Analyses quantitatives de programmes

par génération d'invariants numériques

Laure Gonnord
http://laure.gonnord.org/pro/

Lille1 (USTL)/LIFL
Lille, France

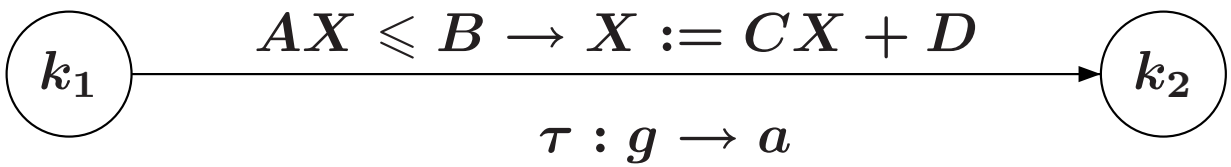
Contexte

Vérification de propriétés de programmes **à nombre d'états infini.**

- Propriétés de **sûreté**.
- Ensemble infini d'états : propriétés en général indécidables.
  - Décision interactive.
  - Classes restreintes + abstractions.
  - Classe indécidable ou trop coûteuse : approximations, ici vérification « conservative » (surapproximations).

Modèle - Notations

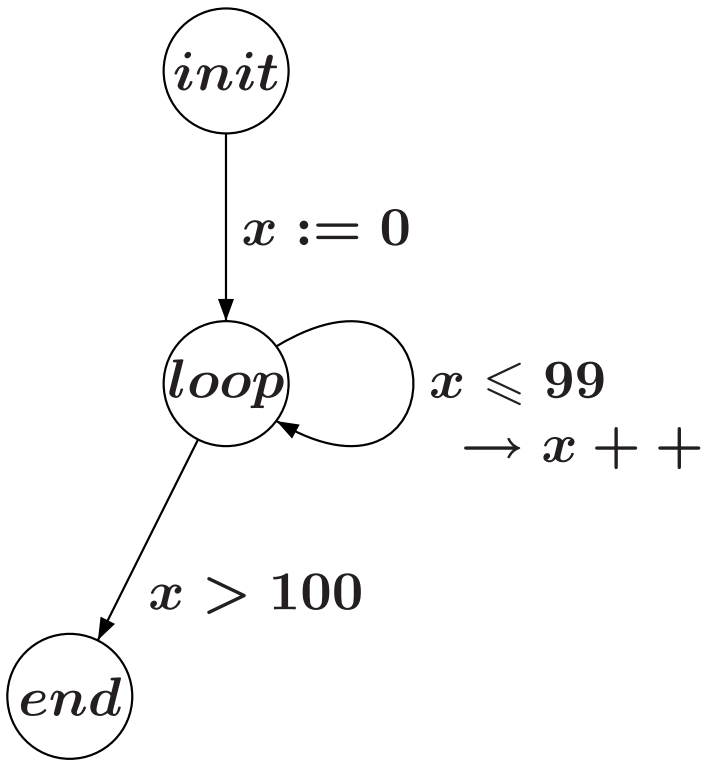
Vérification de propriétés **numériques** sur des graphes de flot de contrôle avec conditions et actions **affines** :



ou encore « automates interprétés », automates « à compteurs ».

- A, C matrices, B, D vecteurs.
- Sémantique « naturelle ».
- On veut des invariants pour **chaque point de contrôle**.

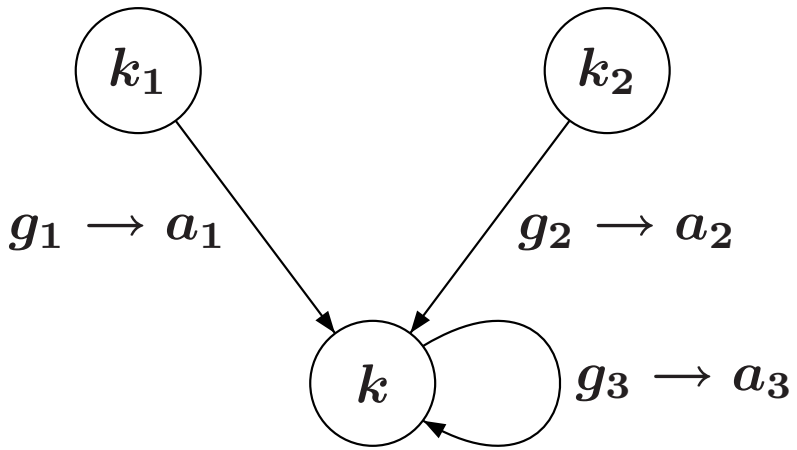
Modèle - Exemple



- $0 \leqslant x \leqslant 142$ est un **invariant** pour ”loop”.

Formalisation du problème

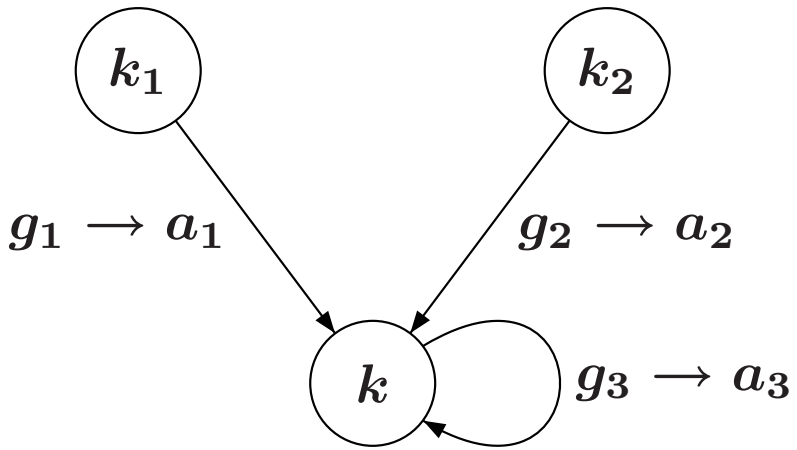
$\mathcal{R}_k$ = ensemble des **valuations** au point k :



$$\mathcal{R}_k = a_1(\mathcal{R}_{k_1} \cap g_1) \cup a_2(\mathcal{R}_{k_2} \cap g_2) \cup a_3(\mathcal{R}_k \cap g_3)$$

Formalisation du problème

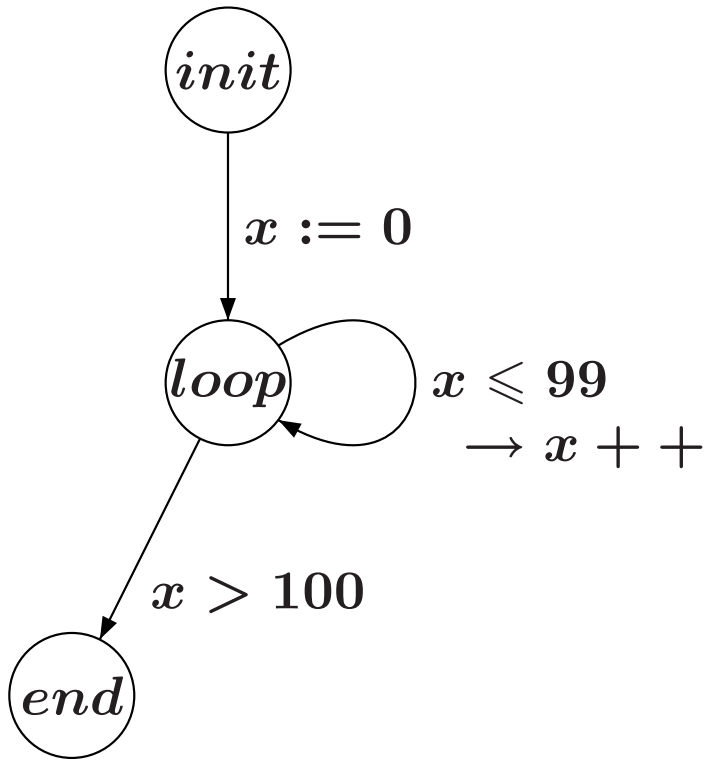
$\mathcal{R}_k$ = ensemble des **valuations** au point k :



$$\mathcal{R}_k = a_1(\mathcal{R}_{k_1} \cap g_1) \cup a_2(\mathcal{R}_{k_2} \cap g_2) \cup a_3(\mathcal{R}_k \cap g_3)$$

- Système d'équations $\mathcal{R}_k = F(\mathcal{R}_k)$, **point fixe**.

Point fixe - Exemple



$\mathcal{R}_{init} = \mathbb{R} = \top$, $R^1_{loop} = \{0\}$, puis $\{0, 1\}$, . . . le **plus petit** point fixe est $\{0, 1 \ldots 100\}$.

