La bibliothèque BDDAPRON de domaines abstraits logico-numériques

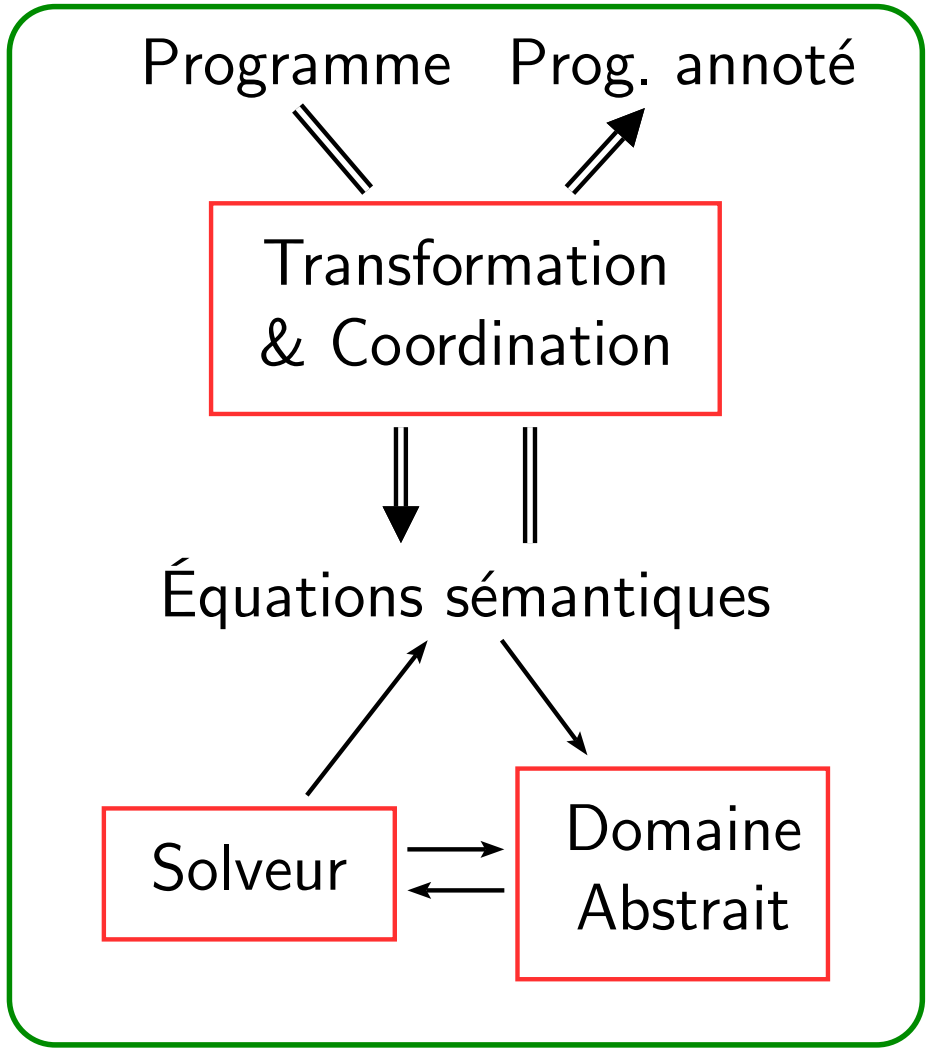
Bertrand Jeannet

INRIA Rhône-Alpes

8 juin 2010

Outline

Structure d'un analyseur statique



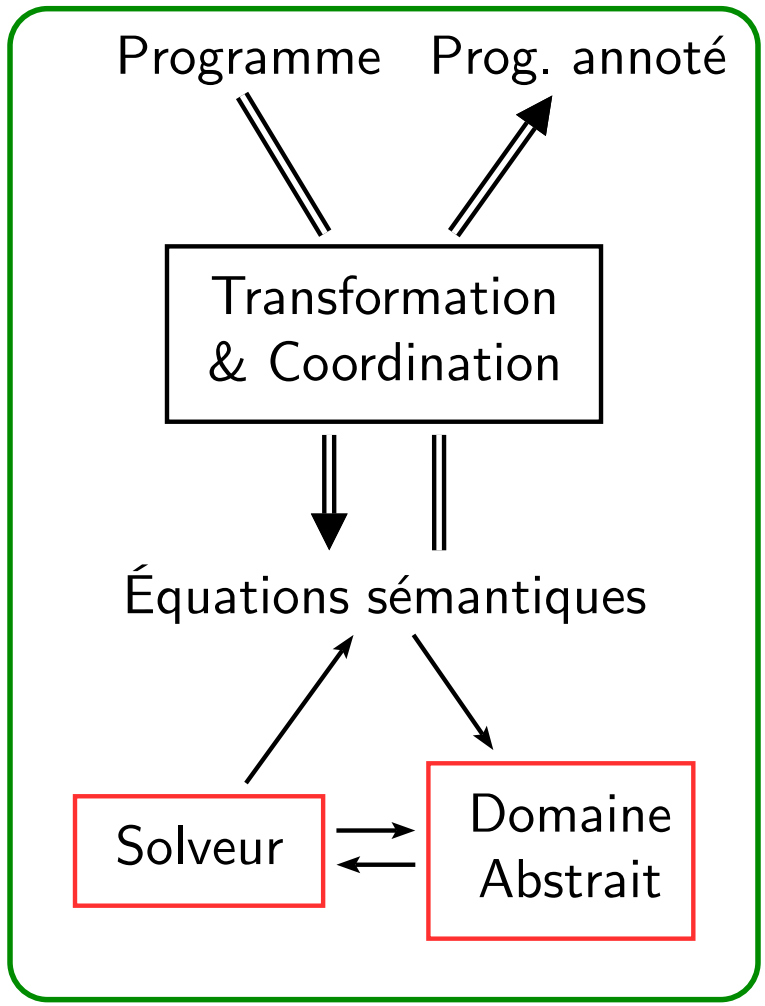
Modularité d'un analyseur statique

Objectif idéal

Chaque composant : librairie indépendante

- API bien identifiée
- Pas de dépendances implicites

$\Longrightarrow$ Composants substituables



En pratique :

Dans quelle mesure est-on robuste à

- Nouveau langage d'entrée ?
- Nouveau type de donnée ?
- Nouveau domaine abstrait ?

"Un composant est réutilisable quand il a été réutilisé au moins deux/trois fois" Anonyme, XX$^{\text{ième}}$ siècle

Outline

Plan

Implantation d'un treillis abstrait $\mathcal{D}^\sharp$

- Représentation des éléments abstraits
- Opérations logiques/ensemblistes :
  - conjonction ($\cap^\sharp$), disjonction ($\cup^\sharp$)
  - test du vide et de l'inclusion
  - introduction/élimination d'une variable
- Définition d'une sémantique concrète
