IS, un domaine numérique abstrait pour l'analyse de programmes manipulant des adresses

Mathias Péron
stage réalisé sous la direction de Nicolas Halbwachs

master informatique Systèmes et Logiciels

Contexte et Objectifs

- contexte : validation de logiciels
- domaine : vérification automatique
  - analyse *statique*
  - propriétés de *sûreté*
- objectif : analyser des programmes manipulant des adresses

  prouver des propriétés dépendantes des *accès aux objets adressés*.
  - adresses dans les bornes
  - exclusivité d'accès : adresses mémoire, de composants (SoCs)

Contexte et Objectifs

- contexte : validation de logiciels
- domaine : vérification automatique
  - analyse *statique*
  - propriétés de *sûreté*
- objectif : analyser des programmes manipulant des adresses

  prouver des propriétés dépendantes des *accès aux objets adressés*.
  - adresses dans les bornes
  - exclusivité d'accès : adresses mémoire, de composants (SoCs)

Contexte et Objectifs

- contexte : validation de logiciels
- domaine : vérification automatique
  - analyse *statique*
  - propriétés de *sûreté*
- objectif : analyser des programmes manipulant des adresses

  prouver des propriétés dépendantes des *accès aux objets adressés*.
  - adresses dans les bornes
  - exclusivité d'accès : adresses mémoire, de composants (SoCs)

Contexte et Objectifs

- contexte : validation de logiciels
- domaine : vérification automatique
  - analyse *statique*
  - propriétés de *sûreté*
- objectif : analyser des programmes manipulant des adresses

  prouver des propriétés dépendantes des *accès aux objets adressés*. $\triangleright$ par des *entiers*
  - adresses dans les bornes
  - exclusivité d'accès : adresses mémoire, de composants (SoCs)

Contexte et Objectifs

- contexte : validation de logiciels
- domaine : vérification automatique
  - analyse *statique*
  - propriétés de *sûreté*
- objectif : analyser des programmes manipulant des adresses

  prouver des propriétés dépendantes des *accès aux objets adressés*. ▷ par des *entiers*
  - adresses dans les bornes

    ▷ *intervalles*
  - exclusivité d'accès : adresses mémoire, de composants (SoCs)

Contexte et Objectifs

- contexte : validation de logiciels
- domaine : vérification automatique
  - analyse *statique*
  - propriétés de *sûreté*
- objectif : analyser des programmes manipulant des adresses

  prouver des propriétés dépendantes des *accès aux objets adressés*. ▷ par des *entiers*
  - adresses dans les bornes

    ▷ *intervalles*
  - exclusivité d'accès : adresses mémoire, de composants (SoCs)

    ▷ *égalité* et *non-égalité*

Vérification de systèmes infinis

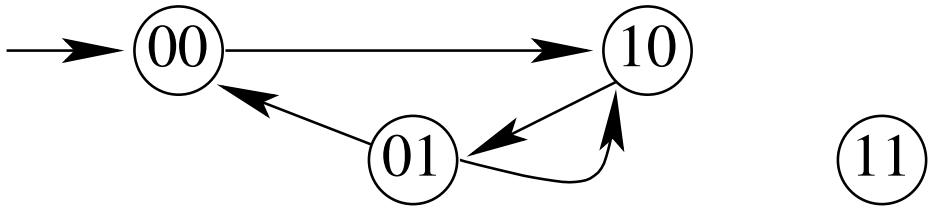
indécidabilité de la vérification de systèmes infinis

- méthodes déductives : theorem proving
- vérification automatique
  - forme abstraite du programme : *système de transition*
  - vérification des propriétés de sûreté $\equiv$ calcul des états accessibles depuis les états initiaux.

Vérification de systèmes infinis

indécidabilité de la vérification de systèmes infinis

- méthodes déductives : theorem proving
- vérification automatique
  - forme abstraite du programme : *système de transition*

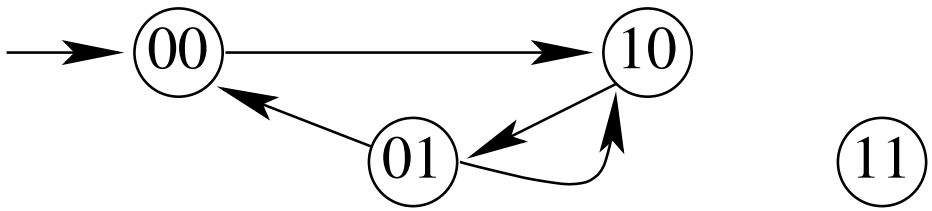


  - vérification des propriétés de sûreté ≡ calcul des états accessibles depuis les états initiaux.

Vérification de systèmes infinis

indécidabilité de la vérification de systèmes infinis

- méthodes déductives : theorem proving
- vérification automatique
  - forme abstraite du programme : *système de transition*

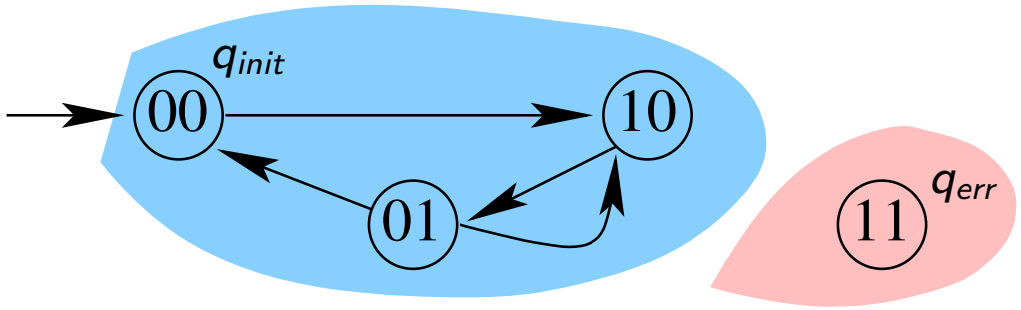


  - vérification des propriétés de sûreté $\equiv$ calcul des états accessibles depuis les états initiaux.

Vérification de systèmes infinis

indécidabilité de la vérification de systèmes infinis

- méthodes déductives : theorem proving
- vérification automatique
  - forme abstraite du programme : *système de transition*

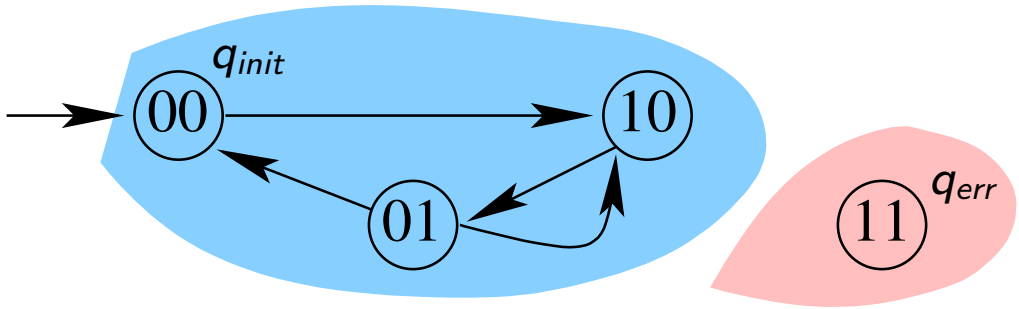


  - vérification des propriétés de sûreté $\equiv$ calcul des états accessibles depuis les états initiaux.
    $\triangleright$ analyse en avant : $accessibles(Q_{init}) \cap Q_{err} = \emptyset$

Vérification de systèmes infinis

indécidabilité de la vérification de systèmes infinis

- méthodes déductives : theorem proving
- vérification automatique
  - forme abstraite du programme : *système de transition*



  - vérification des propriétés de sûreté $\equiv$ calcul des états accessibles depuis les états initiaux.
    $\triangleright$ analyse en avant : $accessibles(Q_{init}) \cap Q_{err} = \emptyset$
    $\triangleright$ les états accessibles sont la solution d'une équation de point-fixe *(post-conditions)*

Vérification de systèmes infinis

- model checking
  ▷ calcul exact du point-fixe.
  ▷ difficulté de trouver un modèle fini adéquat pour chaque programme
- interprétation abstraite
  ▷ calcul d'approximations du point-fixe.
  ▷ permet d'abstraire dans une certaine spécification de propriétés : domaine abstrait
  ▷ une classe de problèmes $\leftrightsquigarrow$ un domaine abstrait

Vérification de systèmes infinis

- model checking
  ▷ calcul exact du point-fixe.
  ▷ difficulté de trouver un modèle fini adéquat pour chaque programme
- interprétation abstraite
  ▷ calcul d'approximations du point-fixe.
  ▷ permet d'abstraire dans une certaine spécification de propriétés : domaine abstrait
  ▷ une classe de problèmes $\leftrightsquigarrow$ un domaine abstrait

Sujet de recherche

- cadre théorique : interprétation abstraite, domaines numériques abstraits.
- sujet : *définir* un domaine abstrait pour une analyse dédiée aux adresses.

  adresses $\equiv$ entiers généraux munis d'opérations restreintes

Sujet de recherche

- cadre théorique : interprétation abstraite, domaines numériques abstraits.
- sujet : *définir* un domaine abstrait pour une analyse dédiée aux adresses.

  adresses $\equiv$ entiers généraux munis d'opérations restreintes

Sujet de recherche

- cadre théorique : interprétation abstraite, domaines numériques abstraits.
- sujet : *définir* un domaine abstrait pour une analyse dédiée aux adresses.

  adresses $\equiv$ entiers généraux munis d'opérations restreintes

  $\triangleright$ un nouveau domaine moins coûteux que les domaines numériques existants

Sujet de recherche

- cadre théorique : interprétation abstraite, domaines numériques abstraits.
- sujet : *définir* un domaine abstrait pour une analyse dédiée aux adresses.

  adresses $\equiv$ entiers généraux munis d'opérations restreintes

  $\triangleright$ un nouveau domaine moins coûteux que les domaines numériques existants
  $\triangleright$ permettant d'exprimer des propriétés d'égalité/non-égalité

Sujet de recherche

- cadre théorique : interprétation abstraite, domaines numériques abstraits.
- sujet : *définir* un domaine abstrait pour une analyse dédiée aux adresses.

  adresses $\equiv$ entiers généraux munis d'opérations restreintes

  $\triangleright$ un nouveau domaine moins coûteux que les domaines numériques existants
  $\triangleright$ permettant d'exprimer des propriétés d'égalité/non-égalité
  $\blacktriangleright$ **intervalles** $+ \{=, \neq\}$