Calcul du point fixe

- Représentation des valuations, calcul

- Convergence de la résolution

Calcul du point fixe

- Représentation des valuations, calcul **polyèdres convexes** :

$$P_k = \text{ if } k = k_{init} \text{ then } \top else \bigsqcup_{(k,g,a,k')} a(P_{k'} \sqcap g)$$

- Convergence de la résolution

Calcul du point fixe

- Représentation des valuations, calcul **polyèdres convexes** :

$$P_k = \text{ if } k = k_{init} \text{ then } \top else \bigsqcup_{(k,g,a,k')} a(P_{k'} \sqcap g)$$

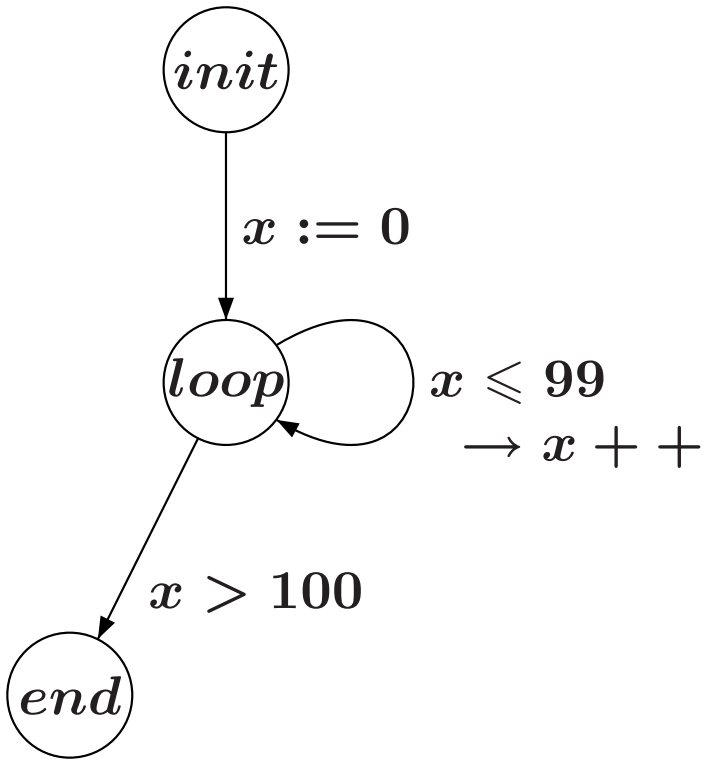
- Convergence de la résolution **utilisation d'un opérateur d'élargissement**, ie on remplace :

$$R_0, R_1 = F(R_0), R_2 = F(F(R_0)), \ldots \text{non convergente}$$

par

$$P_0, P_1 = P_0 \nabla F(P_0), P_1 \nabla F(P_1) \ldots \text{convergente}$$

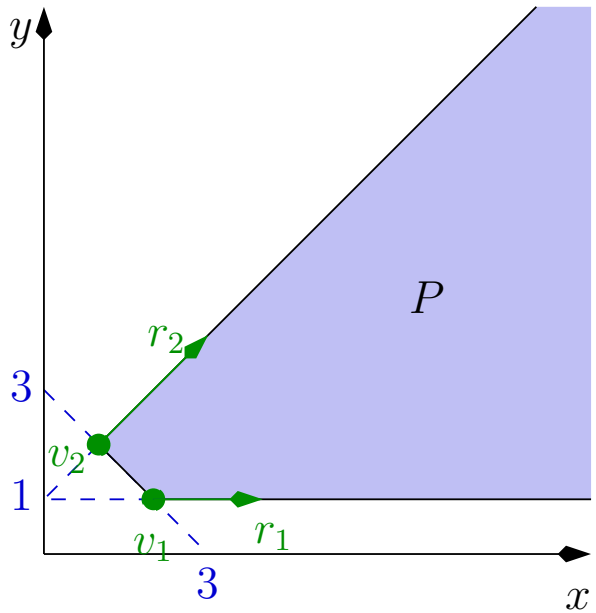
Sur l'exemple



- $P^{fin}_{loop} = \{0 \leqslant x \leqslant 100\}$ si on se débrouille bien

Résolution du système de point-fixe - 1

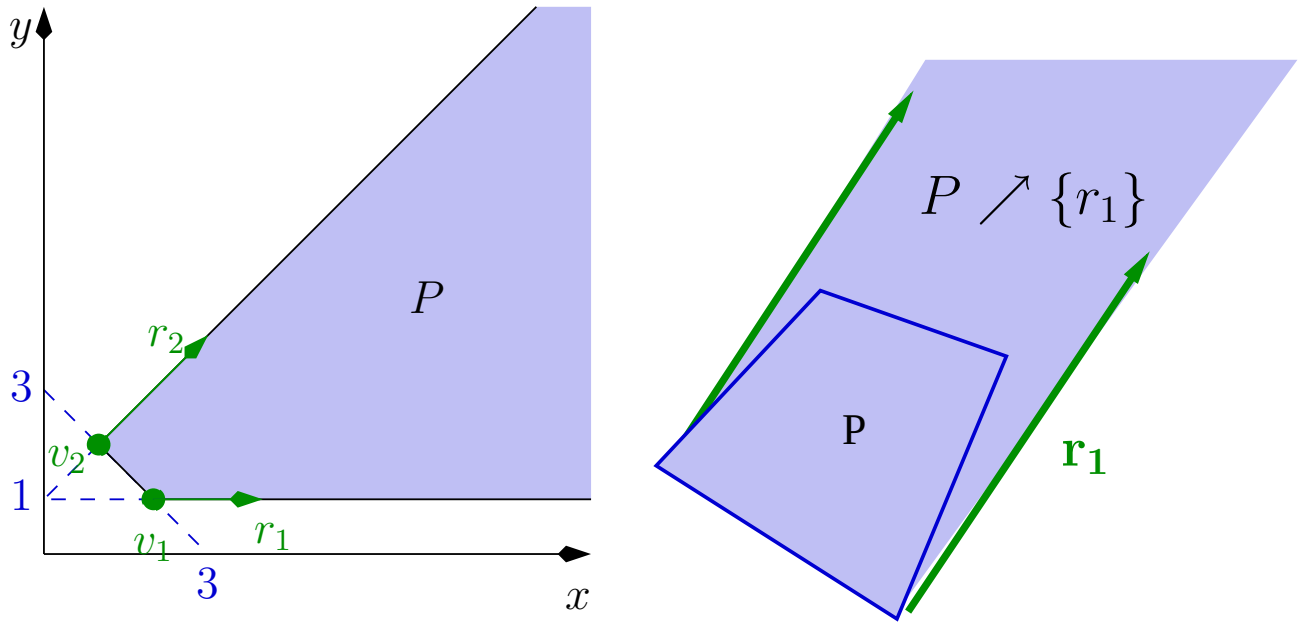
Représentation par polyèdres convexes :



- Algorithmique disponible et efficace (test du vide, intersection, union, transformation affine. . .)

Résolution du système de point-fixe - 1

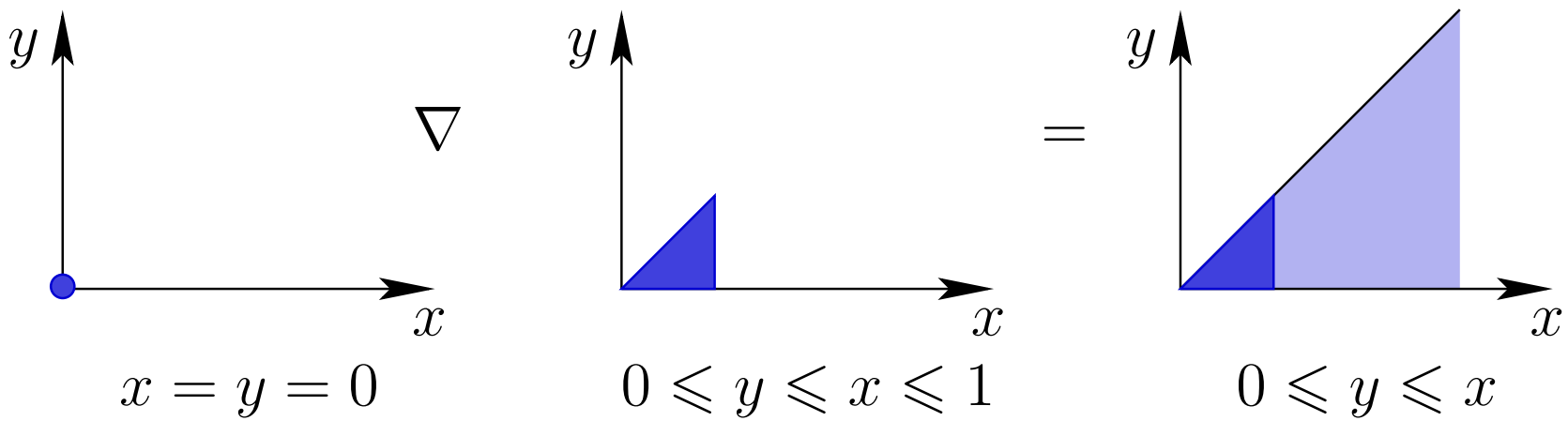
Représentation par polyèdres convexes :



- Algorithmique disponible et efficace (test du vide, intersection, union, transformation affine. . . .)
- Ajout de rayon.

