

Des domaines abstraits temporels pour prouver les spécifications temporelles des systèmes temps-réel

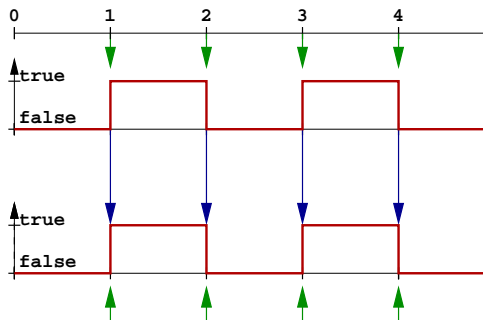
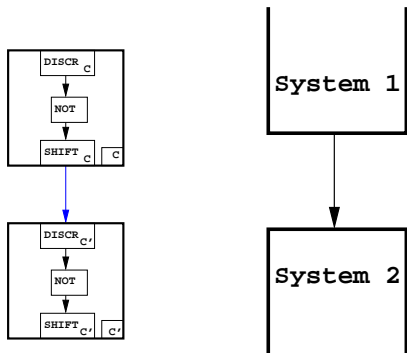
Julien Bertrane
bertrane@di.ens.fr

ENS

5 février 2010

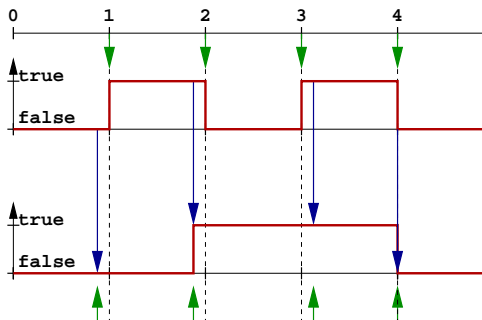
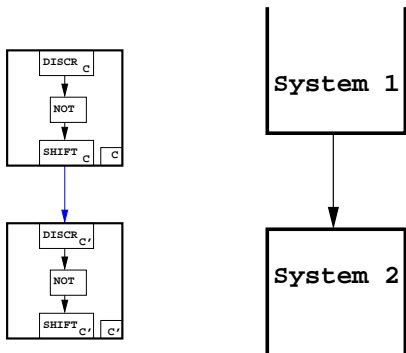
Introduction

Champ de l'étude



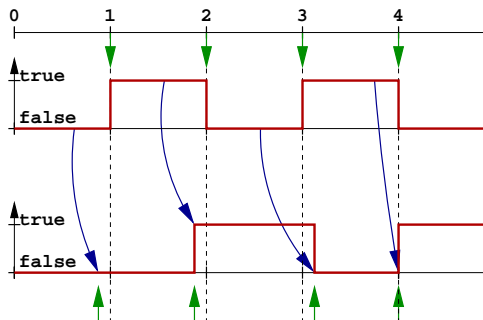
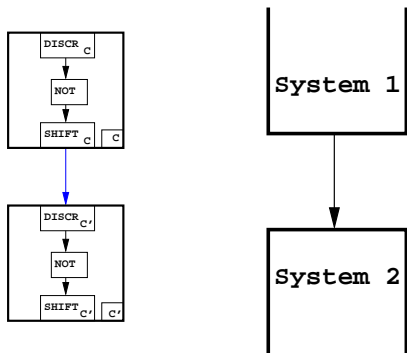
- Certaines imperfections matérielles inévitables sont rarement considérées :

Champ de l'étude



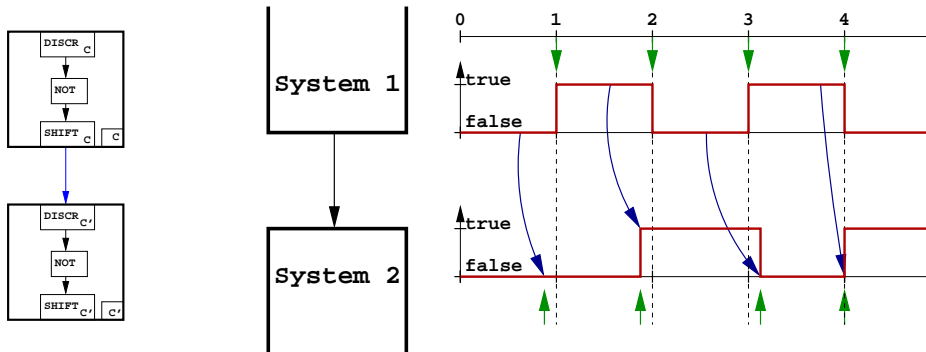
- Certaines imperfections matérielles inévitables sont rarement considérées :
 - ▶ Les horloges des unités de contrôle se **désynchronisent**

Champ de l'étude



- Certaines imperfections matérielles inévitables sont rarement considérées :
 - ▶ Les horloges des unités de contrôle se **désynchronisent**
 - ▶ Les communications **ne peuvent pas** être considérées **instantanées**

Champ de l'étude

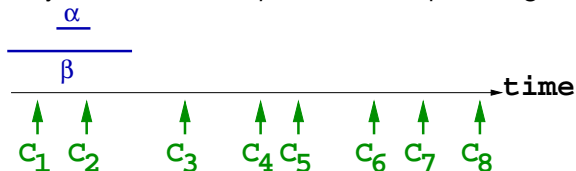


- Certaines imperfections matérielles inévitables sont rarement considérées :
 - ▶ Les horloges des unités de contrôle se **désynchronisent**
 - ▶ Les communications **ne peuvent pas** être considérées **instantanées**
 - ▶ Les **délais** de communications **ne peuvent pas** être considérés **constants**

Hypothèses sur la sémantique

- **Synchronie imparfaite** :

- ▶ Désynchronisation : la *période* de chaque horloge est dans $[\alpha, \beta]$.

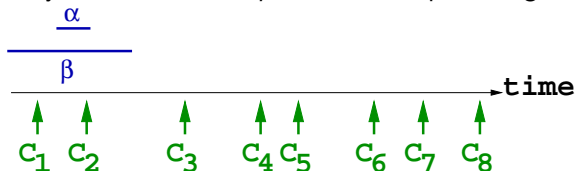


- ▶ Une horloge c est de paramètre $[\alpha, \beta]$ ssi $c_{n+1} - c_n \in [\alpha, \beta]$

Hypothèses sur la sémantique

- **Synchronie imparfaite** :

- ▶ Désynchronisation : la *période* de chaque horloge est dans $[\alpha, \beta]$.



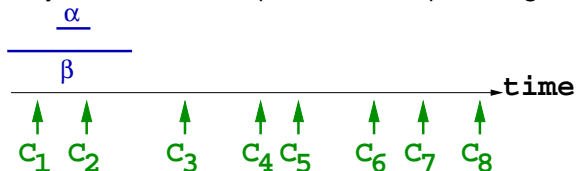
- ▶ Une horloge c est de paramètre $[\alpha, \beta]$ ssi $c_{n+1} - c_n \in [\alpha, \beta]$

- Transmissions **en série** entre systèmes imparfaitement synchrones

Hypothèses sur la sémantique

- **Synchronie imparfaite** :

- ▶ Désynchronisation : la *période* de chaque horloge est dans $[\alpha, \beta]$.



- ▶ Une horloge c est de paramètre $[\alpha, \beta]$ ssi $c_{n+1} - c_n \in [\alpha, \beta]$

- Transmissions **en série** entre systèmes imparfaitement synchrones
- **Blackboard** à l'entrée de chaque unité de contrôle

Spécifications à prouver

- Spécifications de **Sûreté**
 - ▶ **pour tout comportement, à chaque instant t , $s(t) \neq true$**

Spécifications à prouver

- Spécifications de **Sûreté**

- ▶ **pour tout comportement, à chaque instant t , $s(t) \neq true$**

- Spécifications **temporelles**

- ▶ **pour tout comportement, il n'y a pas d'instant t tel que :**

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

Spécifications à prouver

- Spécifications de **Sûreté**

- ▶ **pour tout comportement, à chaque instant t , $s(t) \neq true$**

- Spécifications **temporelles**

- ▶ **pour tout comportement, il n'y a pas d'instant t tel que :**

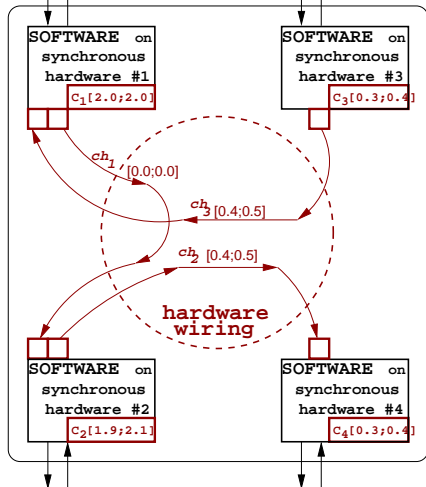
$$\text{pour tout } t' \in [t, t + \alpha], s(t') = true$$

- Spécifications **quantitatives**

- ▶ Les **sorties** de 2 systèmes redondants restent **égaux au moins 75% du temps** durant tout intervalle de temps durant au moins δ .

Système type à vérifier : quantifications des imperfections matérielles

HARDWARE (environment, sensors, actuators)

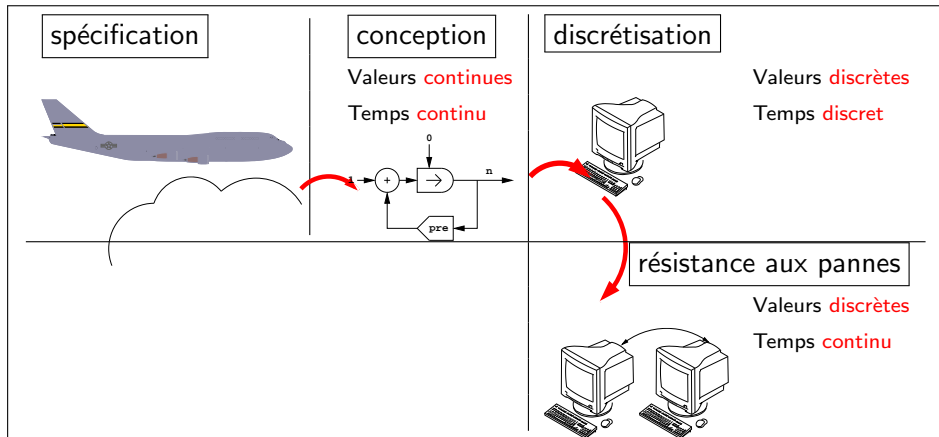


HARDWARE (environment, sensors, actuators)

Sémantique

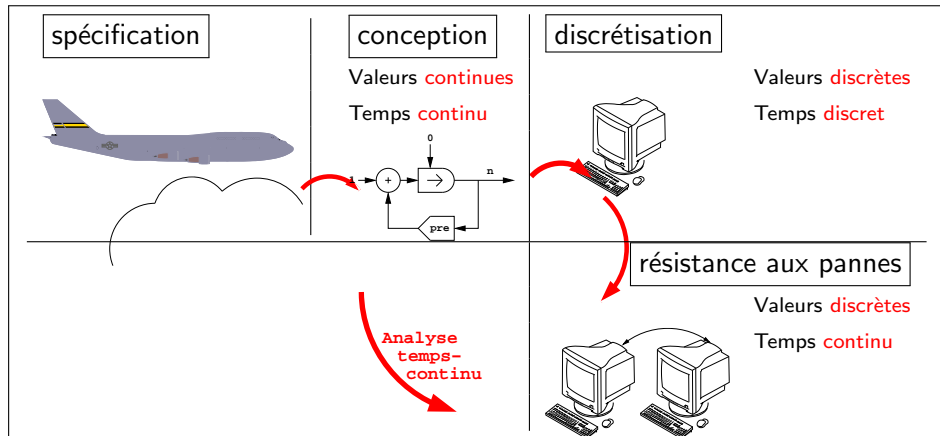
Différences entre conception et analyse statique

- Développement des systèmes embarqués



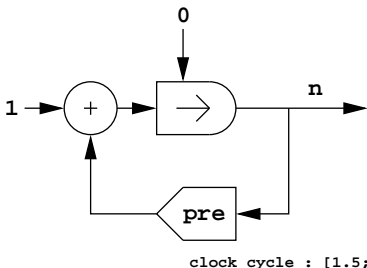
Différences entre conception et analyse statique

- **Preuve automatique** des systèmes embarqués



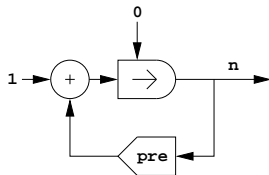
Quelle sémantique ?

