Vérification de propriétés réactives sur des macro-définitions.

Guillaume Capron

École Polytechnique

20 Septembre 2005

1. Introduction.
2. Formalisation de la spécification.
3. Abstraction de l'ensemble d'états.
4. Sémantique.
5. Représentation des ensembles de traces.

But

Vérification des **propriétés réactives** de symboles utilisés dans SCADE pour construire des programmes réactifs.

But

Vérification des **propriétés réactives** de symboles utilisés dans SCADE pour construire des programmes réactifs.

Moyen

analyse statique par interprétation abstraite en utilisant une représentation compacte de la sémantique.

But

Vérification des **propriétés réactives** de symboles utilisés dans SCADE pour construire des programmes réactifs.

Moyen

analyse statique par interprétation abstraite en utilisant une représentation compacte de la sémantique.

Structure d'un programme réactif (écrit en C)

initialisation des variables
while (1) {
...
symbols $(I_1, \ldots, I_i, O_1, \ldots, O_j, K_1, \ldots, K_m)$;
...
}

- Propriétés réactives décrites dans une documentation

- Propriétés réactives décrites dans une documentation
- Variables d'entrées/sorties booléennes

- Propriétés réactives décrites dans une documentation
- Variables d'entrées/sorties booléennes
- Aucune connaissance de l'environnement d'exécution de la macro

- Propriétés réactives décrites dans une documentation
- Variables d'entrées/sorties booléennes
- Aucune connaissance de l'environnement d'exécution de la macro

Nous devons vérifier le diagramme suivant

$$\begin{array}{ccc} S & \supseteq & C \\ \| & & \cap| \\ S^\sharp & \supseteq & C^\sharp \end{array}$$

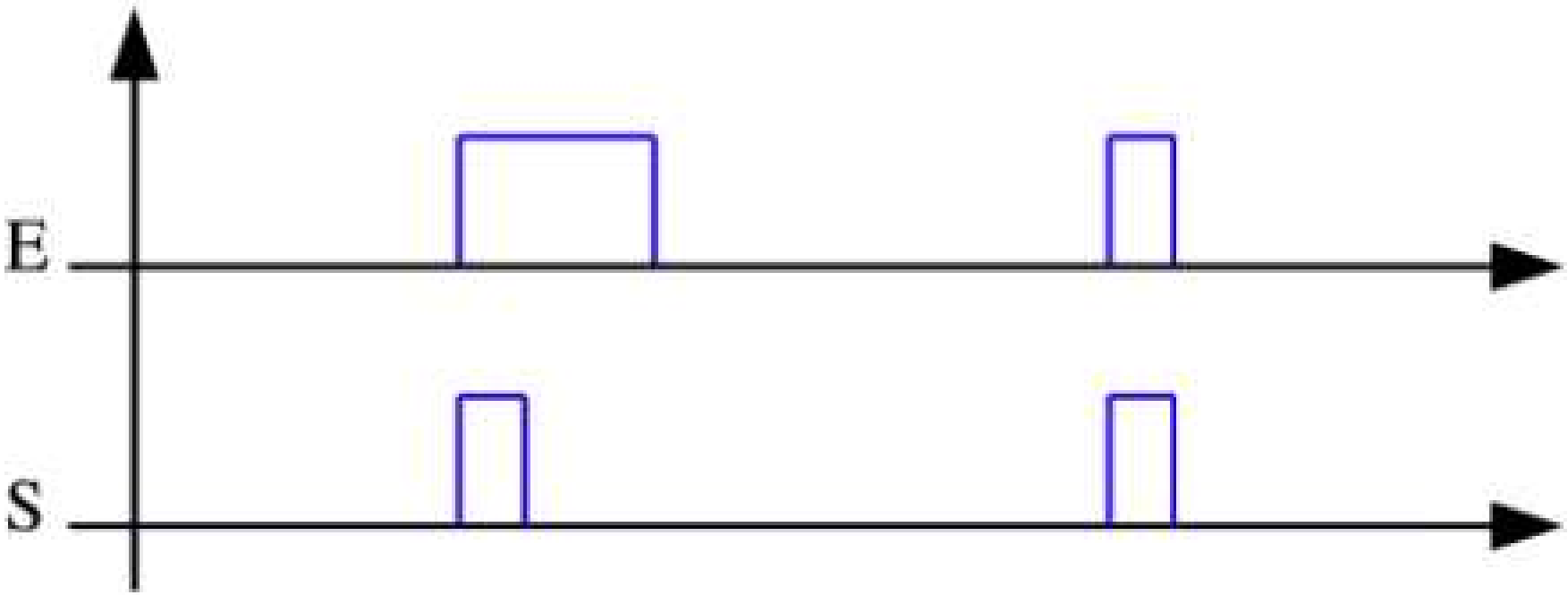
Exemple

```
1 #define pulse (bool E, bool S) {
2     static bool R;
3     S = E & R;
4     R = !E;
5  }
```

