

Recherche de solutions dans les systèmes de contraintes avec flottants

Exploration préliminaire des stratégies de recherche

Heytem ZITOUN

I3S

Plan

- Contexte
- Travaux préliminaires
- Pistes

Problématique

- Contexte :
 - Recherche de contre exemple sur les flottants difficile
 - Exemple : ABS
- Problème : Pas de stratégie efficace sur les flottants
- Difficultés
 - Flottants \neq entiers
 - Énumération complexe
 - Flottants \neq Réels
 - Pas de test d'existences
 - Arithmétique des flottants (absorption, cancellation ...)

Stratégie de recherche SAT/FD

- Objectifs :
 - Guider efficacement la recherche après le filtrage
 - Les mauvaises stratégies ont un impact fort sur le temps de résolution

- Critères:
 - Choix des variables
 - Choix des valeurs

- 2 types de stratégies :
 - **Statique** : ordre fixé avant l'exploration
 - **Dynamique** : ordre varie durant l'exploration
 - ... (nogood)

Stratégie de recherche : illustration

■ $X = [1,2]$ $Y = [1 \dots 1000]$ $C : X + Y = 3$

■ Stratégies :

■ 1) 2 ordres de variables

■ Ordre variable : Y puis X

■ Ordre variable : X puis Y

■ 2) 2 ordres de valeurs

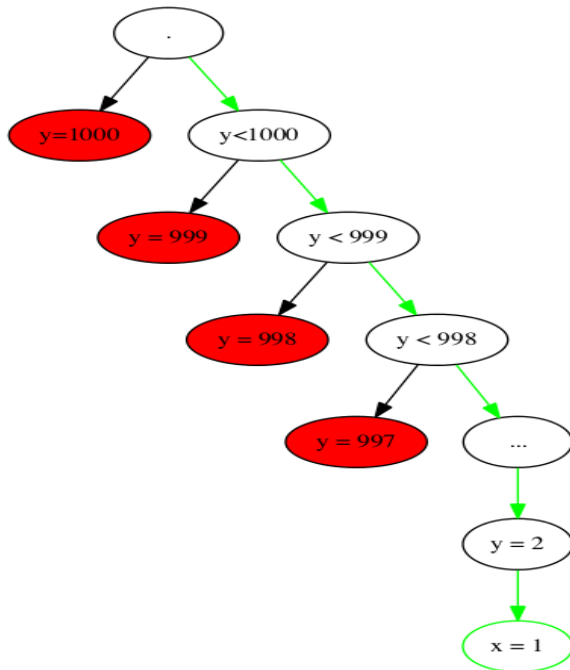
■ Ordre valeur : décroissant

■ Ordre valeur : croissant

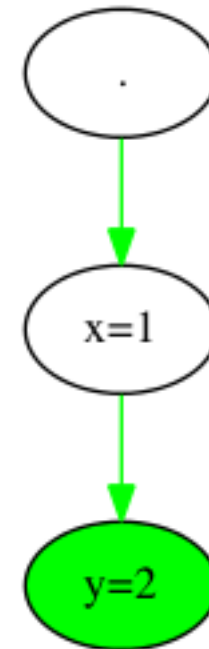
Stratégie de recherche : illustration

$X = [1,2]$ $Y = [1 \dots 1000]$ $C : X + Y = 3$

Stratégie 1



Stratégie 2



Stratégies existantes sur les entiers

■ Statique :

- Min-domaine : ordre des variables choisies selon la taille croissante de leur domaine
- Max-degree : ordre des variables choisies selon le nombre de contraintes où elles apparaissent

■ exemple :

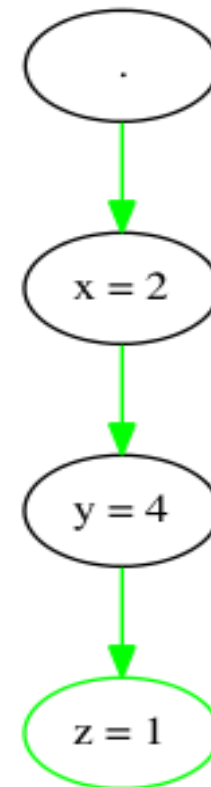
- $X = [2,3]$ $Y = [1,5]$ $Z = [1,3]$
- $C1 : X+Y > 5$
- $C2 : X+Z < 4$
- $C3 : 2Y + 3X > 2$
- Max-domaine : X puis Z puis Y
- Max-degree : X puis Y puis Z

First-Fail

- **Principe** : choisir en premier la variable/valeur qui va menée à un échec
 - Max-degree (i.e variable la plus contrainte)
 - Min-domaine (i.e variable avec le plus petit domaine)
- First-Fail **statique** :
 - $X = \{2,3\}$ $Y = \{1,2,3,4,5\}$ $Z = \{1,2,3\}$
 - C1 : $X+Y > 5$
 - C2 : $X+Z < 5$
 - C3 : $2Y + 3X > 2$
 - Min-domaine : X puis Z puis Y ($|D(X)| = 2$, $|D(Y)| = 5$, $|D(Z)| = 3$)
 - Max-degree : X puis Y puis Z ($d(X) = 3$, $d(Y) = 2$, $d(Z) = 1$)