  - $[\![\textbf{expr}]\!] : \mathcal{D} \longrightarrow \wp(\mathbb{D})$
  - $[\![\textbf{cond}]\!] : \mathcal{D} \longrightarrow \mathcal{D}$
  - $[\![\textbf{instr}]\!] : \mathcal{D} \longrightarrow \mathcal{D}$
- Et abstraction dans $\mathcal{D}^\sharp$
  - $[\![\textbf{cond}]\!]^\sharp : \mathcal{D}^\sharp \longrightarrow \mathcal{D}^\sharp$
  - $[\![\textbf{instr}]\!]^\sharp : \mathcal{D}^\sharp \longrightarrow \mathcal{D}^\sharp$
- Élargissement, Projection, Extraction de propriétés, ...

Plan

Domaines abstraits numériques I

Un cas important : variables numériques

$\mathcal{D}^\sharp$ abstrait $\wp(\mathbf{Env})$ avec $\mathbf{Env} = \mathbf{Var} \longrightarrow \mathcal{N}$

$\mathbb{Z}$ ou $\mathbb{R}$

Objectifs de la librairie APRON

- Domaines abstraits "sur la table", à travers une API commune et de haut niveau
  - Faciliter la conception d'analyseurs
  - Faciliter la comparaison de domaines

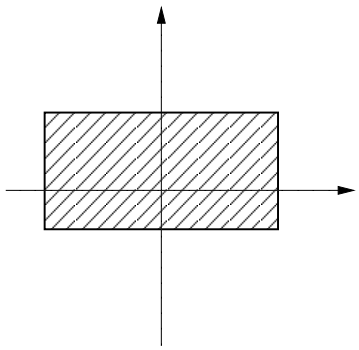
  (argument de modularité)
- Une plateforme pour l'intégration de nouveaux domaines
  - Boite à outils pour les implémenteurs