Résolution du système de point-fixe - 2

Élargissement : $P \nabla Q$: extrapolation de la limite.
Le système de contrainte de $P \nabla Q$ est obtenu en enlevant du système de Q les contraintes non saturées par P :

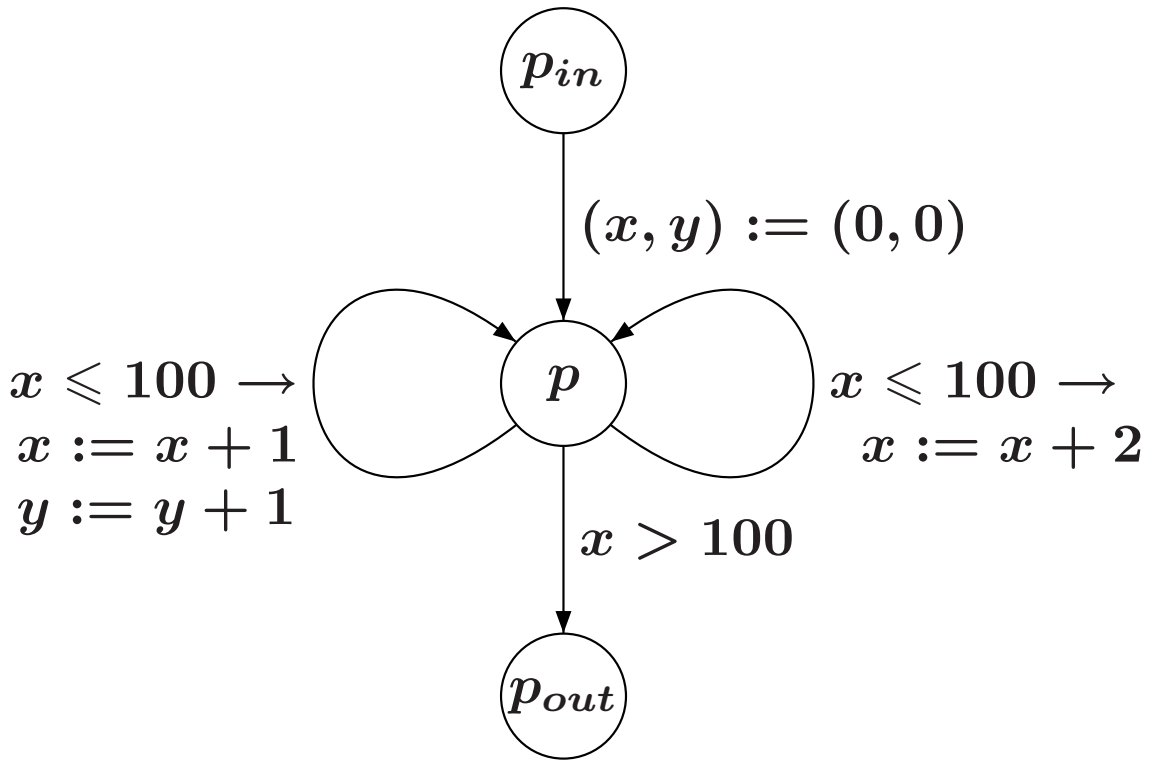


$x = y = 0$ $\quad$ $0 \leqslant y \leqslant x \leqslant 1$ $\quad$ $0 \leqslant y \leqslant x$

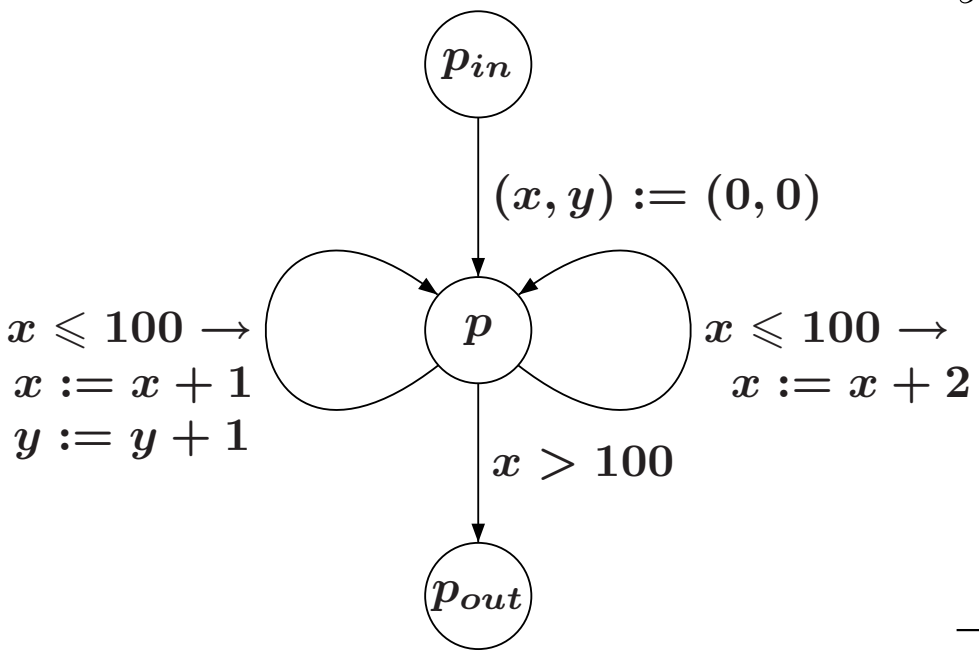
Astuce (!) : $\{x = y = 0\} = \{0 \leqslant y \leqslant x \leqslant 0\}$

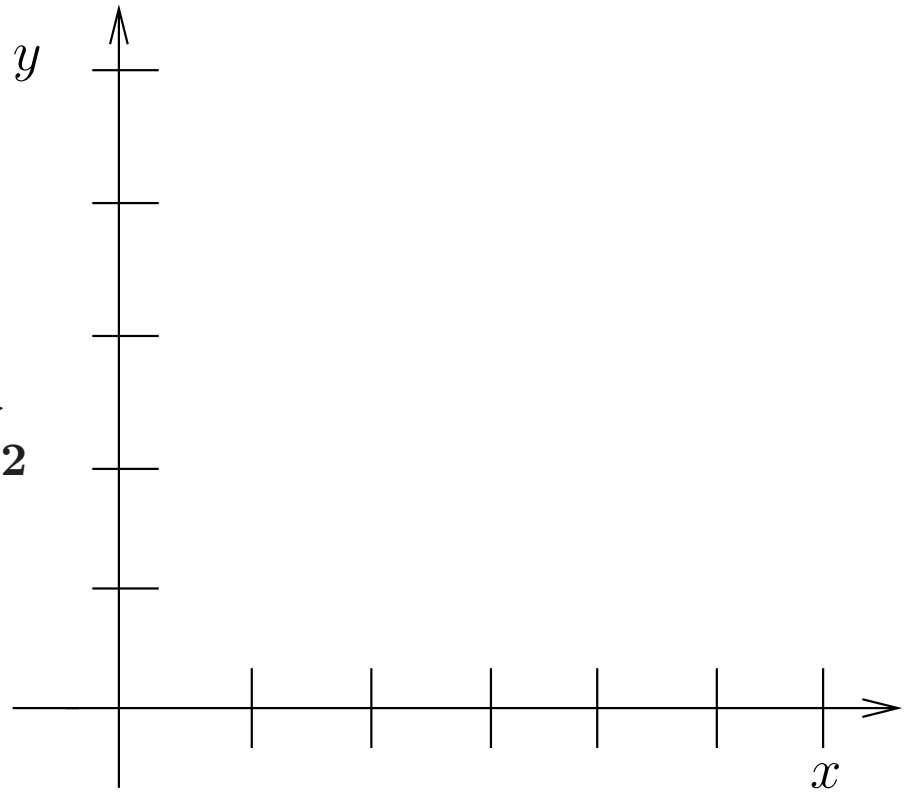
Un exemple - 1

```
x:=0;y:=0
while (x<=100) do
  read(b);
  if b then
        x:=x+2
      else begin
        x:=x+1;
        y:=y+1;
      end;
  endif
endwhile
```