Sujet de recherche

- cadre théorique : interprétation abstraite, domaines numériques abstraits.
- sujet : *définir* un domaine abstrait pour une analyse dédiée aux adresses.

  adresses $\equiv$ entiers généraux munis d'opérations restreintes

  $\triangleright$ un nouveau domaine moins coûteux que les domaines numériques existants
  $\triangleright$ permettant d'exprimer des propriétés d'égalité/non-égalité
  $\blacktriangleright$ **intervalles** $+ \{=, \neq\} + \{\leq, <\}$

Plan de l'exposé

Plan de l'exposé

Plan de l'exposé

Interprétation abstraite

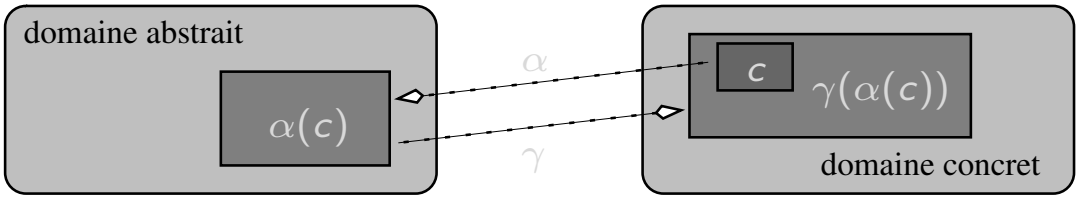
- théorie de l'approximation du comportement *dynamique* de programmes.
  ▷ une approximation pertinente de la sémantique des programmes
  ▷ permet de construire un analyseur *cohérent* à partir de la sémantique du programme
- vérification conservative
  - abstraction α
  - concrétisation γ

Interprétation abstraite

- théorie de l'approximation du comportement *dynamique* de programmes.
  ▷ une approximation pertinente de la sémantique des programmes
  ▷ permet de construire un analyseur *cohérent* à partir de la sémantique du programme
- vérification conservative
  - abstraction α
  - concrétisation γ

Interprétation abstraite

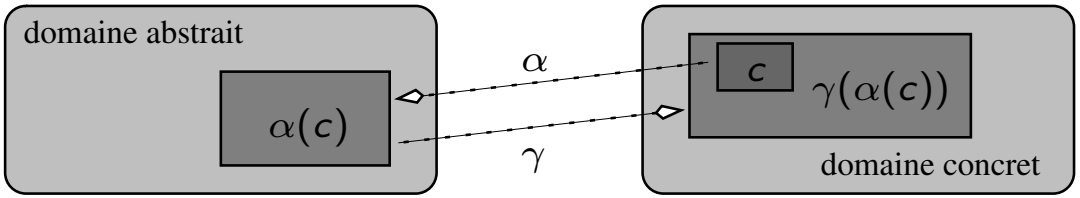
- théorie de l'approximation du comportement *dynamique* de programmes.
  ▷ une approximation pertinente de la sémantique des programmes
  ▷ permet de construire un analyseur *cohérent* à partir de la sémantique du programme
- vérification conservative



- abstraction α
- concrétisation γ

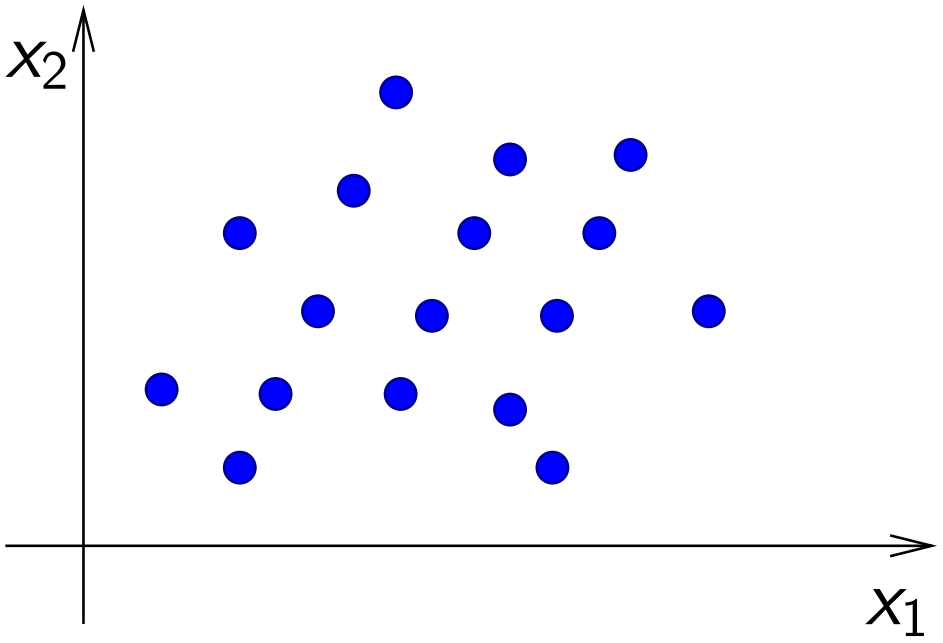
Interprétation abstraite

- théorie de l'approximation du comportement *dynamique* de programmes.
  ▷ une approximation pertinente de la sémantique des programmes
  ▷ permet de construire un analyseur *cohérent* à partir de la sémantique du programme
- vérification conservative