Les synchrones programmes peuvent être compilés en C

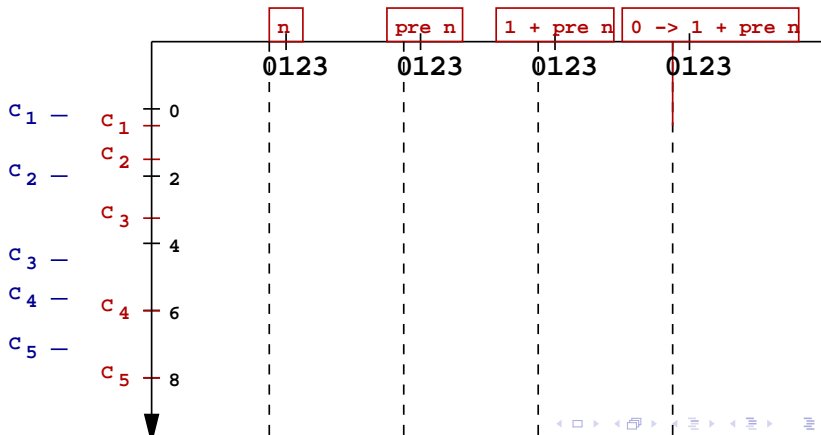


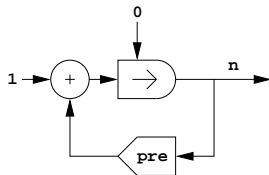
```
while true{
  switch (global_state->current_state){
    case 0:
      global_state->n = 0;
      global_state->current_state = 1;
      break;
    case 1:
      global_state->n = (global_state->n)+ 1;
      global_state->current_state = 1;
      break;}}
```

- En C, il est difficile de savoir à quel instant un événement se produit (optimisation à la compilation, hardware, ...)
- la sémantique du binaire est temps-**réel**
- En Lustre, **chaque point** du système a une valeur à chaque cycle
- Proposition : donner a une valeur à **chaque instant** (réel) à **chaque point du système**

