **pas instantanées**
 - ▶ Le temps de communication n'est **pas constant**

Hypothèses pour ce modèle

- **Quasi-synchronie** :
 - ▶ désynchronisation : la durée de chaque cycle (période entre deux **ticks**) est dans $[\alpha, \beta]$, $\alpha > 0$.
 - ▶ différent du quasi-synchronisme introduit par P. Caspi

Hypothèses pour ce modèle

- **Quasi-synchronie** :
 - ▶ désynchronisation : la durée de chaque cycle (période entre deux **ticks**) est dans $[\alpha, \beta]$, $\alpha > 0$.
 - ▶ différent du quasi-synchronisme introduit par P. Caspi
- Transmission en **série** entre 2 systèmes synchrones

Hypothèses pour ce modèle

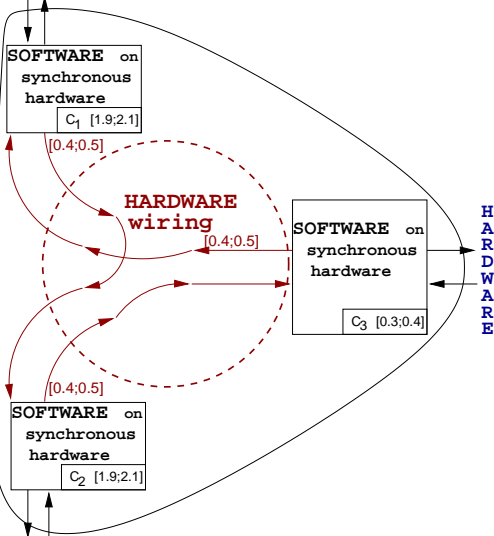
- **Quasi-synchronie** :
 - ▶ désynchronisation : la durée de chaque cycle (période entre deux **ticks**) est dans $[\alpha, \beta]$, $\alpha > 0$.
 - ▶ différent du quasi-synchronisme introduit par P. Caspi
- Transmission en **série** entre 2 systèmes synchrones
- **blackboard** à l'entrée de chaque système

Hypothèses pour ce modèle

- **Quasi-synchronie** :
 - ▶ désynchronisation : la durée de chaque cycle (période entre deux **ticks**) est dans $[\alpha, \beta]$, $\alpha > 0$.
 - ▶ différent du quasi-synchronisme introduit par P. Caspi
- Transmission en **série** entre 2 systèmes synchrones
- **blackboard** à l'entrée de chaque système
- **à l'initialisation** toutes les variables contiennent 0 ou *false*

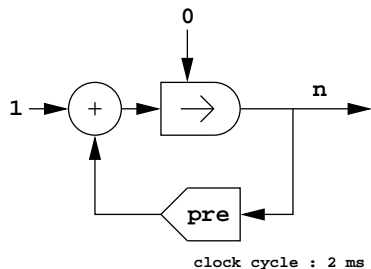
Système type à analyser : détails

HARDWARE (environment, sensors, actuators)



HARDWARE (environment, sensors, actuators)

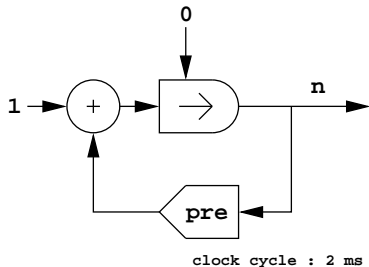
Quelles sémantiques ?



- $n = 0 \rightarrow (1 + pre\ n)$
- $a_t = pre(b_t) \Leftrightarrow \begin{array}{l} a_0 \text{ undefined} \\ a_{t+1} = b_t \end{array}$
- $a_t = b_t \rightarrow c_t \Leftrightarrow \begin{array}{l} a_0 = b_0 \\ a_{t+1} = c_{t+1} \end{array}$

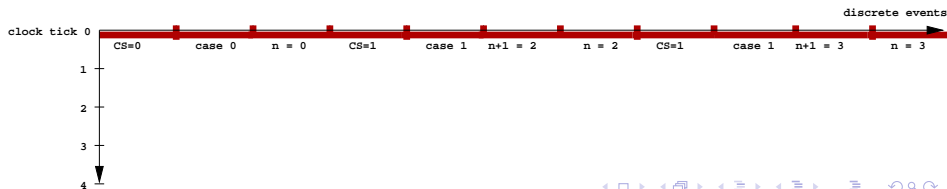
Quelles sémantiques ?

La sémantique classique d'un programme est + ou - celle du C

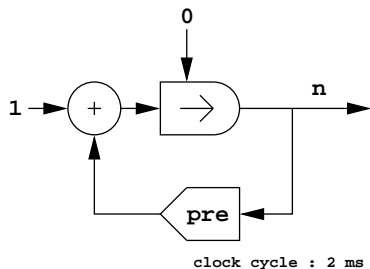


```
switch (global_state->current_state){
case 0 :
global_state->n= 0 ;
global_state->current_state = 1 ; break ;
break ;

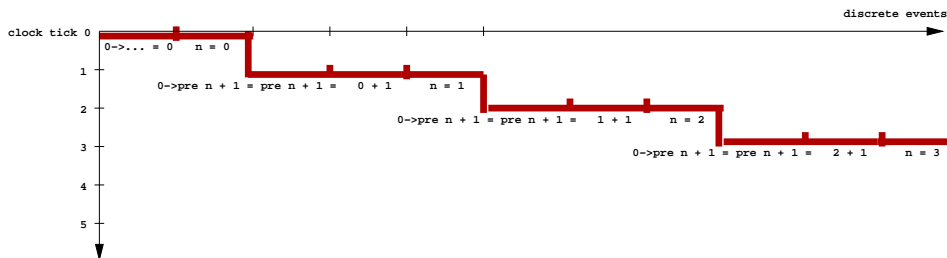
case 1 :
global_state->n= (global_state->n)+ 1 ;
global_state->current_state = 1 ; break ;
break ;}
```



quelles sémantiques ?

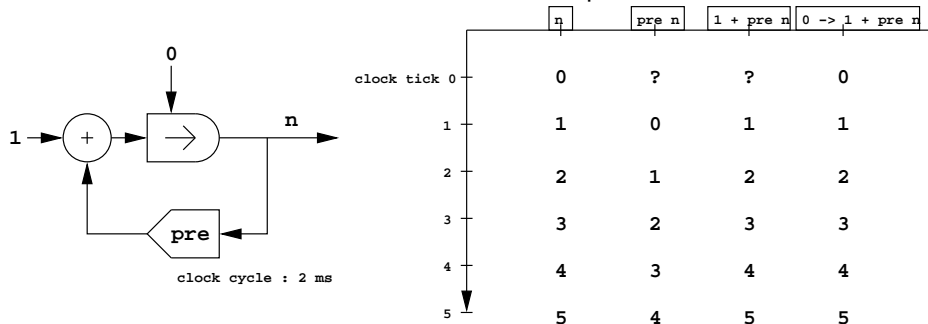


- Pour soft embarqué, il est important de respecter des timings
- en C, on a aucune garantie sur ces timings. Ils dépendent
 - ▶ de l'ordinateur exécutant le code
 - ▶ du compilateur, ...
- On peut définir des cycles d'horloge rythmant l'exécution



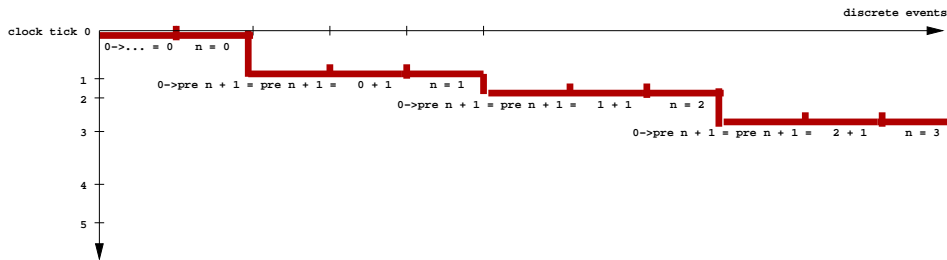
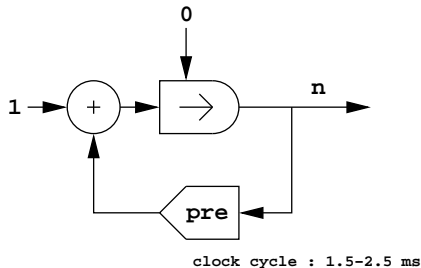
Avec quelles sémantiques doit-on travailler