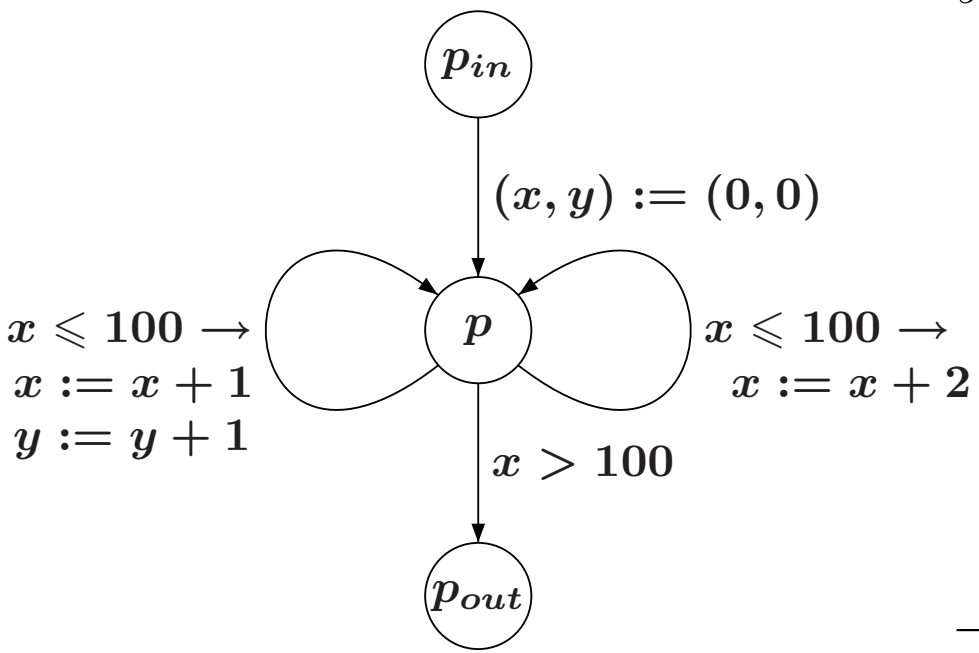


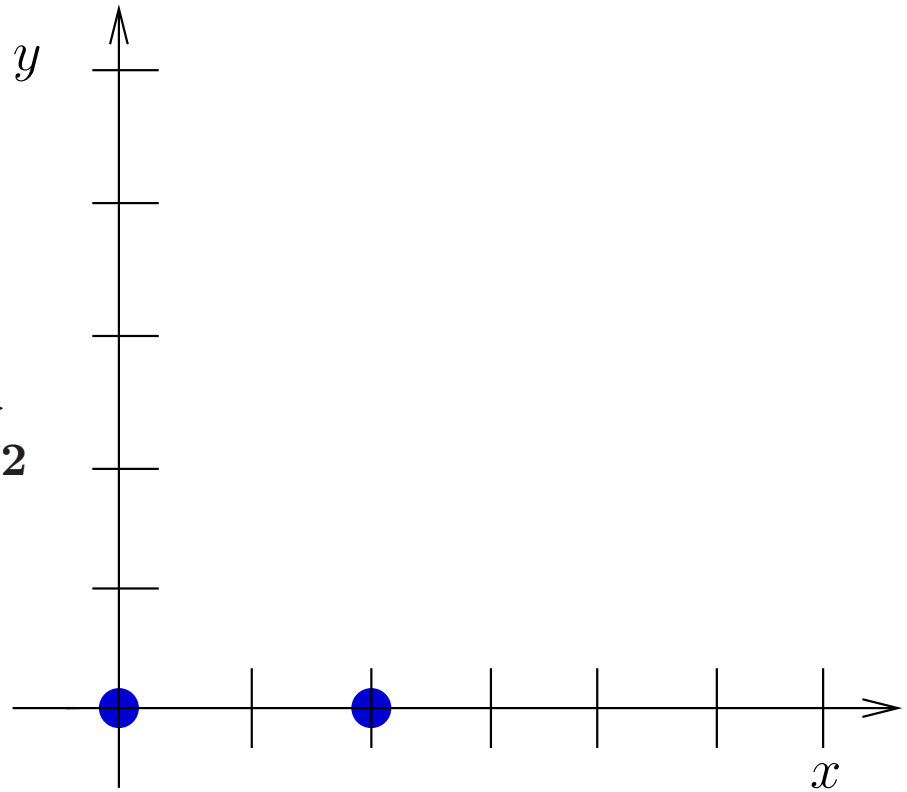
Un exemple - 2


p_{in}
$(x, y) := (0, 0)$
p
$x \leqslant 100 \rightarrow$
$x := x + 1$
$y := y + 1$
$x \leqslant 100 \rightarrow$
$x := x + 2$
$x > 100$
p_{out}

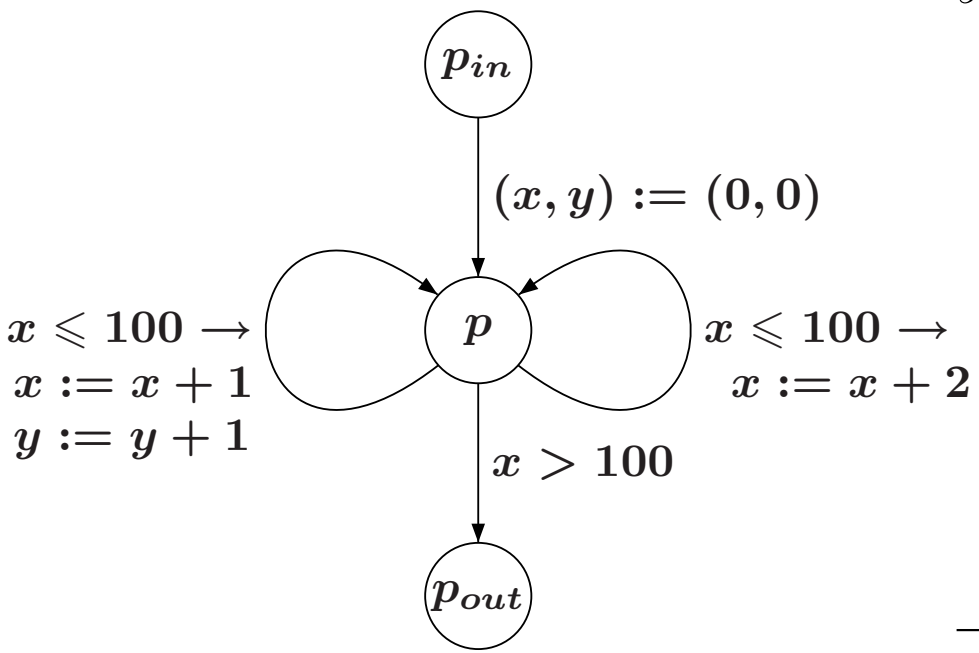

y
x

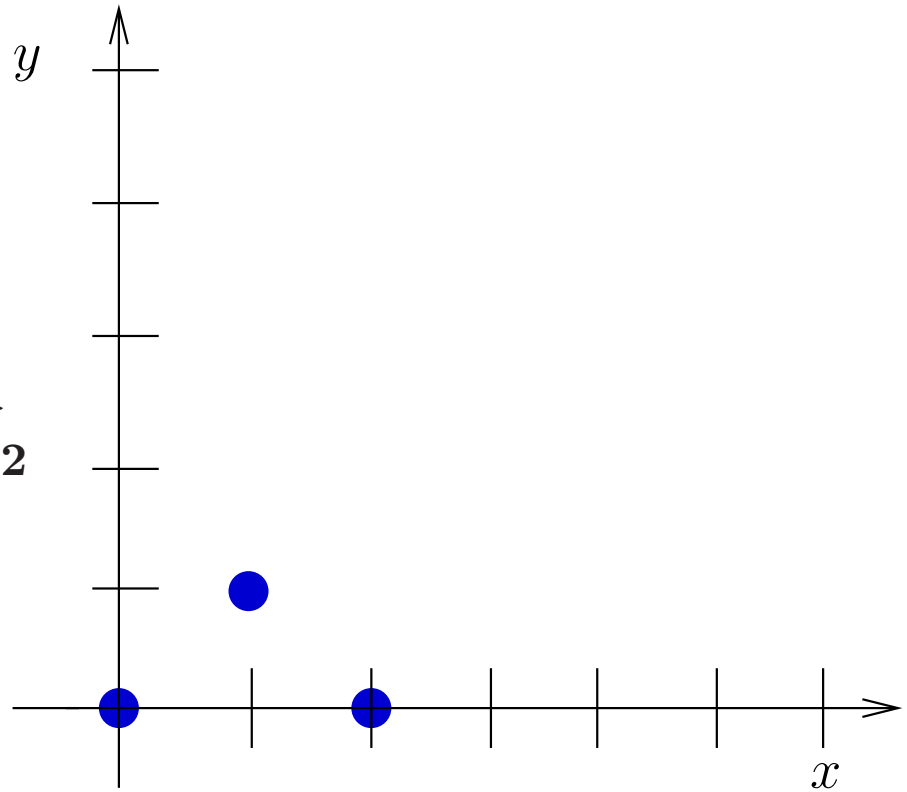
Un exemple - 2


p_{in}
$(x, y) := (0, 0)$
p
$x \leqslant 100 \rightarrow$
$x := x + 1$
$y := y + 1$
$x \leqslant 100 \rightarrow$
$x := x + 2$
$x > 100$
p_{out}