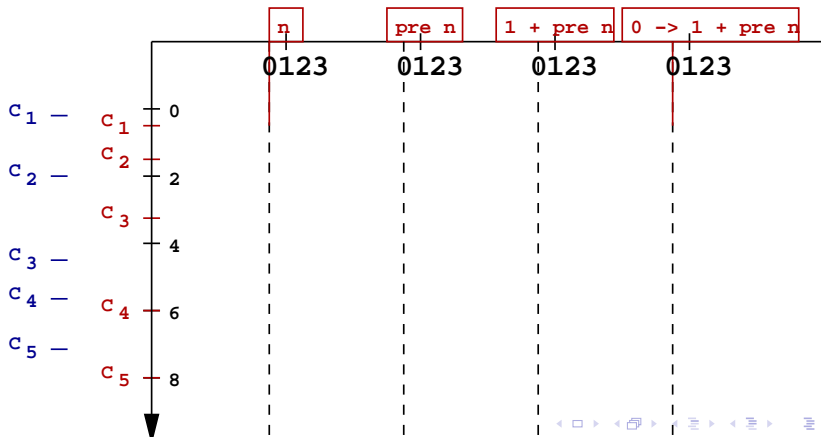


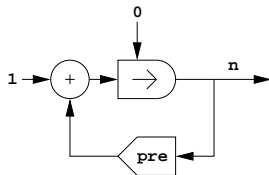
clock cycle : 1.5-2.5 ms



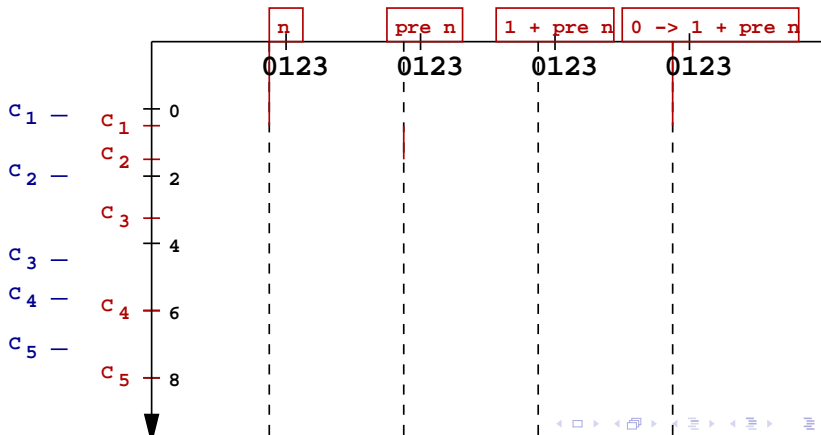


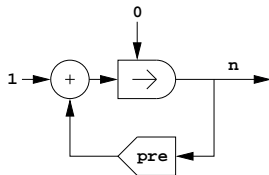
clock cycle : 1.5-2.5 ms



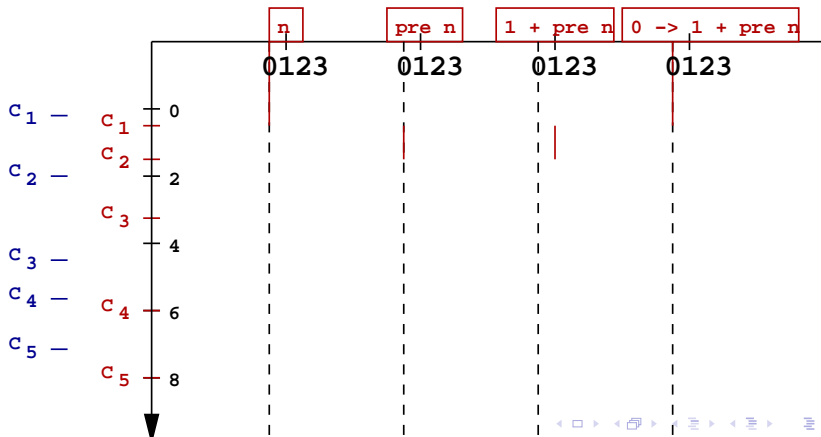


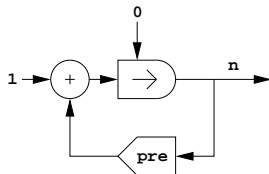
clock cycle : 1.5-2.5 ms



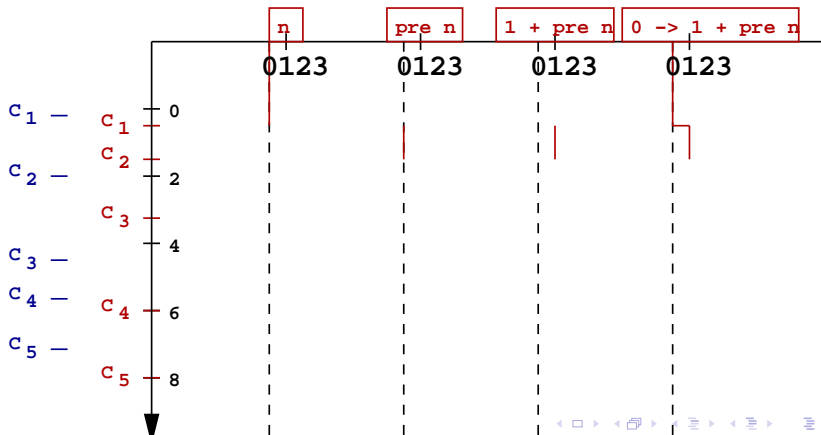


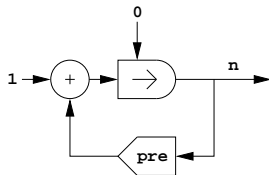
clock cycle : 1.5-2.5 ms



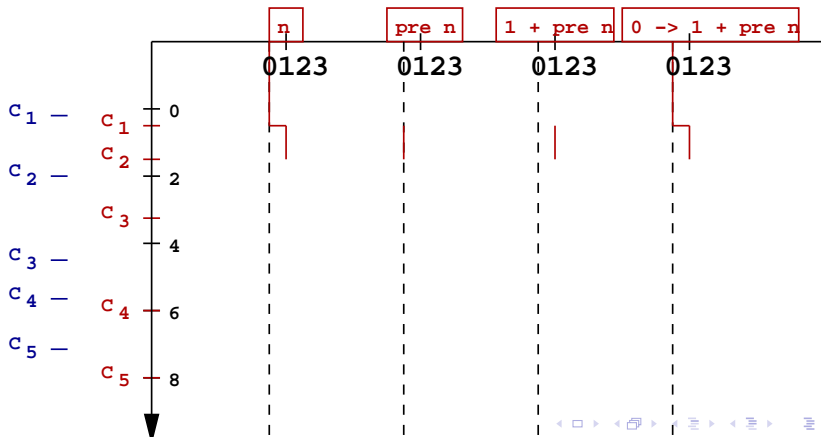


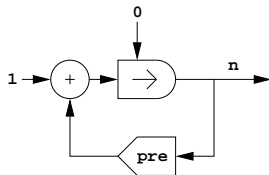
clock cycle : 1.5-2.5 ms



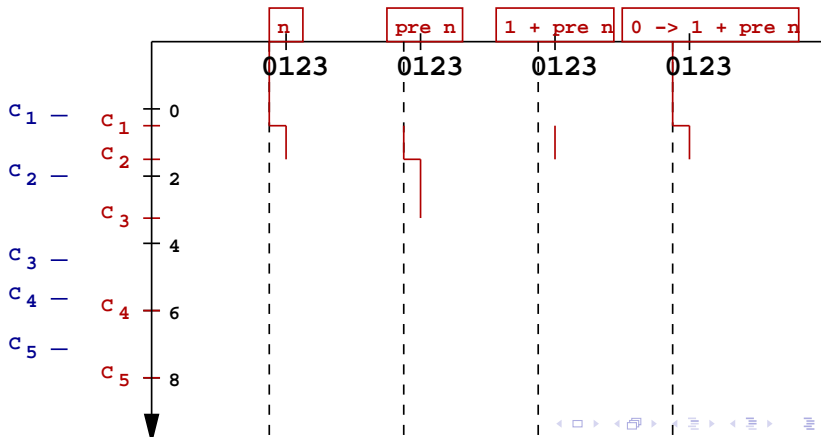


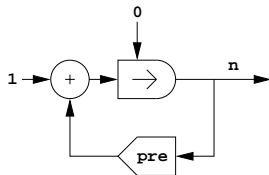
clock cycle : 1.5-2.5 ms



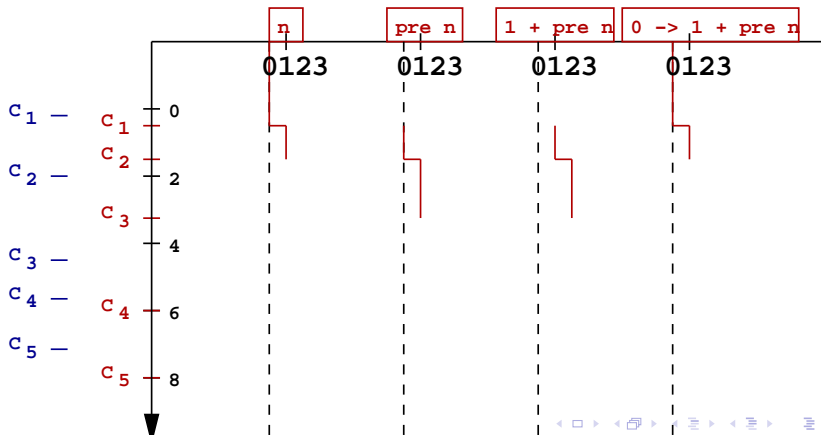


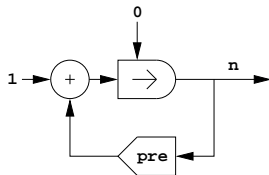
clock cycle : 1.5-2.5 ms



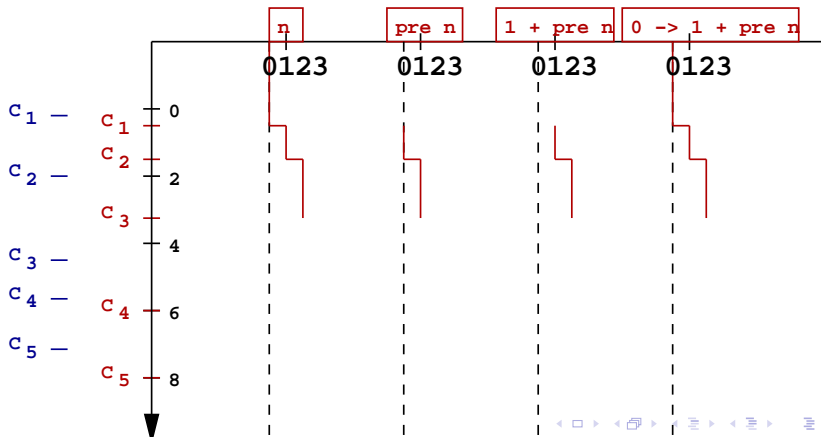


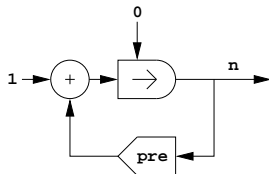
clock cycle : 1.5-2.5 ms



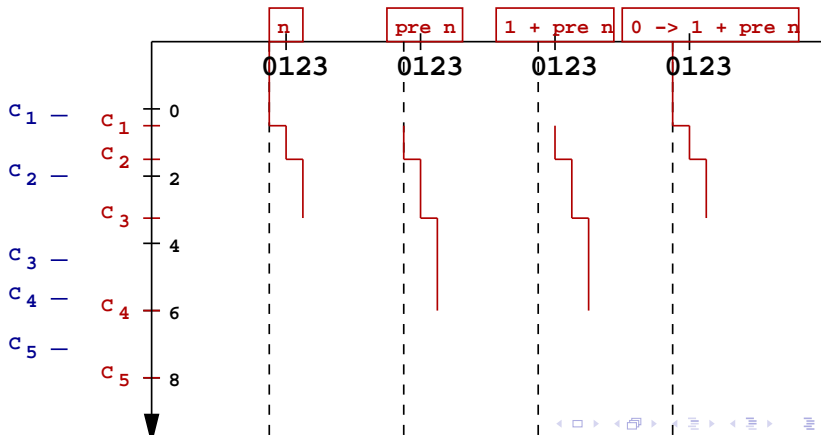


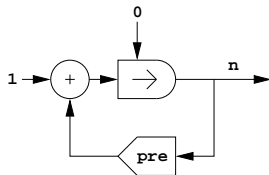
clock cycle : 1.5-2.5 ms



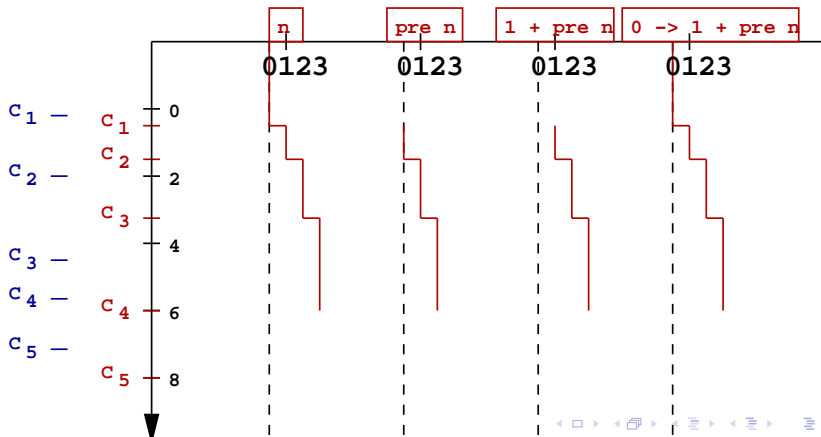


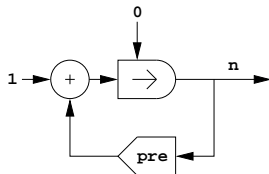
clock cycle : 1.5-2.5 ms



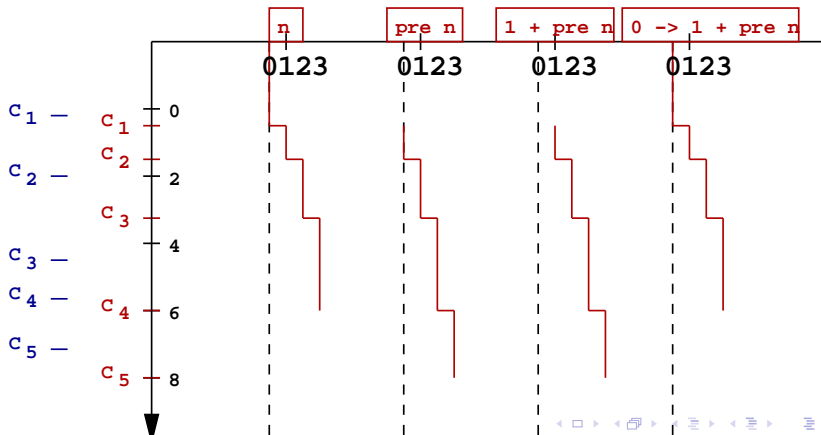


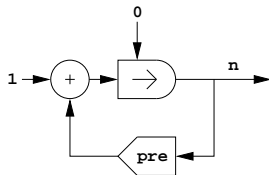
clock cycle : 1.5-2.5 ms



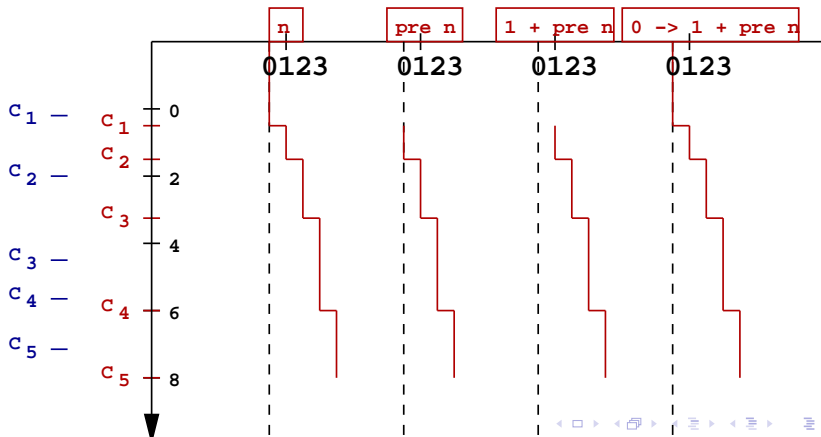


clock cycle : 1.5-2.5 ms

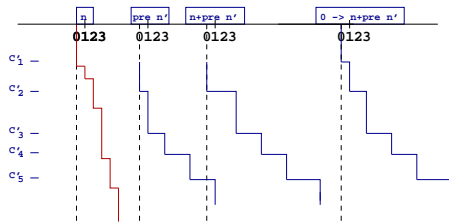
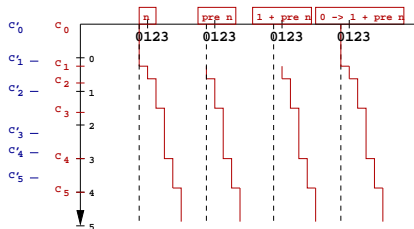
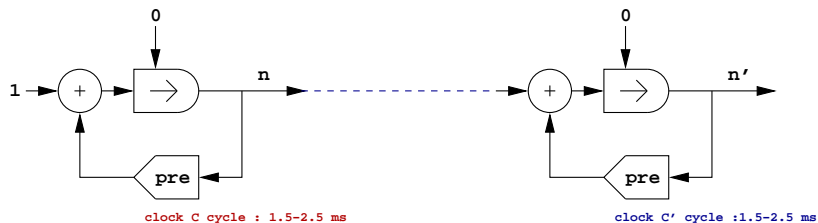




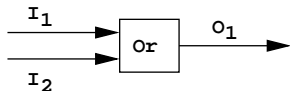
clock cycle : 1.5-2.5 ms



Une sémantique non-standard

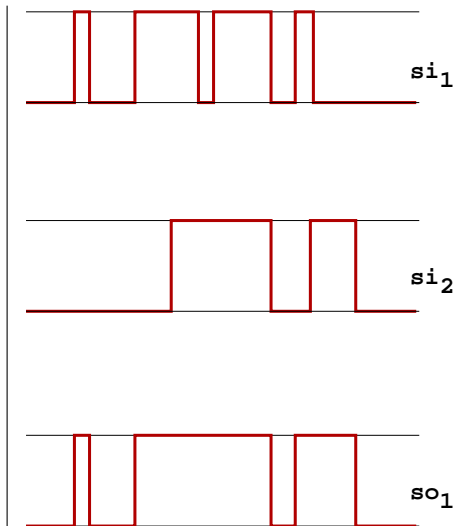


Sémantique des opérateurs non temporels

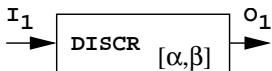


$$so_1(t) = \begin{cases} \bullet \text{ true} & \text{if } si_1(t) = \text{true} \\ & \text{or } si_2(t) = \text{true} \\ \bullet \text{ false} & \text{else} \end{cases}$$

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



Sémantique des opérateurs temporels

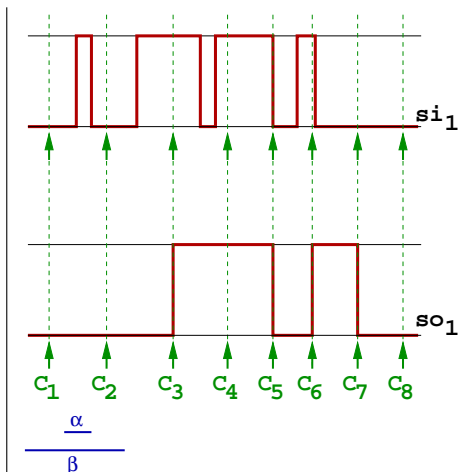


Il existe une horloge C de
paramètre $[\alpha, \beta]$

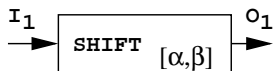
i.e. : $c_{n+1} - c_n \in [\alpha, \beta]$ telle que

$$so_1(t) = \begin{cases} \bullet \text{ false} & \text{if } t < c(0) \\ \bullet \text{ } si_1(c_n) & \text{if } t \in [c_n, c_{n+1}) \end{cases}$$

$$so_1 \triangleq \Psi_{DISCR_{[\alpha, \beta]}}(si_1)$$



Sémantique des opérateurs temporels

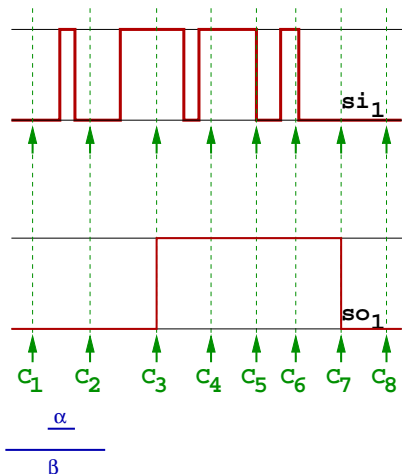


Il existe une horloge C de
paramètre $[\alpha, \beta]$

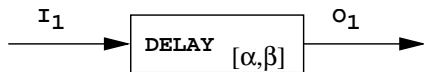
i.e. : $c_{n+1} - c_n \in [\alpha, \beta]$ telle que

$$so_1(t) = \begin{cases} \bullet \text{ false} & \text{if } t < c(0) \\ \bullet \lim_{\substack{t \rightarrow c_n \\ t < c_n}} si_1(t) & \text{if } t \in [c_n, c_{n+1}) \end{cases}$$

$$so_1 \triangleq \Psi_{\text{SHIFT}_{[\alpha, \beta]}}(si_1)$$



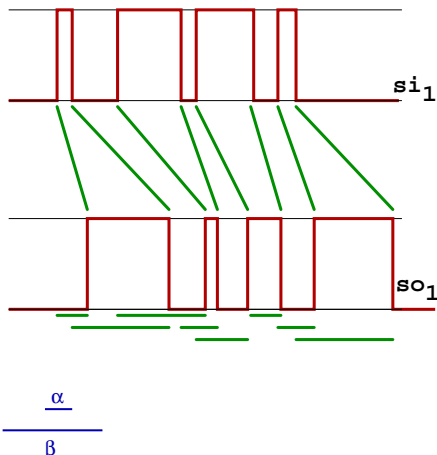
Sémantique des opérateurs temporels



Il existe une fonction δ
 bijection croissante de $\mathbb{R} \rightarrow \mathbb{R}$
 de paramètre $[\alpha, \beta]$ i.e. :
 $\forall t \in \mathbb{R}, \delta(t) - t \in [\alpha, \beta]$ telle que

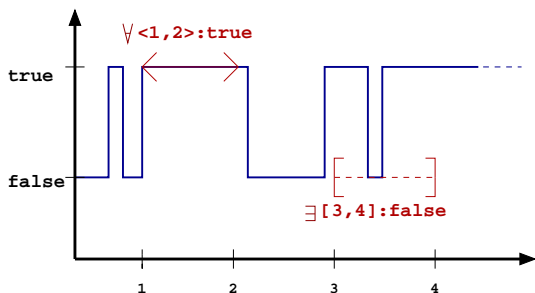
$$so_1(\delta(t)) = si_1(t)$$

$$so_1 \triangleq \Psi_{\text{DELAY}_{[\alpha, \beta]}}(si_1)$$



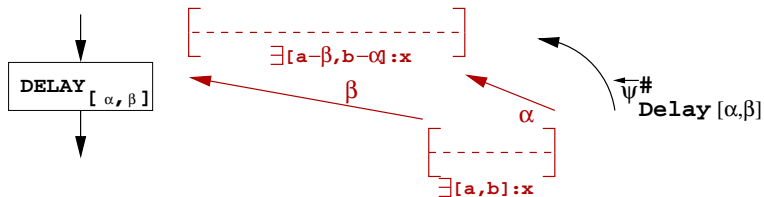
Domaines abstraits temporels

1er domaine abstrait : les contraintes



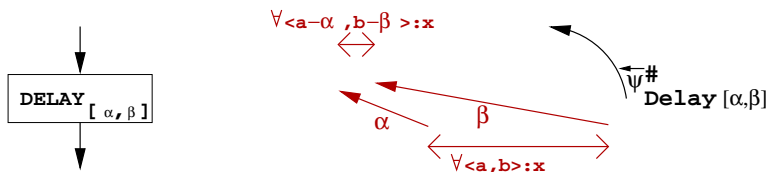
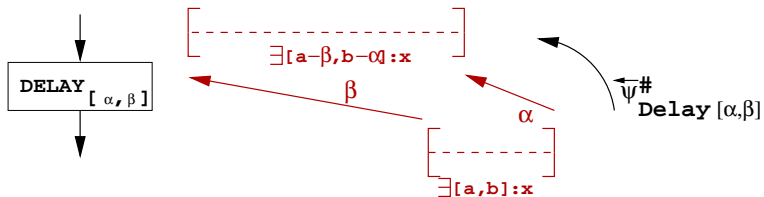
- La contrainte $\exists[a; b] : x$ garantit que les signaux prennent la valeur x **au moins une fois** pendant $[a; b]$.
- La contrainte $\forall\langle a; b \rangle : x$ garantit que les signaux prennent la valeur x **durant tout l'intervalle** $[a; b]$.