On est donc conduit à définir une autre sémantique : celle du Lustre



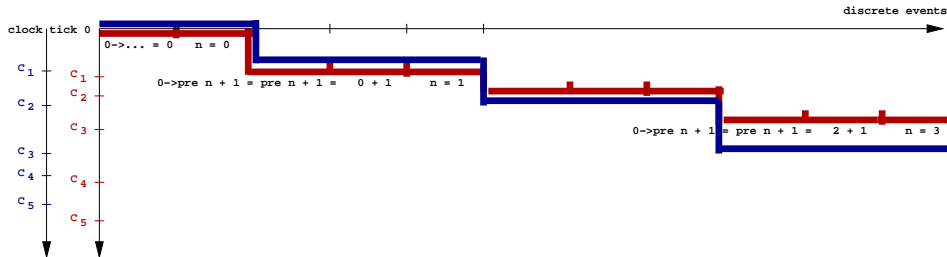
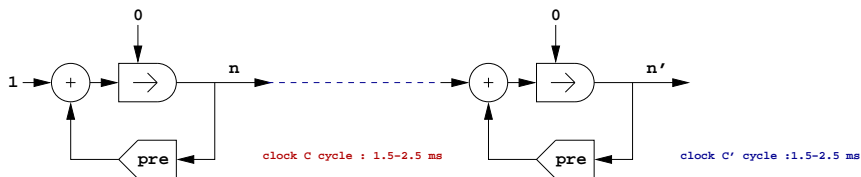
Avec quelles sémantiques doit-on travailler

- Et si l'horloge est imprécise ?



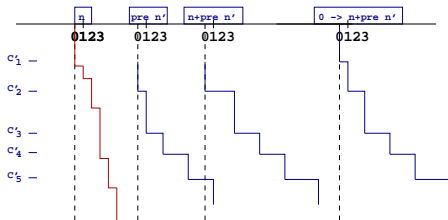
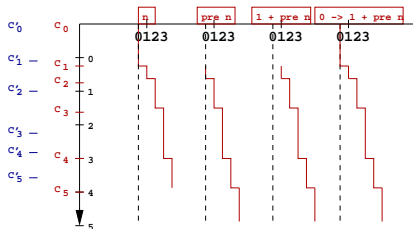
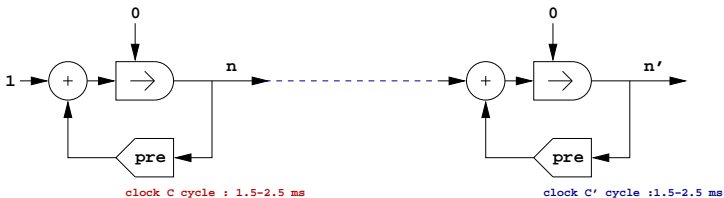
Avec quelles sémantiques doit-on travailler

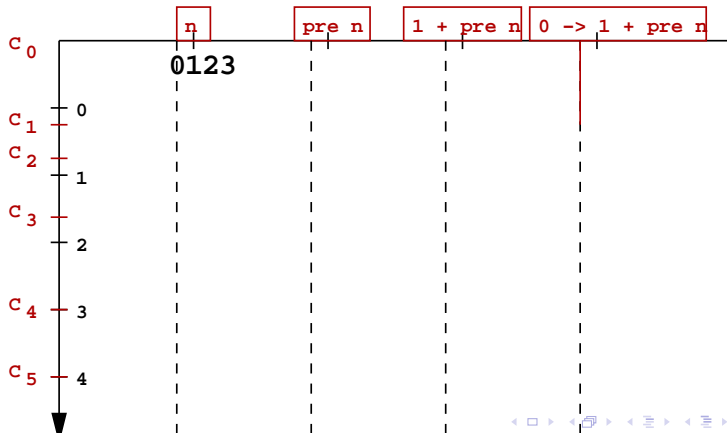
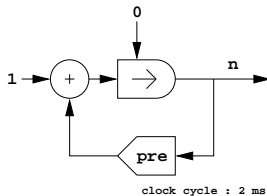
Plusieurs système, chacun ayant son horloge imprécise :

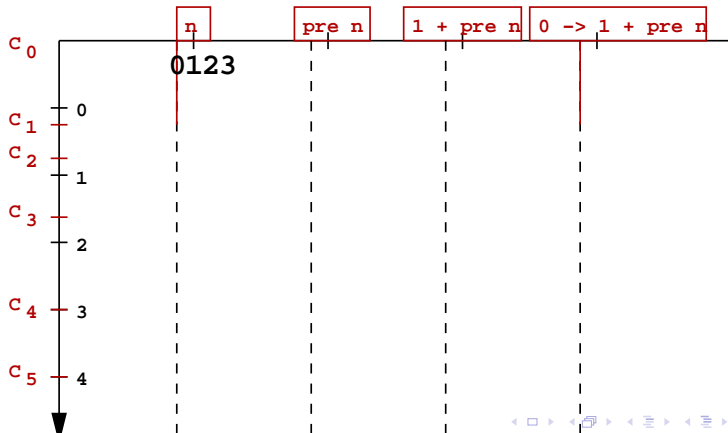
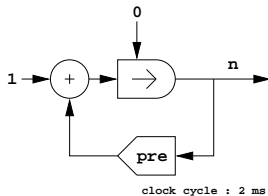


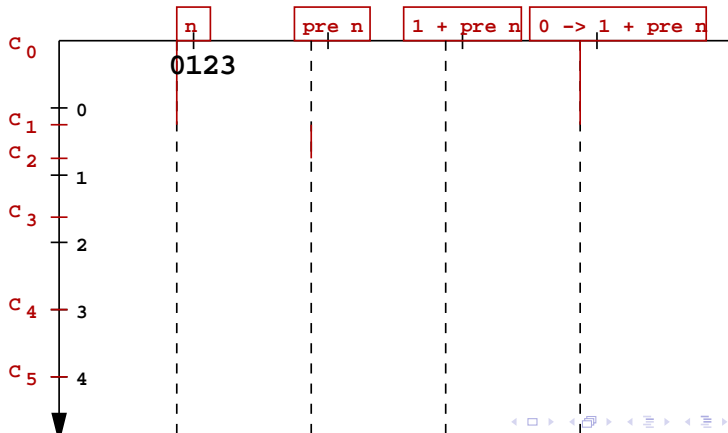
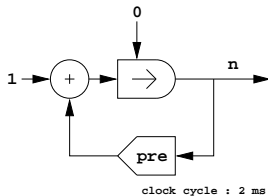
Avec quelles sémantiques doit-on travailler

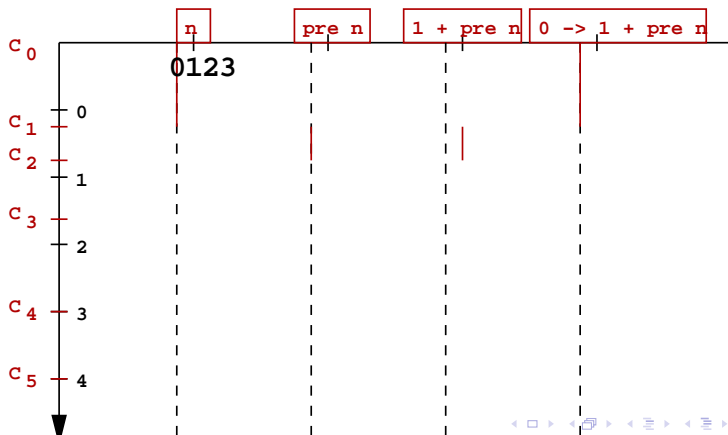
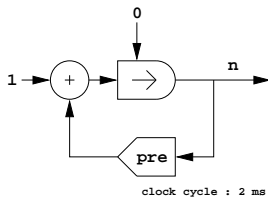
Une sémantique totalement continue permet d'étudier la communication de plusieurs systèmes