Domaines abstraits numériques II

Applications

- Découvrir des propriétés numériques sur des variables
- Prouver l'absence d'une large classe d'erreur à l'exécution
  - Division par zéro, dépassement de capacité, accès incorrect à un tableau
- Paramétriser une nanalyse non-numérique
  - Analyse de pointeur, de forme

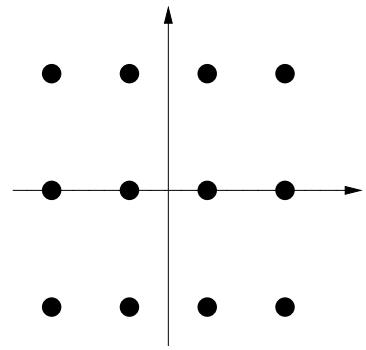
Quelques domaines numériques existants



Intervalles

$X_i \in [a_i, b_i]$

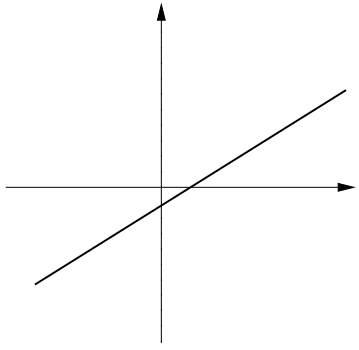
[Cousot-Cousot-76]



Congruences simples

$X_i \equiv a_i\,[b_i]$

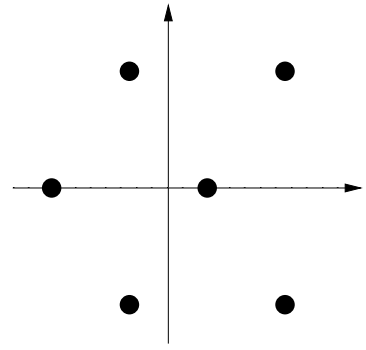
[Granger-89]



Égalités linéaires

$\sum_i \alpha_i X_i = \beta$

[Karr-76]

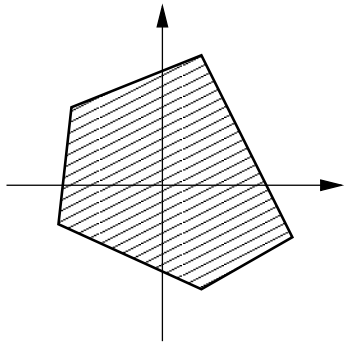


Congruences Linéaires

$\sum_i \alpha_i X_i \equiv \beta\,[\gamma]$

[Granger-91]

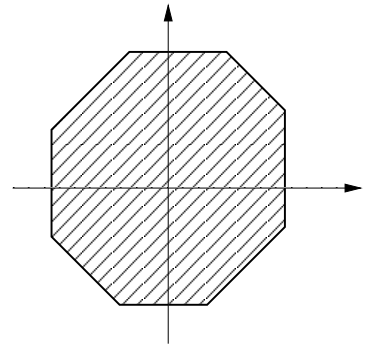
Quelques domaines numériques existants (cont.)



Polyèdres

$\sum_i \alpha_i X_i \geq \beta$

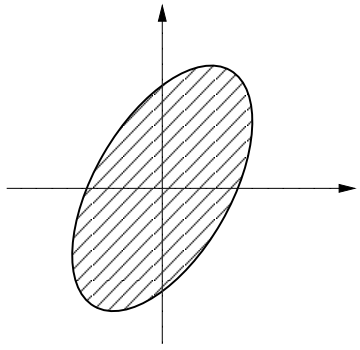
[Cousot-Halbwachs-78]



Octagones

$\pm X_i \pm X_j \leq \beta$

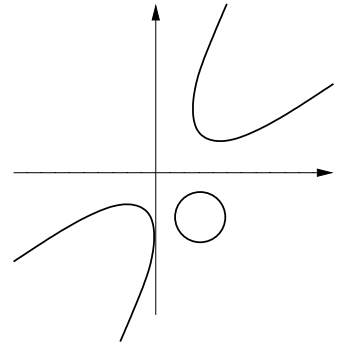
[Miné-01]