Opérateurs abstraits et Contraintes : un exemple



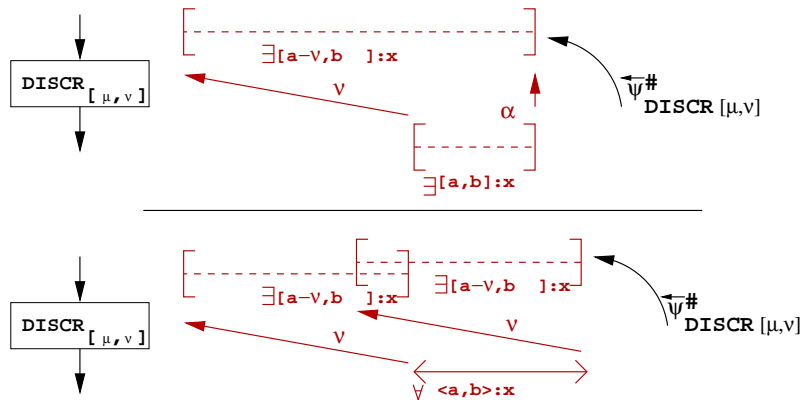
- $$\overleftarrow{\Psi}^{\#}_{\text{DELAY}[\alpha, \beta]}(\exists[a; b] : x) \triangleq \exists[a - \beta; b - \alpha] : x$$

Opérateurs abstraits et Contraintes : un exemple



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

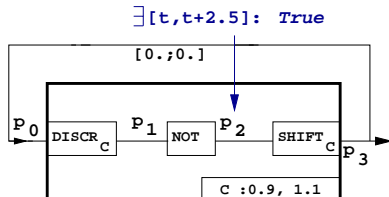
Opérateurs abstraits et Contraintes : un exemple

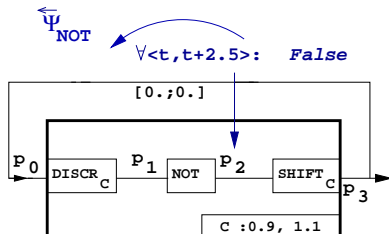


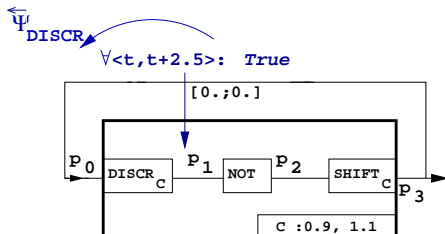
- $\overleftarrow{\Psi}^{\#}_{DISCR_{[\mu, \nu]}}(\exists[a; b] : x) \triangleq \exists[a - \nu; b] : x$
- $\overleftarrow{\Psi}^{\#}_{DISCR_{[\mu, \nu]}}(\forall \langle a; b \rangle : x) \triangleq \bigwedge_{u \in [a, b]} \exists[u - \nu; u] : x$

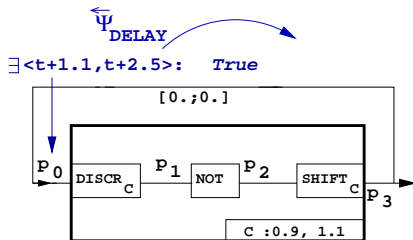
Analyse dans le domaine abstrait des **Contraintes**

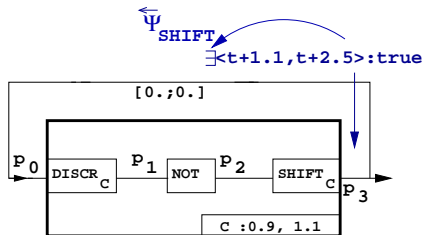
- Prouver la propriété abstraite suivante :



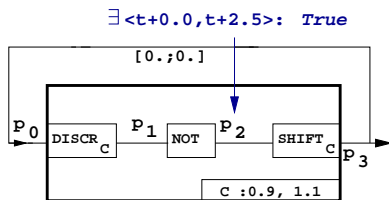
Analyse dans le domaine abstrait des **Contraintes**

Analyse dans le domaine abstrait des **Contraintes**

Analyse dans le domaine abstrait des **Contraintes**

Analyse dans le domaine abstrait des **Contraintes**

Analyse dans le domaine abstrait des **Contraintes**



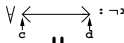
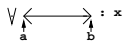
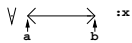
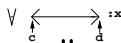
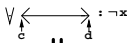
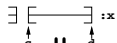
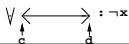
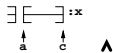
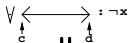
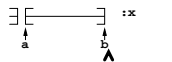
Au point p_2 : 2 contraintes doivent être satisfaites :

- $\forall \langle t; t + 2.5 \rangle : \text{False}$
- $\exists [t; t + 2.5] : \text{True}$

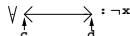
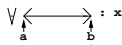
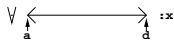
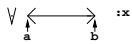
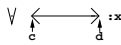
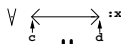
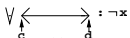
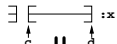
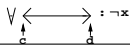
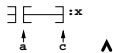
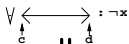
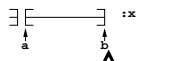
c'est impossible. L'hypothèse initiale est invalidée : $\forall \langle t; t + 2.5 \rangle : \text{False}$.

- Donc $\exists [t; t + 2.5] : \text{True}$ est certifié.

Opérations sur les contraintes : intersection abstraite

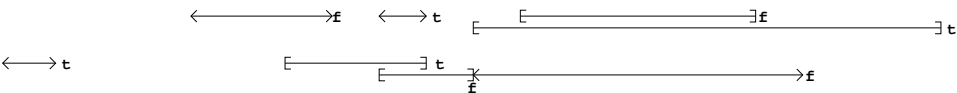


Opérations sur les contraintes : intersection abstraite

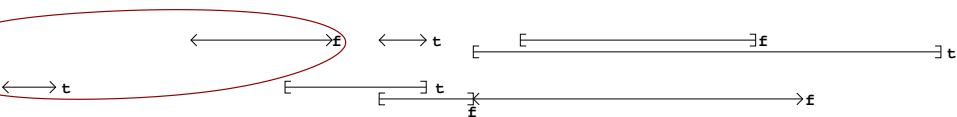


- 2 objectifs : terminer ! (tester $\subseteq \# \emptyset \#$) et analyse plus rapide

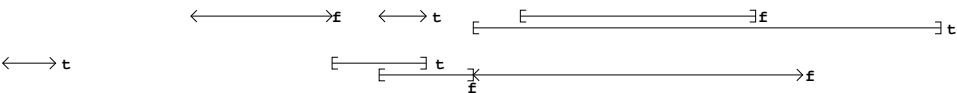
Une conjonction optimisée (linéaire!) pour les Contraintes



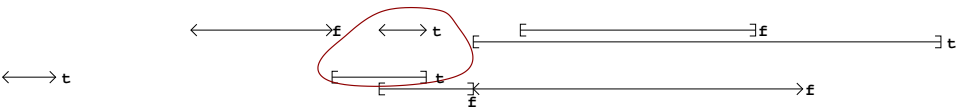
Une conjonction optimisée (linéaire!) pour les Contraintes



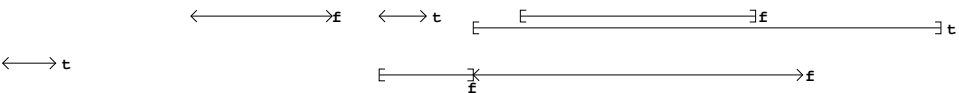
Une conjonction optimisée (linéaire!) pour les Contraintes



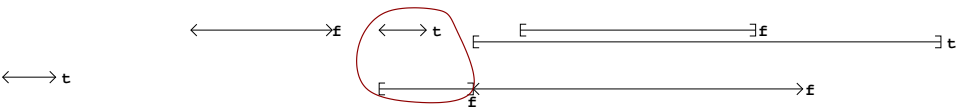
Une conjonction optimisée (linéaire!) pour les Contraintes



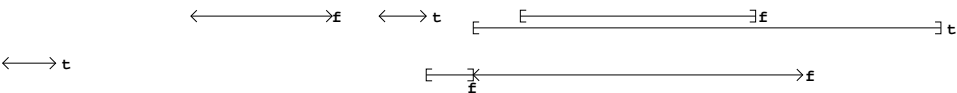
Une conjonction optimisée (linéaire!) pour les Contraintes



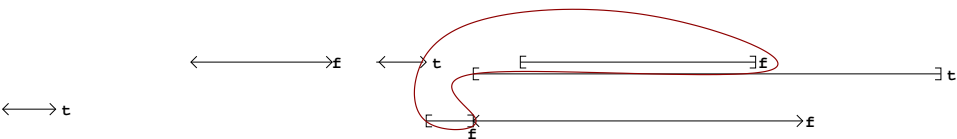
Une conjonction optimisée (linéaire!) pour les Contraintes



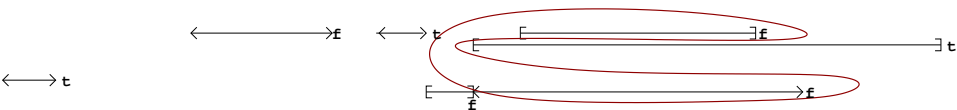
Une conjonction optimisée (linéaire!) pour les Contraintes



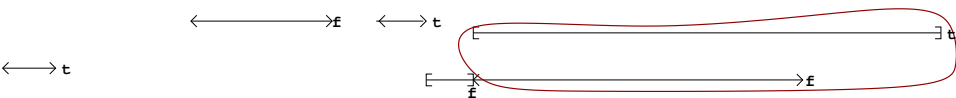
Une conjonction optimisée (linéaire!) pour les Contraintes



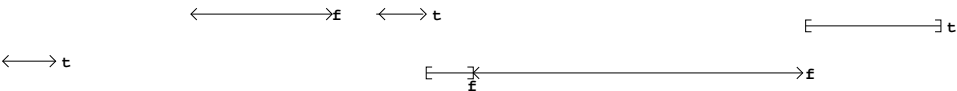
Une conjonction optimisée (linéaire!) pour les Contraintes



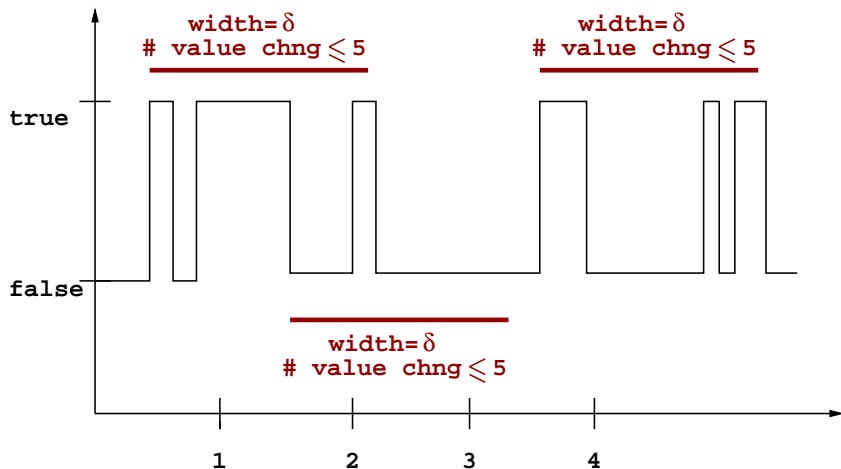
Une conjonction optimisée (linéaire!) pour les Contraintes



Une conjonction optimisée (linéaire!) pour les Contraintes

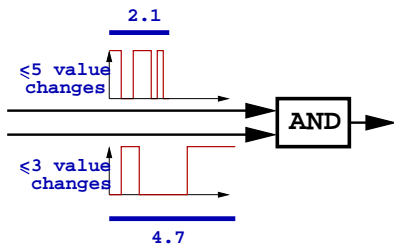


2ème Domaine Abstrait : domaine du comptage des changements de valeur

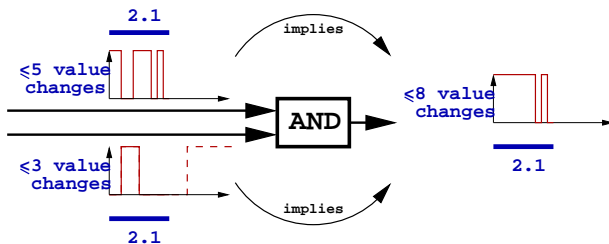


- Permet l'expression de spécifications de "stabilité".