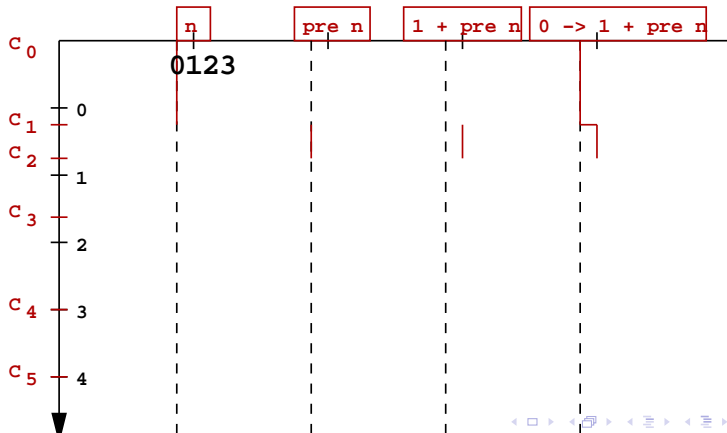
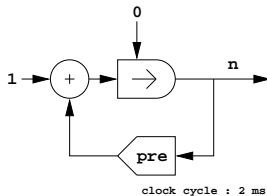


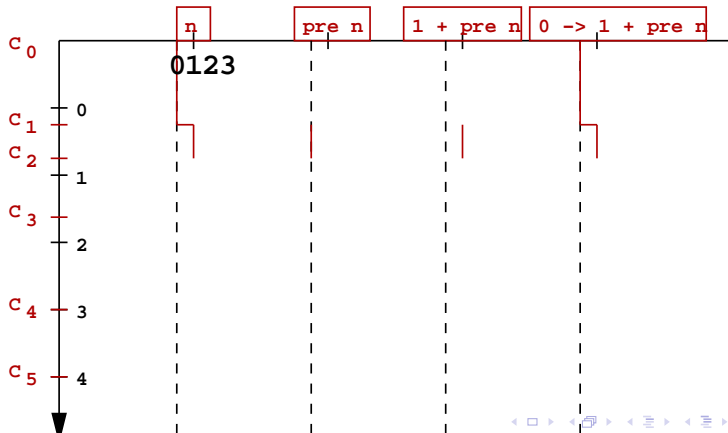
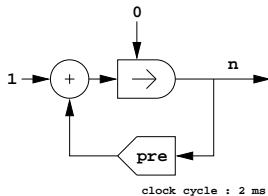


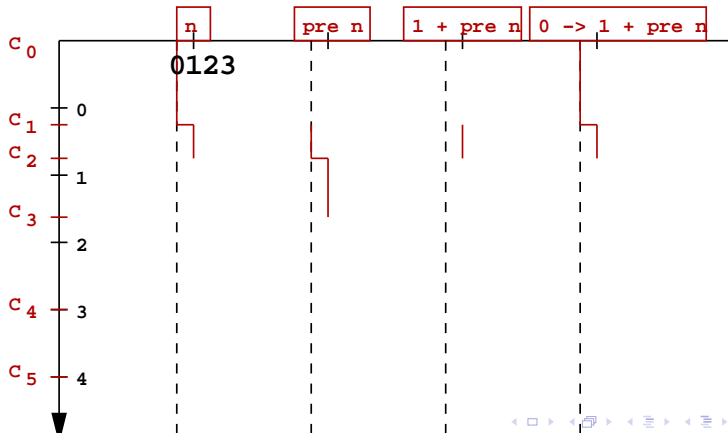
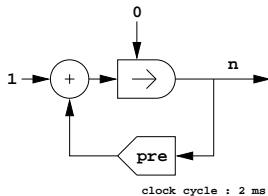


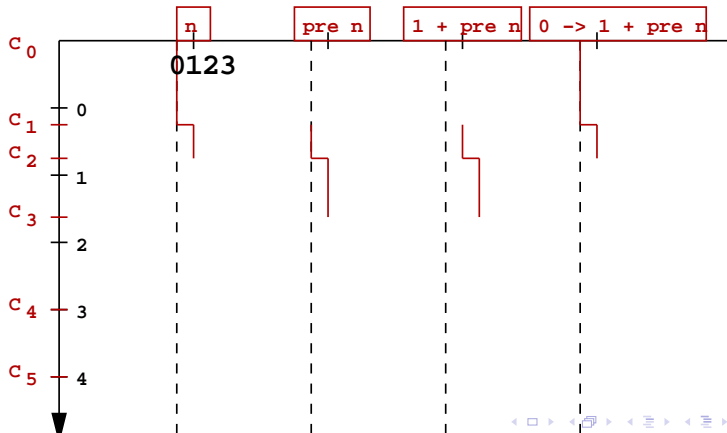
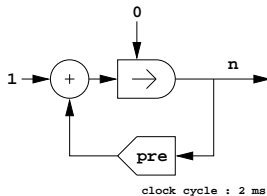


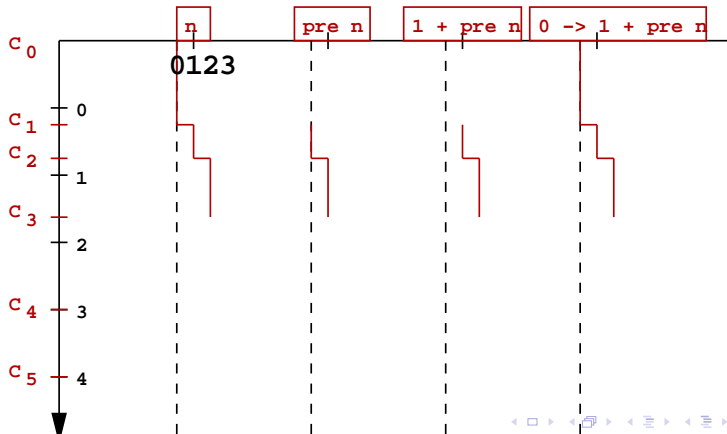
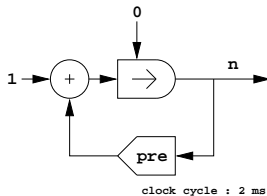


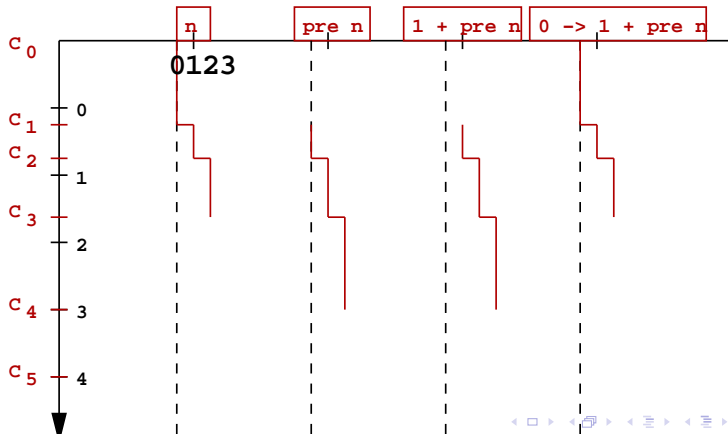
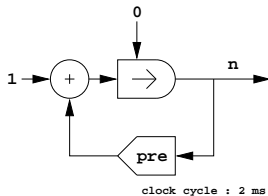


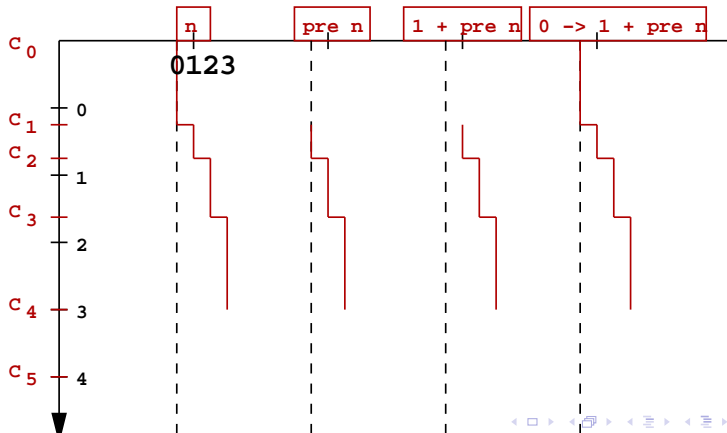
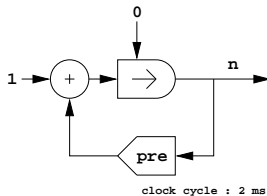