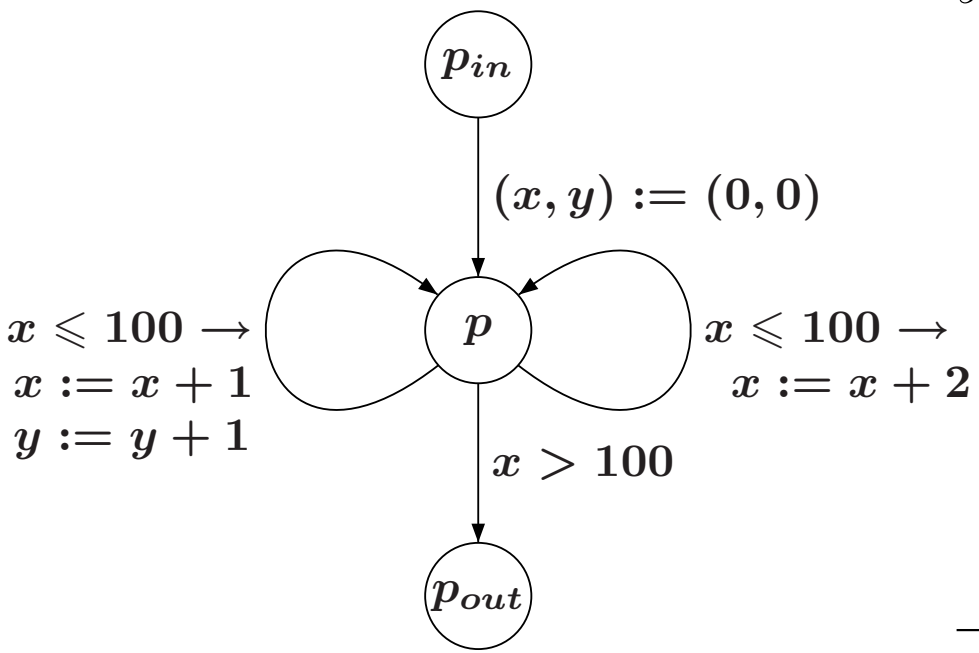

y
x

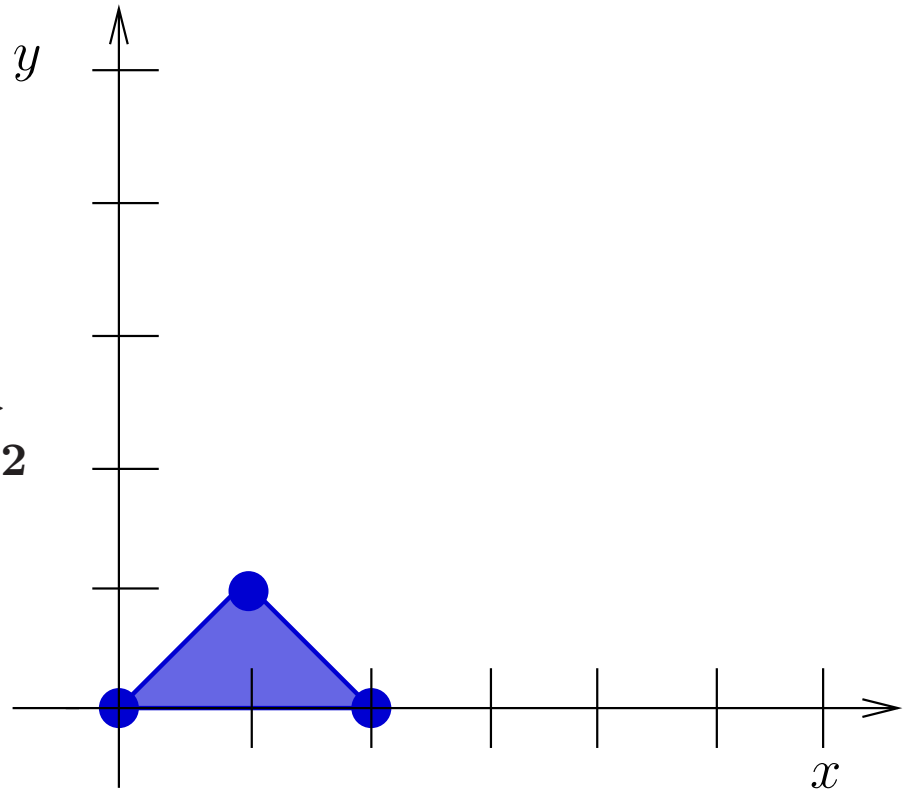
Un exemple - 2


p_{in}
$(x, y) := (0, 0)$
p
$x \leqslant 100 \rightarrow$
$x := x + 1$
$y := y + 1$
$x \leqslant 100 \rightarrow$
$x := x + 2$
$x > 100$
p_{out}

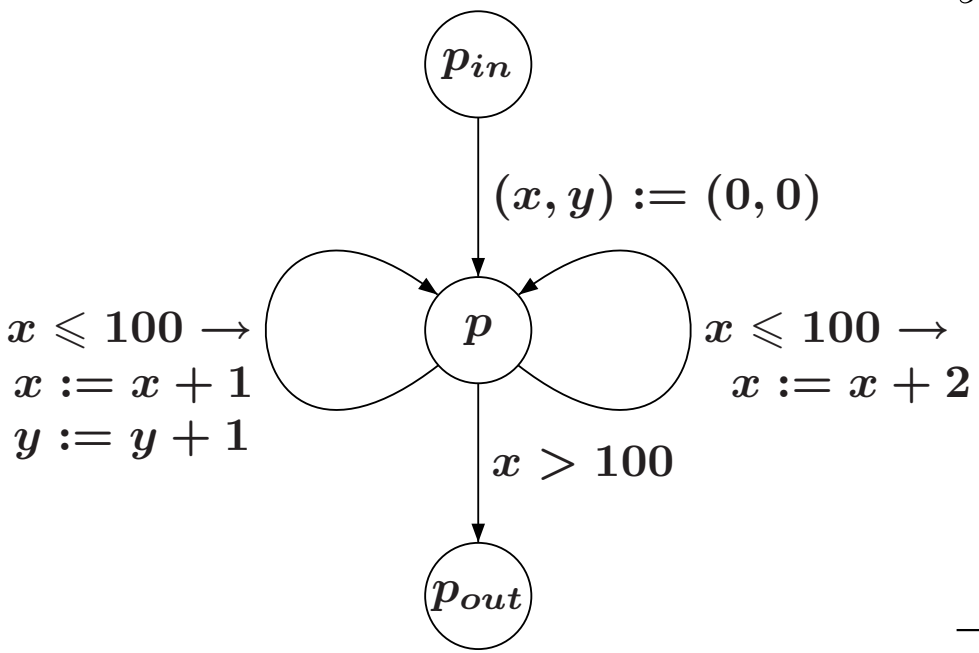

y
x

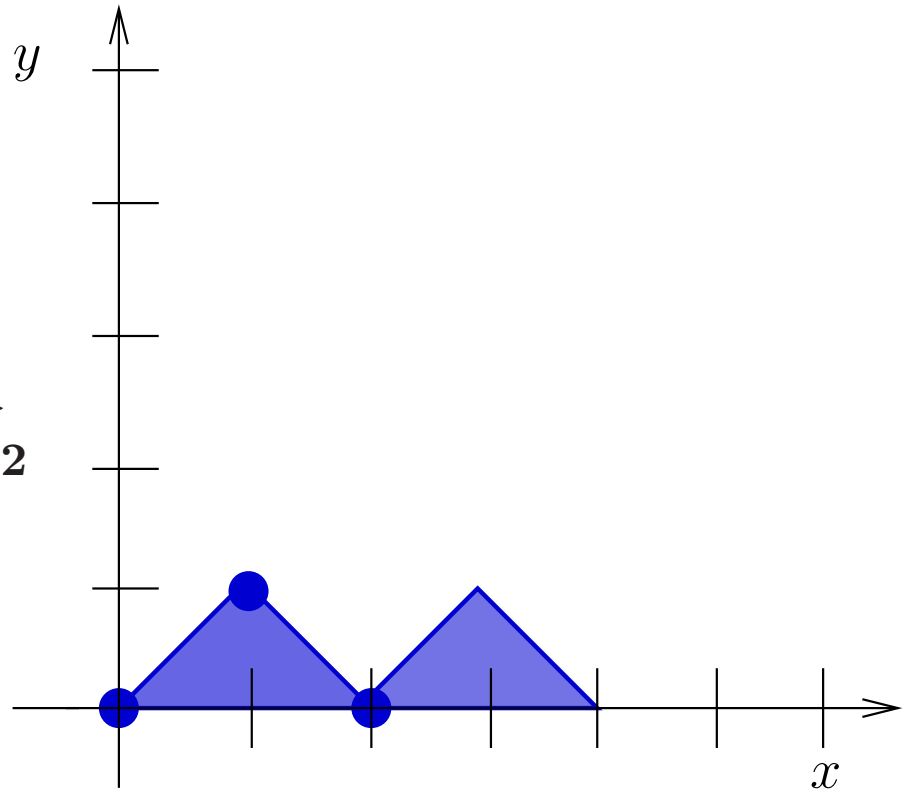
Un exemple - 2


p_{in}
$(x, y) := (0, 0)$
p
$x \leqslant 100 \rightarrow$
$x := x + 1$
$y := y + 1$
$x \leqslant 100 \rightarrow$
$x := x + 2$
$x > 100$
p_{out}

