Max-plus automata for the Worst-Case Complexity Analysis

Laure Daviaud
LIP, ENS Lyon

Work in Progress

DICE, 2 avril 2016
Purpose: Worst-case complexity analysis

Worst-Case complexity
[MFCS’14, Colcombet, D., Zuleger]

Program → Abstraction
Size-change → Max-plus automaton

Work in progress
Purpose: Worst-case complexity analysis

Worst-Case complexity
[MFCS’14, Colcombet, D., Zuleger]

Program

Abstraction
Size-change

Max-plus automaton

Work in progress
Max-plus automata: worst case analysis

Syntax:
Non deterministic finite automaton for which each transition is labelled by a non negative integer (weight).

Semantic:
Weight of a run = sum of the weights of the transitions.

\[ A^* \rightarrow N \cup \{-\infty\} \]

\[ w \mapsto \] Maximum of the weights of accepting runs labelled by \( w \) (\(-\infty\) if no such run)
Max-plus automata : worst case analysis

Syntax : Non deterministic finite automaton for which each transition is labelled by a non negative integer (weight).

Semantic : Weight of a run = sum of the weights of the transitions.

A∗ → N ∪ {−∞} w ↦→ Maximum of the weights of accepting runs labelled by w (−∞ if no such run)
Max-plus automata: worst case analysis

Syntax:
Non deterministic finite automaton
for which each transition is labelled
by a non negative integer (weight).

Semantic:
Weight of a run
= sum of
the weights of the transitions.

\( A^* \rightarrow N \cup \{-\infty\} \)
\( w \mapsto \) Maximum of the weights
of accepting runs
\( (-\infty \text{ if no such run}) \)

Max-plus automata: worst case analysis
Max-plus automata: worst case analysis
Max-plus automata: worst case analysis

Syntax:

Non deterministic finite automaton
for which each transition is labelled
by a non negative integer (weight).

Semantic:

Weight of a run
\(=\) sum of
the weights of the transitions.

\(A^* \rightarrow N \cup \{ -\infty \}\)

\(w \mapsto \) Maximum of the weights
of accepting runs
\((-\infty\) if no such run)
Max-plus automata: worst case analysis

Syntax:

Non deterministic finite automaton for which each transition is labelled by a non negative integer (weight).

Semantic:

Weight of a run \(=\) sum of the weights of the transitions.

\[ A^* \rightarrow \mathbb{N} \cup \{-\infty\} \]

\[ w \mapsto \text{Maximum of the weights of accepting runs labelled by } w \]

\((-\infty \text{ if no such run})\)
What is the computed function?

\[ a : 0, \quad b : 1 \]

\[ q_1 \]

\[ a, b : 0 \]

\[ q_2 \]

\[ b : 0 \]

\[ q_3 \]

\[ a : 1 \]

\[ q_4 \]

\[ a, b : 0 \]
What is the computed function?

\[ a^n b a^{n_1} b \cdots b a^{n_k} \mapsto \max(n_0, n_1, \ldots, n_k, k) \]
Max-plus automata: undecidability results

UNDECIDABLE [Krob '92]

- Equivalence of max-plus automata
- Comparison of functions: for all $w$, $f(w) \leq g(w)$

Generally speaking, getting precise descriptions of functions computed by max-plus automata is very difficult...
Purpose: Worst-case complexity analysis

Worst-Case complexity
[MFCS’14, Colcombet, D., Zuleger]

Program ➔ Abstraction

Size-change

Max-plus automaton

Work in progress
Idea: Look at the decrease of the variables

Input x, y :
while x>=0 {
    y--;
    if y=0 {
        x--;  
        y=random();
    }
}
Size-change abstraction on an exemple

Idea: Look at the decrease of the variables

Input $x, y$ :

while $x \geq 0$ {
    $y--$;  
    if $y=0$ {
        $x--$;  
        $y=$random();  
    }
}