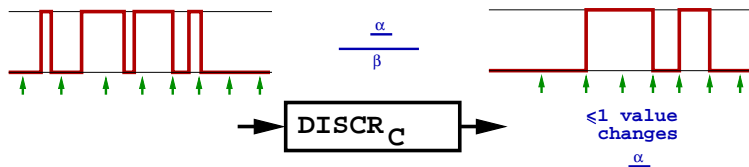
Un opérateur abstrait **non-temporel**



Un opérateur abstrait **non-temporel**

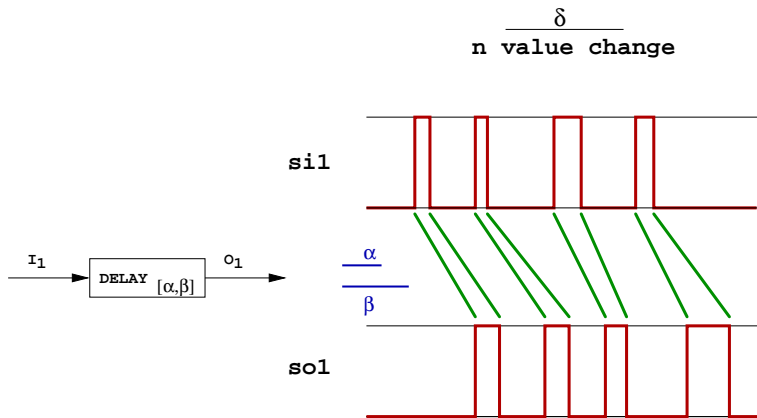


Un abstrait opérateur temporel

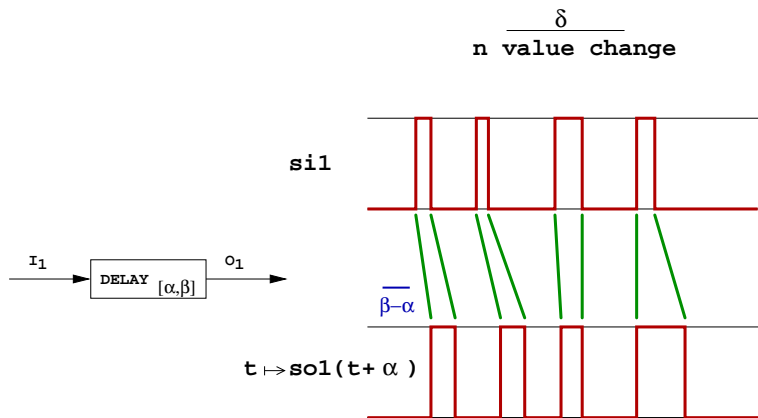


- $[\alpha, \beta]$ paramètre de horloge C
- $\forall A, \vec{\Psi}_{\text{DISCR}_{[\alpha, \beta]}}^{\#}(A) \triangleq (\leq 1, \alpha)$

Un abstrait opérateur temporel

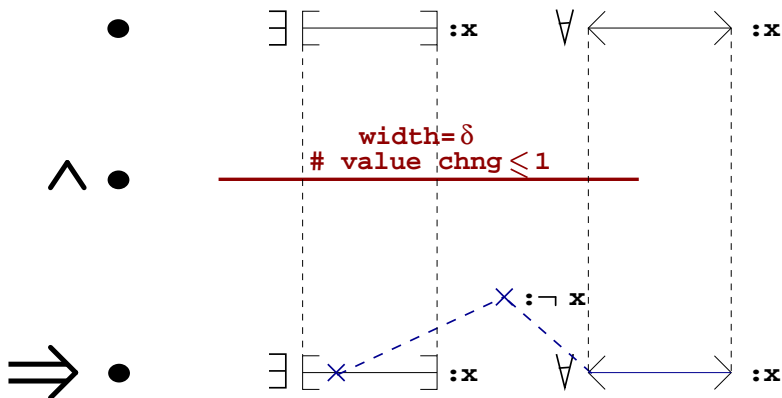


Un abstrait opérateur temporel



$$\vec{\Psi}^\#(n, \delta)_{\mathcal{N}} \triangleq (n, \delta - \beta + \alpha)$$

Produit Réduit Contraintes - Comptage des changements de valeur



Produit Réduit Contraintes - Comptage des changements de valeur

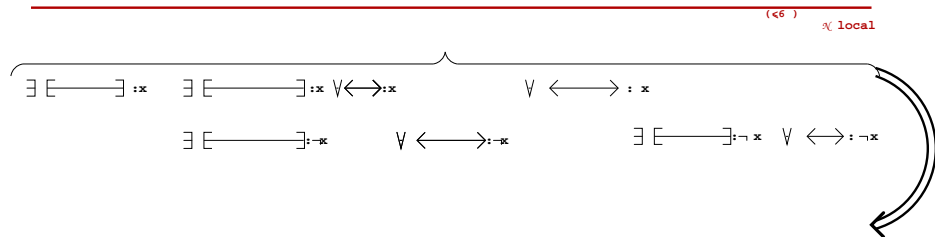
$$\bullet \quad \exists [\text{---}] : \mathbf{x} \quad \forall \langle \text{---} \rangle : \mathbf{x}$$

$$\wedge \bullet \quad \underline{\text{width} = \delta}$$

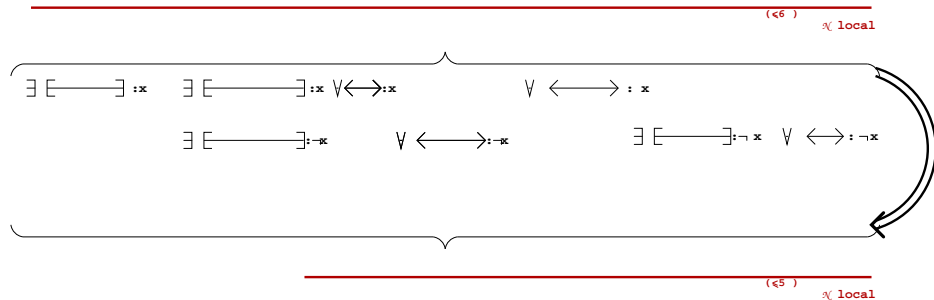
$$\underline{\# \text{ value chng} \leq 1}$$

$$\Rightarrow \bullet \quad \forall \langle \text{---} \rangle : \mathbf{x}$$

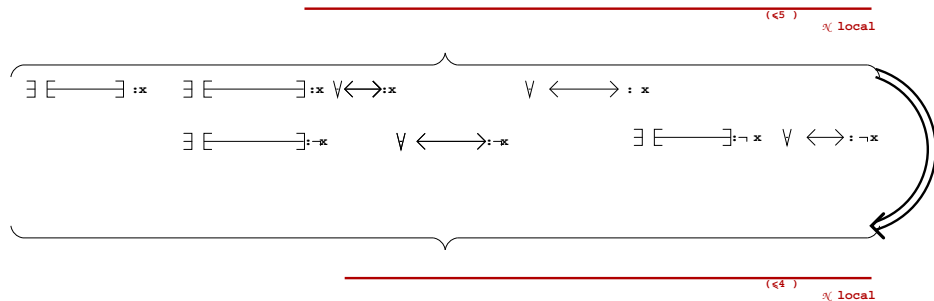
Réduction en temps linéaire : un exemple



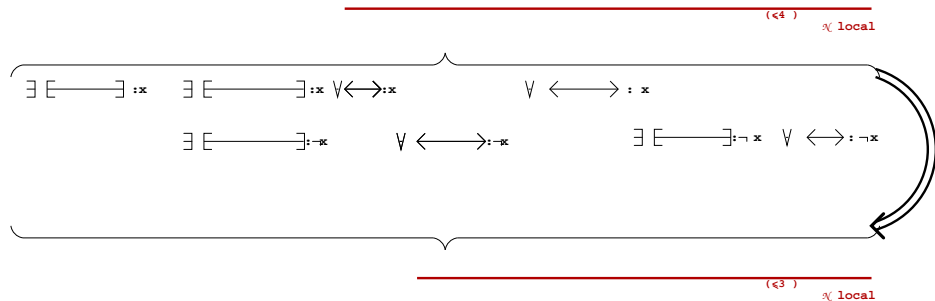
Réduction en temps linéaire : un exemple



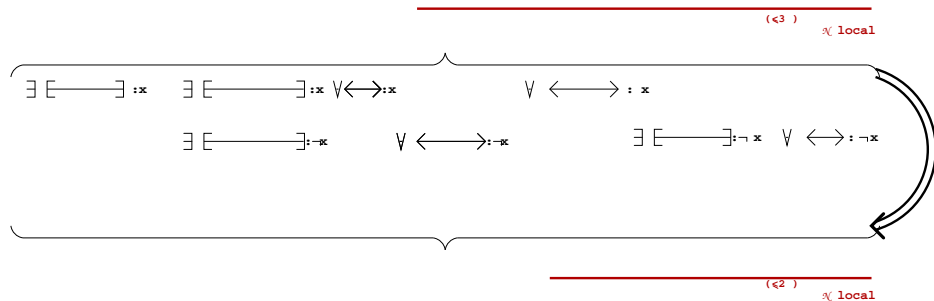
Réduction en temps linéaire : un exemple



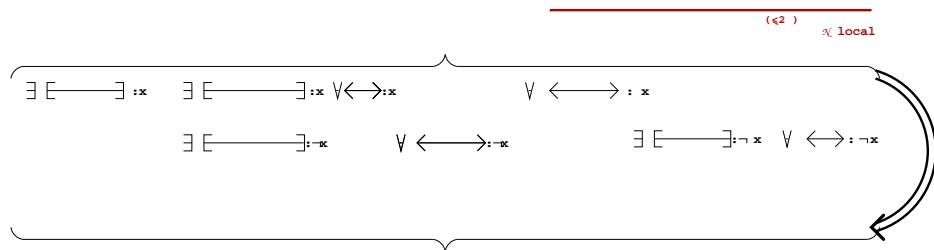
Réduction en temps linéaire : un exemple



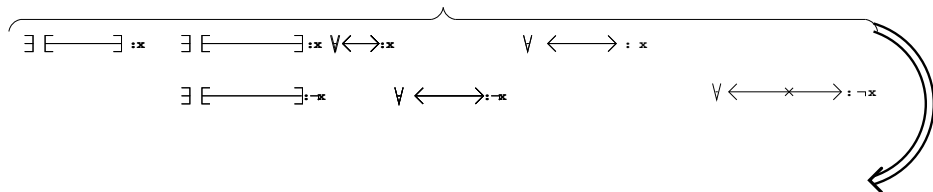
Réduction en temps linéaire : un exemple



Réduction en temps linéaire : un exemple



Réduction en temps linéaire : un exemple



Exemple de génération automatique

- Des générateurs **systematique de domaines** (optimisé temporellement pour la disjonction)

$$\exists [\text{---}] :x$$

$$\exists [\text{---}] :x$$

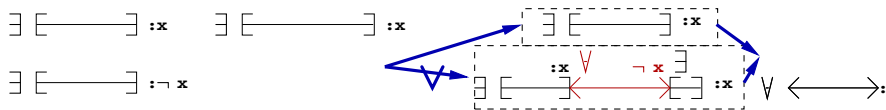
$$\exists [\text{---}] :x$$

$$\exists [\text{---}] : \neg x$$

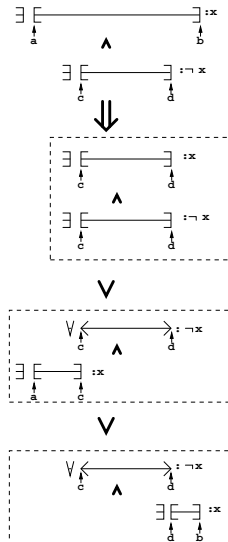
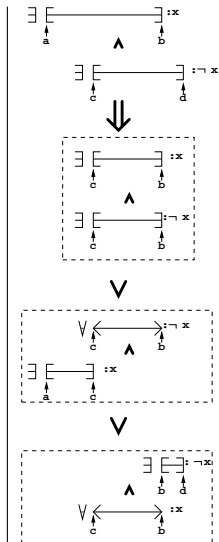
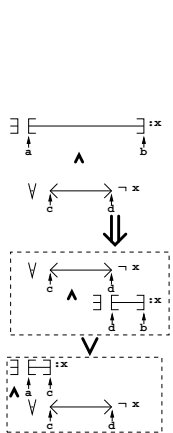
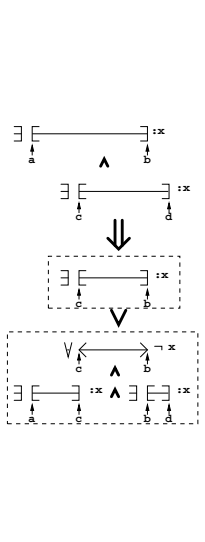
$$\exists [\text{---}] :x \vee \leftarrow \rightarrow :$$