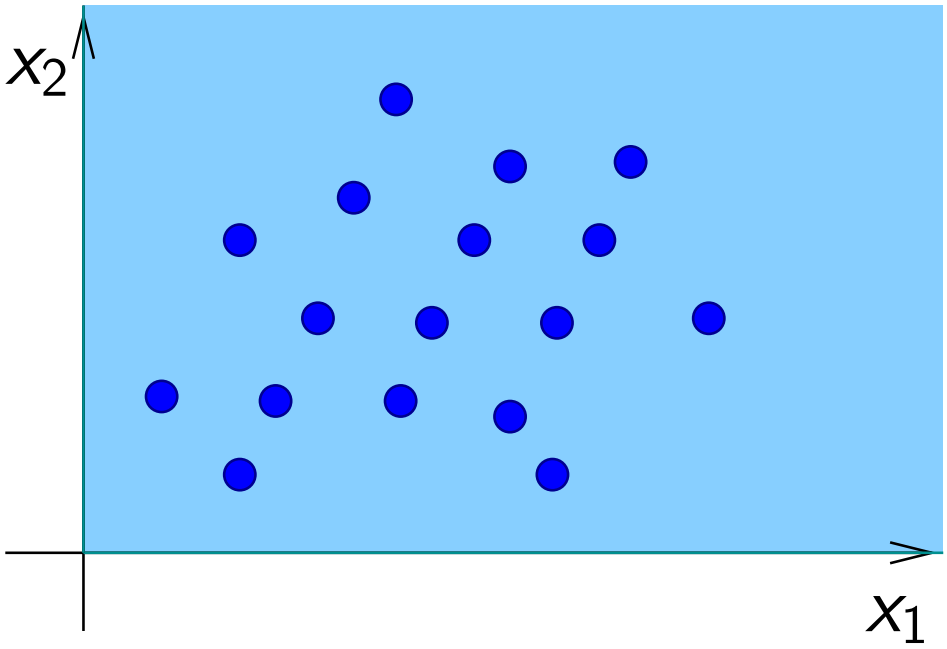


- abstraction α
- concrétisation γ

Domaines numériques abstraits

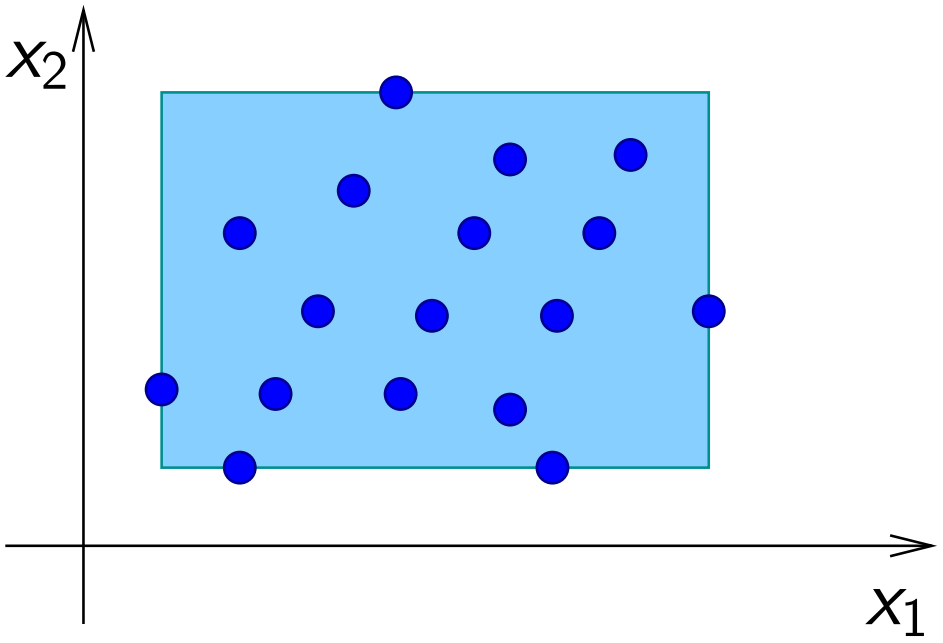


Domaines numériques abstraits



parité $0 \leq x_i$

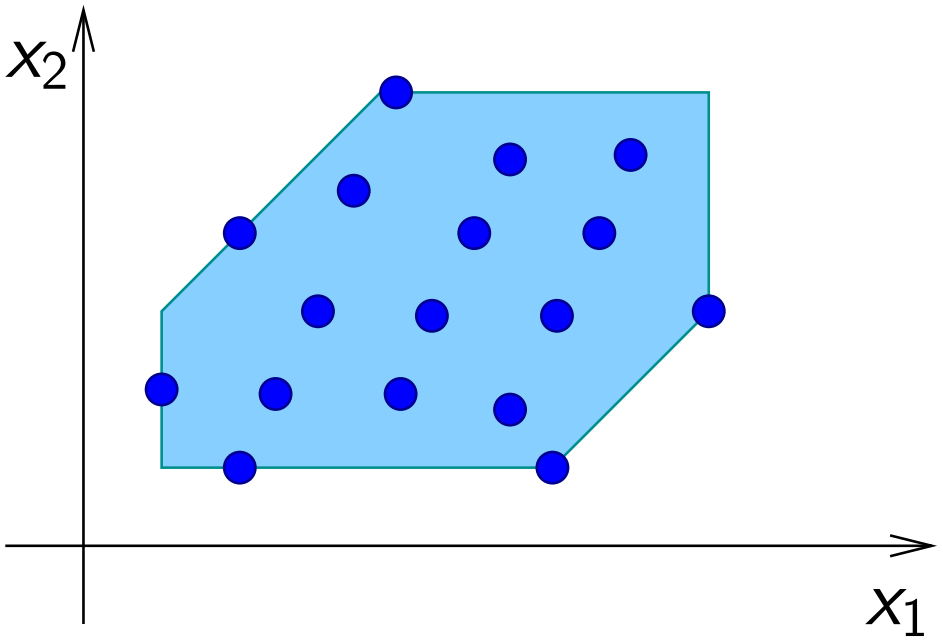
Domaines numériques abstraits



parité

intervalles $lb \leq x_i \leq ub$

Domaines numériques abstraits

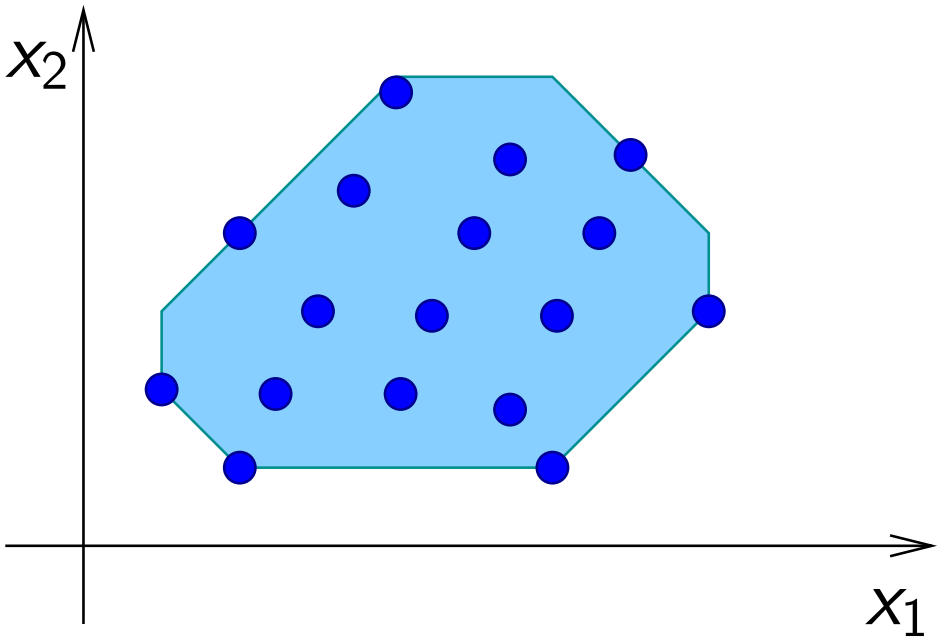


parité

intervalles

zones $x_i - x_j \leq c$

Domaines numériques abstraits



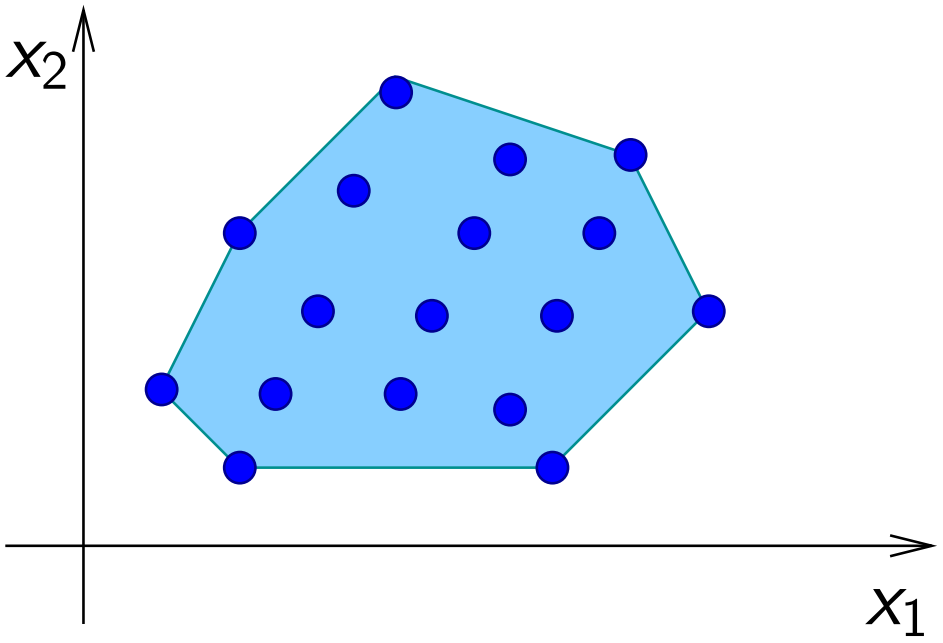
parité

intervalles

zones

octogones $x_i \pm x_j \leq c$

Domaines numériques abstraits



parité

intervalles

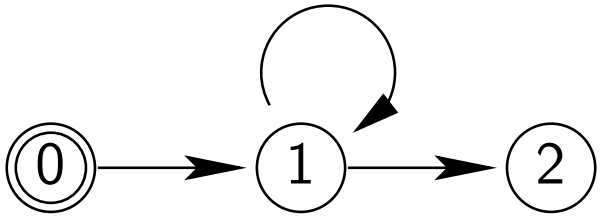
zones

octogones

polyèdres convexes $\sum a_i x_i \leq c_i$

Analyse statique

- modèles des programmes : les *automates interprétés*



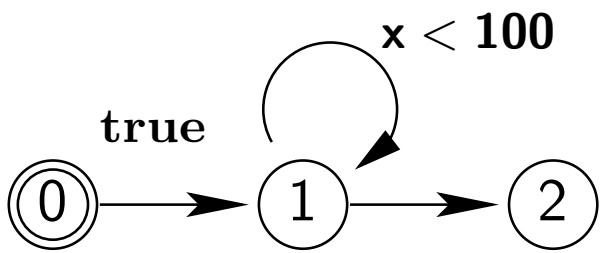
  - points de contrôles $k \in K$
  - commandes gardées
    - gardes $\mathbf{g}$
    - actions a

- sémantique collective
- résolution itérative du *système d'équations* de point-fixe.

  $\triangleright$ analyse d'accessibilité : $R^{\#}_{k_{err}} \stackrel{?}{=} \mathit{vide}$ $\triangleright$ vérification conservative

Analyse statique

- modèles des programmes : les *automates interprétés*



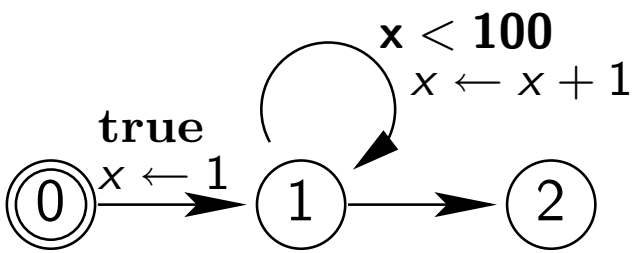
  - points de contrôles $k \in K$
  - commandes gardées
    - gardes **g**
    - actions *a*

- sémantique collective
- résolution itérative du *système d'équations* de point-fixe.

  ▷ analyse d'accessibilité : $R^{\#}_{k_{err}} \stackrel{?}{=} vide$ ▷ vérification conservative

Analyse statique

- modèles des programmes : les *automates interprétés*



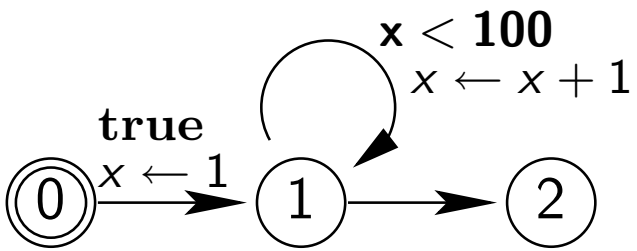
  - points de contrôles $k \in K$
  - commandes gardées
    - gardes $\mathbf{g}$
    - actions a

- sémantique collective
- résolution itérative du *système d'équations* de point-fixe.

  $\triangleright$ analyse d'accessibilité : $R^{\#}_{k_{err}} \stackrel{?}{=} vide$ $\triangleright$ vérification conservative

Analyse statique

- modèles des programmes : les *automates interprétés*



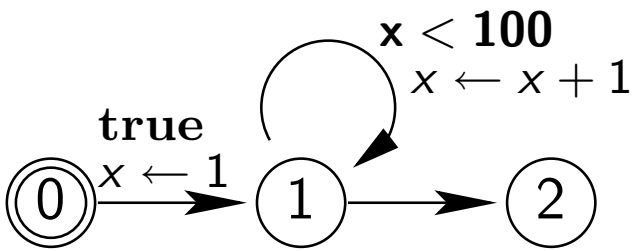
  - points de contrôles $k \in K$
  - commandes gardées
    - gardes $\mathbf{g}$
    - actions a

- sémantique collective
- résolution itérative du *système d'équations* de point-fixe.

  ▷ analyse d'accessibilité : $R^{\#}_{k_{err}} \stackrel{?}{=} vide$ ▷ vérification conservative

Analyse statique

- modèles des programmes : les *automates interprétés*

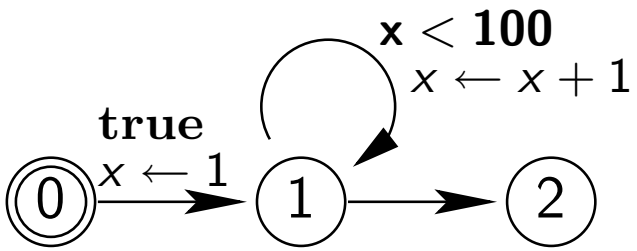


  - points de contrôles $k \in K$
  - commandes gardées
    - gardes $\mathbf{g}$
    - actions a

- sémantique collective
  $R_k = \{\textit{états accessibles en } k\}$
- résolution itérative du *système d'équations* de point-fixe.
  ▷ analyse d'accessibilité : $R^{\#}_{k_{err}} \stackrel{?}{=} \textit{vide}$ ▷ vérification conservative

Analyse statique

- modèles des programmes : les *automates interprétés*



  - points de contrôles $k \in K$
  - commandes gardées
    - gardes $\mathbf{g}$
    - actions a

- sémantique collective

  $R_k = \{\textit{états accessibles en } k\}$

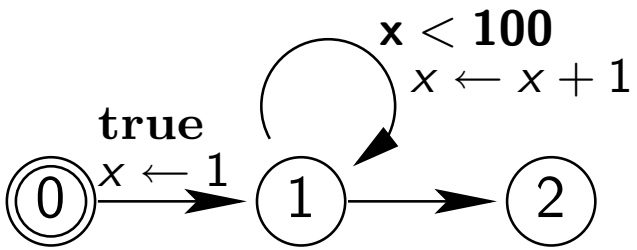
$$R_k = \bigcup_{(k',g,a,k)} a(R_{k'} \cap \overline{g})$$

- résolution itérative du *système d'équations* de point-fixe.