$t_1$: $x \geq x'$, $y > y'$
Idea: Look at the decrease of the variables

Input \( x, y \):

\[
\text{while } x \geq 0 \{
    \text{\color{red}y--;}
    \text{if } y = 0 \{
        \text{\color{blue}x--;}
        \text{\color{blue}y = \text{random}();}
    \}
\}
\]

\(t_1\): \( x \geq x', y > y' \)

\(t_2\): \( x > x' \)
Size-change abstraction

- Finite number of variables (values in $\mathbb{N}$)
- Transition: conjunction of a finite number of predicates of the form $x_i > x'_j$ or $x_i \geq x'_j$
- Trace: sequence of transitions and valuations compatible

$t_1$: $x \geq x', y > y'$

$t_2$: $x > x'$
Size-change abstraction

\[ t_1: x \geq x', y > y' \]

\[ t_2: x > x' \]

- Finite number of variables (values in \( \mathbb{N} \))
- Transition: conjunction of a finite number of predicates of the form \( x_i > x_j' \) or \( x_i \geq x_j' \)
- Trace: sequence of transitions and valuations compatible

\[(5, 5) \xrightarrow{t_1} (5, 4) \xrightarrow{t_2} (4, 8) \xrightarrow{t_1} (3, 6) \xrightarrow{t_1} (3, 2) \xrightarrow{t_2} (2, 10) \ldots\]
Size-change abstraction

$t_1: x \geq x', y > y'$

$t_2: x > x'$

- Finite number of variables (values in $\mathbb{N}$)
- Transition: conjunction of a finite number of predicates of the form $x_i > x_j'$ or $x_i \geq x_j'$
- Trace: sequence of transitions and valuations compatible

$\begin{align*}
(5, 5) &\xrightarrow{t_1}(5, 4) \xrightarrow{t_2}(4, 8) \xrightarrow{t_1}(3, 6) \xrightarrow{t_1}(3, 2) \xrightarrow{t_2}(2, 10) \ldots
\end{align*}$

Terminating sca: no infinite trace
Size-change abstraction

\[ t_1: \; x \geq x', \; y > y' \]

\[ t_2: \; x > x' \]

- Finite number of variables (values in \( \mathbb{N} \))
- Transition: conjunction of a finite number of predicates of the form \( x_i > x'_j \) or \( x_i \geq x'_j \)
- Trace: sequence of transitions and valuations compatible

\[
(5, 5) \xrightarrow{t_1} (5, 4) \xrightarrow{t_2} (4, 8) \xrightarrow{t_1} (3, 6) \xrightarrow{t_1} (3, 2) \xrightarrow{t_2} (2, 10) \ldots
\]

Terminating sca: no infinite trace

**Theorem [Lee, Jones, Ben-Amram]**

It is decidable whether a given sca instance is terminating.
Size-change abstraction: longest traces

$t_1: x \geq x', y > y'$

- Terminating
- Traces of unbounded lengths

$t_2: x > x'$
Size-change abstraction: longest traces

$t_1: x \geq x', y > y'$

- Terminating
- Traces of unbounded lengths

$t_2: x > x'$

Restriction to $[0, n]$, what is the length of the longest trace?
Size-change abstraction: longest traces

$t_1: x \geq x', y > y'$

- Terminating
- Traces of unbounded lengths

$t_2: x > x'$

Restriction to $[0, n]$, what is the length of the longest trace?

\[

t_2: (n, n) \xrightarrow{t_1} (n, n-1) \xrightarrow{t_1} \cdots \xrightarrow{t_1} (n, 0) \\
(0, n) \xrightarrow{t_2} (n-1, n) \xrightarrow{t_1} (n-1, n-1) \xrightarrow{t_1} \cdots \xrightarrow{t_1} (n-1, 0) \\
(0, n-1) \xrightarrow{t_1} \cdots \xrightarrow{t_1} (0, 0)
\]
Given a terminating sca instance, there is a computable rational \( \alpha \geq 1 \) such that

\[
f = \Theta(n^{\alpha})
\]

where \( f \) associates a positive integer \( n \) to the length of the longest traces if the variables are restricted to be in \([0, n]\).
Purpose: Worst-case complexity analysis