Ellipsoïdes

$\alpha X^2 + \beta Y^2 + \gamma XY \leq \delta$

[Feret-04]



Variétés

$P(\vec{X}) = 0,\ P \in \mathbb{R}[\mathrm{Var}]$

[Sankaranarayanan-Sipma-Manna-04

APRON : Une sémantique concrète générale I

Types de données : entiers, réels, (types flottants)

$\mathcal{D} = \boxed{\mathbb{Z}^p \times \mathbb{R}^q}$ / $\mathcal{D} = \boxed{\mathbf{Var} \longrightarrow \mathbb{Z} \uplus \mathbb{R}}$

Main concepts/datatypes

Dimensions/Variables, Dimensionalité/Environnements
Coefficients, Expressions, Contraintes, Valeurs Abstraites

Coefficients

$$\begin{array}{rcl} \langle \text{scalar} \rangle & ::= & \langle \text{mpq} \rangle \mid \langle \text{float} \rangle \mid \langle \text{mpfr} \rangle \\ \langle \text{coeff} \rangle & ::= & \langle \text{scalar} \rangle \mid [\langle \text{scalar} \rangle, \langle \text{scalar} \rangle] \end{array}$$

Variables (niveau utilisateur)

- En C : type défini par l'utilisateur (par défaut : string)
- En OCaml : string

APRON : Une sémantique concrète générale II

Expressions

Incluent les expressions non-linéaires et flottantes

- Par ex., affectation $\boldsymbol{y := 2x^2z + \sqrt{yz} \ +_{f,+\infty} \ e}$
- Incluent la norme IEEE754 (sauf NaN)

$$
\begin{array}{rcl}
\langle \text{expr} \rangle & ::= & \langle \text{coeff} \rangle \mid \langle \text{var} \rangle \\
& \mid & (\text{“}-\text{”} \mid \textbf{sqrt} \mid \textbf{cast})_{\textit{typ},\textit{rmode}} \ \langle \text{expr} \rangle \\
& \mid & \langle \text{expr} \rangle \ (\text{“}+\text{”} \mid \text{“}-\text{”} \mid \text{“}*\text{”} \mid \text{“}/\text{”} \mid \text{“}\%\text{”})_{\textit{typ},\textit{rmode}} \ \langle \text{expr} \rangle \\
\langle \text{typ} \rangle & ::= & \cdot \mid \text{“i”nt} \mid \text{“f”oat} \mid \text{“d”ouble} \\
& \mid & \text{“e”xtended} \mid \text{“l”ong double} \\
\langle \text{mode} \rangle & ::= & \text{“n”ear} \mid \text{“0”} \mid \text{“}+\infty\text{”} \mid \text{“}-\infty\text{”}
\end{array}
$$

APRON : Une sémantique concrète générale III

Contraintes

$$\begin{array}{lll} \langle \text{cons} \rangle & ::= & \langle \text{expr} \rangle \ (\text{“>”} \mid \text{“>=”} \mid \text{“=”}) \ 0 \\ & \mid & \langle \text{expr} \rangle = \langle \text{scalar} \rangle \ \textbf{mod} \ \langle \text{scalar} \rangle \end{array}$$

APRON : Une sémantique concrète générale IV

Valeurs Abstraites

APRON : Une sémantique concrète générale V

- Typées par une dimensionalité $p + q$, ou un environnement normalisé de variables
- Opérations principales
  - Opérations classiques de treillis : $=^\sharp$, $\subseteq^\sharp$, $\cup^\sharp$, $\cap^\sharp$
  - Opérations "géométriques" :
    - projection/oubli de dimension(s)
    - changement de dimensionalité/environnements
    - fold/expand
  - Opérations liées à la sémantique d'un langage :
    - Intersection avec des contraintes
    - Affectation (parallèle) $[\![\vec{x} = \vec{e}]\!]^\sharp(X)$
    - Substitution (parallèle) $[\![\vec{x}/\vec{e}]\!]^\sharp(X)$
  - Tests et extractions de propriétés
    - Satisfaction d'une contrainte par une valeur abstraite
    - Extraction de la liste de contrainte, ...