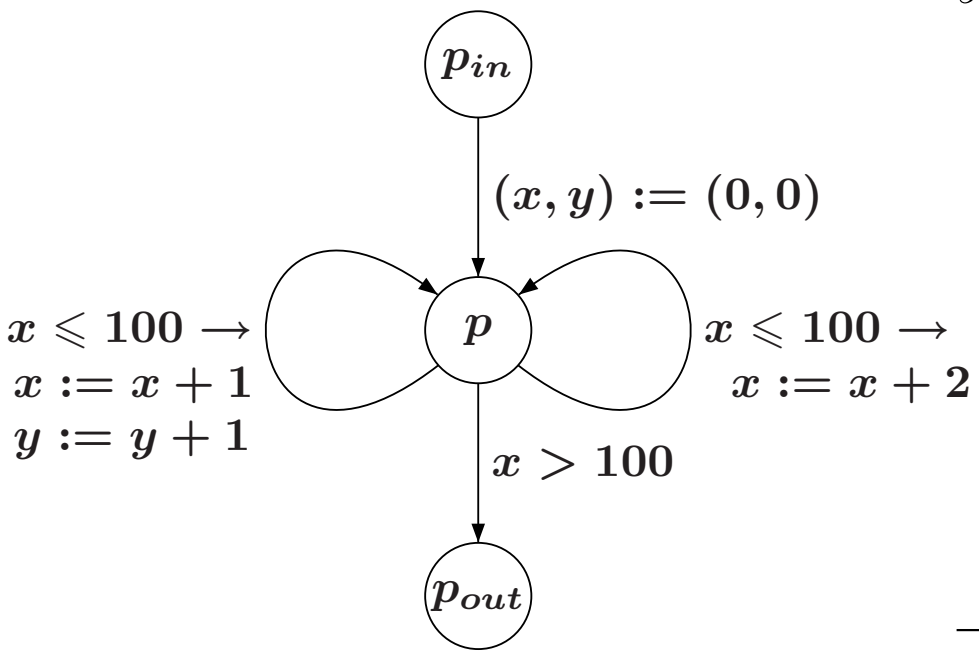

y
x

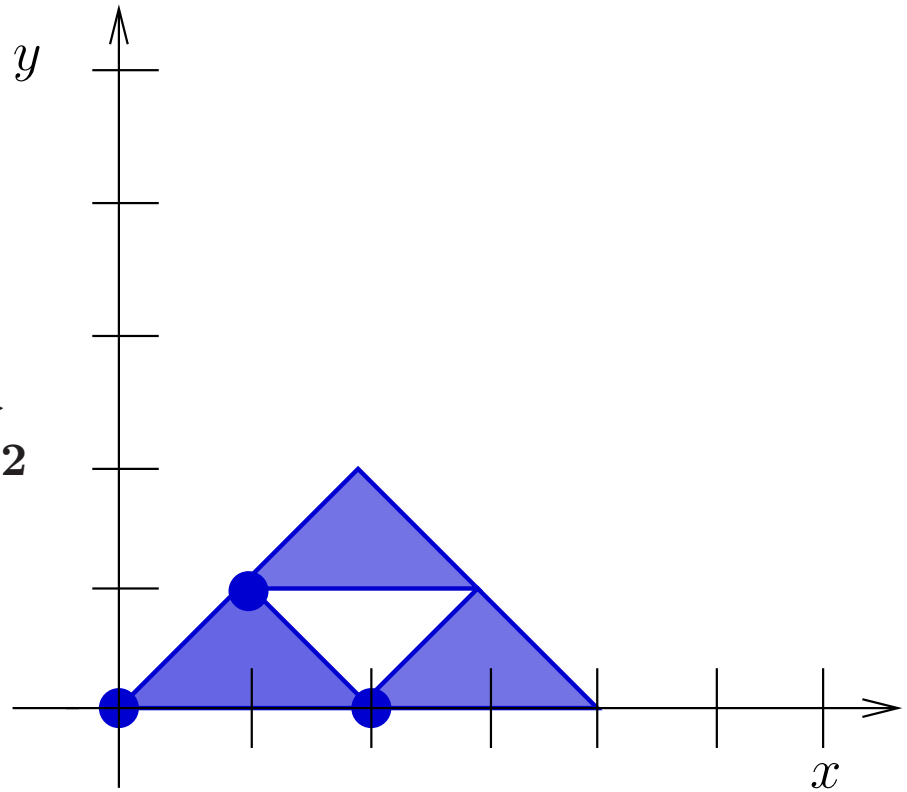
Un exemple - 2


p_{in}
$(x, y) := (0, 0)$
p
$x \leqslant 100 \rightarrow$
$x := x + 1$
$y := y + 1$
$x \leqslant 100 \rightarrow$
$x := x + 2$
$x > 100$
p_{out}

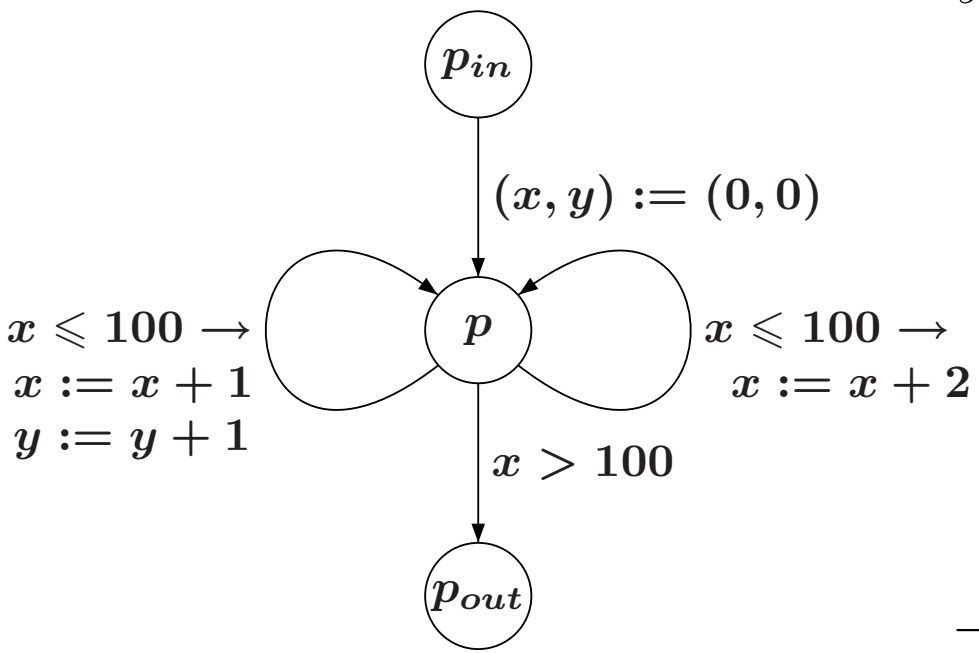

y
x

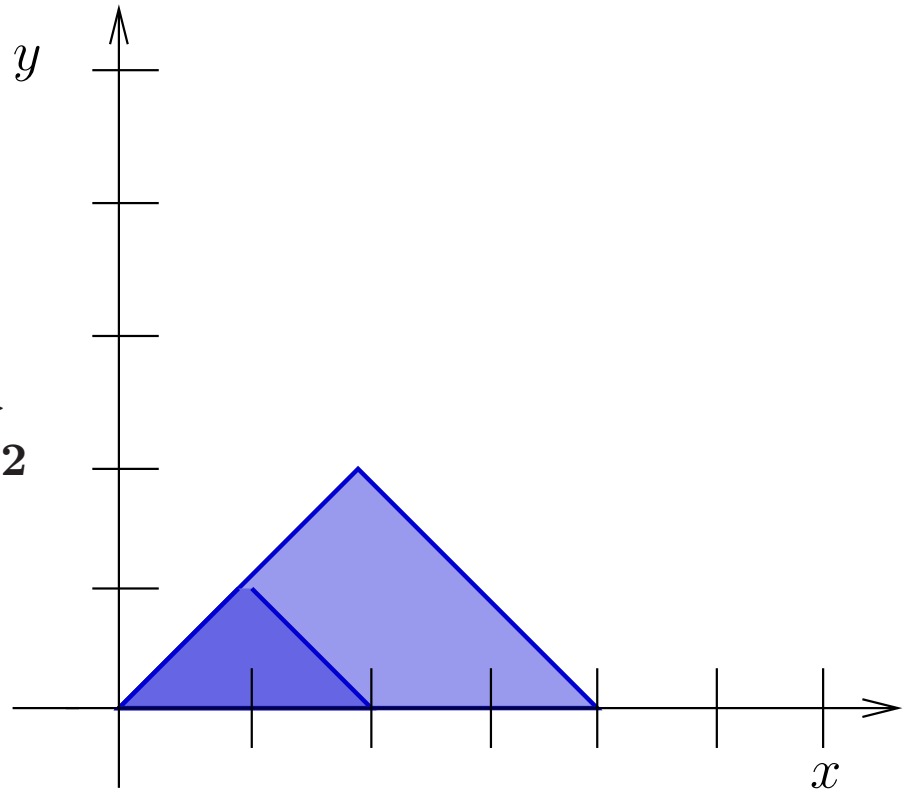
Un exemple - 2


p_{in}
$(x, y) := (0, 0)$
p
$x \leqslant 100 \rightarrow$
$x := x + 1$
$y := y + 1$
$x \leqslant 100 \rightarrow$
$x := x + 2$
$x > 100$
p_{out}

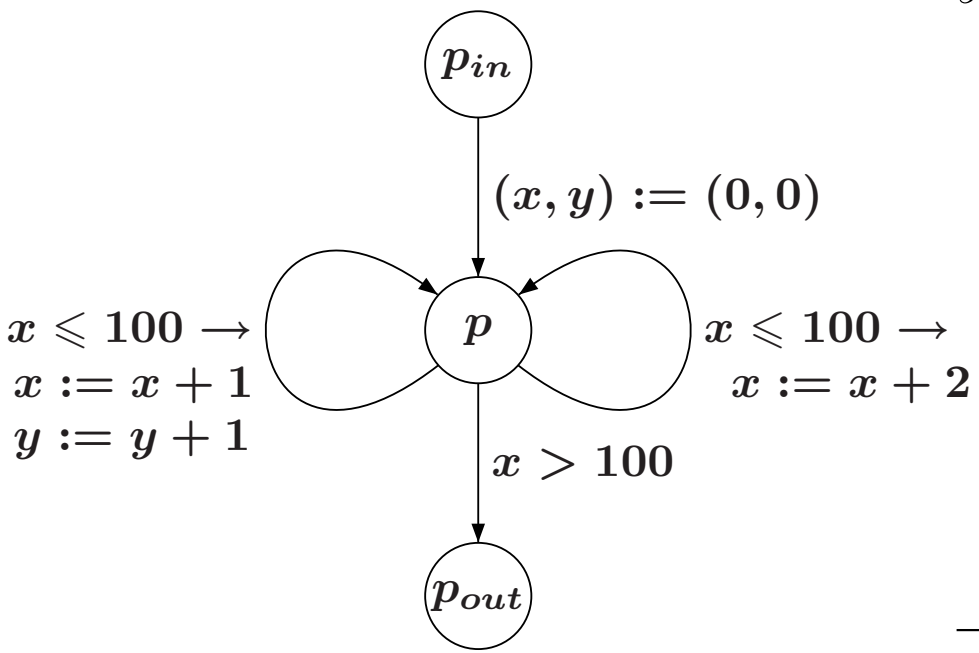

y
x

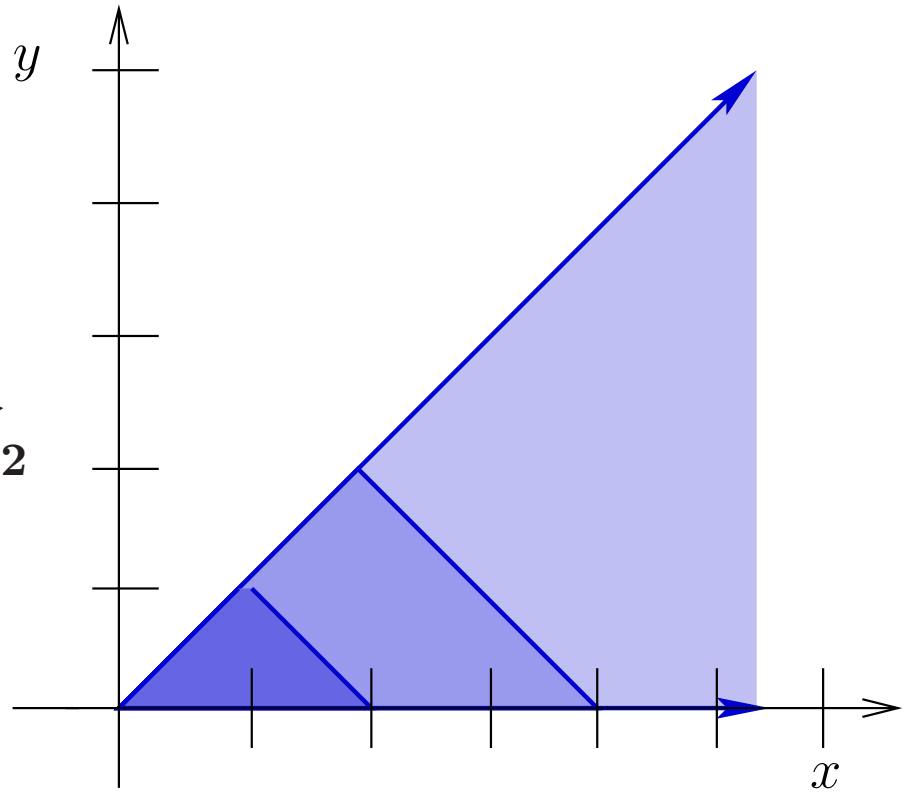
Un exemple - 2


p_{in}
$(x, y) := (0, 0)$
p
$x \leqslant 100 \rightarrow$
$x := x + 1$
$y := y + 1$
$x \leqslant 100 \rightarrow$
$x := x + 2$
$x > 100$
p_{out}