Worst-Case complexity
[MFCS’14, Colcombet, D., Zuleger]

Program → Abstraction
Size-change → Max-plus automaton

Work in progress
From sca to max-plus automata

\[ t_1: x \geq x', y > y' \]

\[ t_2: x > x' \]
From sca to max-plus automata

\[ t_1: x \geq x', \quad y > y' \]

\[ t_2: x > x' \]
From sca to max-plus automata

\[ t_1: x \geq x', y > y' \]

\[ t_2: x > x' \]

\[ t_1: 0 \]

\[ t_1: 1 \]

\[ x \]

\[ y \]
From sca to max-plus automata

\[ t_1: \ x \geq x', \ y > y' \]

\[ t_2: \ x > x' \]
From sca to max-plus automata

$t_1: x \geq x', y > y'$

$t_2: x > x'$
From sca to max-plus automata

\[ t_1: x \geq x', \quad y > y' \]

\[ t_2: x > x' \]

\[ (2, 2) \xrightarrow{t_1}(2, 1) \xrightarrow{t_1}(2, 0) \xrightarrow{t_2}(1, 2) \xrightarrow{t_1}(1, 1) \xrightarrow{t_1}(1, 0) \xrightarrow{t_2}(0, 2) \xrightarrow{t_1}(0, 1) \xrightarrow{t_1}(0, 0) \]

\[ x \geq x', \quad x \geq x', \quad x > x', \quad x \geq x', \quad x > x', \quad x > x', \quad x > x', \quad x > x' \]
From sca to max-plus automata

(2, 2) \xrightarrow{t_1}(2, 1) \xrightarrow{t_1}(2, 0) \xrightarrow{t_2}(1, 2) \xrightarrow{t_1}(1, 1) \xrightarrow{t_1}(1, 0) \xrightarrow{t_2}(0, 2) \xrightarrow{t_1}(0, 1) \xrightarrow{t_1}(0, 0)

\begin{align*}
x &\geq x' \\
y &> y' \\
x &> x' \\
y &> y'
\end{align*}

Longest trace when variables do not exceed $n$

\begin{align*} 
\quad &\quad \rightarrow \text{Sequences of inequalities with at most } n \text{ strict inequalities} \\
\rightarrow &\quad \text{Runs of weight at most } n \\
\quad &\quad \text{Longest word of weight at most } n
\end{align*}
Max-plus automata: main theorem

\[ g : \mathbb{N} \rightarrow \mathbb{N} \cup \{+\infty\} \]
\[ n \mapsto \sup_{f(w) \leq n} |w| \]

There is an algorithm with input a max-plus automaton, that computes a rational \( \alpha \geq 1 \) such that:

\[ g(n) = \Theta(n^\alpha) \]
Purpose: Worst-case complexity analysis

Program → Abstraction (Size-change) → Max-plus automaton

Worst-Case complexity
[MFCS’14, Colcombet, D., Zuleger]

Work in progress
Work in progress

Two directions

- Compositionality
- Decrease and increase of the variables
Compositionality

\[ f : (\alpha, \beta) \] - asymptotic equivalents of the decrease of the variable AND the time taken by the execution

Input \( x, y \):

```java
Input x, y :
while (true) {
    if (cond) {
        x--;
        y=f(x); }
    else {
        y--; }
}
```

Diagram: [Diagram showing transitions and time intervals]
**Problem:** Give an asymptotic equivalent of the length of the longest trace in the program when variables are restricted to be in \([-n, n]\).

Study of max-plus automata with negative weights that compute positive functions.

Main result no longer true in general but...