Analyse de programmes

Rapport projet 1 sur l'analyse de polyèdres

Gaëtan Ramack

Université de Namur
Faculté d'informatique
Année académique 2023-2024

1 Description générale de l'analyseur

L'analyseur a pour but de pouvoir afficher le polytope convexe à 2 dimensions, autrement dit le polygone convexe représentant l'ensemble des valeurs possibles que pourrait prendre des points sujets à des contraintes linéaires sur un plan cartésien. Le programme prend en entrée le path vers un fichier respectant la syntaxe de Geom, une grammaire destinée à décrire des points et leurs contraintes linéaires.

Afin de déclarer un point, il faut lister ses contraintes. Les contraintes des différents points doivent être linéaires, et être décrites selon une syntaxe spécifique afin de ne pas trop alourdir les processus de parsing et de conversion en matrice d'inégalités. Tout d'abord, seules les inégalités non-strictes sont autorisées et seules les variables *"x"* et *"y"* peuvent êtres définies. Au moins une variable doit être définie et les variables restent du côté gauche de l'inégalité. La partie droite de l'inégalité ne peut contenir qu'un nombre.

Exemple de déclaration d'un point :

```
Constraint POINTNAME {
    x + y + 2 <= 6;
    -3x -7y - 2 >= 0;
    x + y >= 1;
    x >= 0;
    y <= 20;
}
```

Exemple de déclaration d'un interval

```
    Interval x of POINTNAME = [3, 10];
    Interval y of POINTNAME = [0, 10];
```

2 points ne peuvent avoir le même nom (*POINTNAME* dans l'exemple), doivent contenir au moins 1 contrainte et ne peuvent être définis qu'une fois. Toute déclaration d'intervale doit être faite après avoir la déclaration de tous les points. La valeur de gauche d'un interval doit toujours être inférieure ou égale à celle de droite. Les 2 parties de l'analyse sont forward.

2 Grammaire et Syntaxe Geom

2.1 Lexer

```
WS: [ \n\t\r]+ -> channel(HIDDEN);

MAIN: 'Main';
FUNC: 'Function';
CONS: 'Constraint';
LPAR: '(';
RPAR: ')';
LBRA: '{';
RBRA: '}';
LA: '[';
RA: ']';
COMMA: ',';
SEMICOLON: ';';
IF:'if';
THEN:'then';
ELSE:'else';
POINT: 'Point';
POLYHEDRON: 'Polyhedron';
NOT: '!';

EQUAL: '=';
PLUS: '+';
MINUS: '-';
```

```
MUL: '*';
DIV: '/';
MOD:'%';
G: '>';
OR: 'or';
AND: 'and';
L: '<';
GE: '>=';
LE: '<=';
DIFF: '!=';
BOOLEAN: 'true' | 'false' ;

ID: LETTER (LETTER | DIGIT)*;
VARID: LETTER (LETTER | DIGIT)*;

DQUOTE: '"';
STRING: DQUOTE(~[\\,\r\n])+DQUOTE;
CHAR: '\''(DIGIT | LETTER | ':' | '.' | '&' | '{' | '\\' | SEMICOLON)+'\'';
NUMERIC: DIGIT+;
fragment DIGIT: [0-9] ;
fragment LETTER: [A-Za-z] ;
```