Sa spécification, donnée par la documentation, peut s'écrire de la manière suivante :

```
1 #define pulse (bool E, bool S) {
2    S = E & !PRE E;
3  }
```

Exemple



Extension du langage C

$$op[v_1, v_2]\{exp\}$$

où $op \in \{\vee, \wedge\}$, v_1 et $v_2 \in \mathbb{N}$ et exp est une expression booléenne.

Extension du langage C

$$op[v_1, v_2]\{exp\}$$

où $op \in \{\vee, \wedge\}$, v_1 et $v_2 \in \mathbb{N}$ et exp est une expression booléenne.

Sémantique

$$[\![\vee[v_1, v_2]\{P\}]\!]_{t,\sigma} = 0 \iff \forall i \in [v_1, v_2], [\![P]\!]_{t-i,\sigma} = 0$$

$$[\![\wedge[v_1, v_2]\{P\}]\!]_{t,\sigma} = 0 \iff \exists i \in [v_1, v_2], [\![P]\!]_{t-i,\sigma} = 0$$

Introduction de deux alias

$$\mathrm{PRE}[v]\{X\} \Longleftrightarrow op[v, v]\{X\} \text{ où } X \in \mathbb{V}$$

$$\mathrm{PRE}\{X\} \Longleftrightarrow op[1, 1]\{X\} \text{ où } X \in \mathbb{V}$$

Introduction de deux alias

$$\mathrm{PRE}[v]\{X\} \Longleftrightarrow op[v, v]\{X\} \text{ où } X \in \mathbb{V}$$

$$\mathrm{PRE}\{X\} \Longleftrightarrow op[1, 1]\{X\} \text{ où } X \in \mathbb{V}$$