  ▷ analyse d'accessibilité : $R^{\#}_{k_{err}} \stackrel{?}{=} \textit{vide}$ ▷ vérification conservative

Analyse statique

- modèles des programmes : les *automates interprétés*



  - points de contrôles $k \in K$
  - commandes gardées
    - gardes $\mathbf{g}$
    - actions a

- sémantique collective *abstraite*

$R_k^\# = \{\textit{approximation des états accessibles en } k\}$

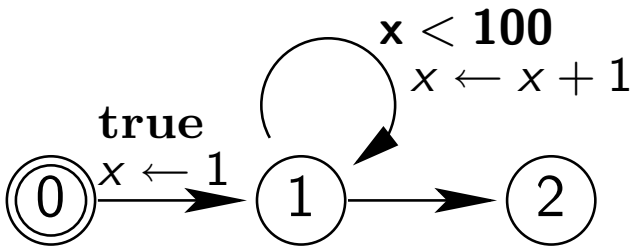
$$R_k^\# = \bigsqcup_{(k',g,a,k)}^{\#} a^\#(R_{k'}^\# \sqcap^\# \overline{g^\#})$$

- résolution itérative du *système d'équations* de point-fixe.

  ▷ analyse d'accessibilité : $R_{k_{err}}^\# \stackrel{?}{=} \textit{vide}$ ▷ vérification conservative

Analyse statique

- modèles des programmes : les *automates interprétés*



  - points de contrôles $k \in K$
  - commandes gardées
    - gardes $\mathbf{g}$
    - actions a

- sémantique collective

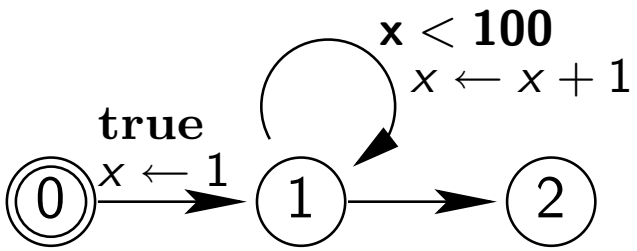
$$\textit{domaine concret} \begin{cases} R_0 = \mathbb{Z} \\ R_1 = a_1(R_0) \cup a_2(R_1 \cap (x < 100)) \\ R_2 = R_1 \cap (x \geq 100) \end{cases}$$

- résolution itérative du *système d'équations* de point-fixe.

  ▷ analyse d'accessibilité : $R^{\#}_{k_{err}} \stackrel{?}{=} \textit{vide}$ ▷ vérification conservative

Analyse statique

- modèles des programmes : les *automates interprétés*



  - points de contrôles $k \in K$
  - commandes gardées
    - gardes $\mathbf{g}$
    - actions a

- sémantique collective

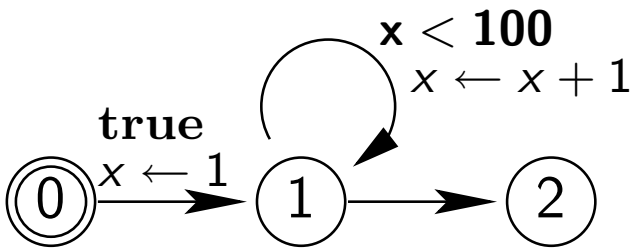
$$\textit{domaine abstrait} \begin{cases} R_0^\# = [-\infty, +\infty] \\ R_1^\# = [1,1] \sqcup ((R_1^\# \sqcap [-\infty, 99]) \oplus [1,1]) \\ R_2^\# = R_1^\# \sqcap [100, +\infty] \end{cases}$$

- résolution itérative du *système d'équations* de point-fixe.

  ▷ analyse d'accessibilité : $R_{k_{err}}^\# \stackrel{?}{=} \textit{vide}$ ▷ vérification conservative

Analyse statique

- modèles des programmes : les *automates interprétés*



  - points de contrôles $k \in K$
  - commandes gardées
    - gardes $\mathbf{g}$
    - actions a

- sémantique collective

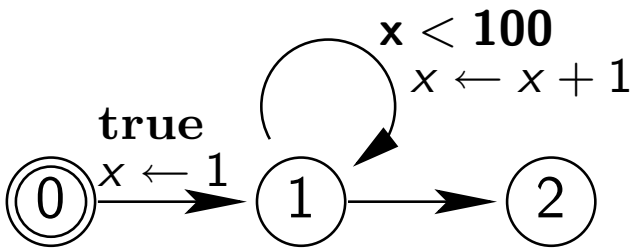
$$\textit{domaine abstrait} \begin{cases} R_0^\# = [-\infty, +\infty] \\ R_1^\# = [1,1] \sqcup ((R_1^\# \sqcap [-\infty, 99]) \oplus [1,1]) \\ R_2^\# = R_1^\# \sqcap [100, +\infty] \end{cases}$$

- résolution itérative du *système d'équations* de point-fixe.

  ▷ analyse d'accessibilité : $R^\#_{k_{err}} \stackrel{?}{=} \textit{vide}$ ▷ vérification conservative

Analyse statique

- modèles des programmes : les *automates interprétés*



  - points de contrôles $k \in K$
  - commandes gardées
    - gardes $\mathbf{g}$
    - actions a

- sémantique collective

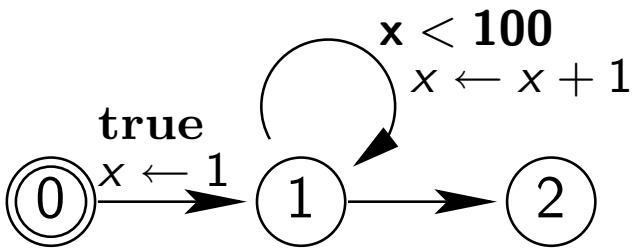
$$\textit{domaine abstrait} \begin{cases} R_0^\# = [-\infty, +\infty] \\ R_1^\# = [1,1] \sqcup ((R_1^\# \sqcap [-\infty, 99]) \oplus [1,1]) \\ R_2^\# = R_1^\# \sqcap [100, +\infty] \end{cases}$$

- résolution itérative du *système d'équations* de point-fixe.

  ▷ analyse d'accessibilité : $R_{k_{err}}^\# \stackrel{?}{=} \textit{vide}$ ▷ vérification conservative

Analyse statique

- modèles des programmes : les *automates interprétés*



  - points de contrôles $k \in K$
  - commandes gardées
    - gardes $\mathbf{g}$
    - actions a

- sémantique collective

$$\textit{domaine abstrait} \begin{cases} R_0^\# = [-\infty, +\infty] \\ R_1^\# = [1,1] \sqcup ((R_1^\# \sqcap [-\infty, 99]) \oplus [1,1]) \\ R_2^\# = R_1^\# \sqcap [100, +\infty] \end{cases}$$

- résolution itérative du *système d'équations* de point-fixe.

  ▷ analyse d'accessibilité : $R_{k_{err}}^\# \stackrel{?}{=} \textit{vide}$ ▷ vérification conservative

Plan de l'exposé

Valeurs abstraites

On note *Id* l'ensemble des variables.
$\mathcal{N}$ l'espace de leurs valeurs.

- valeurs abstraites définies directement à partir de l'intuition
- cohérence : le 7-uplet intervalle/relations est soumis à des *règles de correction*.
  - $i < j \Rightarrow i \leq j$
  - $i < j \wedge j \leq k \Rightarrow i < k$
  - $I(i) = I(j) \in \mathit{singleton} \Rightarrow i = j$
  - $i \leq j \Rightarrow \mathrm{lb}(I(i)) \leq \mathrm{lb}(I(j))$

Valeurs abstraites

On note Id l'ensemble des variables.
$\mathcal{N}$ l'espace de leurs valeurs.

- valeurs abstraites définies directement à partir de l'intuition

$$Id \xrightarrow{I} \mathcal{N}^2$$

- cohérence : le 7-uplet intervalle/relations est soumis à des *règles de correction*.
  - $i < j \Rightarrow i \leq j$
  - $i < j \wedge j \leq k \Rightarrow i < k$
  - $I(i) = I(j) \in \mathit{singleton} \Rightarrow i = j$
  - $i \leq j \Rightarrow \mathrm{lb}(I(i)) \leq \mathrm{lb}(I(j))$

Valeurs abstraites

On note Id l'ensemble des variables.
$\mathcal{N}$ l'espace de leurs valeurs.

- valeurs abstraites définies directement à partir de l'intuition

$$(Id \xrightarrow{I} \mathcal{N}^2, =, \neq, <, \leq, \geq, >)$$

- cohérence : le 7-uplet intervalle/relations est soumis à des *règles de correction*.
  - $i < j \Rightarrow i \leq j$
  - $i < j \wedge j \leq k \Rightarrow i < k$
  - $I(i) = I(j) \in singleton \Rightarrow i = j$
  - $i \leq j \Rightarrow \text{lb}(I(i)) \leq \text{lb}(I(j))$

Valeurs abstraites

On note Id l'ensemble des variables.
$\mathcal{N}$ l'espace de leurs valeurs.

- valeurs abstraites définies directement à partir de l'intuition

$$(Id \xrightarrow{I} \mathcal{N}^2, =, \neq, <, \leq, \geq, >)$$

- cohérence : le 7-uplet intervalle/relations est soumis à des *règles de correction*.
  - $i < j \Rightarrow i \leq j$
  - $i < j \wedge j \leq k \Rightarrow i < k$
  - $I(i) = I(j) \in singleton \Rightarrow i = j$
  - $i \leq j \Rightarrow \mathrm{lb}(I(i)) \leq \mathrm{lb}(I(j))$

Valeurs abstraites

On note Id l'ensemble des variables.
$\mathcal{N}$ l'espace de leurs valeurs.