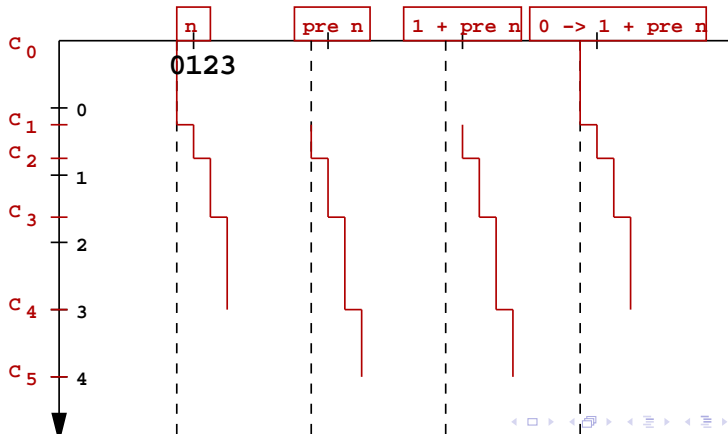
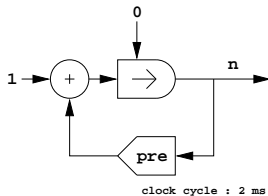


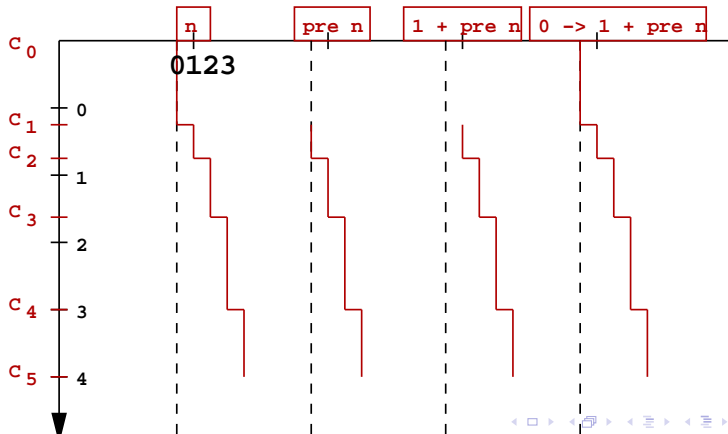
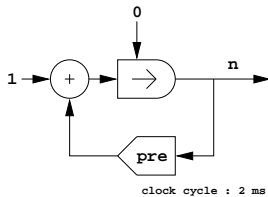




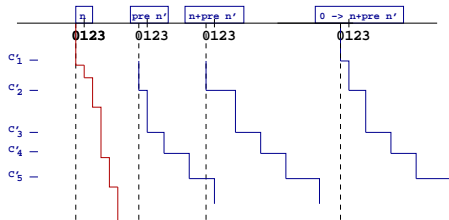
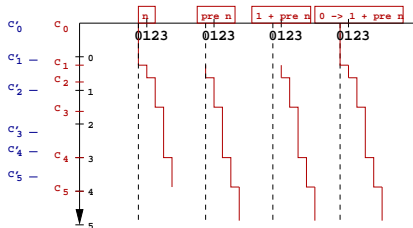
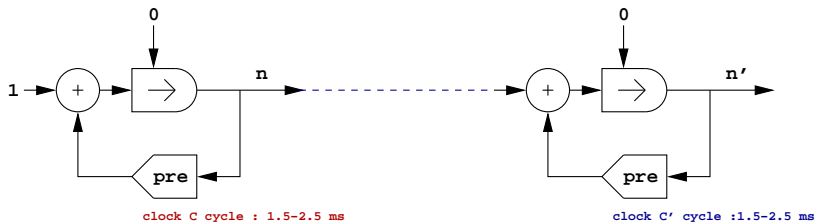




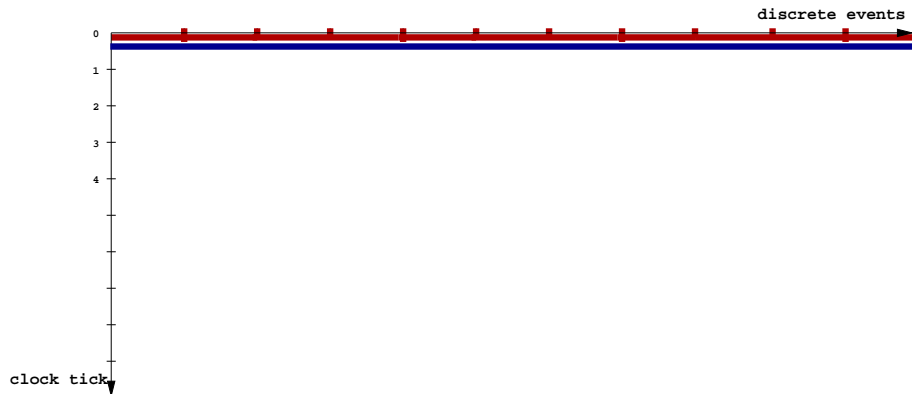




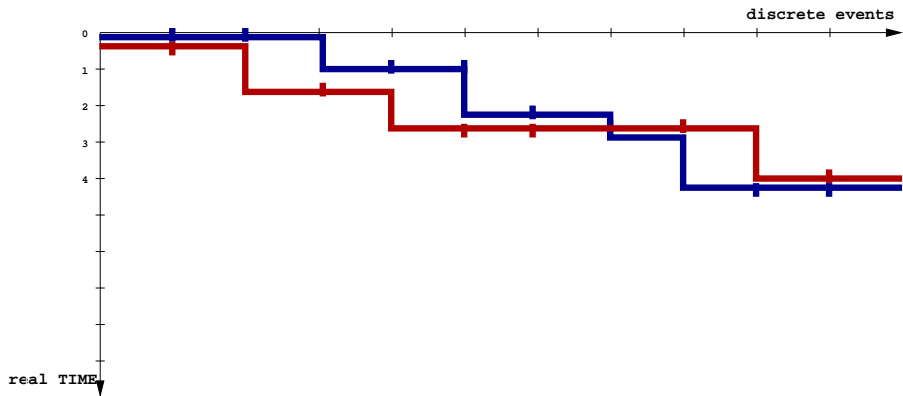
Avec quelles sémantiques doit-on travailler



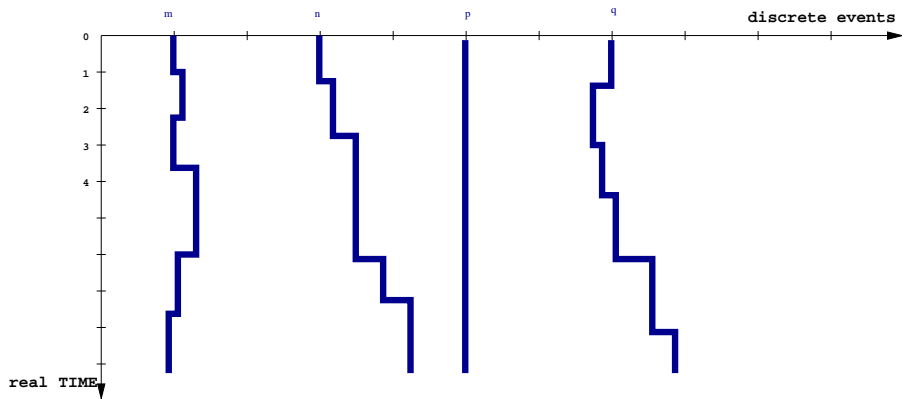
Résumé des modifications de la sémantique



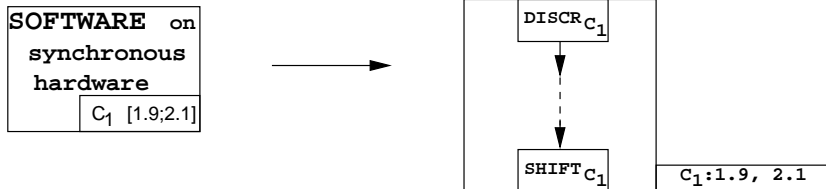
Résumé des modifications de la sémantique



Résumé des modifications de la sémantique

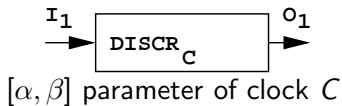


Comportement d'un système synchrone



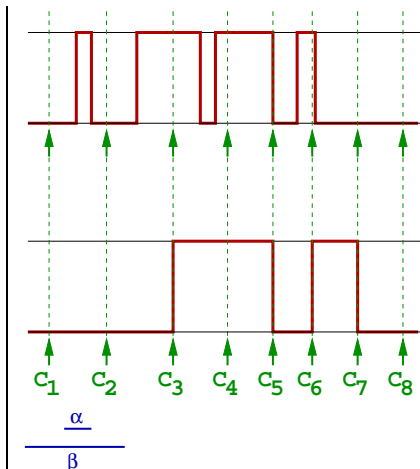
- une horloge est une fonction $:\mathbb{N} \rightarrow \mathbb{R}^+$
- paramétrée par $:[\alpha, \beta]$, avec $\alpha, \beta \in \mathbb{R}^+$ et $0 < \alpha \leq \beta$
- une horloge c satisfait $[\alpha, \beta]$ ssi $c_{n+1} - c_n \in [\alpha, \beta]$
- $DISCR_{C_1}$ modélise la lecture periodique du blackboard à l'entrée du système
- $SHIFT_{C_1}$ modélise l'attente du prochain tick d'horloge, et l'émission du résultat lors de ce tick