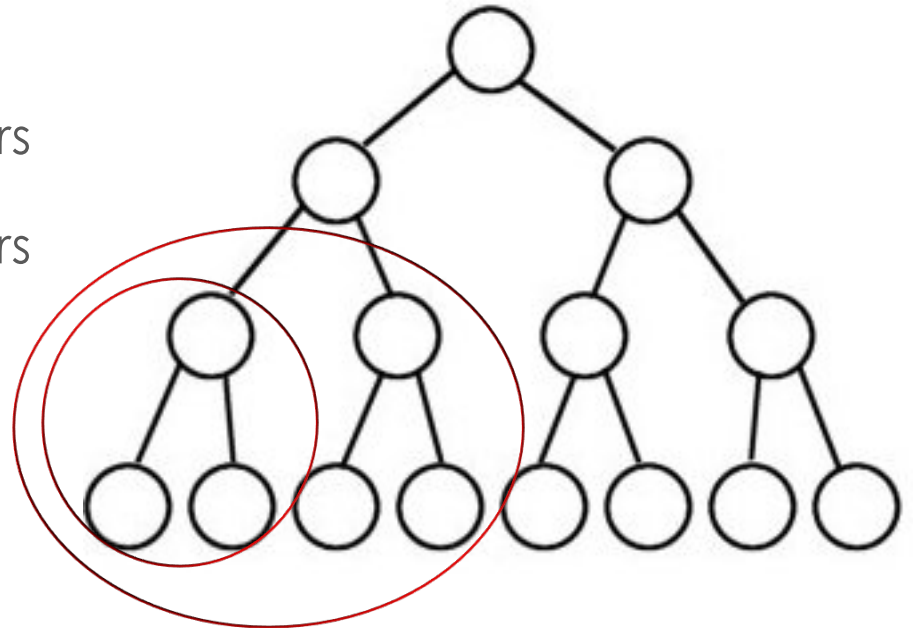
First-Fail

- First-Fail **dynamique** : Min-domain
- Problème :
 - $X = \{2,3\}$ $Y = \{1,2,3,4\}$ $Z = \{1,2,3\}$
 - $C1 : X+Y > 5$
 - $C2 : X+Z < 5$
- $X = 2 \rightarrow Y = \{4\}$ et $Z = \{1,2\}$
- $X = 3 \rightarrow Y = \{3,4\}$ et $Z = \{1\}$



Impact-based Search (IBS)

- Intuition : favoriser les **coupes** dans l'espace de recherche
- Exploration complète
- Choisir $X = 1$ tue 3 valeurs
- Choisir $X = 2$ tue 6 valeurs
- Choix : $X = 2$



- Impact** = mesure l'influence du choix d'une variable/valeur sur l'espace de recherche

IBS : formellement

□ Soient X_1 une variable $a \in D(X_1)$

□ Impact de l'assignement de a à X_1 ($I(X_1 = a)$)

$$I(X_1 = a) = 1 - \frac{P_{after}}{P_{before}}$$

□ P_{after} : produit cartésien des domaines avant affectation

□ P_{before} : produit cartésien des domaines après filtrage

□ Choix de valeur : impact maximal

□ Choix de variable : $\max \left(\frac{\sum_{a \in D(X_i)} I(X_i = a)}{|D(X_i)|} \right)$ (i.e max des moyennes)

□ Remarque : coûteux mais permet un gain dans la combinatoire

IBS : Calcul d'impact

$X = [2,3]$ $Y = [2,4]$

$C1 : X + Y > 5$ $I(X_1 = a) = 1 - \frac{P_{after}}{P_{before}}$

X	2	3
$I(X=a)$	0.83	0.66

Y	2	3	4
$I(Y=a)$	1	0.83	0.66

▣ $X = 2 \rightarrow Y = [4,4]$

$$I(X = 2) = 1 - \frac{1*1}{2*3} = 0.83$$

▣ $X = 3 \rightarrow Y = [3,4]$

$$I(X = 3) = 1 - \frac{1*2}{2*3} = 0.66$$

▣ $Y = 2 \rightarrow X = \emptyset$

$$I(Y = 2) = 1 - \frac{1*0}{2*3} = 1$$

▣ $Y = 3 \rightarrow X = [3,3]$

$$I(Y = 3) = 1 - \frac{1*1}{2*3} = 0.83$$

▣ $Y = 4 \rightarrow X = [3,4]$

$$I(Y = 4) = 1 - \frac{1*2}{2*3} = 0.66$$

IBS : sélection

X	2	3
$I(X=a)$	0.83	0.66

Y	2	3	4
$I(Y=a)$	1	0.83	0.66

■ Sélection de la variable : $\max\left(\frac{\sum_{a \in D(X_i)} I(X_i=a)}{|D(X_i)|}\right)$