APRON : Généricité I

API et types de données

- Modélisent la sémantique concrète, de manière complète
- Indépendants de l'implantation du domaine abstrait

Domaines abstraits échangeables $\Longrightarrow$ même sémantique concrète

Object orientation

- Valeur abstraite = type de donnée abstrait
- Vrai domaine sous-jacent : défini par un "manager"
  - Code dépendant du domaine localisé dans l'allocation du manager
  - Options contrôlant le compromis précision/efficacité

APRON : Généricité II

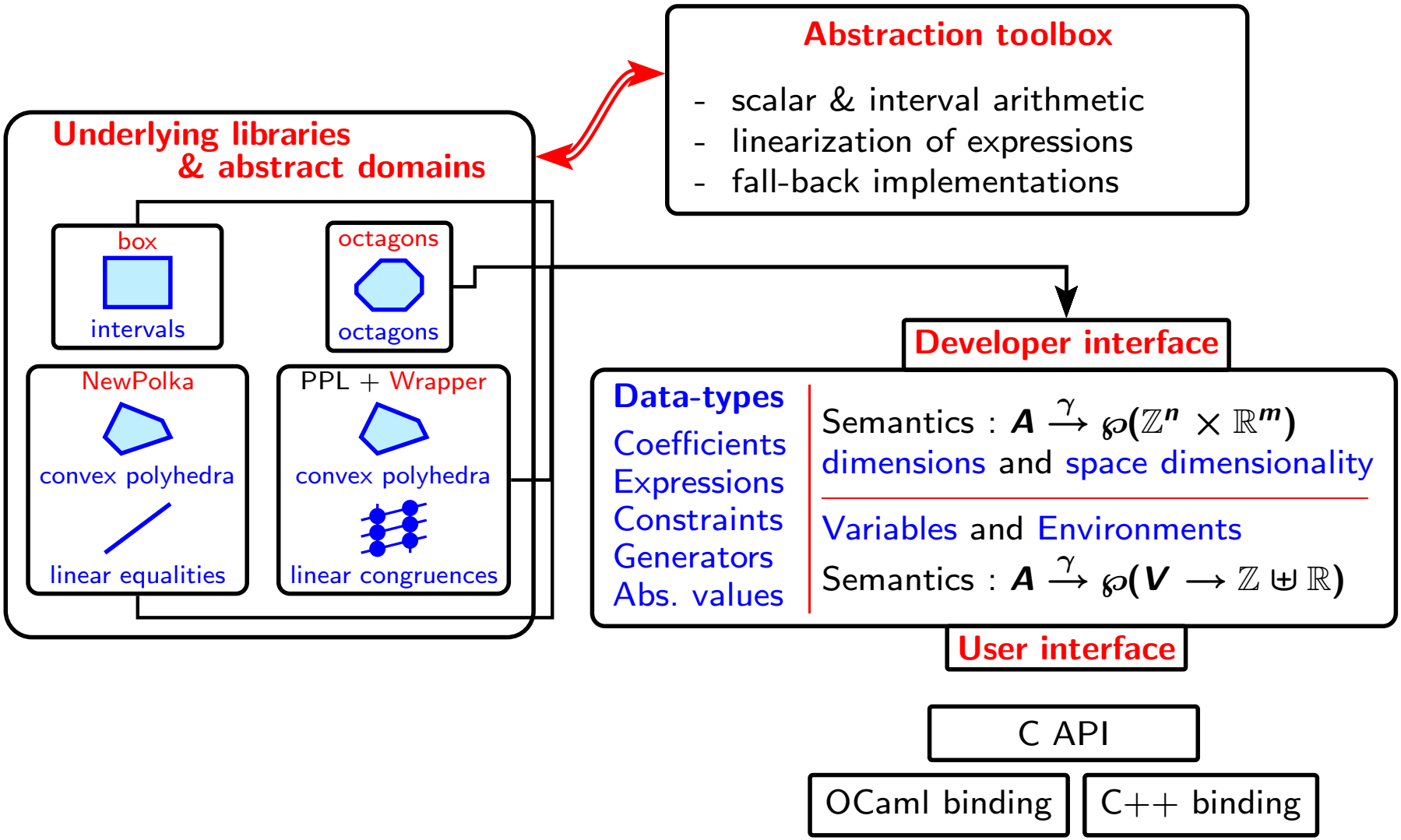
Exemple

```
# let man = Oct.manager_alloc ();;
val man : Oct.t Apron.Manager.t = <abstr>

# let val = Apron.Abstract1.top man env;;
val val : Oct.t Apron.Abstract1.t = <abstr>

# let val = Apron.Abstract1.assign_texpr man val var expr;;
```