- valeurs abstraites définies directement à partir de l'intuition

$$(Id \xrightarrow{I} \mathcal{N}^2, =, \neq, <, \leq, \geq, >)$$

- cohérence : le 7-uplet intervalle/relations est soumis à des *règles de correction*.
  - $i < j \Rightarrow i \leq j$
  - $i < j \wedge j \leq k \Rightarrow i < k$
  - $I(i) = I(j) \in singleton \Rightarrow i = j$
  - $i \leq j \Rightarrow \mathrm{lb}(I(i)) \leq \mathrm{lb}(I(j))$

Valeurs abstraites

On note Id l'ensemble des variables.
$\mathcal{N}$ l'espace de leurs valeurs.

- valeurs abstraites définies directement à partir de l'intuition

$$(Id \xrightarrow{I} \mathcal{N}^2, =, \neq, <, \leq, \geq, >)$$

- cohérence : le 7-uplet intervalle/relations est soumis à des *règles de correction*.
  - $i < j \Rightarrow i \leq j$
  - $i < j \wedge j \leq k \Rightarrow i < k$
  - $I(i) = I(j) \in singleton \Rightarrow i = j$
  - $i \leq j \Rightarrow \mathrm{lb}(I(i)) \leq \mathrm{lb}(I(j))$

Valeurs abstraites

On note Id l'ensemble des variables.
$\mathcal{N}$ l'espace de leurs valeurs.

- valeurs abstraites définies directement à partir de l'intuition

$$(Id \xrightarrow{I} \mathcal{N}^2, =, \neq, <, \leq, \geq, >)$$

- cohérence : le 7-uplet intervalle/relations est soumis à des *règles de correction*.
  - $i < j \Rightarrow i \leq j$
  - $i < j \wedge j \leq k \Rightarrow i < k$
  - $I(i) = I(j) \in singleton \Rightarrow i = j$
  - $i \leq j \Rightarrow \mathrm{lb}(I(i)) \leq \mathrm{lb}(I(j))$

Définition minimale

- on peut simplifier
- on définit l'ensemble des règles de correction

[Meet] $\neg(j \neq i \wedge j \leq i \wedge i \leq j)$
[Eq] $j = i \Leftrightarrow (\forall s \in S^{\#} \quad I\ s\ i \Leftrightarrow I\ s\ j) \wedge (I(i) = I(j))$
[Neq] $\neg(i \neq i)$
[Rfl] $i \leq i$
[AntiRfl] $j \neq i \Leftrightarrow i \neq j$
[TransEq] $\forall s \in \{\leq, \geq\} \quad I\ s\ j \wedge j\ s\ i \Rightarrow I\ s\ i$
[TransNeq1] $\forall s \in \{<, \leq\} \quad I\ s\ j \wedge j < i \Rightarrow I < i$
[TransNeq2] $I < j \wedge j \leq i \Rightarrow I < i$
[C2I] $\forall s \in S^{\bowtie} \quad i\ s\ j \Rightarrow (\mathrm{lb}(I(i))\ s\ \mathrm{lb}(I(j)) \wedge (\mathrm{ub}(I(i))\ s\ \mathrm{ub}(I(j))$
[C2Ineq] $i \neq j \quad I(j) \in \mathrm{Singl} \Rightarrow I(i) \sqcap^I I(j) \neq \mathrm{lb}(I(i)) \neq \mathrm{ub}(I(i))$
[I2Ceq] $I(i) = I(j) \in \mathrm{Singl} \Rightarrow i = j$
[I2Cempty] $I(i) \sqcap^I I(j) = \bot^I \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) < \mathrm{lb}(I(j)) \wedge i < j) \\ \vee\ (\mathrm{lb}(I(i)) > \mathrm{ub}(I(j)) \wedge i > j) \end{cases}$
[I2Csingl] $I(i) \sqcap^I I(j) \in \mathrm{Singl} \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) = \mathrm{lb}(I(j)) \wedge i \leq j) \\ \vee\ (\mathrm{lb}(I(i)) = \mathrm{ub}(I(j)) \wedge i \geq j) \end{cases}$

Définition minimale

- on peut simplifier

$$(Id \xrightarrow{I} \mathcal{N}^2, =, \neq, <, \leq, \geq, >)$$

- on définit l'ensemble des règles de correction

[Meet] $\neg(j \neq i \wedge j \leq i \wedge i \leq j)$
[Eq] $j = i \Leftrightarrow (\forall s \in S^{\#} \quad I\ s\ i \Leftrightarrow I\ s\ j) \wedge (I(i) = I(j))$
[Neq] $\neg(i \neq i)$
[Rfl] $i \leq i$
[AntiRfl] $j \neq i \Leftrightarrow i \neq j$
[TransEq] $\forall s \in \{\leq, \geq\} \quad I\ s\ j \wedge j\ s\ i \Rightarrow I\ s\ i$
[TransNeq1] $\forall s \in \{<, \leq\} \quad I\ s\ j \wedge j < i \Rightarrow I < i$
[TransNeq2] $I < j \wedge j \leq i \Rightarrow I < i$
[C2I] $\forall s \in S^{\bowtie} \quad i\ s\ j \Rightarrow (\mathrm{lb}(I(i))\ s\ \mathrm{lb}(I(j)) \wedge (\mathrm{ub}(I(i))\ s\ \mathrm{ub}(I(j))$
[C2Ineq] $i \neq j \quad I(j) \in \mathrm{Singl} \Rightarrow I(i) \sqcap^I I(j) \neq \mathrm{lb}(I(i)) \neq \mathrm{ub}(I(i))$
[I2Ceq] $I(i) = I(j) \in \mathrm{Singl} \Rightarrow i = j$
[I2Cempty] $I(i) \sqcap^I I(j) = \bot^I \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) < \mathrm{lb}(I(j)) \wedge i < j) \\ \vee\ (\mathrm{lb}(I(i)) > \mathrm{ub}(I(j)) \wedge i > j) \end{cases}$
[I2Csingl] $I(i) \sqcap^I I(j) \in \mathrm{Singl} \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) = \mathrm{lb}(I(j)) \wedge i \leq j) \\ \vee\ (\mathrm{lb}(I(i)) = \mathrm{ub}(I(j)) \wedge i \geq j) \end{cases}$

Définition minimale

- on peut simplifier

$$(Id \xrightarrow{I} \mathcal{N}^2, =, \neq, <, \leq)$$

- on définit l'ensemble des règles de correction

[Meet] $\neg(j \neq i \wedge j \leq i \wedge i \leq j)$
[Eq] $j = i \Leftrightarrow (\forall s \in S^{\#} \quad I\ \mathbf{s}\ i \Leftrightarrow I\ \mathbf{s}\ j) \wedge (I(i) = I(j))$
[Neq] $\neg(i \neq i)$
[Rfl] $i \leq i$
[AntiRfl] $j \neq i \Leftrightarrow i \neq j$
[TransEq] $\forall s \in \{\leq, \geq\} \quad I\ \mathbf{s}\ j \wedge j\ \mathbf{s}\ i \Rightarrow I\ \mathbf{s}\ i$
[TransNeq1] $\forall s \in \{<, \leq\} \quad I\ \mathbf{s}\ j \wedge j < i \Rightarrow I < i$
[TransNeq2] $I < j \wedge j \leq i \Rightarrow I < i$
[C2I] $\forall s \in S^{\bowtie} \quad i\ \mathbf{s}\ j \Rightarrow (\mathrm{lb}(I(i))\ \mathrm{s}\ \mathrm{lb}(I(j)) \wedge (\mathrm{ub}(I(i))\ \mathrm{s}\ \mathrm{ub}(I(j))$
[C2Ineq] $i \neq j \quad I(j) \in \mathrm{Singl} \Rightarrow I(i) \sqcap^I I(j) \neq \mathrm{lb}(I(i)) \neq \mathrm{ub}(I(i))$
[I2Ceq] $I(i) = I(j) \in \mathrm{Singl} \Rightarrow i = j$
[I2Cempty] $I(i) \sqcap^I I(j) = \bot^I \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) < \mathrm{lb}(I(j)) \wedge i < j) \\ \vee\ (\mathrm{lb}(I(i)) > \mathrm{ub}(I(j)) \wedge i > j) \end{cases}$
[I2Csingl] $I(i) \sqcap^I I(j) \in \mathrm{Singl} \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) = \mathrm{lb}(I(j)) \wedge i \leq j) \\ \vee\ (\mathrm{lb}(I(i)) = \mathrm{ub}(I(j)) \wedge i \geq j) \end{cases}$

Définition minimale

- on peut simplifier
  - $(Id \xrightarrow{I} \mathcal{N}^2, =, <)$
  - $(Id \xrightarrow{I} \mathcal{N}^2, \neq, \leq)$
- on définit l'ensemble des règles de correction

[Meet] $\neg(j \neq i \wedge j \leq i \wedge i \leq j)$
[Eq] $j = i \Leftrightarrow (\forall s \in S^{\#} \quad I\ s\ i \Leftrightarrow I\ s\ j) \wedge (I(i) = I(j))$
[Neq] $\neg(i \neq i)$
[Rfl] $i \leq i$
[AntiRfl] $j \neq i \Leftrightarrow i \neq j$
[TransEq] $\forall s \in \{\leq, \geq\} \quad I\ s\ j \wedge j\ s\ i \Rightarrow I\ s\ i$
[TransNeq1] $\forall s \in \{<, \leq\} \quad I\ s\ j \wedge j < i \Rightarrow I < i$
[TransNeq2] $I < j \wedge j \leq i \Rightarrow I < i$
[C2I] $\forall s \in S^{\bowtie} \quad i\ s\ j \Rightarrow (\mathrm{lb}(I(i))\ s\ \mathrm{lb}(I(j)) \wedge (\mathrm{ub}(I(i))\ s\ \mathrm{ub}(I(j))$
[C2Ineq] $i \neq j \quad I(j) \in \mathrm{Singl} \Rightarrow I(i) \sqcap^I I(j) \neq \mathrm{lb}(I(i)) \neq \mathrm{ub}(I(i))$
[I2Ceq] $I(i) = I(j) \in \mathrm{Singl} \Rightarrow i = j$
[I2Cempty] $I(i) \sqcap^I I(j) = \bot^I \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) < \mathrm{lb}(I(j)) \wedge i < j) \\ \vee\ (\mathrm{lb}(I(i)) > \mathrm{ub}(I(j)) \wedge i > j) \end{cases}$
[I2Csingl] $I(i) \sqcap^I I(j) \in \mathrm{Singl} \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) = \mathrm{lb}(I(j)) \wedge i \leq j) \\ \vee\ (\mathrm{lb}(I(i)) = \mathrm{ub}(I(j)) \wedge i \geq j) \end{cases}$

Définition minimale

- on peut simplifier
  - $(Id \xrightarrow{I} \mathcal{N}^2, =, <)$
    $\triangleright \neq = \{(i,j) \mid i < j \vee i > j\}$
  - $(Id \xrightarrow{I} \mathcal{N}^2, \neq, \leq)$
- on définit l'ensemble des règles de correction