Exemple de génération automatique

- Des générateurs **systematique de domaines** (optimisé temporellement pour la disjonction)

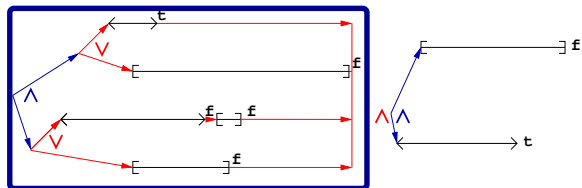
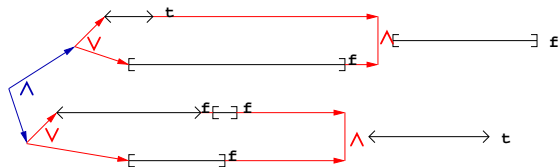


Produit réduit disjonctif : règles additionnelles



Domaines temporel : Conjonction

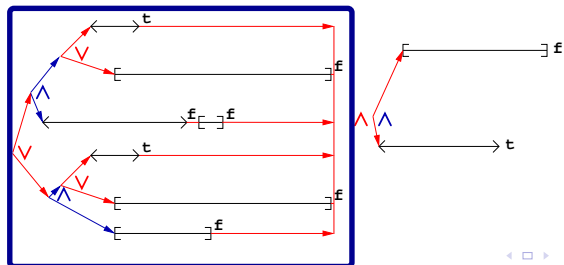
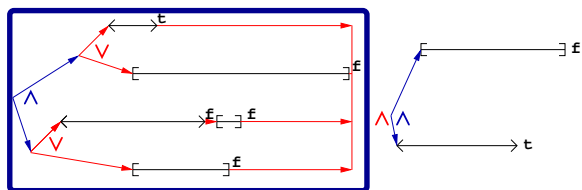
- Comment effectuer une conjonction de 2 éléments



Slicing temporel

Domaines temporel : Conjonction

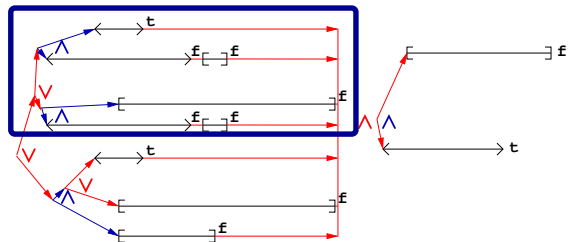
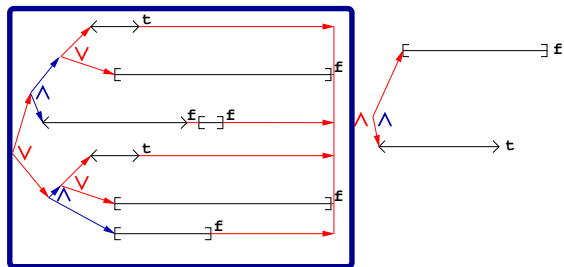
- Comment effectuer une conjonction de 2 éléments



Propagation

Domaines temporel : Conjonction

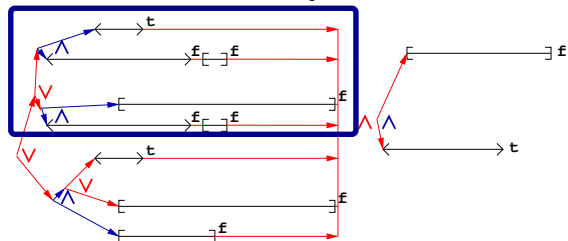
- Comment effectuer une conjonction de 2 éléments



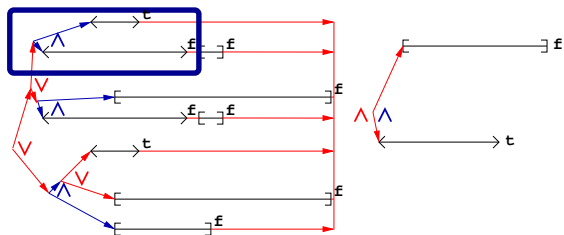
Propagation

Domaines temporel : Conjonction

- Comment effectuer une conjonction de 2 éléments

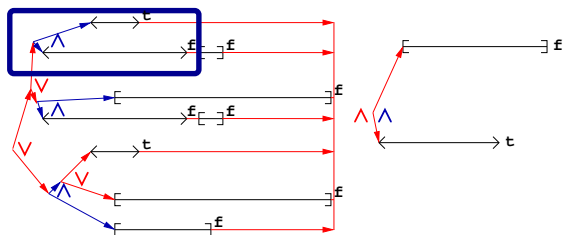


Slicing temporel

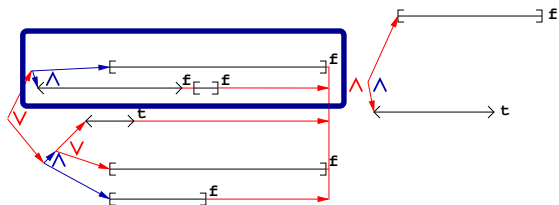


Domaines temporel : Conjonction

- Comment effectuer une conjonction de 2 éléments

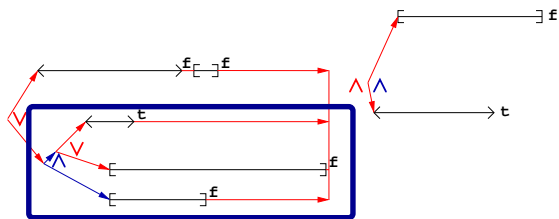


Interaction (\perp)

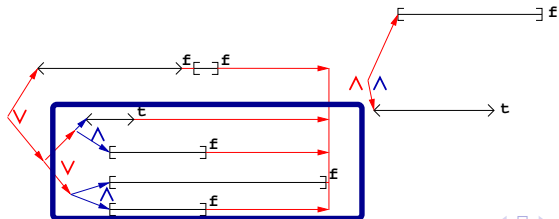


Domaines temporel : Conjonction

- Comment effectuer une conjonction de 2 éléments

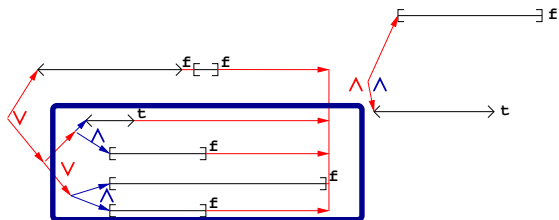


Propagation

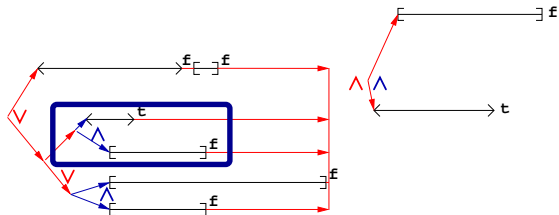


Domaines temporel : Conjonction

- Comment effectuer une conjonction de 2 éléments

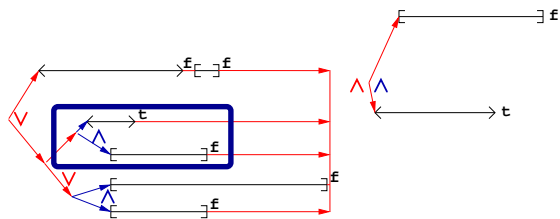


Slicing temporel

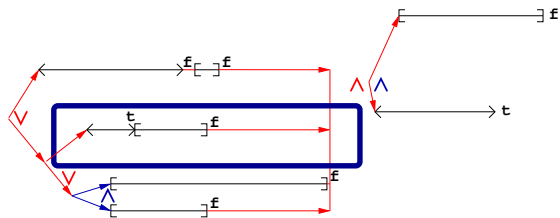


Domaines temporel : Conjonction

- Comment effectuer une conjonction de 2 éléments

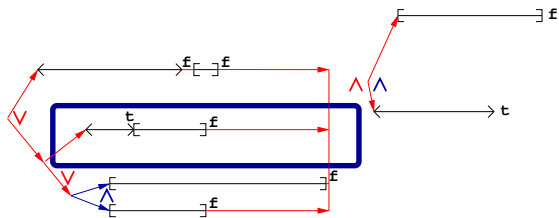


Interaction

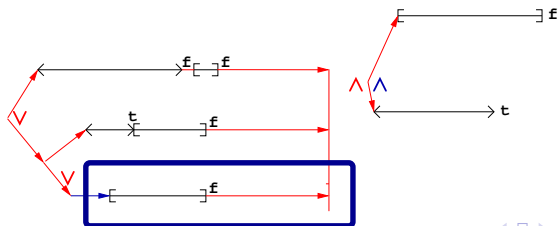


Domaines temporel : Conjonction

- Comment effectuer une conjonction de 2 éléments

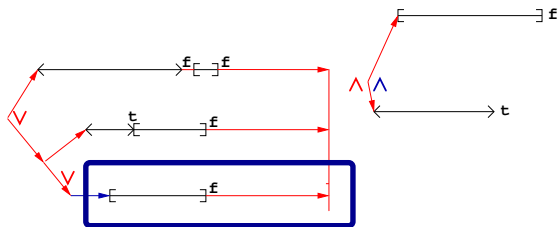


Interaction

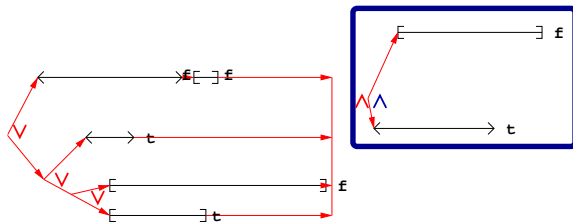


Domaines temporel : Conjonction

- Comment effectuer une conjonction de 2 éléments

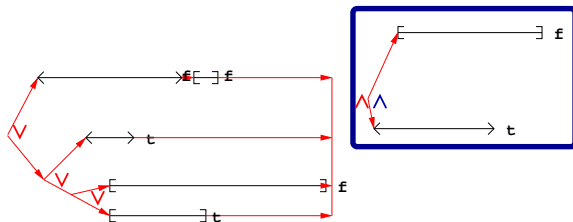


Slicing temporel

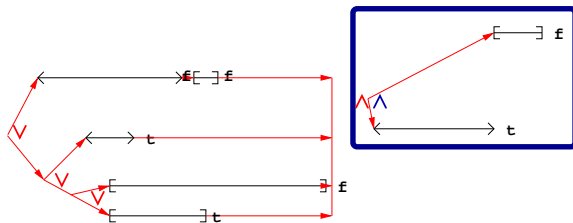


Domaines temporel : Conjonction

- Comment effectuer une conjonction de 2 éléments

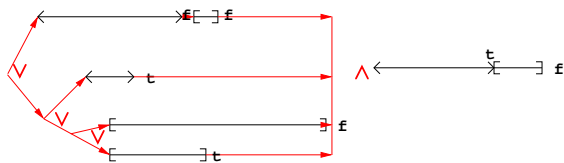
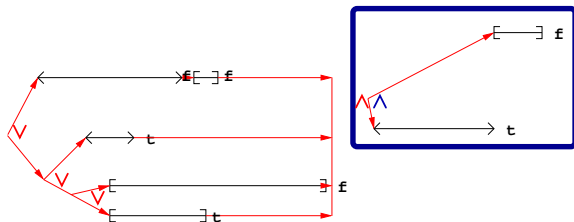