La librairie APRON : structure



Bénéfice

Avantages pour un programmeur client :

- Interface unifiée et de plus haut niveau
  - Les variables (“x”, “y”) remplacent les dimensions (“0”, “1”)
  - Expressions non-linéaires
  - Sémantique des opérations entières et flottantes

Avantages pour l’implanteur d’un nouveau domaine :

- Possibilité de se concentrer sur les fonctionnalités de base, et de bas niveau
- Aide fournie par les fonctionnalités génériques
- La connexion d’une bibliothèque existante est économique (pour les deux domaines de la PPL, 4200 LOC)

Limites de la généricité I

Exemple de domaine ne posant pas problème

- Congruences linéaires de la PPL
- Polyèdres MaxPlus (thèse de X. Allamigeon)

Exemple de domaine posant problème

Les "template constraints" :

- à chaque point de contrôle k, on a un ens. fini d'expressions $\vec{e}(k)$
- l'analyse : infère des bornes $\vec{l}(k) \leq \vec{e}(k) \leq \vec{u}(k)$ généralise les intervalles, zones, octogones, octaèdres, ...

Comment "passer" les $\vec{e}(k)$?
En fait le choix des $\vec{e}(k)$ définit l'analyse

Limites de la généricité II

Autre exemple de domaine posant problème

Les formes affines (Goubault, Martel et Putot) ou zonotopes (Girard)

- Polyèdres centrés symmétriques

$$\begin{aligned} x_1 &= x_1^0 + 3\epsilon_1 - 2\epsilon_2 + \epsilon_3 \\ x_2 &= x_2^0 + 2\epsilon_1 + 3\epsilon_2 + 2\epsilon_3 \\ -1 &\leq \epsilon_1, \epsilon_2, \epsilon_3 \leq 1 \end{aligned}$$

- Les symboles de bruit ϵ_i sont attachés à certains points du CFG
- Gestion fine de ces symboles (introduction, oubli, fusion, ...) liée au CFG (boucles) et expressions non linéaires

Une solution acceptable (mais non optimale) bientôt livrée

Exploitation de la généricité

"Foncteurs génériques"

- Produit réduit de domaines numériques
  - Paramétré par une fonction de réduction
  - Ex : combinaison polyèdres/congruences linéaires
  - Exemple de réduction :
    $x \in [1, 5] \wedge x \equiv 1 \mod 3 \implies$
    $x \in [4, 4] \wedge x \equiv 4 \mod 0$
- Unions finies de valeurs abstraites, paramétrée par
  - Une fonction de fusion
    (pour borner le cardinal de l'union)
  - Un opérateur d'élargissement adapté

(Envisagé)

Ajout de nouveaux types de donnée

Combiner types finis et numériques : BDDAPRON

Plan

BDDAPRON : Motivations

- Le moindre programme manipule des variables booléennes !
- On ne peut pas se contenter d'analyses séparées :
  les variables booléennes et numériques interagissent :

```
b := x>y ;
x := 0 ;
...
if b
  then y := x+1 ;
  else z := y+x ;
endif ;
z := if b then y+1 else y ;
```

- Il faut distinguer :
  - L'abstraction des types de données
    (booléens,numériques)
  - Leur couplage
    (indépendent du domaine numérique considéré)

BDDAPRON : Sémantique concrète : types de donnée

Types finis

- Booléens
- Types énumérés
- Entiers (signés ou non) codés sur N bits

Types numériques (APRON)

- Entiers
- Réels (permet d'émuler les flottants)

Sémantique d'une valeur abstraite

$\gamma : \boldsymbol{A} \longrightarrow (\boldsymbol{V} \longrightarrow (\mathbb{B} \uplus \mathbb{E}^{\boldsymbol{t}} \uplus \mathbb{I}^{\boldsymbol{n}}) \uplus (\mathbb{Z} \uplus \mathbb{R}))$