2.2 Grammaire

```
main: MAIN LBRA (consdecl)+ (intervaldecl)* RBRA;

consdecl: CONS ID ':' LBRA lindecl* RBRA;
intervaldecl: 'Interval' ID 'of' ID EQUAL LA numericValue COMMA numericValue RA SEMICOLON;
lindecl: leftexpr rightexpr? (linop numericValue)? int_comp_op numericValue SEMICOLON;
numericValue: MINUS? NUMERIC (COMMA NUMERIC)?;

leftexpr: (numericValue | MINUS)? ID;
rightexpr: linop numericValue? ID;
booleq: DIFF | EQUAL;
int_comp_op: G | GE | L | LE;
booldecl: BOOLEAN;
num_op: PLUS | MINUS | DIV | MUL | MOD;
linop: PLUS | MINUS;

exprent:
LPAR exprent RPAR
| MINUS LPAR exprent RPAR
|numericValue num_op numericValue
|numericValue num_op MINUS? ID
| MINUS? ID num_op numericValue
| MINUS? ID num_op MINUS? ID
| exprent num_op exprent
| MINUS? ID
| numericValue
;

consexpr: exprent;

exprcomp: ID |
| booldecl
| exprent
;

exprbool:
ID
```

```
| NOT exprbool
| ID int_comp_op exprent
| ID int_comp_op numericValue
| numericValue int_comp_op ID
| exprent int_comp_op ID
| exprent int_comp_op exprent
| (exprcomp) booleq (exprcomp)
| exprbool AND exprbool
| exprbool OR exprbool
| booldecl
| LPAR exprbool RPAR
;
```

2.3 Semantique

[Linear constraint $\leq$] $\dfrac{(v_1,\sigma)\rightsquigarrow v_1 \quad (v_2,\sigma)\rightsquigarrow v_2 \quad (v_3,\sigma)\rightsquigarrow v_3 \quad (v_4,\sigma)\rightsquigarrow v_4 \quad \sigma'=\sigma[p\mapsto[m_p^T \quad [-v_4-v_3,-v_1,-v_2]]]}{(v_1x+v_2y+v_3\leq v_4,E\bullet\sigma)\rightsquigarrow E\bullet\sigma'}$

[Linear constraint $\geq$] $\dfrac{(v_1,\sigma)\rightsquigarrow v_1 \quad (v_2,\sigma)\rightsquigarrow v_2 \quad (v_3,\sigma)\rightsquigarrow v_3 \quad (v_4,\sigma)\rightsquigarrow v_4 \quad \sigma'=\sigma[p\mapsto[m_p^T \quad [-v_4+v_3,v_1,v_2]]]}{(v_1x+v_2y+v_3\geq v_4,E\bullet\sigma)\rightsquigarrow E\bullet\sigma'}$

[Interval definition for variable x] $\dfrac{(p,\sigma)\rightsquigarrow m_p \quad (v_1,\sigma)\rightsquigarrow v_1 \quad (v_2,\sigma)\rightsquigarrow v_2 \quad \sigma'=\sigma[p\mapsto[m_p^T \quad [[-v_1,1,0],[v_2,1,0]]]]}{(Interval \quad x \quad of \quad p \quad = \quad [v_1,v_2],E\bullet\sigma)\rightsquigarrow E\bullet\sigma'}$

[Interval definition for variable y] $\dfrac{(p,\sigma)\rightsquigarrow m_p \quad (v_1,\sigma)\rightsquigarrow v_1 \quad (v_2,\sigma)\rightsquigarrow v_2 \quad \sigma'=\sigma[p\mapsto[m_p^T \quad [[-v_1,0,1],[v_2,0,1]]]]}{(Interval \quad y \quad of \quad p \quad = \quad [v_1,v_2],E\bullet\sigma)\rightsquigarrow E\bullet\sigma'}$

m_p est la matrice $[b \quad -A]$ (ou H-representation matrix) du point p

3 Définition du domaine et de la lattice

Le domaine abstrait utilisé a pour but de déterminer si un point peut avoir une, aucune, ou plusieurs possibilités de valeurs dans un plan cartésien. La région des valeurs possibles d'un point définis par des contraintes linéaires est un polytope convexe, un ensemble d'intersections de half-spaces qui sont eux définis par des contraintes linéaires inégales.

L'analyseur se sert de la librairie *pycddlib*, un wrapper Python de la librairie cddlib de Komei Fukuda. Cette librairie implémente la Double Description Method de Motzkin et al., permettant de passer de représentation V à H et vice-versa d'un polytope convexe.

3.1 Domaine

Soit $D = \{\top = \{m_p | m \in M \wedge \text{V-representation}(m_p) \neq \{\}, \bot = \{m_p | m \in M \wedge \text{V-representation}(m_p) = \{\}\}$
où M est le set de matrices H-representation des différents points.