Sémantique

$$[\![PRE\ X]\!]_{t,\sigma} \Longleftrightarrow [\![X]\!]_{t-1,\sigma}$$

$$[\![PRE[v]\ X]\!]_{t,\sigma} \Longleftrightarrow [\![X]\!]_{t-v,\sigma}$$

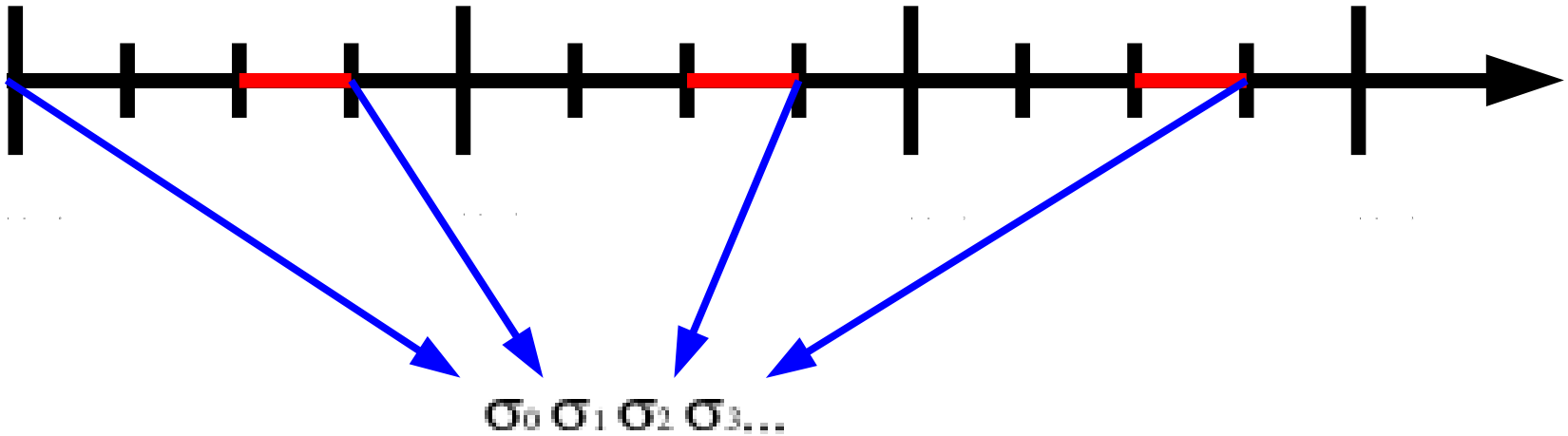
On peut exprimer la spécification suivante :
"La sortie S est vraie si et seulement si l'entrée E a été vraie pendant n cycles"

$$S = \wedge[0, n-1]\{E\}$$

ainsi que la spécification suivante :
"La sortie S est vraie si et seulement si l'entrée E a été vraie au cycle précédent"

$$S = \vee[1,1]\{E\} = PRE[1]\{E\} = PRE\{E\}$$

Etats



Etats concrets

Soit n_1, le nombre de variables entières/réelles du symbole.
Soit n_2 le nombre de variables booléennes du symbole.
Soit $\vec{S_Q} \in \mathbb{Q}^{n_1}$.
Soit $\vec{S_B} \in \mathbb{B}^{n_2}$.
Soit $t \in \mathbb{N}$ le nombre de cycles.

Etat concret

$$(\vec{S_Q}, \vec{S_B})_t$$

Etats abstraits

Soit n_1, le nombre de variables entières/réelles du symbole
Soit n_2 le nombre de variables booléennes du symbole
Soit P le polyèdre associé aux variables du symbole
Soit $\vec{S_B}$ le vecteur des variables du symbole à valeurs dans $\mathbb{B}$
Soit $Poly_X$ le polyèdre dont les variables appartiennent à X

Etat abstrait

$$S^\sharp = (\vec{S_B}, P) \in (\mathbb{B}^{n_2} \rightarrow Poly_X)$$

Fonction de concrétisation

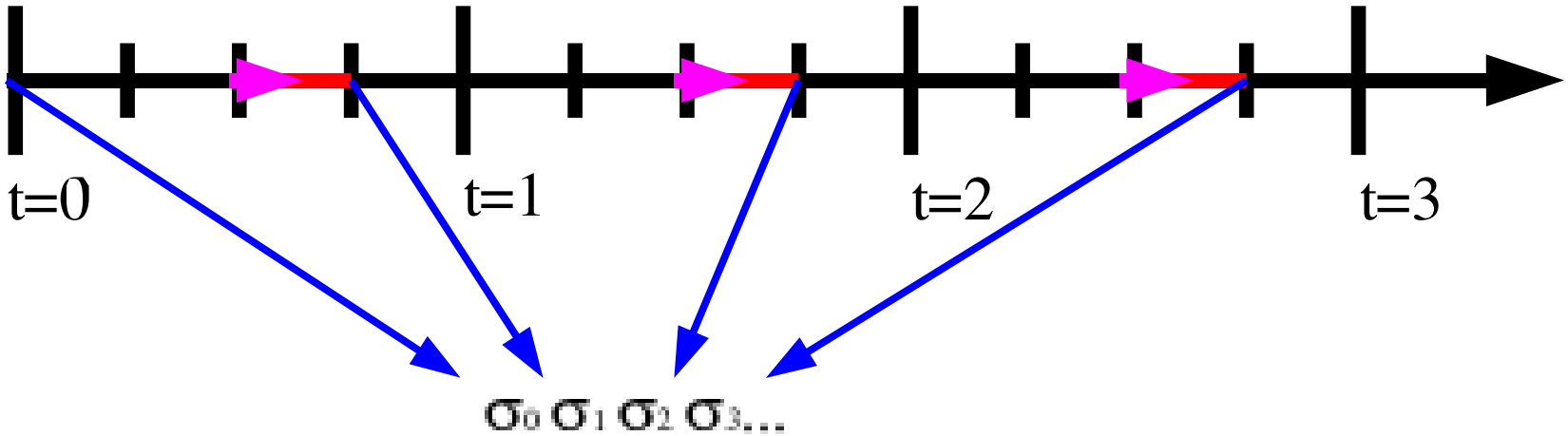
Soit $\gamma(\vec{S^\sharp_Q})$, la fonction de concrétisation pour les polyèdres.