Interaction



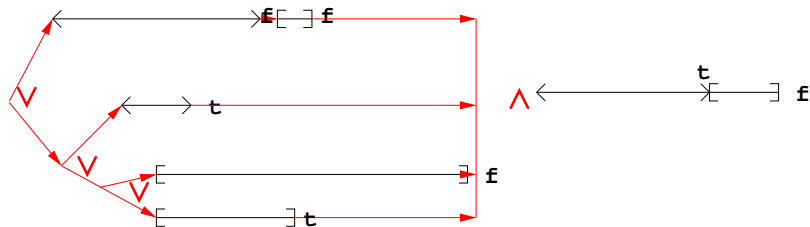
Domaines temporel : Conjonction

- Comment effectuer une conjonction de 2 éléments

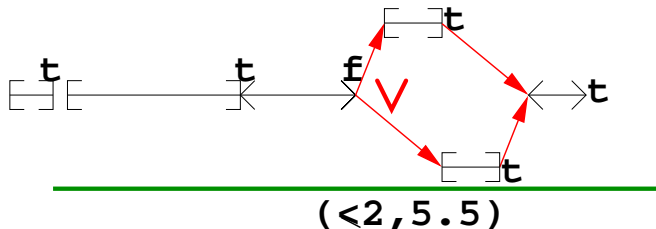


Domaines temporel : Conjonction

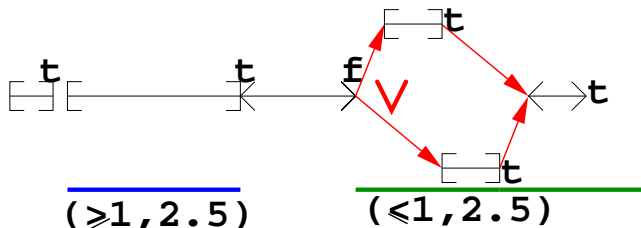
- Comment effectuer une conjonction de 2 éléments



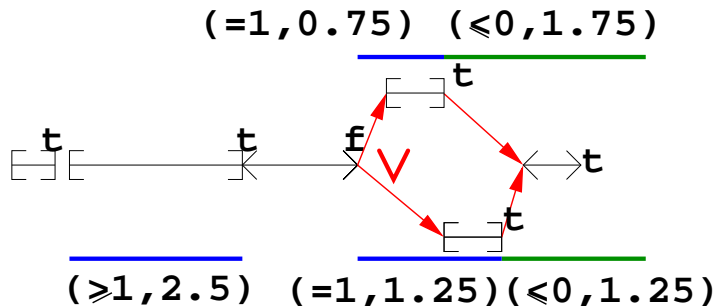
Produit Réduit Disjonctif Contraintes - Comptage des changements de valeurs : un exemple



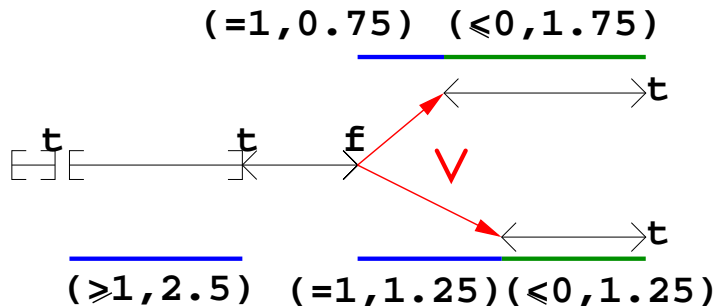
Produit Réduit Disjonctif Contraintes - Comptage des changements de valeurs : un exemple



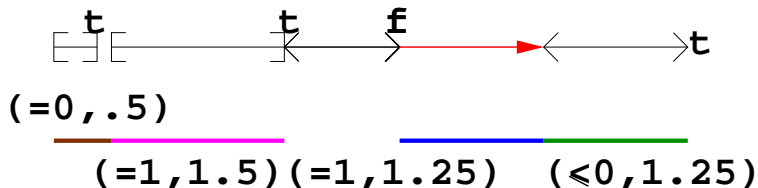
Produit Réduit Disjonctif Contraintes - Comptage des changements de valeurs : un exemple



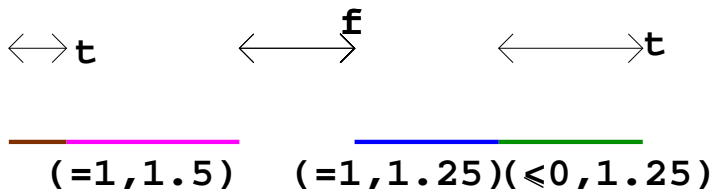
Produit Réduit Disjonctif Contraintes - Comptage des changements de valeurs : un exemple



Produit Réduit Disjonctif Contraintes - Comptage des changements de valeurs : un exemple

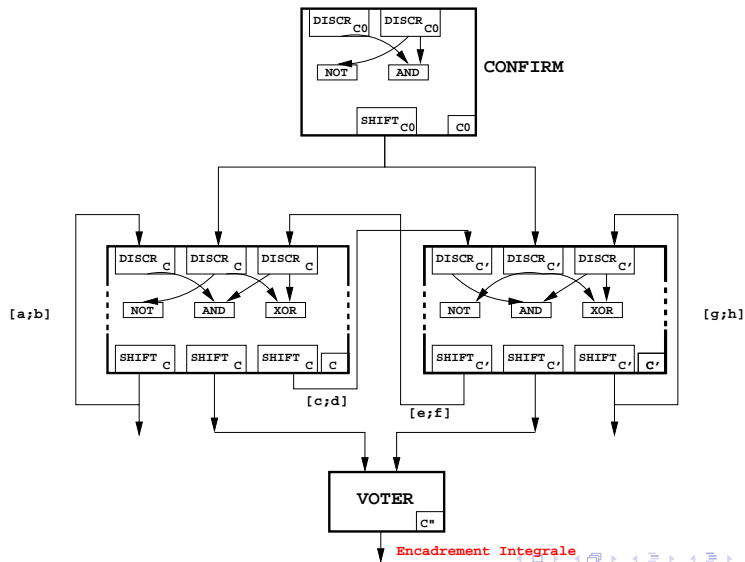


Produit Réduit Disjonctif Contraintes - Comptage des changements de valeurs : un exemple



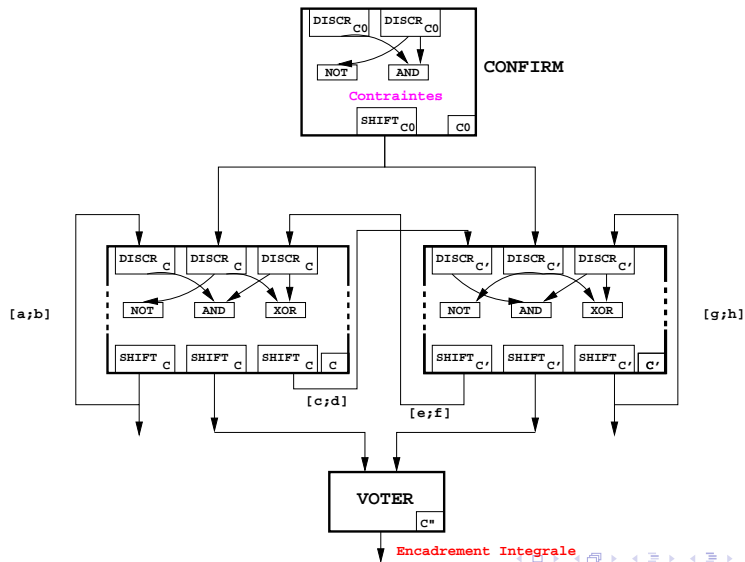
Collaboration entre domaines abstraits temporels

Construction d'une preuve : exemple de 2 unités redondantes



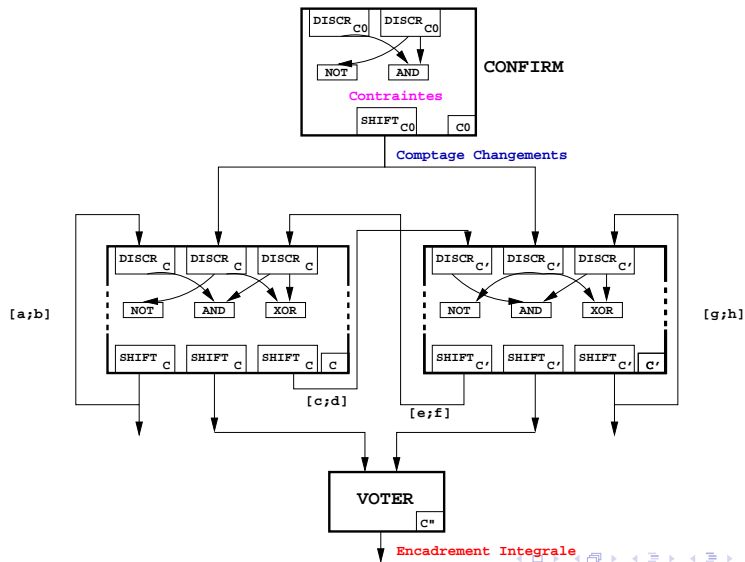
Collaboration entre domaines abstraits temporels

Construction d'une preuve : exemple de 2 unités redondantes



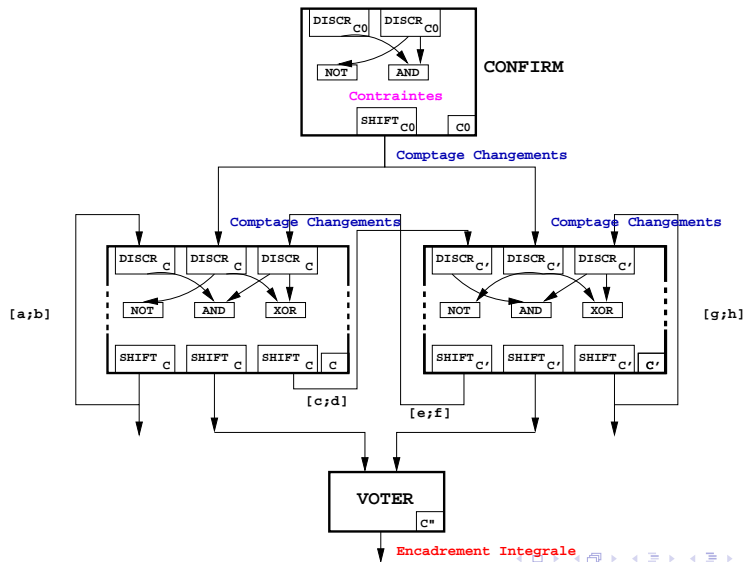
Collaboration entre domaines abstraits temporels

Construction d'une preuve : exemple de 2 unités redondantes



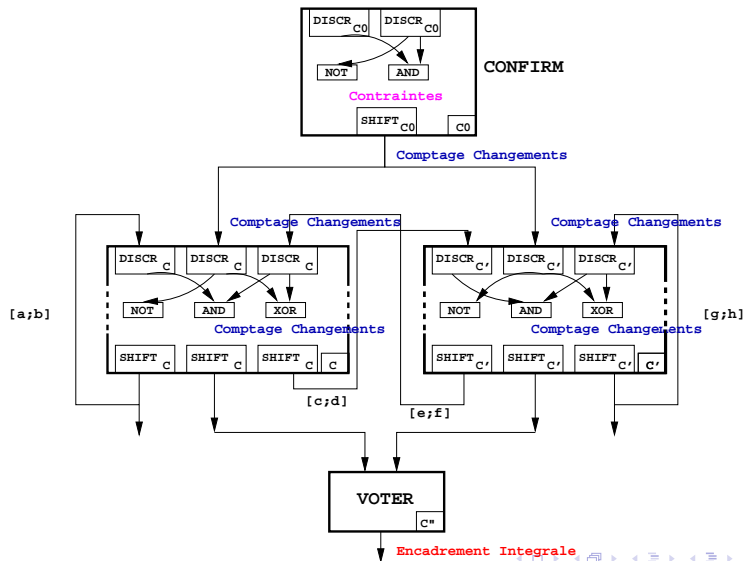
Collaboration entre domaines abstraits temporels

Construction d'une preuve : exemple de 2 unités redondantes



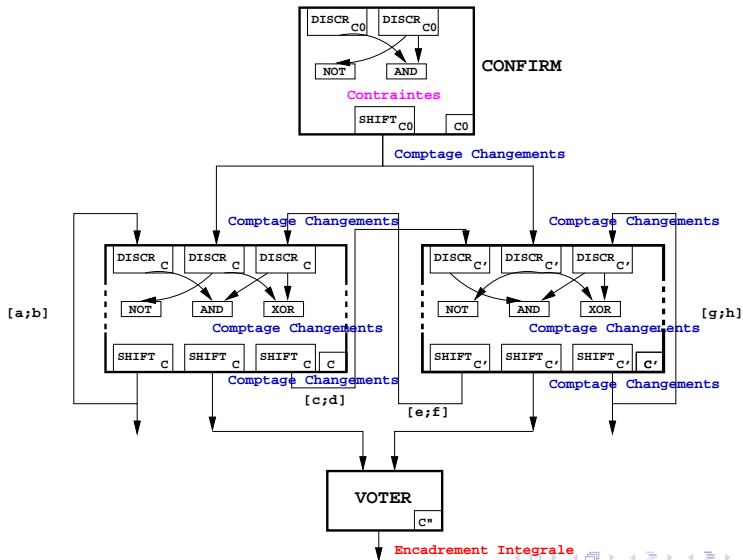
Collaboration entre domaines abstraits temporels

