

# Estimation de WCET haut-niveau avec Interprétation abstraite et Programmation par contraintes

Aurèle Barrière

17 mai - 22 juillet 2016

Équipe : CELTIQUE

Estimation WCET

Préliminaires techniques

CFG et notations

Hypothèses

Interprétation abstraite

Programmation par  
contraintes

Contribution

CSP;

Arbres d'exécution  
symbolique

COP final

Implémentation et  
résultats

Implémentation

Résultats

Analyse d'un exemple

Conclusion

Annexes

# Estimation WCET

WCET : *worst case execution time*

**Analyse haut-niveau** ou analyse de flot du programme. Estimer le nombre d'exécutions de chaque instruction.

**Analyse bas-niveau.** Estimer le temps d'exécution de chaque instruction dans le pire cas.

**Analyse des anomalies.** Prendre en compte les phénomènes de *caches*, *pipelines* et leur influence sur le WCET.

On veut une majoration **sûre** du WCET.

## Estimation WCET

### Préliminaires techniques

CFG et notations  
Hypothèses  
Interprétation abstraite  
Programmation par  
contraintes

### Contribution

CSP;  
Arbres d'exécution  
symbolique  
COP final

### Implémentation et résultats

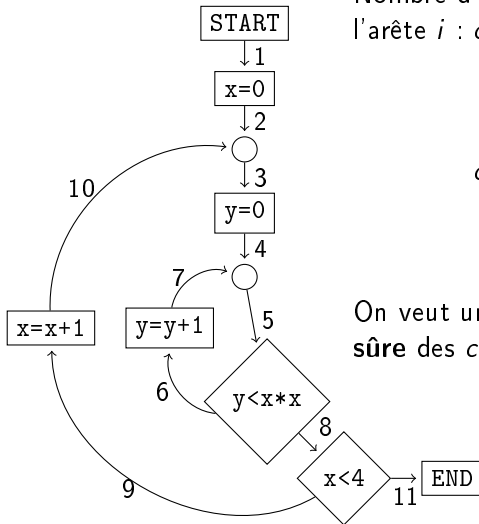
Implémentation  
Résultats  
Analyse d'un exemple

### Conclusion

### Annexes

## CFG et notations

Nombre d'exécutions de  
l'arête  $i$  :  $c_i$ .



$$c_3 = 5$$

On veut une majoration  
**sûre** des  $c_i$ .

Estimation WCET

Préliminaires techniques

**CFG et notations**

Hypothèses

Interprétation abstraite

Programmation par  
contraintes

Contribution

CSP;

Arbres d'exécution  
symbolique

COP final

Implémentation et  
résultats

Implémentation

Résultats

Analyse d'un exemple

Conclusion

Annexes

# Hypothèses

## Estimation WCET

### Préliminaires techniques

CFG et notations

#### **Hypothèses**

Interprétation abstraite

Programmation par  
contraintes

### Contribution

CSP;

Arbres d'exécution  
symbolique

COP final

### Implémentation et résultats

Implémentation

Résultats

Analyse d'un exemple

### Conclusion

### Annexes

On ne considérera que des programmes qui vérifient les hypothèses suivantes :

Les programmes sont déterministes.

Les programmes terminent.

Sous ces hypothèses, un programme ne passe pas deux fois dans la même arête avec le même état mémoire.

## Interprétation abstraite

Sur-approximation de la sémantique d'un programme.  
À chaque arête  $i$ , on associe un ensemble d'états de la mémoire. Tout état de la mémoire accessible appartient à cet ensemble.

### Estimation WCET

#### Préliminaires techniques

CFG et notations

Hypothèses

**Interprétation abstraite**

Programmation par  
contraintes

#### Contribution

CSP;

Arbres d'exécution  
symbolique

COP final

#### Implémentation et résultats

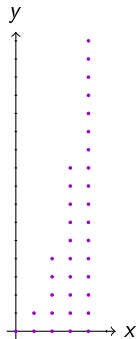
Implémentation

Résultats

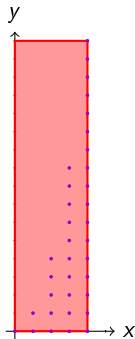
Analyse d'un exemple

#### Conclusion

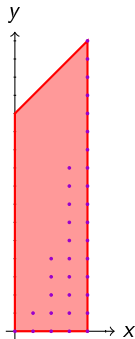
#### Annexes



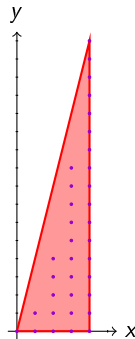
États  
atteignables.