Valeurs abstraites typées par un environnement $\boldsymbol{V} \longrightarrow \mathbf{Type}$

BDDAPRON : Sémantique concrète : expressions

Booléennes

$$\begin{array}{rcl}
\langle \text{bexpr} \rangle & ::= & \textbf{true} \mid \textbf{false} \mid \langle \text{bvar} \rangle \\
& \mid & \langle \text{expr} \rangle = \langle \text{expr} \rangle \\
& \mid & \langle \text{iexpr} \rangle \ (\text{“<”} \mid \text{“<=”}) \ \langle \text{iexpr} \rangle \mid \langle \text{aexpr} \rangle \ (\text{“<”} \mid \text{“<=”}) \ \langle \text{aexpr} \rangle \\
& \mid & \textbf{not} \ \langle \text{bexpr} \rangle \mid \langle \text{bexpr} \rangle \ (\textbf{and} \mid \textbf{or} \mid \ldots) \ \langle \text{bexpr} \rangle
\end{array}$$

Énumérée

$$\begin{array}{rcl}
\langle \text{eexpr} \rangle & ::= & \langle \text{label} \rangle \mid \langle \text{evar} \rangle \\
& \mid & \textbf{if} \ \langle \text{bexpr} \rangle \ \textbf{then} \ \langle \text{eexpr} \rangle \ \textbf{else} \ \langle \text{eexpr} \rangle
\end{array}$$

Entier borné

$$\begin{array}{rcl}
\langle \text{iexpr} \rangle & ::= & \langle \text{cst} \rangle \mid \langle \text{ivar} \rangle \\
& \mid & \text{“–”} \ \langle \text{iexpr} \rangle \mid \langle \text{iexpr} \rangle \ (\text{“+”} \mid \text{“–”} \mid \text{“*”}) \ \langle \text{iexpr} \rangle \\
& \mid & \langle \text{iexpr} \rangle \ (\text{“<<”} \mid \text{“>>”}) \ \text{n} \\
& \mid & \textbf{if} \ \langle \text{bexpr} \rangle \ \textbf{then} \ \langle \text{iexpr} \rangle \ \textbf{else} \ \langle \text{iexpr} \rangle
\end{array}$$

Arithmétique (APRON)

$$\begin{array}{rcl}
\langle \text{aexpr} \rangle & ::= & \langle \text{coeff} \rangle \mid \langle \text{avar} \rangle \\
& \mid & (\text{“–”} \mid \textbf{sqrt} \mid \textbf{cast})_{\textit{typ},\textit{rmode}} \ \langle \text{aexpr} \rangle \\
& \mid & \langle \text{aexpr} \rangle \ (\text{“+”} \mid \text{“–”} \mid \text{“*”} \mid \text{“/”} \mid \text{“\%”})_{\textit{typ},\textit{rmode}} \ \langle \text{aexpr} \rangle \\
& \mid & \textbf{if} \ \langle \text{bexpr} \rangle \ \textbf{then} \ \langle \text{aexpr} \rangle \ \textbf{else} \ \langle \text{aexpr} \rangle
\end{array}$$

BDDAPRON : Sémantique concrète (cont.)

Contraintes

$\Leftrightarrow$ Expressions booléennes

Opérations sur les valeurs abstraites

Les mêmes que pour APRON !

BDDAPRON : Sémantique abstraite

On suppose $\gamma^{\mathcal{N}} : \boldsymbol{A}^{\mathcal{N}} \longrightarrow \wp(\mathbb{R}^{n})$ (domaine APRON)

Choix effectué

- Pas d'abstraction des types finis
- Couplage précis des types finis vers les numériques

$$\begin{array}{c} \wp(\mathbb{B}^{m} \times \mathbb{R}^{n}) \\ \simeq \\ \mathbb{B}^{m} \longrightarrow \wp(\mathbb{R}^{n}) \end{array} \quad \Longleftrightarrow \quad \boxed{\mathbb{B}^{m} \longrightarrow \boldsymbol{A}^{\mathcal{N}}}$$

Autre choix possible (non implémenté)