Construction d'une preuve : exemple de 2 unités redondantes



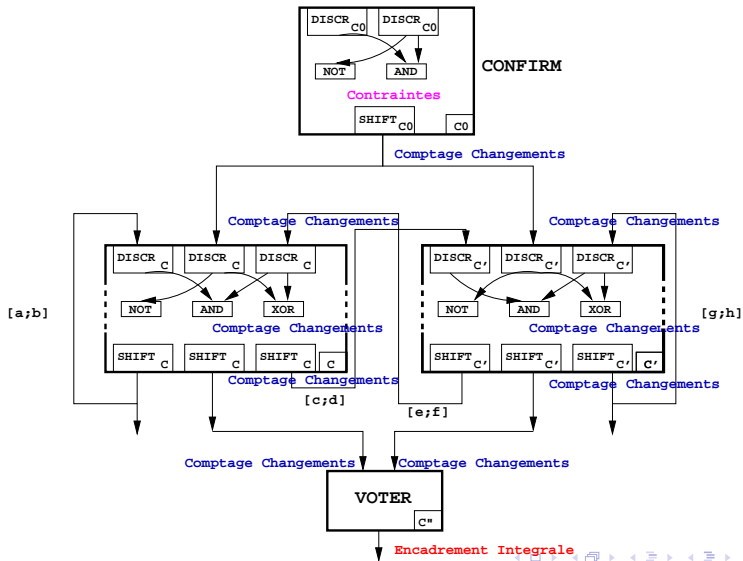
Collaboration entre domaines abstraits temporels

Construction d'une preuve : exemple de 2 unités redondantes



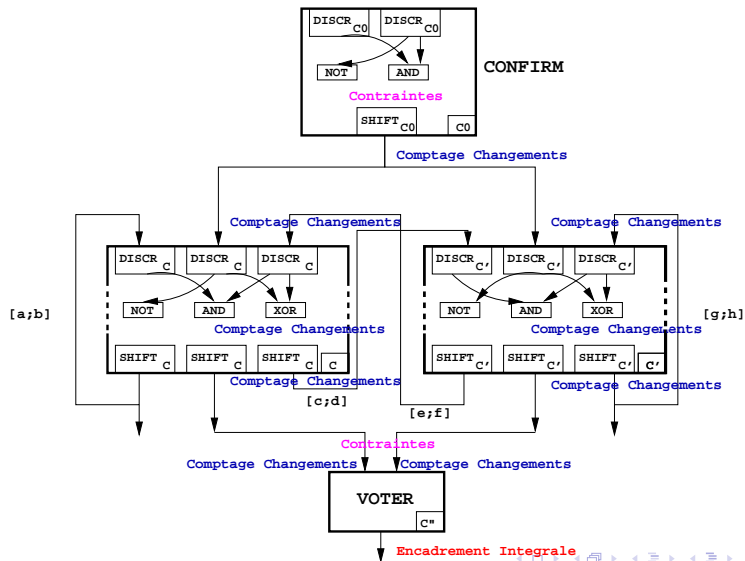
Collaboration entre domaines abstraits temporels

Construction d'une preuve : exemple de 2 unités redondantes



Collaboration entre domaines abstraits temporels

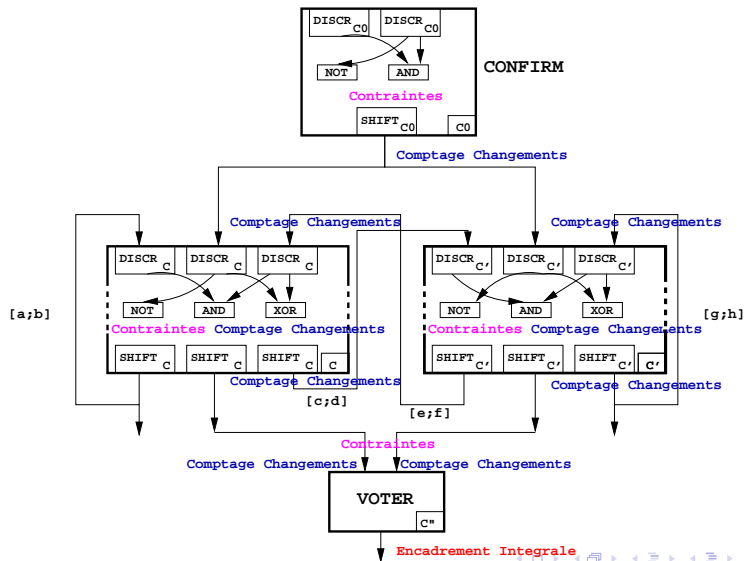
Construction d'une preuve : exemple de 2 unités redondantes



Encadrement Intégrale

Collaboration entre domaines abstraits temporels

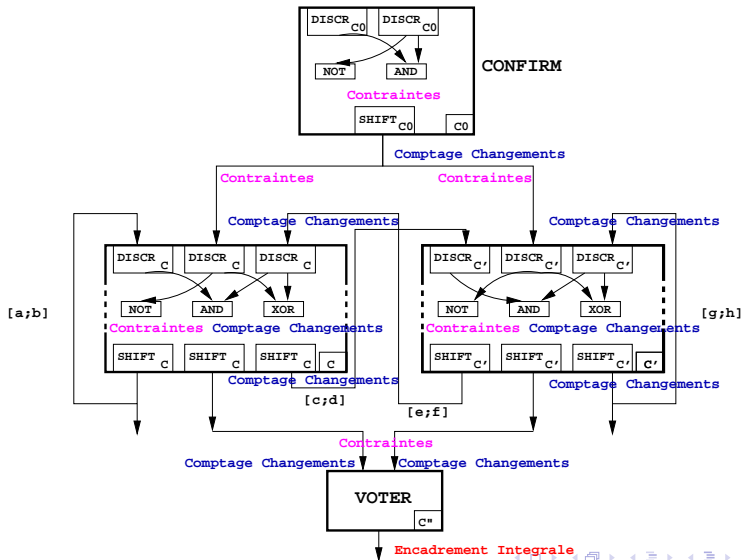
Construction d'une preuve : exemple de 2 unités redondantes



Encadrement Intégrale

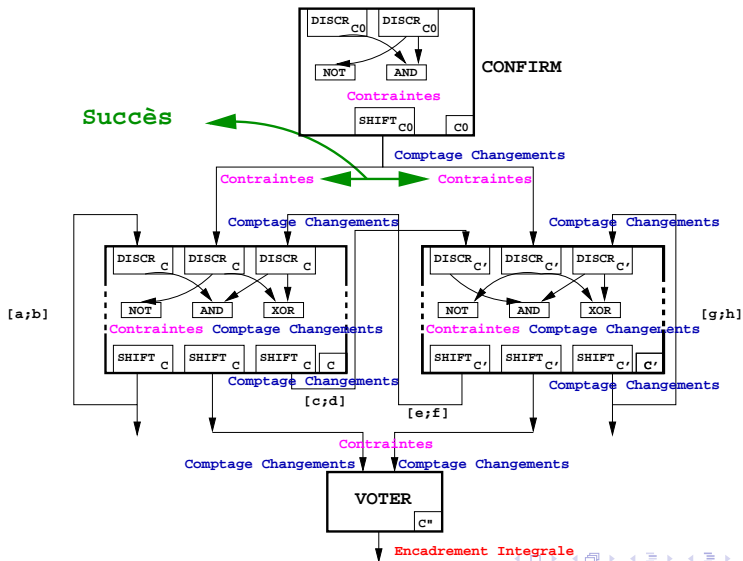
Collaboration entre domaines abstraits temporels

Construction d'une preuve : exemple de 2 unités redondantes



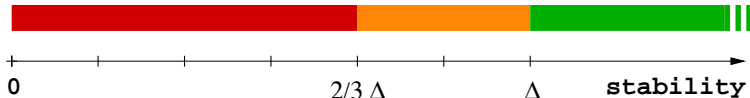
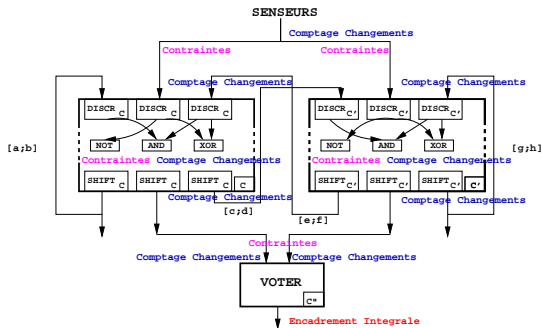
Collaboration entre domaines abstraits temporels

Construction d'une preuve : exemple de 2 unités redondantes



Analyse statique d'ordinateurs redondants (Ex. Airbus)

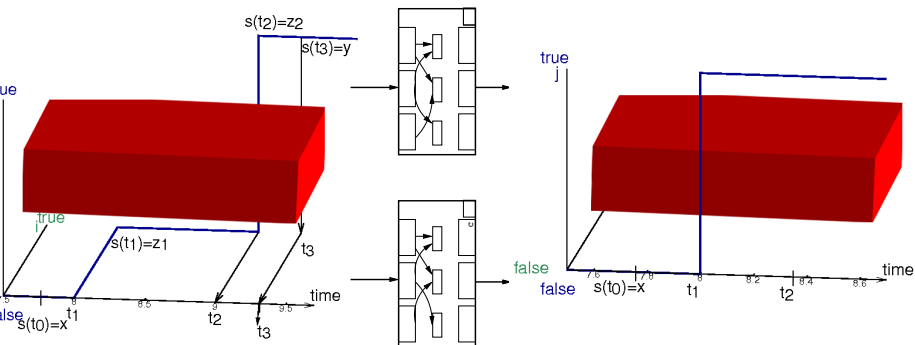
communiquant de façon imparfaite, avec dérive d'horloge



l'analyse pointe vers un contre-exemple | ? | l'analyse prouve la spécification

Un domaine abstrait relationnel

- Définir un domaine abstrait **relationnel** pour analyser les systèmes redondants.



Conclusion

Conclusion

- Les systèmes **synchrones communicants** peuvent avoir des horloges imparfaites et des canaux de communication non-instantanés.
 - ▶ Les **preuves of correction** sont alors très difficiles et doivent considérer des aspects temporels

Conclusion

- Les systèmes **synchrones communicants** peuvent avoir des horloges imparfaites et des canaux de communication non-instantanés.
 - ▶ Les **preuves of correction** sont alors très difficiles et doivent considérer des aspects temporels
- Nous définissons une **sémantique temps-continu**
 - ▶ La **sémantique temps continu** permet une abstraction rapide et précise.
 - ▶ Nous avons besoin de domaines abstraits **temporels** nouveaux.
 - ▶ Les mathématiques continues sont plus simples mais moins utilisées que les mathématiques discrètes !

Conclusion

- Les systèmes **synchrones communicants** peuvent avoir des horloges imparfaites et des canaux de communication non-instantanés.
 - ▶ Les **preuves of correction** sont alors très difficiles et doivent considérer des aspects temporels
- Nous définissons une **sémantique temps-continu**
 - ▶ La **sémantique temps continu** permet une abstraction rapide et précise.
 - ▶ Nous avons besoin de domaines abstraits **temporels** nouveaux.
 - ▶ Les mathématiques continues sont plus simples mais moins utilisées que les mathématiques discrètes !
- Un **produit réduit** peut améliorer la vitesse si les caractéristiques temporelles sont prises en considération.

Conclusion

- Les systèmes **synchrones communicants** peuvent avoir des horloges imparfaites et des canaux de communication non-instantanés.
 - ▶ Les **preuves of correction** sont alors très difficiles et doivent considérer des aspects temporels
- Nous définissons une **sémantique temps-continu**
 - ▶ La **sémantique temps continu** permet une abstraction rapide et précise.
 - ▶ Nous avons besoin de domaines abstraits **temporels** nouveaux.
 - ▶ Les mathématiques continues sont plus simples mais moins utilisées que les mathématiques discrètes !
- Un **produit réduit** peut améliorer la vitesse si les caractéristiques temporelles sont prises en considération.
- Les générateurs **systématique de domaines** (e.g. disjonction temporelle optimisée) doivent aussi être temporels.

Conclusion

- Les systèmes **synchrones communicants** peuvent avoir des horloges imparfaites et des canaux de communication non-instantanés.
 - ▶ Les **preuves of correction** sont alors très difficiles et doivent considérer des aspects temporels
- Nous définissons une **sémantique temps-continu**
 - ▶ La **sémantique temps continu** permet une abstraction rapide et précise.
 - ▶ Nous avons besoin de domaines abstraits **temporels** nouveaux.
 - ▶ Les mathématiques continues sont plus simples mais moins utilisées que les mathématiques discrètes !
- Un **produit réduit** peut améliorer la vitesse si les caractéristiques temporelles sont prises en considération.
- Les générateurs **systématique de domaines** (e.g. disjonction temporelle optimisée) doivent aussi être temporels.
- Les domaines ne doivent pas être trop expressifs : il faut trouver un **équilibre** entre précision, vitesse et occupation mémoire (défini par des contraintes temporelles?).

Questions ?

Transparents : www.di.ens.fr/~bertrane/

Contact : bertrane@di.ens.fr