■ $I(X) = \frac{0.83 + 0.66}{2} = 0.745$

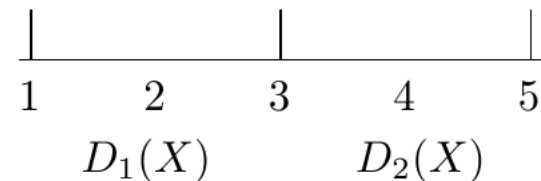
$I(Y) = \frac{1 + 0.83 + 0.66}{3} = 0.83$

■ Sélection : Y

■ Sélection de la variable 2

IBS : initialisation

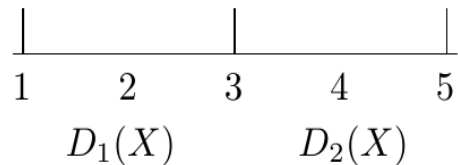
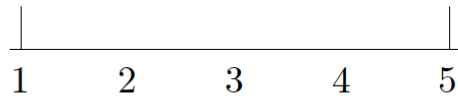
- ▣ Calcul des impacts peut-être **coûteux**
- ▣ Intuition : les valeurs proches ont des chances d'avoir le même impact
- ▣ Soient
 - ▣ $s = 1$ (nombre de bisection)
 - ▣ $D(X) = [1,5]$
 - ▣ $I(X = 2) = 0.5$
 - ▣ $I(X = 4) = 0.2$



1	2	3	4	5
0.5	0.5	0.5	0.2	0.2

Variante d'IBS

- Bisection jusqu'à que la cardinalité d'un sous-domaine soit au moins k
- Exemple : bisection jusqu'à que la taille soit 2



- Question : Que ce passe-t-il quand k varie ?

Expérimentation

- Problème considéré : carré magique
- Carré de taille n (ici 3)
- Propriété :
 - Tous éléments sont différent
 - $\sum \text{lignes} = \sum \text{colonnes} = \sum \text{diagonales}$
 - Domaine des variables $[1, n^2]$

2	7	6	→ 15
9	5	1	→ 15
4	3	8	→ 15

15 ↙ 15 ↓ 15 ↓ 15 ↓ 15 ↘

Expérimentation : paramètres

- Temps en ms
- Timeout : 1 minute
- $S = 3$ correspond à la profondeur de bisection
- DESSIN EXPLICATIF DE $S=3 \rightarrow 2^3$ sous domaine

Expérimentations

Existant sur les flottants

- **std**: standard *filtrage & bisection*
- **fpc**: le domaine est divisé en *5 intervalles*
 - *3 intervalles dégénérés*: le plus petit flottant *l*, le plus grand flottant *r*, et le flottant du milieu *m*
 - intervalles *]l, m[* et *]m, r[*
- **fp3s**: le domaine est divisé en *3 intervalles dégénérés*: le plus petit flottant *l*, le plus grand flottant *r*, et le flottant du milieu *t* *m (méthode incomplète)*

Experimentations

- ▣ **CDFL**: Analyseur de programme prouve l'absence d'erreur à l'exécution
- ▣ **CBMC**: état de l'art des outils de bounded model checking
- ▣ **CPBPV_FP**: outil de bounded model checking utilisant la programmation par contraintes

Expérimentations : programme Heron

- La formule d'Heron calcul l'air d'un triangle à partir de ses longueurs a , b , c (avec a le plus long côté)

- $\sqrt{s * (s - a) * (s - b) * (s - c)}$ avec $s = \frac{a+b+c}{2}$

```
1 /* Pre-condition : a ≥ b and a ≥ c */  
2 float heron(float a, float b, float c) {  
3     float s, squared_area;  
4  
5     squared_area = 0.0f;  
6     if (a <= b + c) {  
7         s = (a + b + c) / 2.0f;  
8         squared_area = s*(s-a)*(s-b)*(s-c);  
9     }  
10    return sqrt(squared_area);  
11 }
```

- Version Optimiser : $\text{squared_area} = ((a+(b+c))*(c-(a-b)) * (c+(a-b))*(a+(b-c)))/16.0f;$

Experimentations

Name	Condition	CDFL	CBMC	std	fpc	fpc3s	s?
heron	area < 10_f^{-5}	3.87 s	0.28 s	> 180 s	0.70 s	0.02 s n	y
	area > $156.25f + 10_f^{-5}$	> 180 s	34.51 s	22.32 s	7.80 s	0.08 s n	y
heron Optimisé	area < 10_f^{-5}	7.61 s	0.93 s	> 180 s	0.15 s	0.01 s n	y
	area > $156.25f + 10_f^{-5}$	> 180 s	> 180 s	8.99 s	30.48 s	0.01 s n	n

std: standard *filtrage & bisection*

fpc: le domaine est divisé en 5 *intervals*

- 3 *intervalles dégénérés*: le plus petit flottant l , le plus grand flottant r , and le flottant du milieu m
- intervalles $]l, m[$ and $]m, r[$

fp3s: le domaine est divisé en 3 *intervalles dégénérés*: le plus petit flottant l , le plus grand flottant r , et le flottant du milieu m

S? : existence d'un cas de test

Stratégie sur les flottants : pistes

- Structure des flottants
 - Enumération d'exposant et mantisse
 - Densité
- Adapté IBS aux flottants

QUESTIONS ?

Merci pour votre
Attention