Sémantique des opérateurs temporels

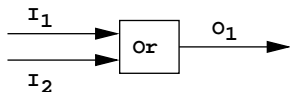


$$s_{O_1}(t) = \begin{cases} \bullet \text{ false} & \text{if } t < c(0) \\ \bullet \text{ } s_{I_1}(c_n) & \text{if } t \in [c_n, c_{n+1}) \end{cases}$$

$$s_{O_1} \triangleq \Psi_{\text{DISCR}_C}(s_{I_1})$$

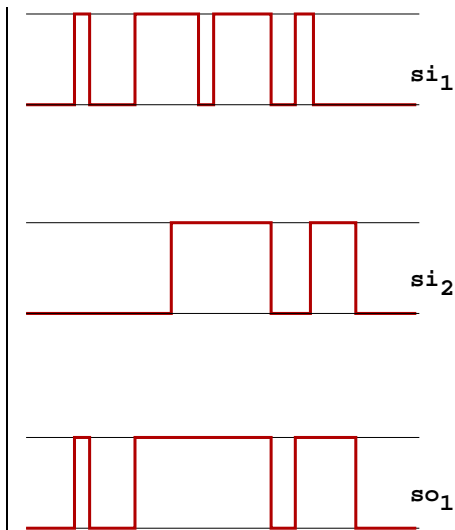


Sémantique des opérateurs atemporels



$$so_1(t) = \begin{cases} \bullet \textit{true} & \text{if } si_1(t) = \textit{true} \\ & \text{or } si_2(t) = \textit{true} \\ \bullet \textit{false} & \text{else} \end{cases}$$

$$so_1 \triangleq \Psi_{OR}(si_1, si_2)$$



semantique concrète

- un point de contrôle \triangle équivalent graphique d'une variable
- L'ensemble des points de contrôle est noté P .

semantique concrète

- un point de contrôle \triangleq equivalent graphique d'une variable
- L'ensemble des points de contrôle est noté P .
- D : ensemble de systèmes synchrones, $[[D]] \subseteq P \rightarrow (\mathbb{R}^+ \rightarrow \mathbb{B})$

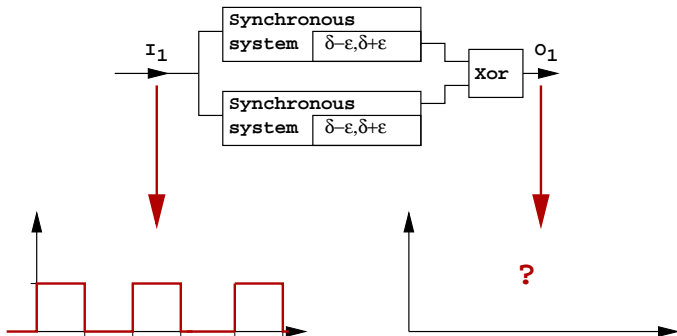
semantique concrète

- un point de contrôle \triangleq equivalent graphique d'une variable
- L'ensemble des points de contrôle est noté P .
- D : ensemble de systèmes synchrones, $\llbracket D \rrbracket \subseteq P \rightarrow (\mathbb{R}^+ \rightarrow \mathbb{B})$

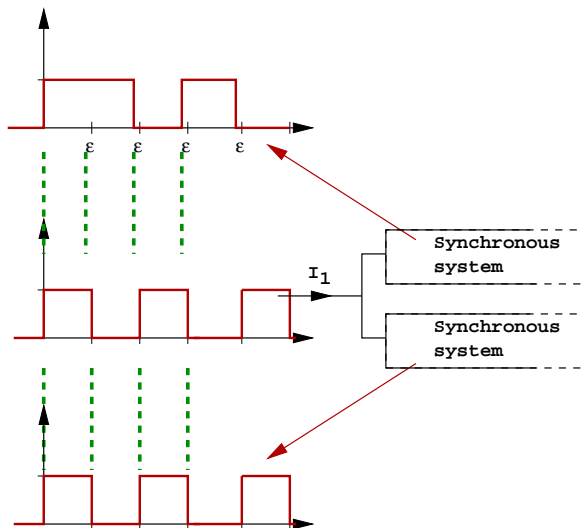
- $\left(\begin{array}{ccc} p_1 & \mapsto & s_1 \\ & \vdots & \\ p_{\#P-1} & \mapsto & s_{\#P-1} \end{array} \right) \in \llbracket D \rrbracket$ **iff** it satisfies the equations of all the gates in D .

Difficultés résultantes

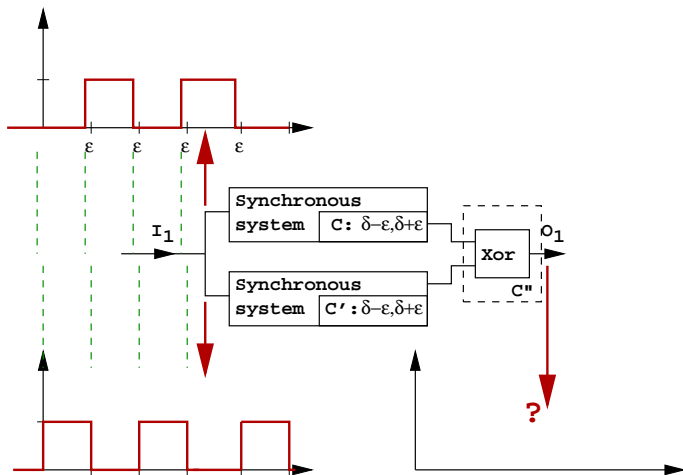
- imprécision de l'horloge (c-à-d *quasi-synchronous* à la place de *synchronous*) \Rightarrow nombre de comportements **non dénombrable**



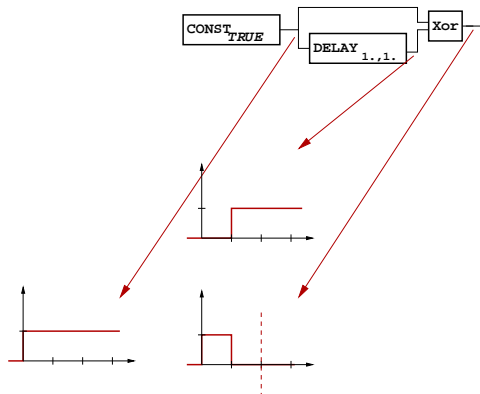
Difficultés résultantes



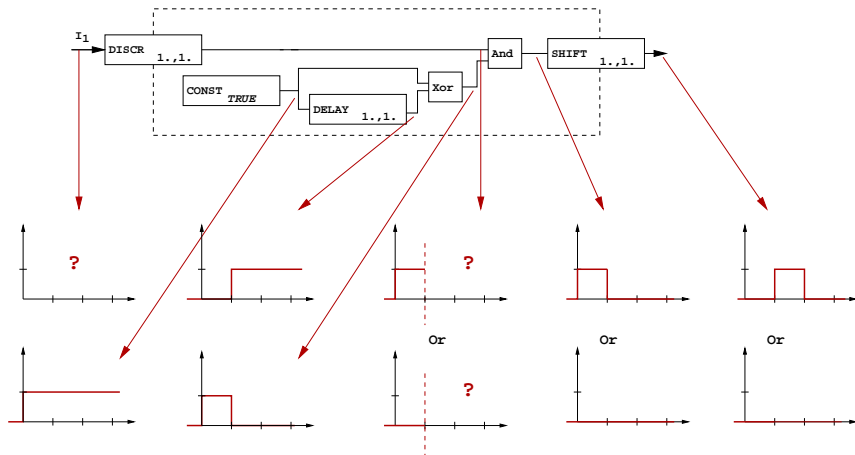
Difficultés résultantes



Analyse par regroupement ?