Intervalle  
 $x \in [a, b]$   
 $y \in [c, d]$



Octogones  
 $\pm x \pm y \leq c$



Polyèdres  
Toute  
contrainte  
linéaire

# Programmation par contraintes

## Estimation WCET

### Préliminaires techniques

CFG et notations

Hypothèses

Interprétation abstraite

Programmation par  
contraintes

### Contribution

CSP;

Arbres d'exécution  
symbolique

COP final

### Implémentation et résultats

Implémentation

Résultats

Analyse d'un exemple

### Conclusion

### Annexes

**CSP** : *constraint satisfaction problem*

$X$  ensemble de variables

$D$  ensemble de domaines pour chaque variable

$C$  ensemble de contraintes sur les variables

**COP** : *constraint optimization problem* Un CSP avec  
une fonction objectif à minimiser.

## Exemple

$X = \{x_1, x_2\}$ ,  $D = \{\llbracket 0, 2 \rrbracket, \llbracket 0, 4 \rrbracket\}$ ,

$C = \{x_1 = 0, x_1^2 + x_2 \geq 7\}$  est un CSP sans solution.

Par contre, si  $D = \{\llbracket 0, 2 \rrbracket, \llbracket 0, 7 \rrbracket\}$ , alors on a la  
solution  $x_1 = 0, x_2 = 7$ .

## Estimation WCET

### Préliminaires techniques

CFG et notations  
Hypothèses  
Interprétation abstraite  
Programmation par  
contraintes

### Contribution

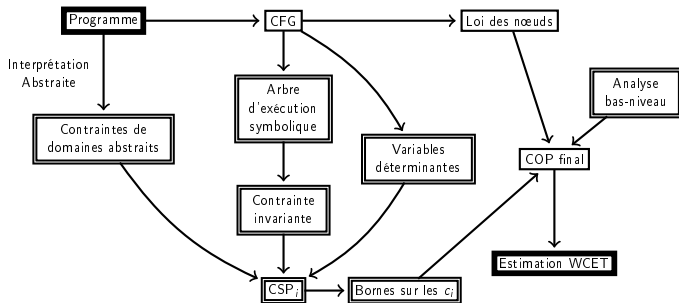
CSP;  
Arbres d'exécution  
symbolique  
COP final

### Implémentation et résultats

Implémentation  
Résultats  
Analyse d'un exemple

### Conclusion

### Annexes



Estimation WCET

Préliminaires techniques

CFG et notations

Hypothèses

Interprétation abstraite

Programmation par  
contraintes

Contribution

**CSP<sub>i</sub>**

Arbres d'exécution  
symbolique

COP final

Implémentation et  
résultats

Implémentation

Résultats

Analyse d'un exemple

Conclusion

Annexes

## CSP<sub>i</sub>

Un CSP dont chaque état mémoire atteignable à l'arête  $i$  est une solution.

## Exemple

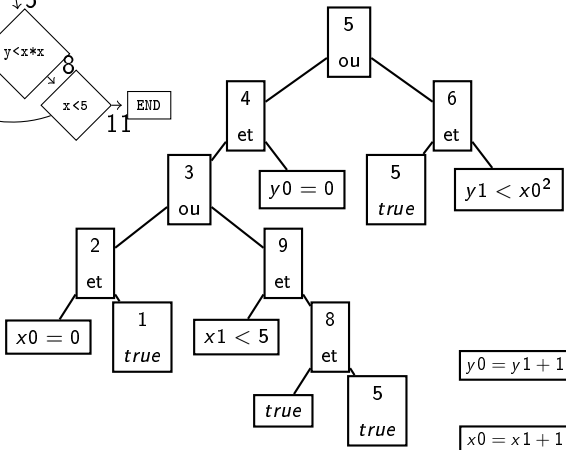
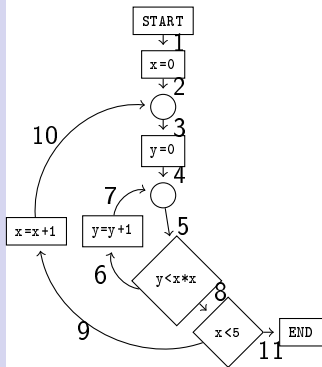
$$X = \{x, y\}$$

$$D = \mathbb{Z} \times \mathbb{Z}$$

$$C = \{x \geq 0, x \leq 4, y \geq 0, y \leq 16, \textcolor{red}{y} \leq \textcolor{red}{x}^2\}$$



# Arbre d'exécution symbolique



Estimation WCET

Préliminaires techniques

CFG et notations

Hypothèses

Interprétation abstraite

Programmation par  
contraintes

Contribution

CSP;

Arbres d'exécution  
symbolique

COP final

Implémentation et  
résultats

Implémentation

Résultats

Analyse d'un exemple

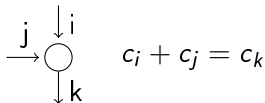
Conclusion

Annexes

## COP final

Pour chaque  $c_i$ , on a trouvé en comptant les solutions du CSP<sub>i</sub> un  $b_i$  tel que  $c_i \leq b_i$ .