3.2 Fonctions d'abstraction & concrétisation

:

— $\alpha(p) = [t \quad V]$ où $[t \quad V]$ est la V-representation matrix du polyhèdre p décris par la H-representation matrix $[b \quad -A]$

m = nombre de contraintes

b est un vecteur colonne $m \times 1$ dont les coordonnées sont les valeurs à droite des inégalités.

A est une matrice $m \times 2$ reprenant les valeurs des coefficients des variables x et y

— $\gamma([t \quad V]) = \gamma(\top) = \{[a_1, b_1], ..., [a_n, b_n]\}$ où $\forall i : 0 \leq i \leq n, \quad a_i, b_i \in \mathbb{Q}$ lorsqu'il existe au moins un générateur du polygone

— $\gamma(\{\}) = \gamma(\bot) = \{\}$ lorsqu'il n'y a aucun intervale de valeur possible pour le set de contraintes linéaires

3.3 Lattice :

3.4 Flow functions

$f[[v_1x + v_2y + v_3 \leq v_4;]](\phi) = \phi[[p \mapsto \top]] = \phi[[p \mapsto [m_p^T \quad [-v_3 - v_4, -v_1, -v_2] \ ] \ ]]$ if V-representation matrix de $[m_p^T \quad [-v_3 - v_4, -v_1, -v_2] \ ]$ est $\neq \{\}$

$f[[v_1x + v_2y + v_3 \geq v_4;]](\phi) = \phi[[p \mapsto \top]] = \phi[[p \mapsto [m_p^T \quad [v_3 - v_4, v_1, v_2] \ ] \ ]]$ if V-representation matrix de $[m_p^T \quad [v_3 - v_4, v_1, v_2] \ ]$ est $\neq \{\}$

$f[[v_1x + v_2y + v_3 \leq v_4;]](\phi) = \phi[[p \mapsto \bot]] = \phi[[p \mapsto [m_p^T \quad [-v_3 - v_4, -v_1, -v_2] \ ] \ ]]$ if V-representation matrix de $[m_p^T \quad [-v_3 - v_4, -v_1, -v_2] \ ]$ est $= \{\}$

$f[[v_1x + v_2y + v_3 \geq v_4;]](\phi) = \phi[[p \mapsto \bot]] = \phi[[p \mapsto [m_p^T \quad [v_3 - v_4, v_1, v_2] \ ] \ ]]$ if V-representation matrix de $[m_p^T \quad [v_3 - v_4, v_1, v_2] \ ]$ est $= \{\}$

$f[[Interval\ x\ of\ p = [v_1, v_2] \ ]](\phi) = \phi[[p \mapsto \top]] = \phi[[p \mapsto [m_p^T \quad [[-v_1, 1, 0], [v_2, 1, 0]]]]]$ if V-representation matrix de $[m_p^T \quad [[-v_1, 1, 0], [v_2, 1, 0]]]$ est $\neq \{\}$

$f[[Interval\ y\ of\ p = [v_1, v_2] \ ]](\phi) = \phi[[p \mapsto \top]] = \phi[[p \mapsto [m_p^T \quad [[-v_1, 0, 1], [v_2, 0, 1]]]]]$ if V-representation matrix de $[m_p^T \quad [[-v_1, 0, 1], [v_2, 0, 1]]]$ est $\neq \{\}$

$f[[Interval\ x\ of\ p = [v_1, v_2] \ ]](\phi) = \phi[[p \mapsto \bot]] = \phi[[p \mapsto [m_p^T \quad [[-v_1, 1, 0], [v_2, 1, 0]]]]]$ if V-representation matrix de $[m_p^T \quad [[-v_1, 1, 0], [v_2, 1, 0]]]$ est $= \{\}$