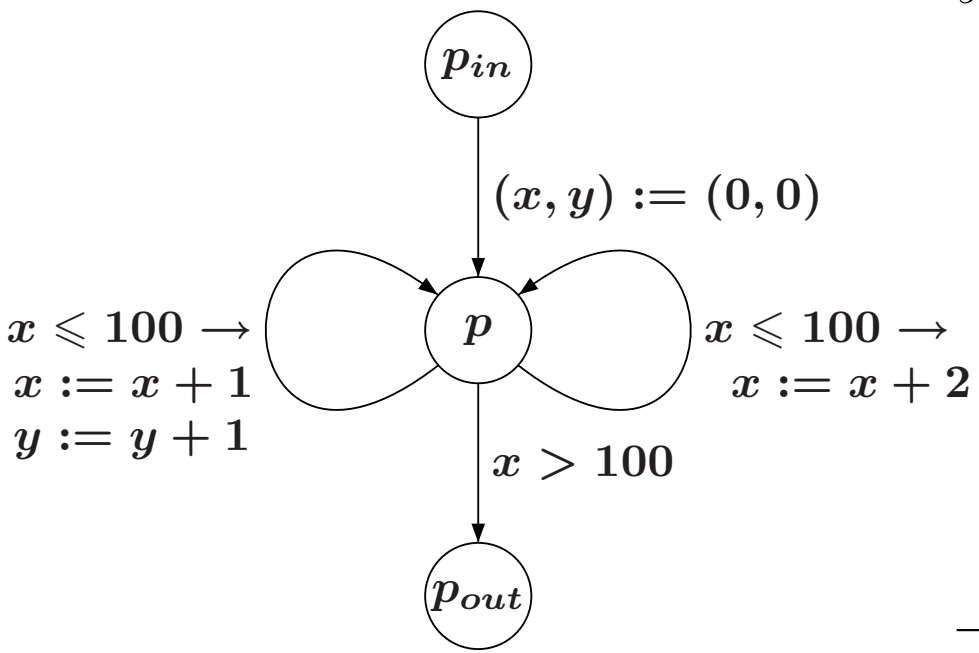

y
x

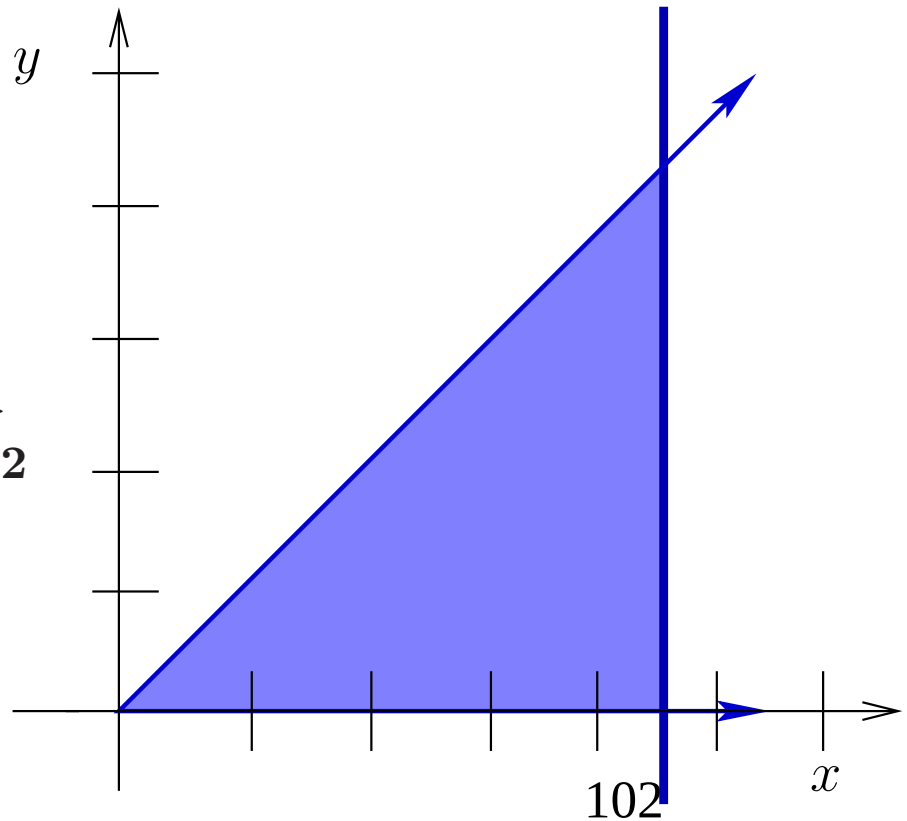
Un exemple - 2


p_{in}
$(x, y) := (0, 0)$
p
$x \leqslant 100 \rightarrow$
$x := x + 1$
$y := y + 1$
$x \leqslant 100 \rightarrow$
$x := x + 2$
$x > 100$
p_{out}


y
x

Un exemple - 2


p_{in}
$(x, y) := (0, 0)$
p
$x \leqslant 100 \rightarrow$
$x := x + 1$
$y := y + 1$
$x \leqslant 100 \rightarrow$
$x := x + 2$
$x > 100$
p_{out}


y
x
102

Les problèmes de l'Analyse des Relations Linéaires

Sources de complexité :

- nombre de points de contrôle.
- nombre de variables numériques.

Sources d'approximation :

- Enveloppe convexe.
- **Élargissement** (cf ma thèse)