$$\gamma : AStates \longrightarrow \mathcal{P}(CStates)$$

Fonction

$$\gamma(\vec{S^\sharp_B}, \vec{S^\sharp_Q}) = \{(\vec{S_B}, \vec{S_Q})_t | \vec{S_Q} \in \gamma(\vec{S^\sharp_Q}) \wedge \vec{S_B} = \vec{S^\sharp_B} \wedge t \in \mathbb{N}\}$$

Etats accessibles

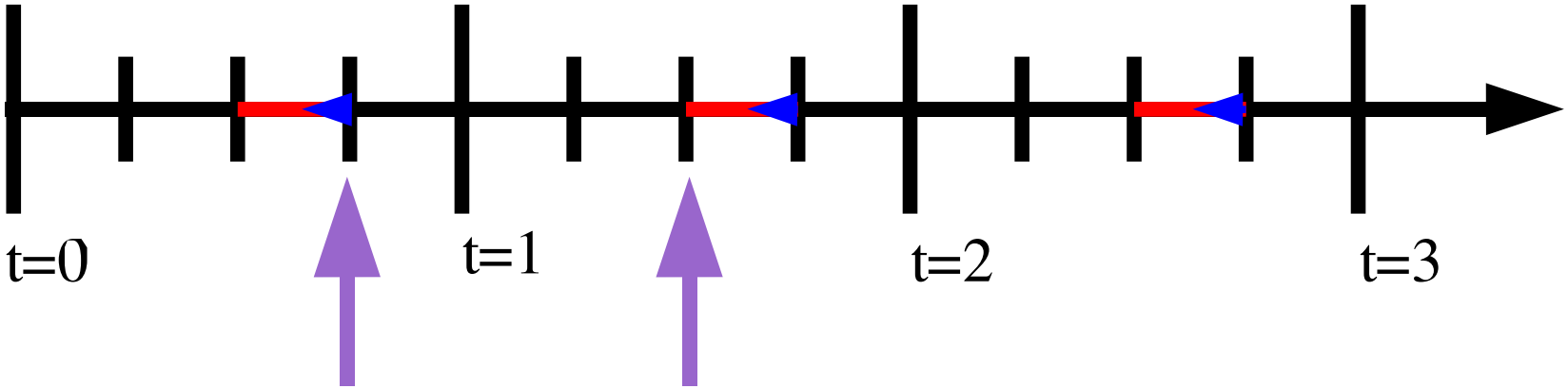


Soit $Init \subseteq S^\sharp$, l'ensemble des états initiaux du symbole.