[Meet] $\neg(j \neq i \wedge j \leq i \wedge i \leq j)$
[Eq] $j = i \Leftrightarrow (\forall s \in S^{\#} \quad I\ s\ i \Leftrightarrow I\ s\ j) \wedge (I(i) = I(j))$
[Neq] $\neg(i \neq i)$
[Rfl] $i \leq i$
[AntiRfl] $j \neq i \Leftrightarrow i \neq j$
[TransEq] $\forall s \in \{\leq, \geq\} \quad I\ s\ j \wedge j\ s\ i \Rightarrow I\ s\ i$
[TransNeq1] $\forall s \in \{<, \leq\} \quad I\ s\ j \wedge j < i \Rightarrow I < i$
[TransNeq2] $I < j \wedge j \leq i \Rightarrow I < i$
[C2I] $\forall s \in S^{\bowtie} \quad i\ s\ j \Rightarrow (\mathrm{lb}(I(i))\ s\ \mathrm{lb}(I(j)) \wedge (\mathrm{ub}(I(i))\ s\ \mathrm{ub}(I(j))$
[C2Ineq] $i \neq j \quad I(j) \in \mathrm{Singl} \Rightarrow I(i) \sqcap^I I(j) \neq \mathrm{lb}(I(i)) \neq \mathrm{ub}(I(i))$
[I2Ceq] $I(i) = I(j) \in \mathrm{Singl} \Rightarrow i = j$
[I2Cempty] $I(i) \sqcap^I I(j) = \bot^I \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) < \mathrm{lb}(I(j)) \wedge i < j) \\ \vee\ (\mathrm{lb}(I(i)) > \mathrm{ub}(I(j)) \wedge i > j) \end{cases}$
[I2Csingl] $I(i) \sqcap^I I(j) \in \mathrm{Singl} \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) = \mathrm{lb}(I(j)) \wedge i \leq j) \\ \vee\ (\mathrm{lb}(I(i)) = \mathrm{ub}(I(j)) \wedge i \geq j) \end{cases}$

Définition minimale

- on peut simplifier
  - $(Id \xrightarrow{I} \mathcal{N}^2, =, <)$
    $\triangleright \quad \neq \equiv \{(i,j) \mid i < j \vee i > j\}$
    $\triangleright$ on peut savoir que $i \neq j$ sans savoir laquelle de i, j est supérieure stricte à l'autre.
  - $(Id \xrightarrow{I} \mathcal{N}^2, \neq, \leq)$

- on définit l'ensemble des règles de correction

[Meet] $\neg(j \neq i \wedge j \leq i \wedge i \leq j)$
[Eq] $j = i \Leftrightarrow (\forall s \in S^{\#} \quad I\ s\ i \Leftrightarrow I\ s\ j) \wedge (I(i) = I(j))$
[Neq] $\neg(i \neq i)$
[Rfl] $i \leq i$
[AntiRfl] $j \neq i \Leftrightarrow i \neq j$
[TransEq] $\forall s \in \{\leq, \geq\} \quad I\ s\ j \wedge j\ s\ i \Rightarrow I\ s\ i$
[TransNeq1] $\forall s \in \{<, \leq\} \quad I\ s\ j \wedge j < i \Rightarrow I < i$
[TransNeq2] $I < j \wedge j \leq i \Rightarrow I < i$
[C2I] $\forall s \in S^{\bowtie} \quad i\ s\ j \Rightarrow (\mathrm{lb}(I(i))\ s\ \mathrm{lb}(I(j)) \wedge (\mathrm{ub}(I(i))\ s\ \mathrm{ub}(I(j))$
[C2Ineq] $i \neq j \quad I(j) \in \mathrm{Singl} \Rightarrow I(i) \sqcap^I I(j) \neq \mathrm{lb}(I(i)) \neq \mathrm{ub}(I(i))$
[I2Ceq] $I(i) = I(j) \in \mathrm{Singl} \Rightarrow i = j$

Définition minimale

- on peut simplifier
  - $(Id \xrightarrow{I} \mathcal{N}^2, =, <)$
  - $(Id \xrightarrow{I} \mathcal{N}^2, \neq, \leq)$
    $\triangleright \quad < \equiv \{(i,j) \mid i \leq j \wedge i \neq j\}$
    $\triangleright \quad = \equiv \{(i,j) \mid i \leq j \wedge i \geq j\}$
- on définit l'ensemble des règles de correction

[Meet] $\neg(j \neq i \wedge j \leq i \wedge i \leq j)$
[Eq] $j = i \Leftrightarrow (\forall s \in S^{\#} \quad I\ s\ i \Leftrightarrow I\ s\ j) \wedge (I(i) = I(j))$
[Neq] $\neg(i \neq i)$
[Rfl] $i \leq i$
[AntiRfl] $j \neq i \Leftrightarrow i \neq j$
[TransEq] $\forall s \in \{\leq, \geq\} \quad I\ s\ j \wedge j\ s\ i \Rightarrow I\ s\ i$
[TransNeq1] $\forall s \in \{<, \leq\} \quad I\ s\ j \wedge j < i \Rightarrow I < i$
[TransNeq2] $I < j \wedge j \leq i \Rightarrow I < i$
[C2I] $\forall s \in S^{\bowtie} \quad i\ s\ j \Rightarrow (\mathrm{lb}(I(i))\ s\ \mathrm{lb}(I(j)) \wedge (\mathrm{ub}(I(i))\ s\ \mathrm{ub}(I(j))$
[C2Ineq] $i \neq j \quad I(j) \in \mathrm{Singl} \Rightarrow I(i) \sqcap^I I(j) \neq \mathrm{lb}(I(i)) \neq \mathrm{ub}(I(i))$
[I2Ceq] $I(i) = I(j) \in \mathrm{Singl} \Rightarrow i = j$
[I2Cempty] $I(i) \sqcap^I I(j) = \bot^I \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) < \mathrm{lb}(I(j)) \wedge i < j) \\ \vee\ (\mathrm{lb}(I(i)) > \mathrm{ub}(I(j)) \wedge i > j) \end{cases}$
[I2Csingl] $I(i) \sqcap^I I(j) \in \mathrm{Singl} \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) = \mathrm{lb}(I(j)) \wedge i \leq j) \\ \vee\ (\mathrm{lb}(I(i)) = \mathrm{ub}(I(i)) \wedge i > j) \end{cases}$

Définition minimale

- on peut simplifier
  - $(Id \xrightarrow{I} \mathcal{N}^2, =, <)$
  - $(Id \xrightarrow{I} \mathcal{N}^2, \neq, \leq)$
    $\triangleright \quad < \equiv \{(i,j) \mid i \leq j \wedge i \neq j\}$
    $\triangleright \quad = \equiv \{(i,j) \mid i \leq j \wedge i \geq j\}$

- on définit l'ensemble des règles de correction

[Meet] $\neg(j \neq i \wedge j \leq i \wedge i \leq j)$
[Eq] $j = i \Leftrightarrow (\forall s \in S^{\#} \quad I\ \mathbf{s}\ i \Leftrightarrow I\ \mathbf{s}\ j) \wedge (I(i) = I(j))$
[Neq] $\neg(i \neq i)$
[Rfl] $i \leq i$
[AntiRfl] $j \neq i \Leftrightarrow i \neq j$
[TransEq] $\forall s \in \{\leq, \geq\} \quad I\ \mathbf{s}\ j \wedge j\ \mathbf{s}\ i \Rightarrow I\ \mathbf{s}\ i$
[TransNeq1] $\forall s \in \{<, \leq\} \quad I\ \mathbf{s}\ j \wedge j < i \Rightarrow I < i$
[TransNeq2] $I < j \wedge j \leq i \Rightarrow I < i$
[C2I] $\forall s \in S^{\bowtie} \quad i\ \mathbf{s}\ j \Rightarrow (\mathrm{lb}(I(i))\ s\ \mathrm{lb}(I(j)) \wedge (\mathrm{ub}(I(i))\ s\ \mathrm{ub}(I(j))$
[C2Ineq] $i \neq j \quad I(j) \in \mathrm{Singl} \Rightarrow I(i) \sqcap^I I(j) \neq \mathrm{lb}(I(i)) \neq \mathrm{ub}(I(i))$
[I2Ceq] $I(i) = I(j) \in \mathrm{Singl} \Rightarrow i = j$
[I2Cempty] $I(i) \sqcap^I I(j) = \bot^I \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) < \mathrm{lb}(I(j)) \wedge i < j) \\ \vee\ (\mathrm{lb}(I(i)) > \mathrm{ub}(I(j)) \wedge i > j) \end{cases}$
[I2Csingl] $I(i) \sqcap^I I(j) \in \mathrm{Singl} \Rightarrow \begin{cases} (\mathrm{ub}(I(i)) = \mathrm{lb}(I(j)) \wedge i \leq j) \\ \vee\ (\mathrm{lb}(I(i)) = \mathrm{ub}(I(i)) \wedge i \geq j) \end{cases}$

Canonisation

- canoniser une valeur abstraite $\equiv$ tirer toutes les conséquences des informations qu'elle contient.