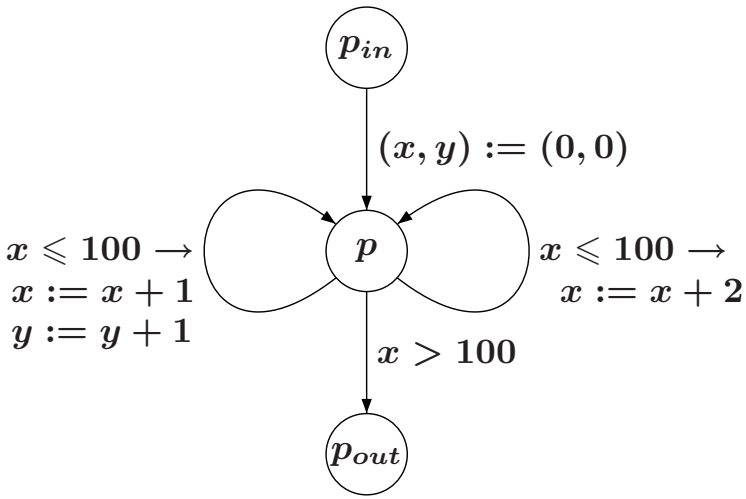
Caractéristiques d'Aspic

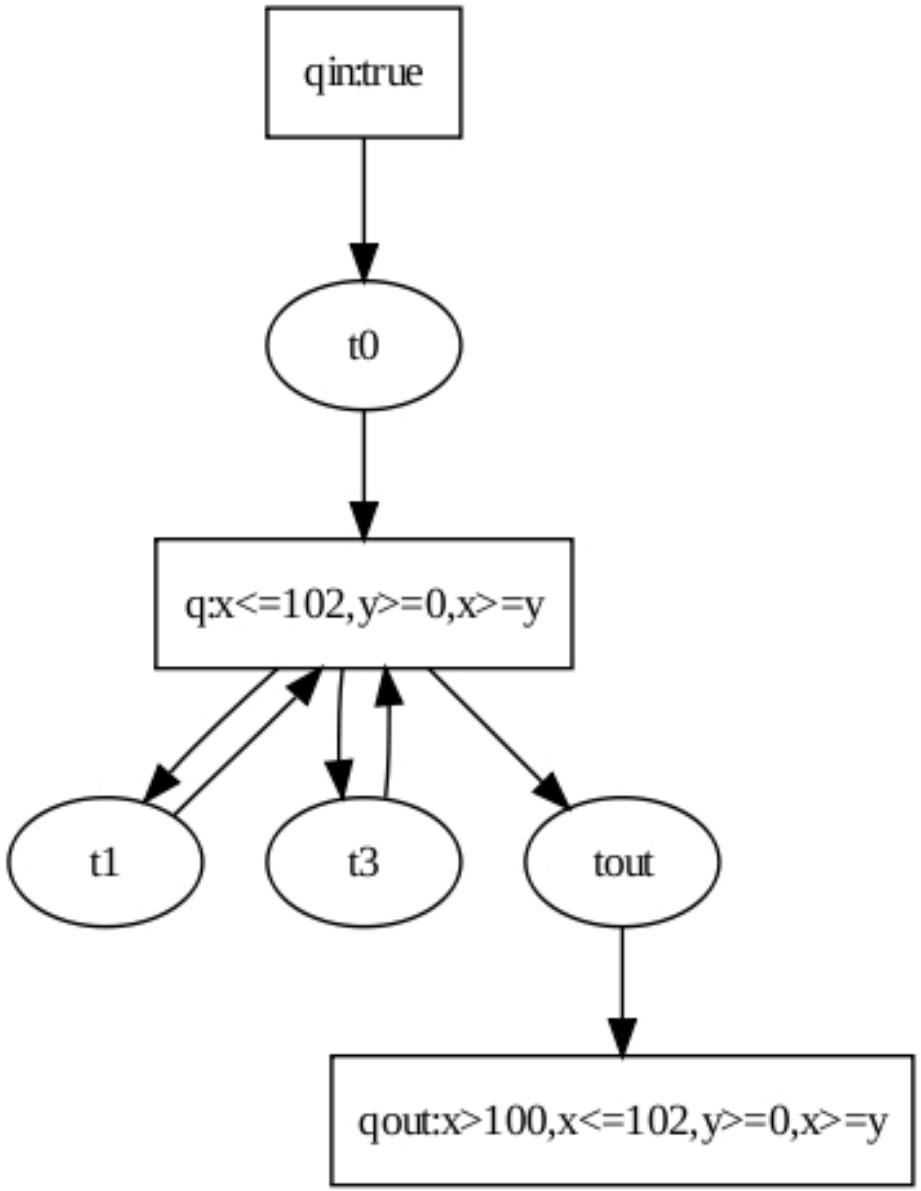
ASPIC : Accelerated Symbolic Polyhedral Invariant Computation

- Un langage textuel d'automates (Fast) avec ou sans but de preuve (formule). Non déterminisme et `x :=?`
- Calcul classique + accélérations.
- Sorties : invariants (+ diagnostic).

▶ `http://laure.gonnord.org/pro/aspic/aspic.html`

Invariants de l'exemple

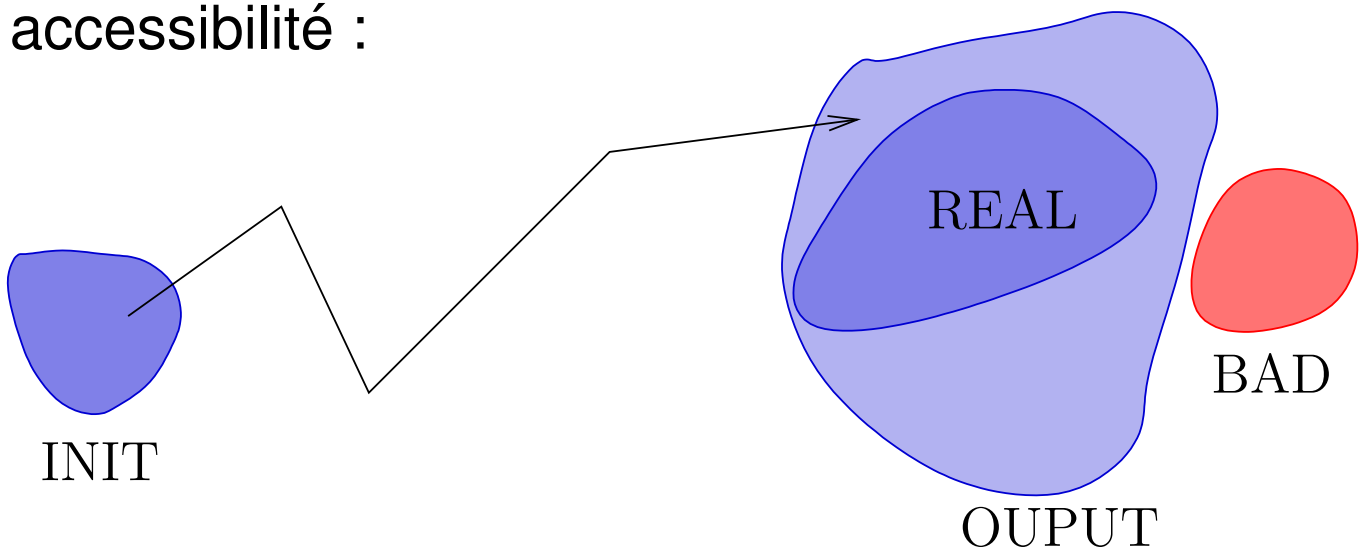




- Démo ?

Applications - 1

- Vérification de programmes **numériques**. On prouve la non accessibilité :



- (non) Accessibilité dans des automates à compteurs (sémantique de SystemC), une centaine de points de contrôle, J. Cornet.

Applications - 2

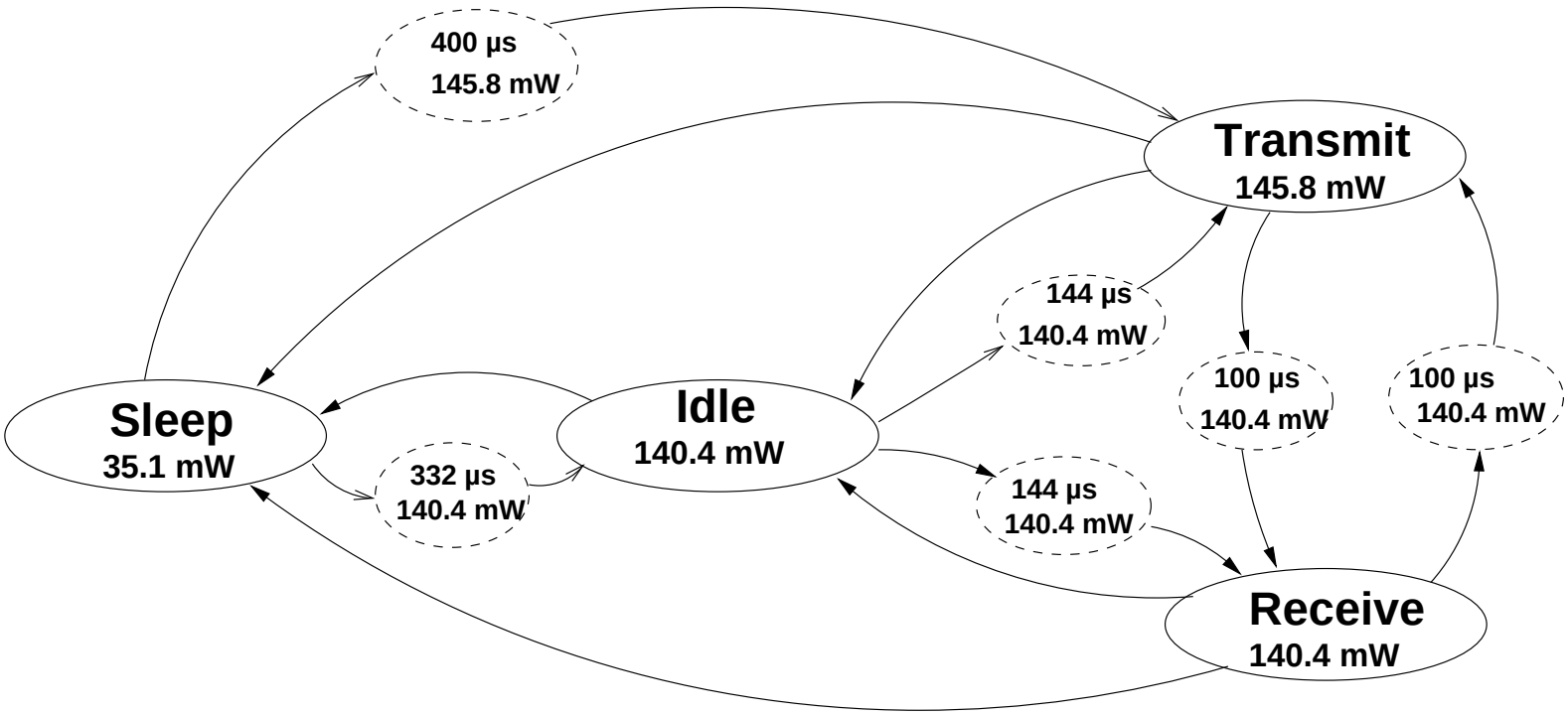
Par encodage dans des automates à compteurs :

- Vérification de programmes manipulant des **listes**, R. Iosif et S. Perarnau.
- Vérification de programmes à **pointeurs**, A. Sangnier et A. Finkel.

Applications - 3

Par encodage toujours :

- Invariants numériques d'automates modélisant une **consommation d'énergie** de (réseaux de) capteurs, L. Samper et F. Maraninchi.



Applications - 4

Travaux avec COMPSYS (ENS Lyon) : calcul de pire temps d'exécution de programmes :

- compilation + analyse statique
- ordonnancement.

- Travail avec A. Darte, P. Feautrier, C. Alias.
- Démo ?

En résumé

L'Analyse des Relations Linéaires :

- Calcule des invariants numériques
- Sur des automates à compteurs
- N'est pas toujours exacte mais **sûre** (surapprox)
- Est performante !
- Peut servir à autre chose !

D'autres analyses

- Approchées : intervalles, octogones, tableaux, ...
- Exactes : ensembles d'entiers, intervalles, booléens, variables actives, propagation de constantes

et plein de combinaisons...

Un **outil** intéressant :

`http://frama-c.cea.fr/`

Fin de l'exposé

Merci.