Ensemble des états accessibles

$$Accessible = lfp(\lambda X.(Init \sqcup post^\sharp(X)))$$

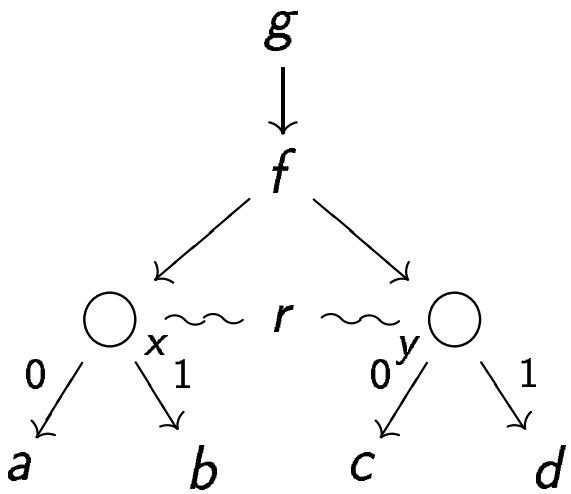
Sémantique

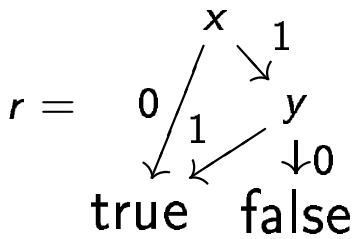


Soit *AInit*, l'ensemble des états abstraits initiaux. Soit *RAStates*, l'ensemble des états abstraits accessibles.

Sémantique

$$Sem_{macro} = \left\{ \overleftarrow{\sigma} \;\middle|\; \begin{array}{c} \sigma_0 \in AInit \wedge \\ |\sigma| = n \in \mathbb{N} \wedge \\ \forall i \in [1, n[, \sigma_i \in RAStates \wedge \\ \forall i \in [1, n[, \sigma_i \in post^\sharp(\sigma_{i-1}) \end{array} \right\}$$





où la relation r est vraie ssi x est 0 ou x et y sont 1

But

Construire les schémas d'arbres correspondant à la spécification et à la macro.

But

Construire les schémas d'arbres correspondant à la spécification et à la macro.

Spécification

Doit être traduite dans le domaine abstrait.
Utilisation de schémas d'arbres pré-définis avec compteurs.

But

Construire les schémas d'arbres correspondant à la spécification et à la macro.

Spécification

Doit être traduite dans le domaine abstrait.
Utilisation de schémas d'arbres pré-définis avec compteurs.

Macro-définition

Utilisation des ensembles d'états et de la fonction prédécesseur.