- on a montré :
  - la terminaison
  - la confluence *(paires critiques)*
  - la correction des formes normales
- on a une représentation canonique et un algorithme de canonisation

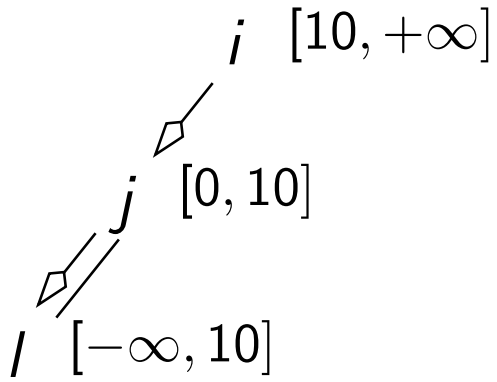
Canonisation

- canoniser une valeur abstraite $\equiv$ tirer toutes les conséquences des informations qu'elle contient.
  $\triangleright$ définition d'un système de réécriture de valeurs abstraites

- on a montré :
  - la terminaison
  - la confluence *(paires critiques)*
  - la correction des formes normales
- on a une représentation canonique et un algorithme de canonisation

Canonisation

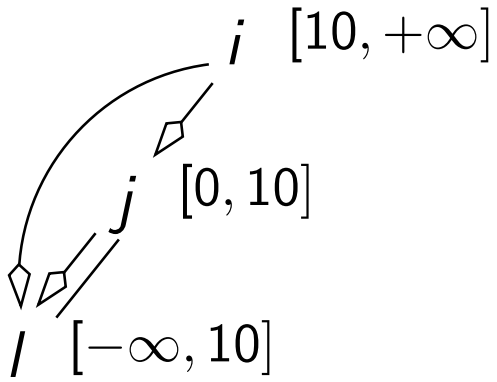
- canoniser une valeur abstraite $\equiv$ tirer toutes les conséquences des informations qu'elle contient.
  $\triangleright$ définition d'un système de réécriture de valeurs abstraites



- on a montré :
  - la terminaison
  - la confluence *(paires critiques)*
  - la correction des formes normales
- on a une représentation canonique et un algorithme de canonisation

Canonisation

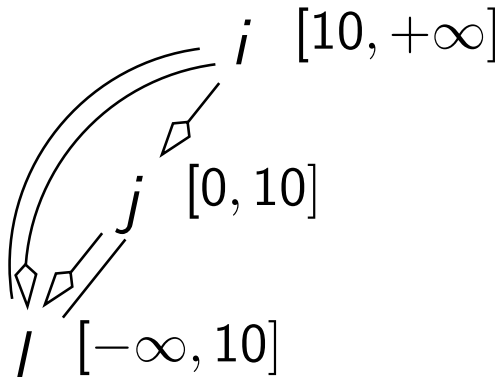
- canoniser une valeur abstraite $\equiv$ tirer toutes les conséquences des informations qu'elle contient.
  $\triangleright$ définition d'un système de réécriture de valeurs abstraites



- on a montré :
  - la terminaison
  - la confluence *(paires critiques)*
  - la correction des formes normales
- on a une représentation canonique et un algorithme de canonisation

Canonisation

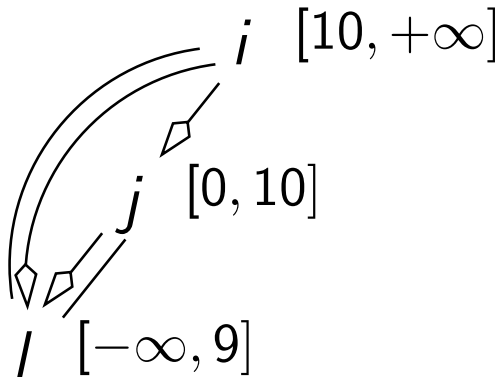
- canoniser une valeur abstraite $\equiv$ tirer toutes les conséquences des informations qu'elle contient.
  $\triangleright$ définition d'un système de réécriture de valeurs abstraites



- on a montré :
  - la terminaison
  - la confluence *(paires critiques)*
  - la correction des formes normales
- on a une représentation canonique et un algorithme de canonisation

Canonisation

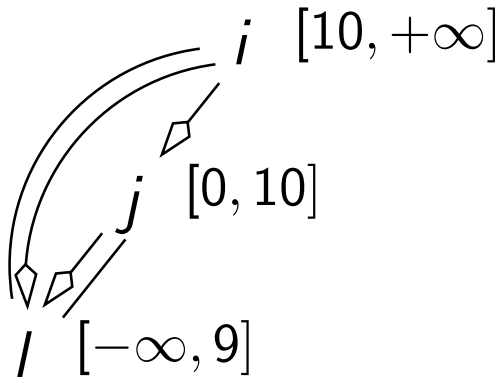
- canoniser une valeur abstraite $\equiv$ tirer toutes les conséquences des informations qu'elle contient.
  $\triangleright$ définition d'un système de réécriture de valeurs abstraites



- on a montré :
  - la terminaison
  - la confluence *(paires critiques)*
  - la correction des formes normales
- on a une représentation canonique et un algorithme de canonisation

Canonisation

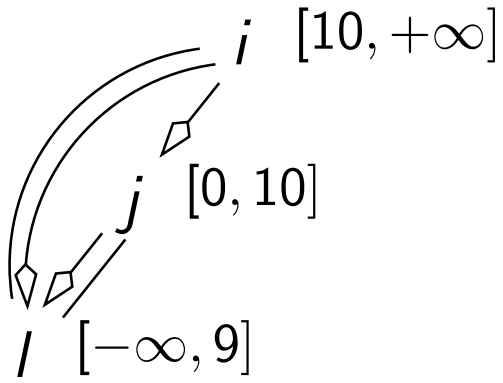
- canoniser une valeur abstraite $\equiv$ tirer toutes les conséquences des informations qu'elle contient.
  $\triangleright$ définition d'un système de réécriture de valeurs abstraites



- on a montré :
  - la terminaison
  - la confluence *(paires critiques)*
  - la correction des formes normales
- on a une représentation canonique et un algorithme de canonisation

Canonisation

- canoniser une valeur abstraite $\equiv$ tirer toutes les conséquences des informations qu'elle contient.
  $\triangleright$ définition d'un système de réécriture de valeurs abstraites



- on a montré :
  - la terminaison
  - la confluence *(paires critiques)*
  - la correction des formes normales
- on a une représentation canonique et un algorithme de canonisation

Opérations de treillis

- la borne inférieure : regrouper toutes les informations connues
  - ▷ intersection des intervalles
  - ▷ union des relations
- la borne supérieure : garder les contraintes communes
  - ▷ borne supérieure des intervalles
  - ▷ intersection des relations

$$\mathcal{A} \sqcup^{\#} \mathcal{B} = \begin{pmatrix} i \to (& [0,20], & \{j\}, & \{i\} &) \\ j \to (& [10,30], & \{i\}, & \{i,j\} &) \end{pmatrix}$$

$$\begin{pmatrix} i \to (& [10,20], & \{j\}, & \{i\} &) \\ j \to (& [15,25], & \{i\}, & \{i,j\} &) \end{pmatrix} \quad \begin{pmatrix} i \to (& [0,15], & \emptyset, & \{i\} &) \\ j \to (& [10,30], & \emptyset, & \{i,j\} &) \end{pmatrix}$$

$$\mathcal{A} \sqcap^{\#} \mathcal{B} = \begin{pmatrix} i \to (& [0,15], & \emptyset, & \{i\} &) \\ j \to (& [15,25], & \emptyset, & \{i,j\} &) \end{pmatrix}$$

Opérations de treillis

- la borne inférieure : regrouper toutes les informations connues
  - ▷ intersection des intervalles
  - ▷ union des relations
- la borne supérieure : garder les contraintes communes
  - ▷ borne supérieure des intervalles
  - ▷ intersection des relations

$$\mathcal{A} \sqcup^{\#} \mathcal{B} = \begin{pmatrix} i \to (& [0,20], & \{j\}, & \{i\} &) \\ j \to (& [10,30], & \{i\}, & \{i,j\} &) \end{pmatrix}$$

$$\begin{pmatrix} i \to (& [10,20], & \{j\}, & \{i\} &) \\ j \to (& [15,25], & \{i\}, & \{i,j\} &) \end{pmatrix} \quad \begin{pmatrix} i \to (& [0,15], & \emptyset, & \{i\} &) \\ j \to (& [10,30], & \emptyset, & \{i,j\} &) \end{pmatrix}$$

$$\mathcal{A} \sqcap^{\#} \mathcal{B} = \begin{pmatrix} i \to (& [0,15], & \emptyset, & \{i\} &) \\ j \to (& [15,25], & \emptyset, & \{i,j\} &) \end{pmatrix}$$

Opérations de treillis

- la borne inférieure : regrouper toutes les informations connues
  - ▷ intersection des intervalles
  - ▷ union des relations
- la borne supérieure : garder les contraintes communes
  - ▷ borne supérieure des intervalles
  - ▷ intersection des relations

$$\mathcal{A} \sqcup^{\#} \mathcal{B} = \begin{pmatrix} i \to (& [0,20], & \{j\}, & \{i\} &) \\ j \to (& [10,30], & \{i\}, & \{i,j\} &) \end{pmatrix}$$

$$\begin{pmatrix} i \to (& [10,20], & \{j\}, & \{i\} &) \\ j \to (& [15,25], & \{i\}, & \{i,j\} &) \end{pmatrix} \quad \begin{pmatrix} i \to (& [0,15], & \emptyset, & \{i\} &) \\ j \to (& [10,30], & \emptyset, & \{i,j\} &) \end{pmatrix}$$

$$\mathcal{A} \sqcap^{\#} \mathcal{B} = \begin{pmatrix} i \to (& [0,15], & \emptyset, & \{i\} &) \\ j \to (& [15,25], & \emptyset, & \{i,j\} &) \end{pmatrix}$$

Opérations de treillis

- la borne inférieure : regrouper toutes les informations connues
  - ▷ intersection des intervalles
  - ▷ union des relations
- la borne supérieure : garder les contraintes communes
  - ▷ borne supérieure des intervalles
  - ▷ intersection des relations

$$\mathcal{A} \sqcup^{\#} \mathcal{B} = \begin{pmatrix} i \to (& [0,20], & \{j\}, & \{i\} &) \\ j \to (& [10,30], & \{i\}, & \{i,j\} &) \end{pmatrix}$$

$$\begin{pmatrix} i \to (& [10,20], & \{j\}, & \{i\} &) \\ j \to (& [15,25], & \{i\}, & \{i,j\} &) \end{pmatrix} \quad \begin{pmatrix} i \to (& [0,15], & \emptyset, & \{i\} &) \\ j \to (& [10,30], & \emptyset, & \{i,j\} &) \end{pmatrix}$$

$$\mathcal{A} \sqcap^{\#} \mathcal{B} = \begin{pmatrix} i \to (& [0,15], & \emptyset, & \{i\} &) \\ j \to (& [15,25], & \emptyset, & \{i,j\} &) \end{pmatrix}$$

Opérations de treillis

- la borne inférieure : regrouper toutes les informations connues
  - ▷ intersection des intervalles
  - ▷ union des relations
- la borne supérieure : garder les contraintes communes
  - ▷ borne supérieure des intervalles
  - ▷ intersection des relations