- Pas d'abstraction des types finis
- Couplage minimal entre types finis et numériques

$$\wp(\mathbb{B}^{m} \times \mathbb{R}^{n}) \Longleftrightarrow \wp(\mathbb{B}^{m}) \times \wp(\mathbb{R}^{n}) \Longleftrightarrow \boxed{\wp(\mathbb{B}^{m}) \times \boldsymbol{A}^{\mathcal{N}}}$$

Représentation des valeurs abstraites : choix 1

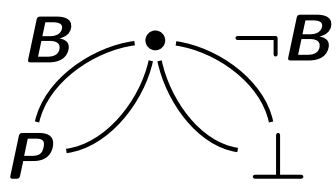
$f \in \mathbb{B}^m \longrightarrow A^{\mathcal{N}}$ représenté par

- Un MTBDD (ADD dans la librairie CUDD) (C)
  - Représente des fonctions $\mathbb{B}^m \longrightarrow$ `C double`
  - Généralise les BDDs ($\mathbb{B}^m \longrightarrow \mathbb{B}$)
- CUDD étendu pour remplacer `double` par des valeurs OCaml (200-300 lignes de C)

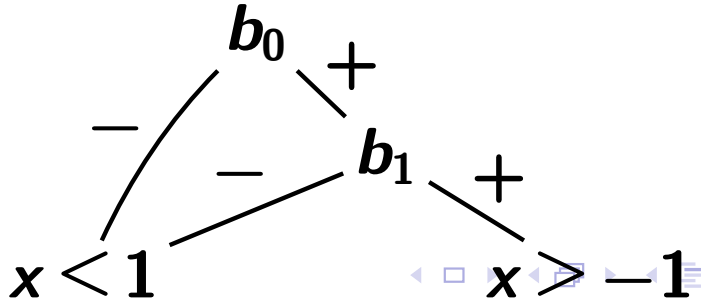
Implications

- Hachage de valeurs abstraites APRON
- Requiert une normalisation forte (cher !)
- Test d'égalité (cher aussi !)

Exemples



Conjonction $B \wedge P$



Représentation des valeurs abstraites : choix 2

$f \in \mathbb{B}^m \longrightarrow A^{\mathcal{N}}$ représenté par

- Une liste de paires $\bigwedge_i B_i \wedge P_i$
- Normalisation : $B_i \wedge B_j = \mathbf{ff}$, $\bigwedge_i B_i = \mathbf{tt}$, et $P_i \neq P_j$
- Sémantiquement équivalent au choix 1

Algorithmiquement

- Gain en mémoire :
  2 listes isomorphes $\bigwedge_i B_i \wedge P'_i$ et $\bigwedge_i B_i \wedge P''_i$
  Partage des gardes (si MTBDDs : graphes disjoints)
- Gain en complexité ?
  Possibilité de manipuler des valeurs non normalisées
  Expérimentalement : bof pour l'instant (25 %)
- "Perte" pour l'implémenteur
  Recoder la mécanique des MTBDDs

Représentation des expressions

Expressions Booléennes

$\boldsymbol{f} : \mathbb{B}^{\boldsymbol{m}} \times \mathbf{ACond}^* \longrightarrow \mathbb{B}$: BDDs avec

décisions : conditions atomiques

feuilles : constantes booléennes

Condition atomique : variable booléenne ou contrainte arithmétique

Expressions de type énuméré ou entier borné

$\boldsymbol{f} : \mathbb{B}^{\boldsymbol{m}} \times \mathbf{ACond}^* \longrightarrow \mathbb{B}^{\boldsymbol{p}}$

tableau de BDDs

Expressions de type arithmétique

$\boldsymbol{f} : \mathbb{B}^{\boldsymbol{m}} \times \mathbf{ACond}^* \longrightarrow \mathbf{AExpr}$: MTBDDs avec

décisions : conditions atomiques

feuilles : expressions arithmétiques atomiques

Expression arithmétique atomique : expression sans if-then-else

BDDAPRON et modularité

Extension générique d'un domaine APRON avec des types finis

2 implémentations différentes de la même sém. abstraite

Manipulation des valeurs abs. : isomorpohes à APRON

Mais changement (extension) de la sémantique concrète

$\Longrightarrow$ modification de Interproc !