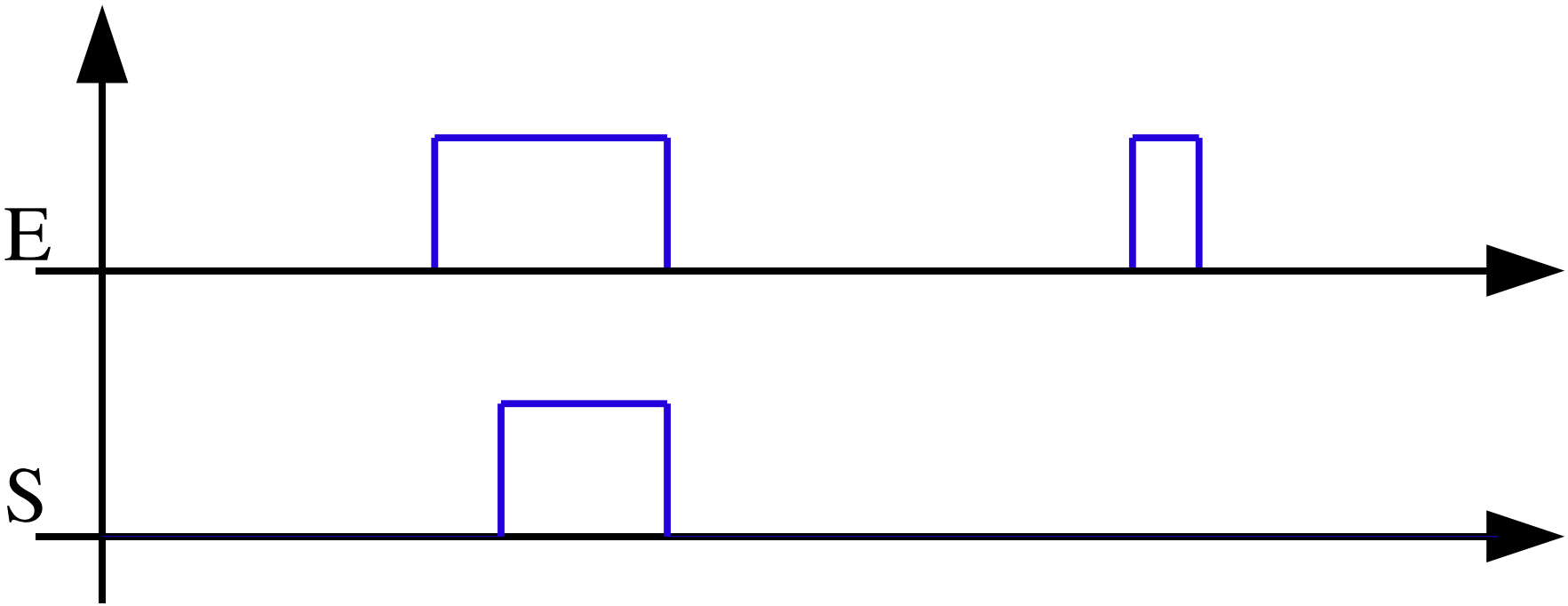
Exemple

```
1 #define Conf (bool E, int C, bool S) {
2    static int R;
3    if (E) R++;
4    else R = 0;
5    S = (bool)(R > C); }
```

Sa spécification :

```
1 #define Conf (bool E, int C, bool S) {
2 S = /\[0, C]{ E }; }
```

Exemple



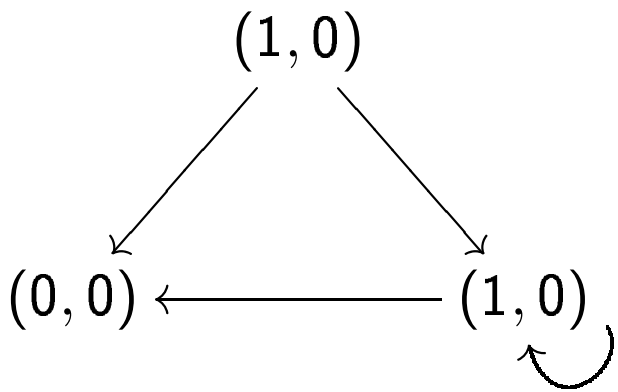
Nous obtenons l'ensemble d'états suivant ($\begin{pmatrix} E \\ S \end{pmatrix}$) :

$$\{ (\{R > C\}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}), (\{R = 0\}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}),$$
$$(\{R \leq C, R > 0\}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}) \}$$

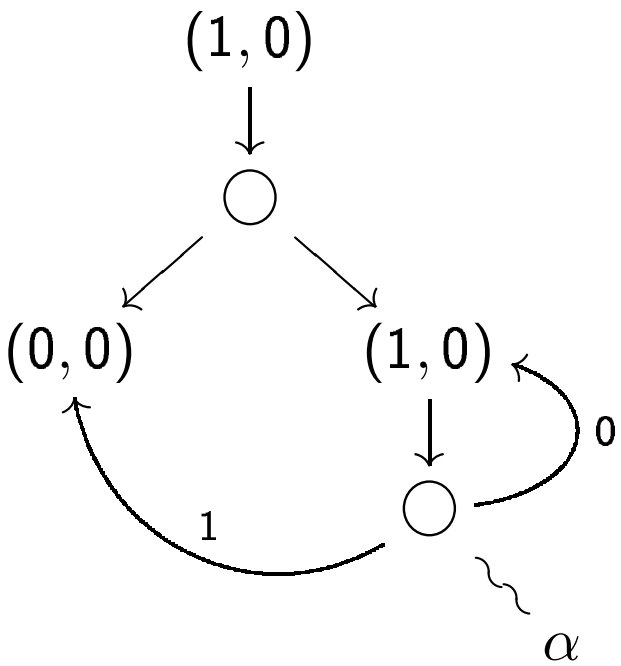
On s'intéresse au dernier état. $Pred^{\sharp}(\{R \leq C, R > 0\}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}) =$