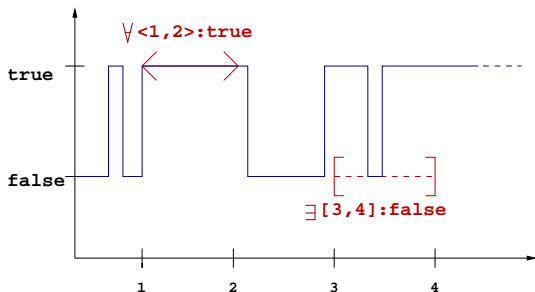


Analyse par regroupement ?



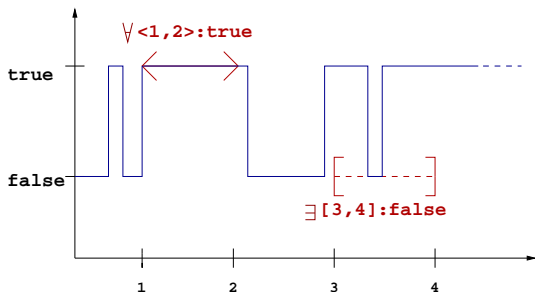
- Manipulation possible de données inconnues ou “infinies”

1er domaine abstrait : les contraintes



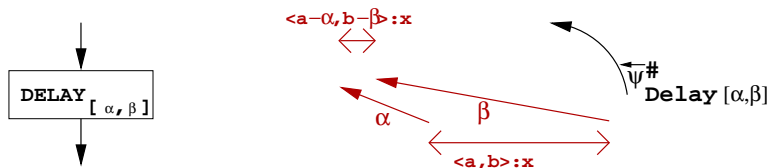
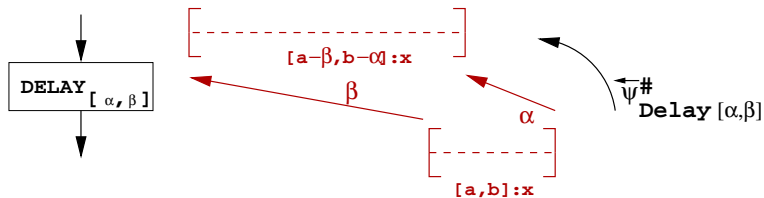
- La constraint $\exists [a; b] : x$ garantit que les signaux prennent la valeur x **au moins une fois** pendant $[a; b]$.
- La constraint $\forall \langle a; b \rangle : x$ garantit que les signaux prennent la valeur x **durant tout l'intervalle** $[a; b]$.

1er domaine abstrait : les contraintes



- La constraint $\exists[a; b] : x$ garantit que les signaux prennent la valeur x **au moins une fois** pendant $[a; b]$.
- La constraint $\forall \langle a; b \rangle : x$ garantit que les signaux prennent la valeur x **durant tout l'intervalle** $[a; b]$.
- Permet d'exprimer de nombreuses **propriétés temporelles**

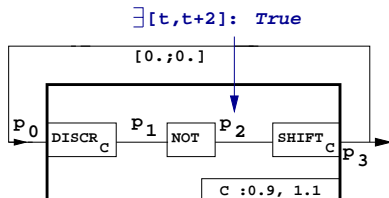
Opérateurs Abstraits et Contraintes : un exemple



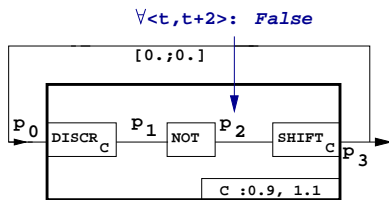
- $\overleftarrow{\Psi}^{\#}_{\text{Delay}[\alpha, \beta]}(\exists \langle a; b \rangle : x) \triangleq \exists \langle a - \beta; b - \alpha \rangle : x$
- $\overleftarrow{\Psi}^{\#}_{\text{Delay}[\alpha, \beta]}(\forall \langle a; b \rangle : x) \triangleq \forall \langle a - \alpha; b - \beta \rangle : x$

Analyse dans le domaine abstrait des **Contraintes**

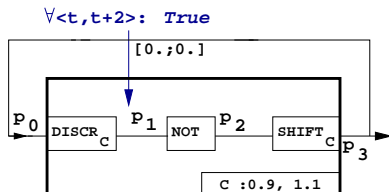
- Prouver la propriété abstraite suivante.



Analyse dans le domaine abstrait des **Contraintes**

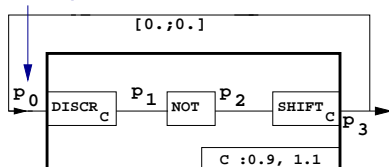


Analyse dans le domaine abstrait des **Contraintes**

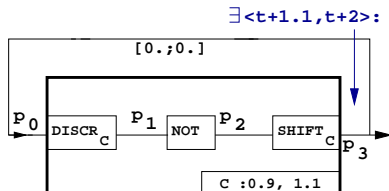


Analyse dans le domaine abstrait des **Contraintes**

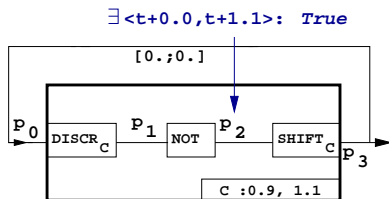
$\exists \langle t+1.1, t+2 \rangle: \text{True}$



Analyse dans le domaine abstrait des **Contraintes**



Analyse dans le domaine abstrait des **Contraintes**



Au point p_2 : 2 contraintes doivent être satisfaites :

- $\forall \langle t; t + 2 \rangle : \text{False}$
- $\exists [t + 0; t + 1.1] : \text{True}$

ce qui est impossible et invalide donc l'hypothèse initiale.

Analyse par interprétation abstraite

- $\llbracket D \rrbracket$: sémantique de l'ensemble de systèmes D .
- $[P]$ est l'ensemble de comportements satisfaisant la propriété P .
- **Ancien objectif** : Prove that $\llbracket D \rrbracket \subseteq [P]$.

Analyse par interprétation abstraite

- $\llbracket D \rrbracket$: sémantique de l'ensemble de systèmes D .
- $[P]$ est l'ensemble de comportements satisfaisant la propriété P .
- **Ancien objectif** : Prove that $\llbracket D \rrbracket \subseteq [P]$.
- **Or** :

$$\llbracket D \rrbracket \cap [\neg P] \subseteq \text{gfp}_{[\neg P]}(\Psi \cap Id)$$

Analyse par interprétation abstraite

- $\llbracket D \rrbracket$: sémantique de l'ensemble de systèmes D .
- $[P]$ est l'ensemble de comportements satisfaisant la propriété P .
- **Ancien objectif** : Prove that $\llbracket D \rrbracket \subseteq [P]$.

- **Or** :

$$\llbracket D \rrbracket \cap [\neg P] \subseteq \text{gfp}_{[\neg P]}(\Psi \cap Id)$$

-

$$\text{gfp}_{[\neg P]}(\Psi \cap Id) \subseteq? \emptyset$$

Analyse par interprétation abstraite

- $\llbracket D \rrbracket$: sémantique de l'ensemble de systèmes D .
- $[P]$ est l'ensemble de comportements satisfaisant la propriété P .
- **Ancien objectif** : Prove that $\llbracket D \rrbracket \subseteq [P]$.