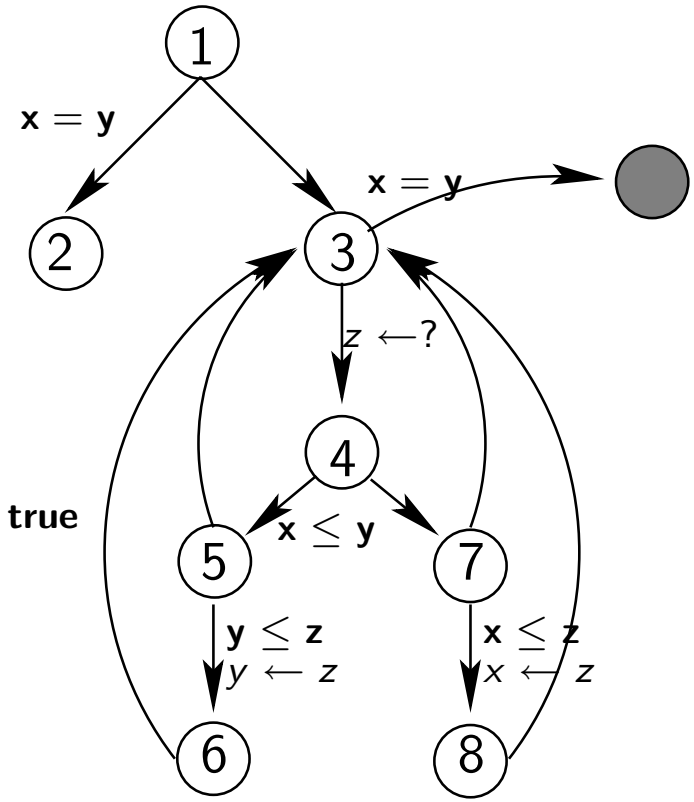
$$\mathcal{A} \sqcup^{\#} \mathcal{B} = \begin{pmatrix} i \to (& [0,20], & \{j\}, & \{i\} &) \\ j \to (& [10,30], & \{i\}, & \{i,j\} &) \end{pmatrix}$$

$$\begin{pmatrix} i \to (& [10,20], & \{j\}, & \{i\} &) \\ j \to (& [15,25], & \{i\}, & \{i,j\} &) \end{pmatrix} \quad \begin{pmatrix} i \to (& [0,15], & \emptyset, & \{i\} &) \\ j \to (& [10,30], & \emptyset, & \{i,j\} &) \end{pmatrix}$$

$$\mathcal{A} \sqcap^{\#} \mathcal{B} = \begin{pmatrix} i \to (& [0,15], & \emptyset, & \{i\} &) \\ j \to (& [15,25], & \emptyset, & \{i,j\} &) \end{pmatrix}$$

Exemple

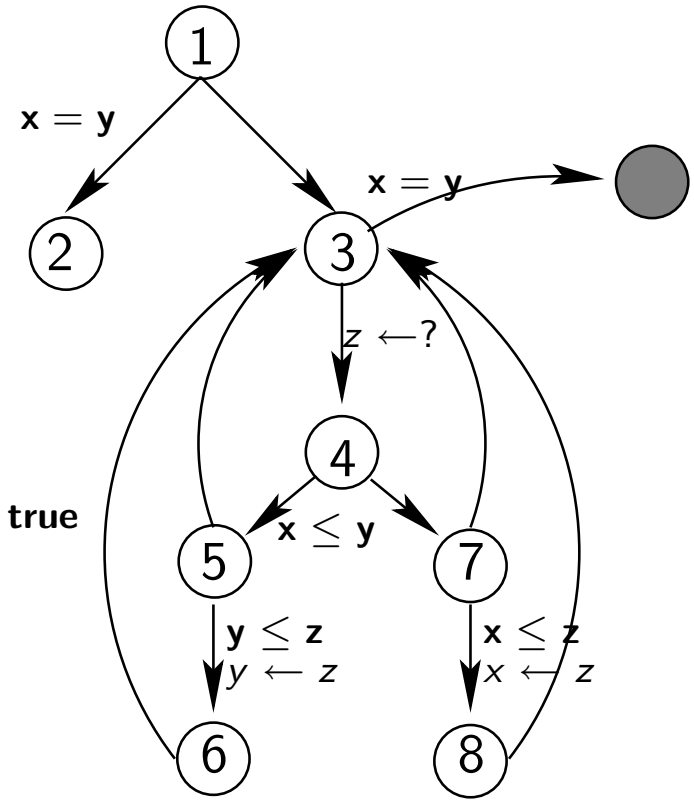
- on donne la syntaxe de nos gardes et de nos actions



- et on donne leur sémantique abstraite

Exemple

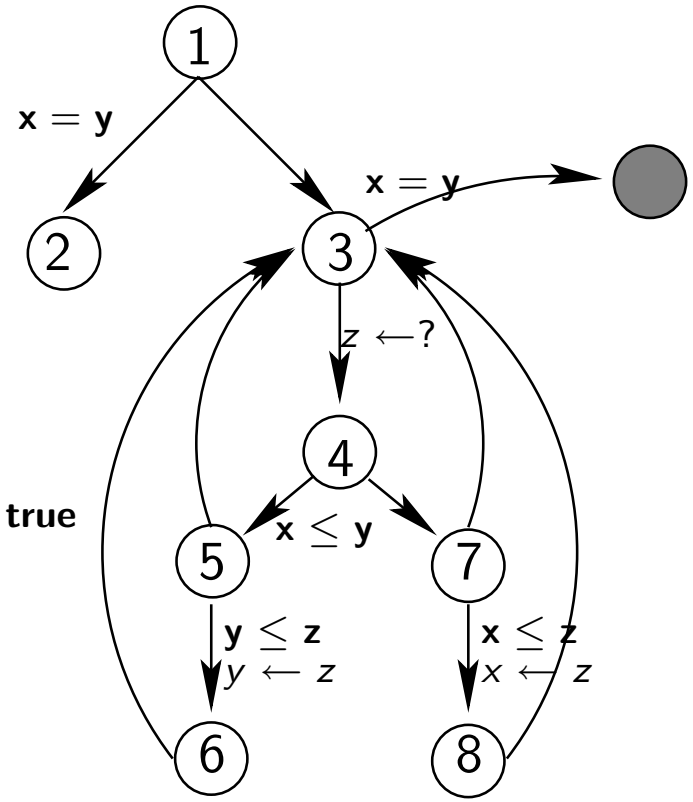
- on donne la syntaxe de nos gardes et de nos actions



- et on donne leur sémantique abstraite

Exemple (suite)

- itération par itération, sur l'automate



Plan de l'exposé

Analyseur

- sous partie de NBAC, outil de vérification de programmes asynchrones, écrit en OCAML
  - ▷ premier utilisateur
- implémentation du sous-treillis abstrait des intervalles
- implémentation de notre domaine abstrait
  - valeurs abstraites
  - opérateurs : borne inférieure, borne supérieure, élargissement, ...
  - système de réécriture de canonisation
- implémentation (en cours) d'un parser OC vers le format interne

Analyseur

- sous partie de NBAC, outil de vérification de programmes asynchrones, écrit en OCAML
  ▷ premier utilisateur
- implémentation du sous-treillis abstrait des intervalles
- implémentation de notre domaine abstrait
  - valeurs abstraites
  - opérateurs : borne inférieure, borne supérieure, élargissement, ...
  - système de réécriture de canonisation
- implémentation (en cours) d'un parser OC vers le format interne

Analyseur

- sous partie de NBAC, outil de vérification de programmes asynchrones, écrit en OCAML
  ▷ premier utilisateur
- implémentation du sous-treillis abstrait des intervalles
- implémentation de notre domaine abstrait
  - valeurs abstraites
  - opérateurs : borne inférieure, borne supérieure, élargissement, ...
  - système de réécriture de canonisation
- implémentation (en cours) d'un parser OC vers le format interne

Analyseur

- sous partie de NBAC, outil de vérification de programmes asynchrones, écrit en OCAML
  ▷ premier utilisateur
- implémentation du sous-treillis abstrait des intervalles
- implémentation de notre domaine abstrait
  - valeurs abstraites
  - opérateurs : borne inférieure, borne supérieure, élargissement, ...
  - système de réécriture de canonisation
- implémentation (en cours) d'un parser OC vers le format interne

Analyseur

- sous partie de NBAC, outil de vérification de programmes asynchrones, écrit en OCAML
  ▷ premier utilisateur
- implémentation du sous-treillis abstrait des intervalles
- implémentation de notre domaine abstrait
  - valeurs abstraites
  - opérateurs : borne inférieure, borne supérieure, élargissement, ...
  - système de réécriture de canonisation
- implémentation (en cours) d'un parser OC vers le format interne

Résultats

- parcours
- bakery

Conclusions

- contributions
  - un nouveau domaine abstrait, pour l'analyse dédiée aux adresses
    ▷ premier à inclure des propriétés de non-égalité
  - implémentation de ce domaine dans le cadre de l'analyseur de NBAC
- recul sur le domaine
  - théorie de l'interprétation abstraite
  - étude des domaines numériques existant
  - difficulté de la conception d'un domaine abstrait

Conclusions

- contributions
  - un nouveau domaine abstrait, pour l'analyse dédiée aux adresses
    ▷ premier à inclure des propriétés de non-égalité
  - implémentation de ce domaine dans le cadre de l'analyseur de NBAC
- recul sur le domaine
  - théorie de l'interprétation abstraite
  - étude des domaines numériques existant
  - difficulté de la conception d'un domaine abstrait

Conclusions

- contributions
  - un nouveau domaine abstrait, pour l'analyse dédiée aux adresses
    ▷ premier à inclure des propriétés de non-égalité
  - implémentation de ce domaine dans le cadre de l'analyseur de NBAC
- recul sur le domaine
  - théorie de l'interprétation abstraite
  - étude des domaines numériques existant
  - difficulté de la conception d'un domaine abstrait

Perspectives

- terminer le parseur
  - ▷ continuer à travailler sur NBAC pour APRON
- améliorer l'algorithmique (DBMs)
- utiliser le domaine abstrait
  - pour l'analyse de programmes numériques généraux
  - comme brique pour définir une *analyse de systèmes concurrents*

Perspectives

- terminer le parseur
  ▷ continuer à travailler sur NBAC pour APRON
- améliorer l'algorithmique (DBMs)
- utiliser le domaine abstrait
  - pour l'analyse de programmes numériques généraux
  - comme brique pour définir une *analyse de systèmes concurrents*

Perspectives

- terminer le parseur
  ▷ continuer à travailler sur NBAC pour APRON
- améliorer l'algorithmique (DBMs)
- utiliser le domaine abstrait
  - pour l'analyse de programmes numériques généraux
  - comme brique pour définir une *analyse de systèmes concurrents*

Perspectives

- terminer le parseur
  ▷ continuer à travailler sur NBAC pour APRON
- améliorer l'algorithmique (DBMs)
- utiliser le domaine abstrait
  - pour l'analyse de programmes numériques généraux
  - comme brique pour définir une *analyse de systèmes concurrents*

Merci pour votre attention.

questions ?