$$\{(\{R \leq C, R > 0\}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}), (\{R = 0\}, \begin{pmatrix} 0 \\ 0 \end{pmatrix})\}$$

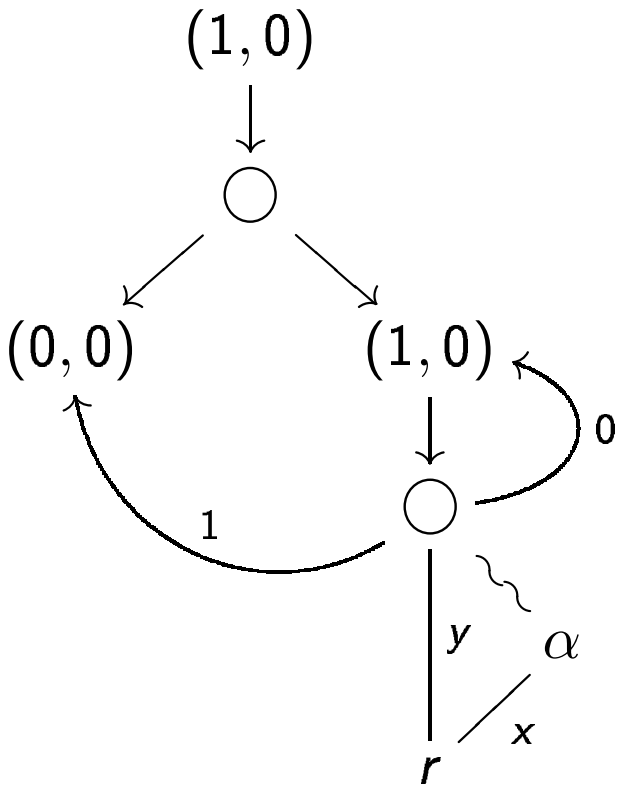
Graphe partiel du prédécesseur