$f[[Interval\ y\ of\ p = [v_1, v_2] \ ]](\phi) = \phi[[p \mapsto \bot]] = \phi[[p \mapsto [m_p^T \quad [[-v_1, 0, 1], [v_2, 0, 1]]]]]$ if V-representation matrix de $[m_p^T \quad [[-v_1, 0, 1], [v_2, 0, 1]]]$ est $= \{\}$

m_p est la matrice $[b \quad -A]$ (ou H-representation matrix) du point p

4 Analyse

4.1 Cas d'erreurs

Etant donné la grammaire de Geom et le parsing des contraintes, il a fallut diviser l'analyse en 2 parties. La première partie est chargée d'analyser la syntaxe afin que les contraintes ne pouvant pas être spécifiées par la grammaire soient respectées. Cette première partie d'analyse se trouve dans le fichier *SyntaxAnalyzerListener.py* et va donc renvoyer une liste d'erreurs ou un message d'erreur pour les cas suivants :

- Lorsqu'une erreur de syntaxe ne respectant pas la grammaire est observée. (exemple : oublier un *";"* à la fin d'une contrainte ou lorsque une déclaration d'interval est faite avant que la déclaration des contraintes des points ne soit faite.)
- Lorsqu'un point a déjà été défini auparavant.
- Lorsqu'une variable autre que x ou y a été entrée dans une contrainte linéaire.
- Lorsqu'une contrainte linéaire comporte 2 x ou y.
- Lorsqu'une contrainte linéaire ne présente aucune variable du côté gauche de l'inégalité.
- Lorsqu'un nom de point inconnu est utilisé dans la déclaration d'interval d'une variable d'un point.
- Lorsqu'un nom de variable est différent de x ou y dans la déclaration d'interval d'une variable d'un point.
- Lorsqu'un interval est == 0.

La seconde partie de l'analyse est quant à elle chargée de déterminer si un point a au moins une réponse possible dans le graphe cartésien. Lorsque la première partie de l'analyse se termine sans renvoyer d'exception, la

seconde partie située dans le fichier *PolytopeAnalyzerListener.py* entre en scène et va se charger de repasser dans l'arbre syntaxique afin de pouvoir récupérer les contraintes et intervals de valeurs définits des différents points. La seule exception renvoyée par cette partie survient lorsque les contraintes d'un point ne permettent pas de solutions possibles.

4.2 GEN/KILL

```
1    Main {
2        Constraint EX : {
3            x >= 2;
4            y >= 1;
5        }
6        Constraint EX2 : {
7           x >= 0;
8           x + 3 <= 10;
9        }
10       Interval x of EX = [3, 15];
11       Interval y of EX = [-30, 0];
12   }
```

Soit $D = \{m_p | m \in M \wedge p \in PP\}$
où M est le set de matrices des différents points et P P le set de points de programmes.

$\top = M$
$\bot = \{\}$

GEN(p) = $\{m_p\}$ if P[p] $\equiv l1 + l2 + l3 \leq l4$
$\{m_p\}$ if P[p] $\equiv l1 + l2 + l3 \geq l4$
$\{m_p\}$ if P[p] $\equiv x$ of $v = [i1, i2]$
$\{m_p\}$ if P[p] $\equiv y$ of $v = [i1, i2]$
$\{\}$ otherwise

KILL(p) = $\{m_p\}$ if P[p] $\equiv l1 + l2 + l3 \leq l4$ and V-representation matrix of convex polygon defined by H-representation matrix $[m_{p-1}^T \quad [-l4 - l3, -l1, -l2]]^T$ contains vertices
$\{m_p\}$ if P[p] $\equiv l1 + l2 + l3 \geq l4$ and V-representation matrix of convex polygon defined by H-representation matrix $[m_{p-1}^T \quad [-l4 + l3, l1, l2]]^T$ contains no vertices
$\{m_p\}$ if P[p] $\equiv x$ of $v = [i1, i2]$ and V-representation matrix of convex polygon defined by H-representation matrix $[m_{p-1}^T \quad [[-i1, 1, 0], [i2, 1, 0]]]^T$ contains no vertices
$\{m_p\}$ if P[p] $\equiv y$ of $v = [i1, i2]$ and V-representation matrix of convex polygon defined by H-representation matrix $[m_{p-1}^T \quad [[-i1, 0, 1], [i2, 0, 1]]]^T$ contains no vertices
$\{\}$ otherwise