$K$  : contraintes de la loi des nœuds sur le CFG.



## COP final

$$\begin{aligned} X &= \{c_i\} & C &= (\bigcap_i c_i \leq b_i) \cap K \\ D_i &= \mathbb{N} & O &= \sum_i c_i \times v_i \end{aligned}$$

Les  $v_i$  sont obtenues par analyse bas-niveau.

## Résultat

Majoration sûre de WCET

# Implémentation

## AbSolute

Solveur de contraintes. Peut compter les solutions, travailler avec des variables entières, minimiser une fonction. Utilise déjà l'interprétation abstraite.

## SawjaCard

Analyseur statique. Peut faire de l'interprétation abstraite avec le domaine des intervalles. Utilise une autre structure de CFG.

## Implémentation modulaire

Les modules qui, à partir d'un CFG, calculent les CSP; peuvent être utilisés avec d'autres solveurs de contraintes (CHOCO), d'autres analyseurs statiques (Interproc).

Estimation WCET

Préliminaires techniques

CFG et notations

Hypothèses

Interprétation abstraite

Programmation par  
contraintes

Contribution

CSP;

Arbres d'exécution  
symbolique

COP final

Implémentation et  
résultats

**Implémentation**

Résultats

Analyse d'un exemple

Conclusion

Annexes

benchmarks -> à faire

## Estimation WCET

### Préliminaires techniques

CFG et notations  
Hypothèses  
Interprétation abstraite  
Programmation par  
contraintes

### Contribution

CSP;  
Arbres d'exécution  
symbolique  
COP final

### Implémentation et résultats

Implémentation  
**Résultats**  
Analyse d'un exemple

### Conclusion

### Annexes

# Analyse d'un exemple

à faire

Estimation WCET

Préliminaires techniques

CFG et notations

Hypothèses

Interprétation abstraite

Programmation par  
contraintes

Contribution

CSP;

Arbres d'exécution  
symbolique

COP final

Implémentation et  
résultats

Implémentation

Résultats

**Analyse d'un exemple**

Conclusion

Annexes

## Conclusion

On tire profit de l'expressivité des langages de contraintes et de la précision de la programmation par contraintes pour affiner la première approximation faite par l'interprétation abstraite.

Dès que le programme exhibe dans son code des contraintes (même non linéaires), on peut les récupérer.

Agrandir le langage étudié.

On ne propose pas une expression paramétrique du WCET.

### Estimation WCET

#### Préliminaires techniques

CFG et notations  
Hypothèses  
Interprétation abstraite  
Programmation par  
contraintes

#### Contribution

CSP;  
Arbres d'exécution  
symbolique  
COP final

#### Implémentation et résultats

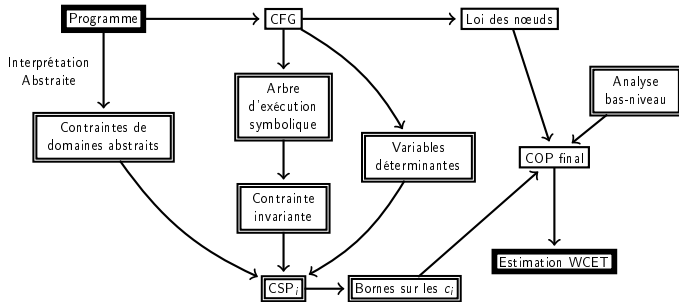
Implémentation  
Résultats  
Analyse d'un exemple

#### Conclusion

#### Annexes

# Estimation WCET haut-niveau, avec Interprétation abstraite et Programmation par contraintes

Aurèle Barrière



## Estimation WCET

## Préliminaires techniques

### CFG et notations

### Hypothèses

### Interprétation abstraite

### Programmation par contraintes

## Contribution

### CSP<sub>i</sub>

### Arbres d'exécution symbolique

### COP final

## Implémentation et résultats

### Implémentation

### Résultats

### Analyse d'un exemple

## Conclusion

## Annexes

# Variables déterminantes

## Estimation WCET

### Préliminaires techniques

CFG et notations  
Hypothèses  
Interprétation abstraite  
Programmation par  
contraintes

### Contribution

CSP;  
Arbres d'exécution  
symbolique  
COP final

### Implémentation et résultats

Implémentation  
Résultats  
Analyse d'un exemple

### Conclusion

### Annexes