- **Or** :

$$\llbracket D \rrbracket \cap [\neg P] \subseteq \text{gfp}_{[\neg P]}(\Psi \cap Id)$$

-

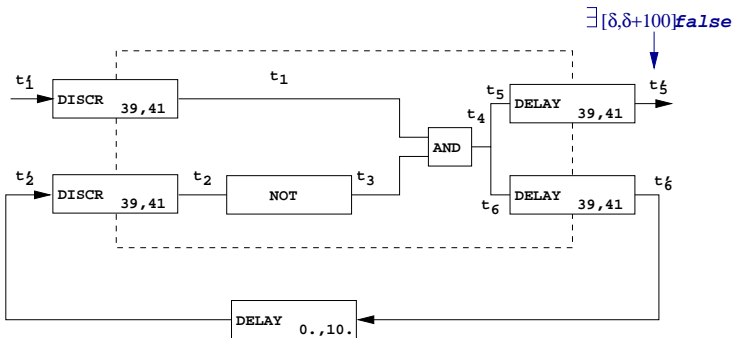
$$\text{gfp}_{[\neg P]}(\Psi \cap Id) \subseteq^? \emptyset$$

- **Prouvé dans le cas où** :

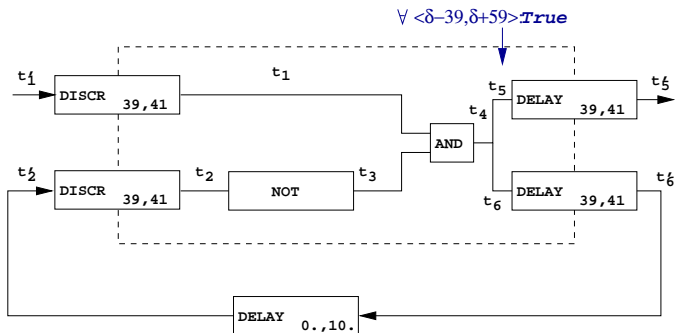
$$\text{gfp}_{[\neg P]}(\Psi^\# \cap Id^\#) \subseteq^\# \emptyset^\# = \perp$$

Exemple d'analyse : Itération jusqu'au point fixe vide

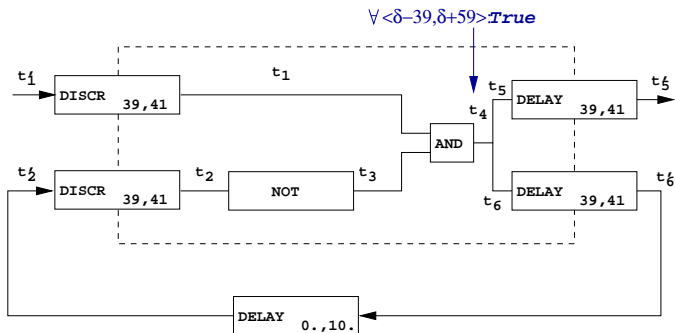
- prouver la propriété abstraite suivante



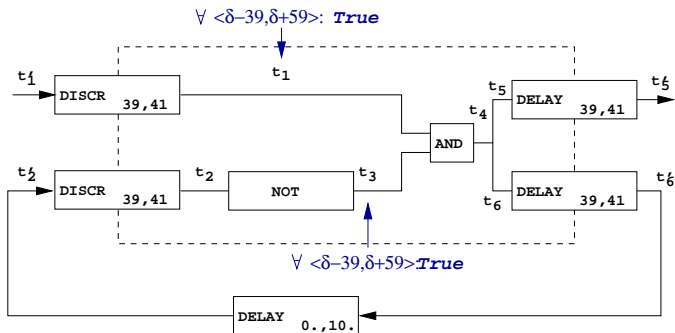
Exemple : Itération jusqu'au point fixe (vide ?)



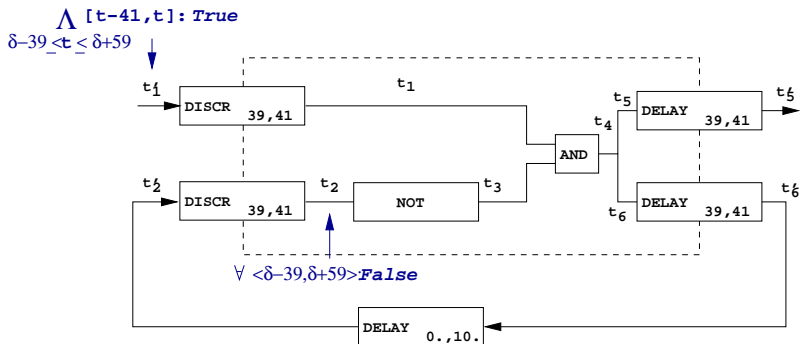
Exemple : Itération jusqu'au point fixe (vide ?)



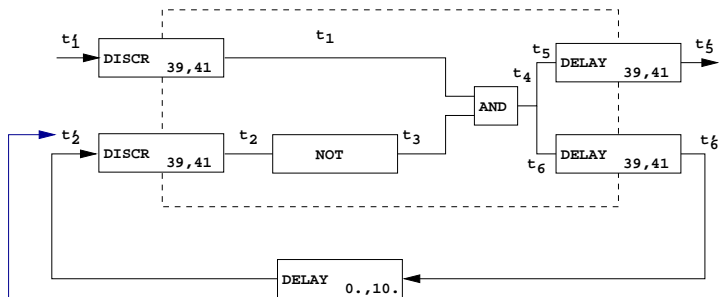
Exemple : Itération jusqu'au point fixe (vide ?)



Exemple : Itération jusqu'au point fixe (vide ?)

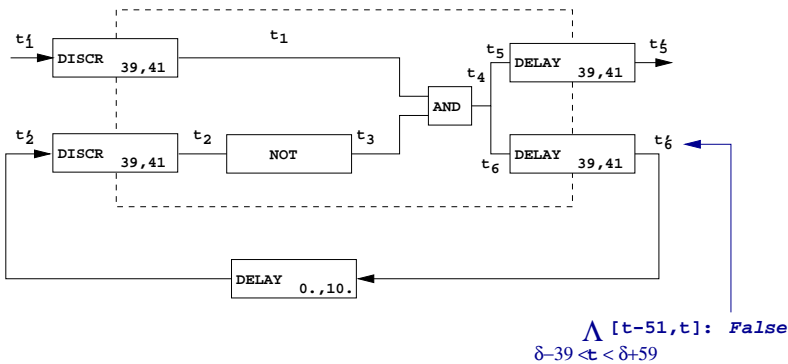


Exemple : Itération jusqu'au point fixe (vide ?)

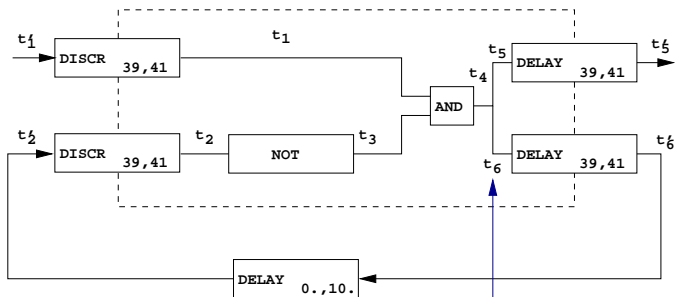


$\bigwedge [t-41, t]: \text{False}$
 $\delta-39 \leq t \leq \delta+59$

Exemple : Itération jusqu'au point fixe (vide ?)



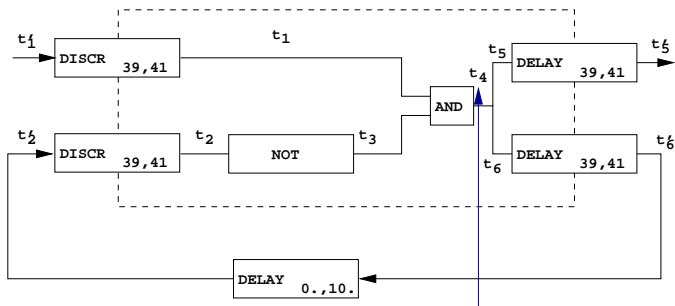
Exemple : Itération jusqu'au point fixe (vide ?)



$$\bigwedge [t-92, t-39]: \text{False}$$

$$\delta-39 \leq t \leq \delta+59$$

Exemple : Itération jusqu'au point fixe (vide ?)



$$\Delta [t-92, t-39]: \text{False}$$

$$\delta-39 \leq t \leq \delta+59$$

Exemple : Itération jusqu'à un point fixe vide

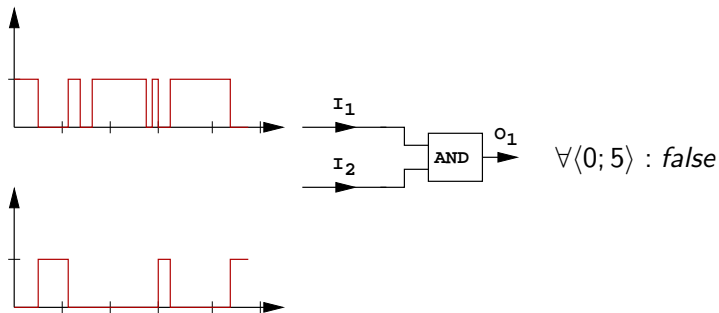
Les signaux doivent donc satisfaire au point de contrôle t_4 :

$$\langle \delta - 39, \delta + 59 \rangle : \textit{True} \textbf{ and } \bigwedge_{\delta - 39 \leq t \leq \delta + 59} ([t - 92, t - 39] : \textit{False})$$