KILL(p) = $\{m_i | m_i \in M, i \neq p\}$

p	IN(p)	KILL(p)	GEN(p)	OUT(p)
3	$\{\}$	$\{\}$	$\{EX_3\}$	$\{EX_3\}$
4	$\{EX_3\}$	$\{EX_3\}$	$\{EX_4\}$	$\{EX_4\}$
7	$\{EX_4\}$	$\{\}$	$\{EX2_7\}$	$\{EX_4, EX2_7\}$
8	$\{EX_4, EX2_7\}$	$\{EX2_7\}$	$\{EX2_8\}$	$\{EX_4, EX2_8\}$
10	$\{EX_4, EX2_8\}$	$\{EX_4\}$	$\{EX_{10}\}$	$\{EX_{10}, EX2_8\}$
11	$\{EX_{10}, EX2_8\}$	$\{EX_{10}, EX_{11}\}$	$\{EX_{11}\}$	$\{EX2_8\}$

5 Exemple with worklist

prenons le cas de problem3 :

```
1    Main {
2        Constraint EX : {
3            x >= 2;
4            y >= 1;
5        }
6        Constraint EX2 : {
7           x >= 0;
8           x + 3 <= 10;
9        }
10       Interval x of EX = [3, 15];
11   }
```

Dans le tableau qui suit, res(EX) représente la V-representation matrix de EX & res(EX2) représente la V-representation matrix de EX2.

p	WorkList	ϕ_p(EX)	ϕ_p(EX2)	res(EX)	res(EX2)
3	4	$\perp$	$\perp$	$\begin{matrix}0&1&0\\1&2&0\\0&0&1\end{matrix}$	$\perp$
4	7	$\begin{matrix}0&1&0\\1&2&0\\0&0&1\end{matrix}$	$\perp$	$\begin{matrix}0&1&0\\0&0&1\\1&2&1\end{matrix}$	$\perp$
7	8	$\begin{matrix}0&1&0\\0&0&1\\1&2&1\end{matrix}$	$\perp$	$\begin{matrix}0&1&0\\1&2&0\\0&0&1\end{matrix}$	$\begin{matrix}0&1&0\\0&0&1\end{matrix}$
8	10	$\begin{matrix}0&1&0\\0&0&1\\1&2&1\end{matrix}$	$\begin{matrix}0&1&0\\0&0&1\end{matrix}$	$\begin{matrix}0&1&0\\1&2&0\\0&0&1\end{matrix}$	$\begin{matrix}1&0&0\\1&7&0\\0&0&1\end{matrix}$
10	11	$\begin{matrix}0&1&0\\0&0&1\\1&2&1\end{matrix}$	$\begin{matrix}1&0&0\\1&7&0\\0&0&1\end{matrix}$	$\begin{matrix}1&3&1\\0&1&0\\0&0&1\end{matrix}$	$\begin{matrix}1&0&0\\1&7&0\\0&0&1\end{matrix}$
11	/	$\begin{matrix}1&3&1\\0&1&0\\0&0&1\end{matrix}$	$\begin{matrix}1&0&0\\1&7&0\\0&0&1\end{matrix}$	$\begin{matrix}1&3&1\\0&1&0\\0&0&1\end{matrix}$	$\begin{matrix}1&0&0\\1&7&0\\0&0&1\end{matrix}$

6 Instructions d'installation

La librairie *pycddlib* ne marche que sur les versions de python entre 3.7 et 3.11. Commande d'installation des packages nécessaires :

```
pip install pycddlib matplotlib antlr4-tools antlr4-python3-runtime sympy sympy-plot-backends
```

Exécuter cette commande dans la source du projet (Il est nécessaire sur Windows d'ajouter le path du dossier scripts de python si cela n'est pas déjà fait) :

```
antlr4
```

Cette commande permet de télécharger antlr4 et Java 11 si cela est nécessaire. Si tout est installé correctement, il suffit juste d'exécuter *analyzer.py* avec comme argument le path vers le fichier à analyser.