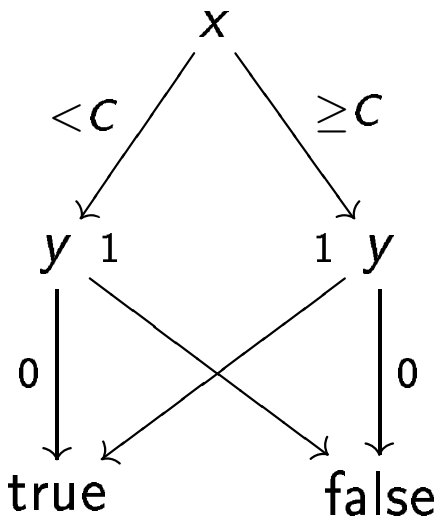
Ajout des noeuds de choix et compteurs



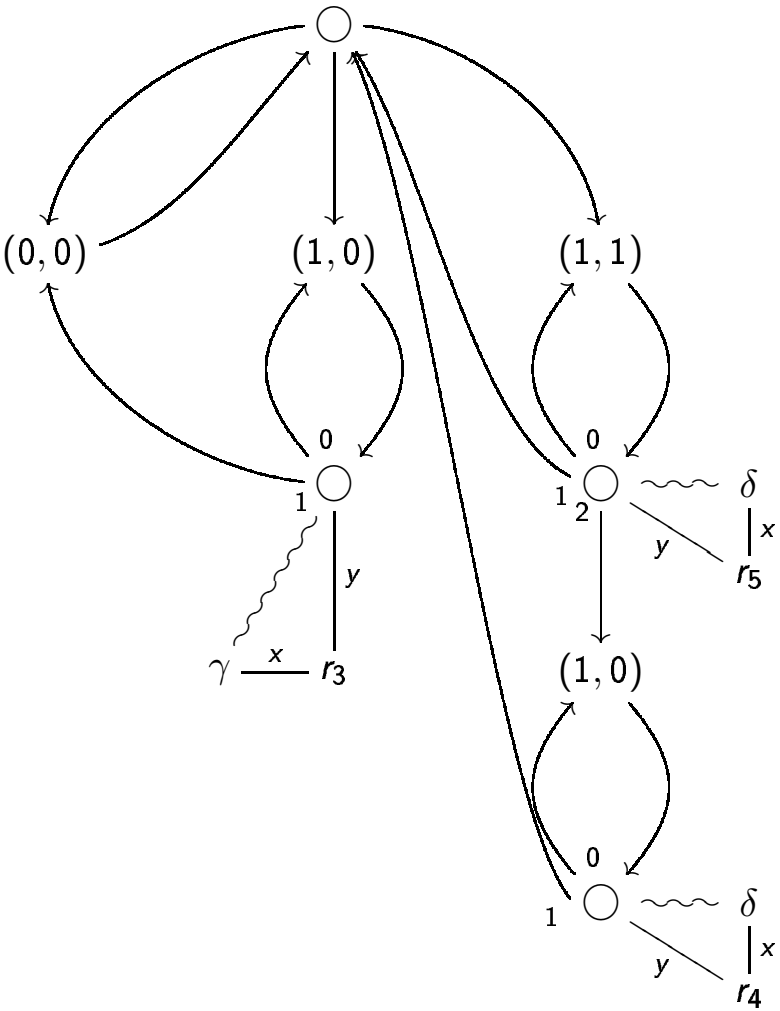
Ajout des relations

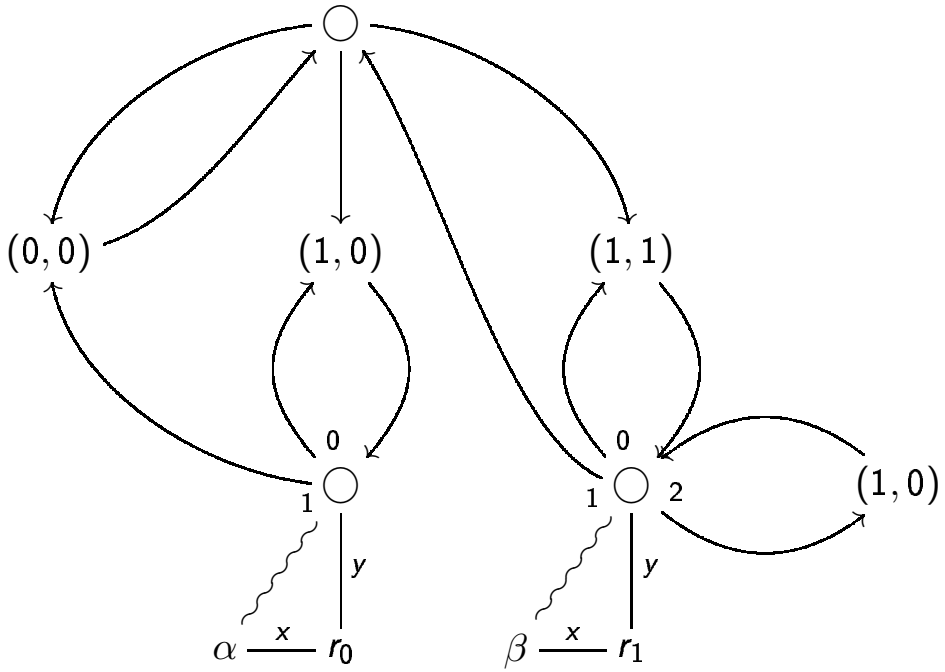


$$r(x,y) =$$



Union et test





ON VERRA ;-)