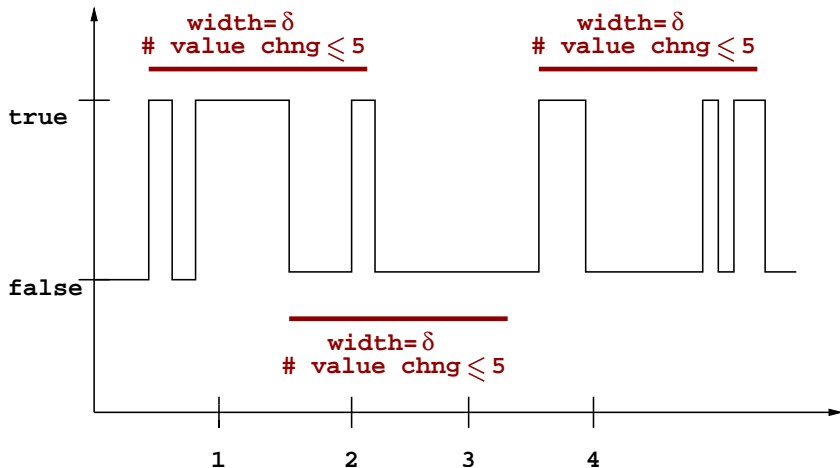
ce qui implique $[\delta - 33 = \delta + 59 - 92, \delta + 20 = \delta + 59 - 39] : \textit{False}$

Faiblesses du domaine des contraintes

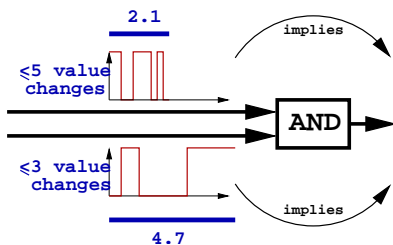
- Domaine précis dans le cas de : DELAY, DISCR, SHIFT, NOT,
- Grande perte de précision dans le cas de : AND, OR, XOR



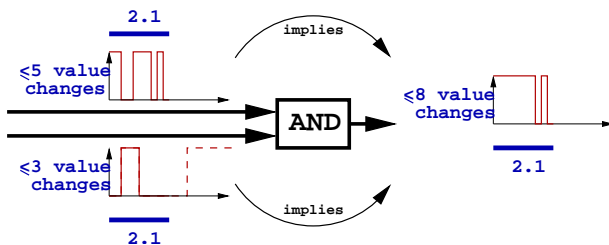
2ème Domaine Abstrait : Domaine du comptage des Changements de valeurs



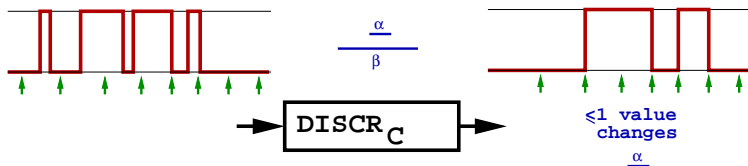
un opérateur abstrait **indépendant** du temps dans le domaine du comptage des Changements de valeurs



un opérateur abstrait **indépendant** du temps dans le domaine du comptage des Changements de valeurs



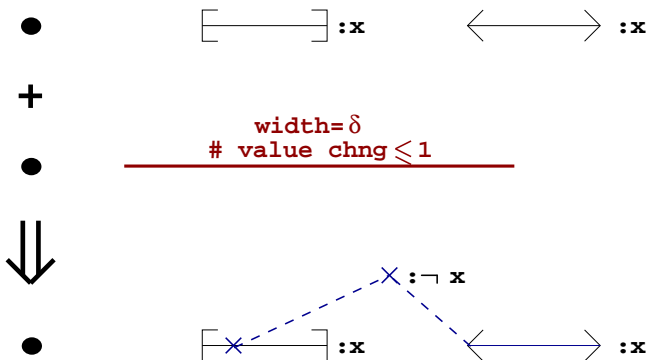
un opérateur abstrait **dépendant** du temps dans le domaine du comptage des Changements de valeurs



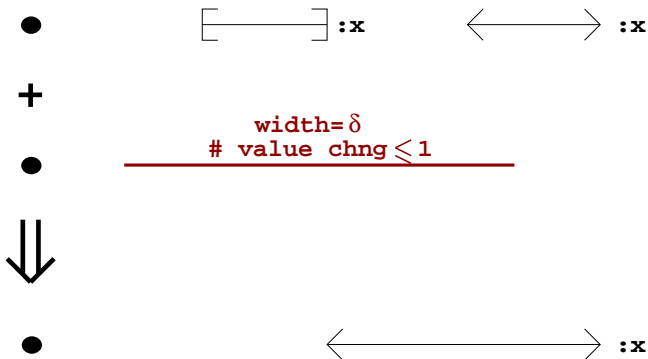
- $[\alpha, \beta]$ paramètre de l'horloge C
- $\vec{\Psi}_{\text{DISCR}_{[\alpha, \beta]}}^{\#}(\text{any input}) \triangleq (1, \alpha)$

Produit Réduit

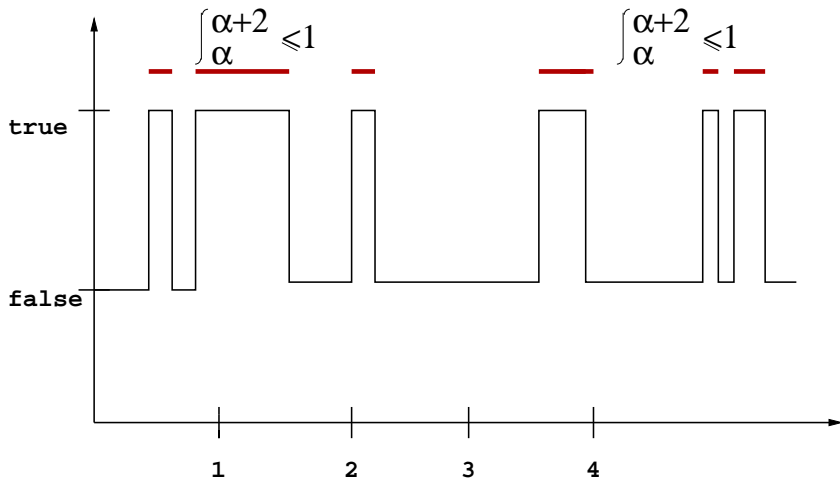
Contraintes-nombre de changements de valeur



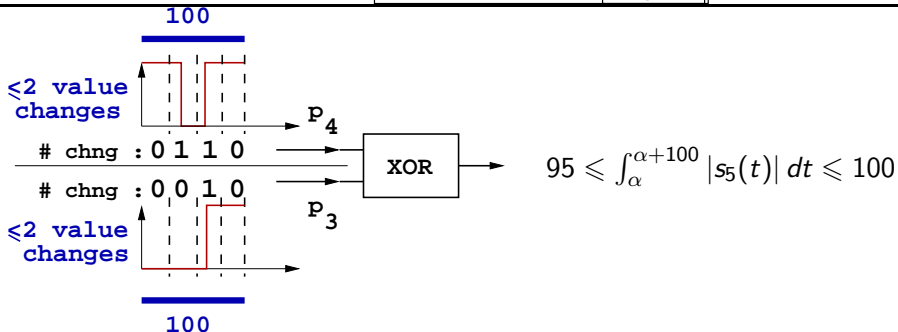
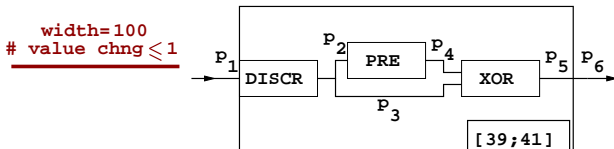
Produit Réduit Contraintes-nombre de changements de valeur



3ème domaine abstrait : encadrement des intégrales



- Permet d'exprimer les spécifications quantitatives

Exemple de **Coopération** entre les 3 domaines abstraits

Résultats

- Sous certaines hypothèses, certaines des techniques proposées par P. Caspi sont validées
- Analyse possible quand divergence de l'horloge reste mesurée
- Stabilité des signaux à l'entrée très importante (les contre-exemples sont souvent des signaux alternants)

Conclusion

- Modèle réaliste de l'exécution des systèmes synchrones communicant à horloge imparfaite
- La syntaxe permet des annotations quantifiant les imperfections matérielles
- La sémantique est à "temps continu" (\neq des sémantiques classiques)
- Modularité de l'analyse vis à vis des domaines abstraits