Algorithmic Theory of WQOs

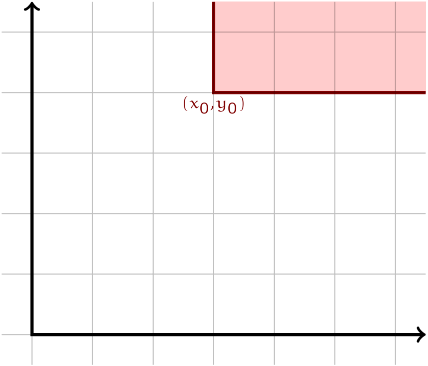
S. Schmitz
joint work with D. Figueira, S. Figueira, S. Haddad, and Ph. Schnoebelen

LSV, ENS Cachan & CNRS

Ker-Lann, November 27, 2012

A One-Player Game

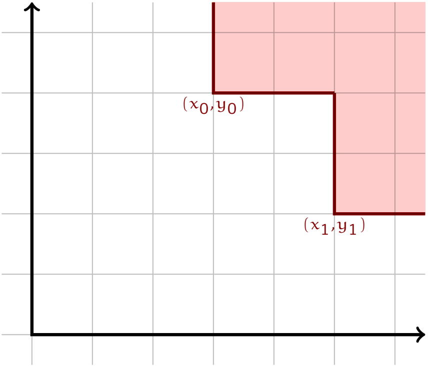
- over $\mathbb{Q}_{\geqslant 0} \times \mathbb{Q}_{\geqslant 0}$
- given initially (x_0, y_0)
- Eloise plays (x_j, y_j) s.t. $\forall 0 \leqslant i < j, x_i > x_j$ or $y_i > x_j$



- Can Eloise win, i.e. play indefinitely?
- If no, how long can she last?

A One-Player Game

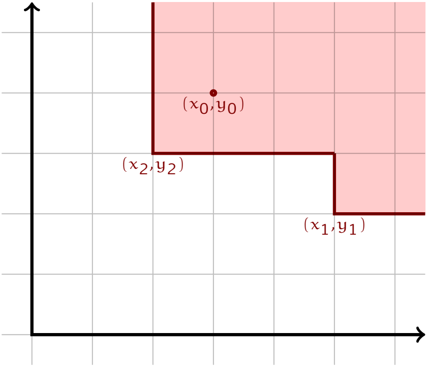
- over $\mathbb{Q}_{\geqslant 0} \times \mathbb{Q}_{\geqslant 0}$
- given initially (x_0, y_0)
- Eloise plays (x_j, y_j) s.t. $\forall 0 \leqslant i < j, x_i > x_j$ or $y_i > x_j$



- Can Eloise win, i.e. play indefinitely?
- If no, how long can she last?

A One-Player Game

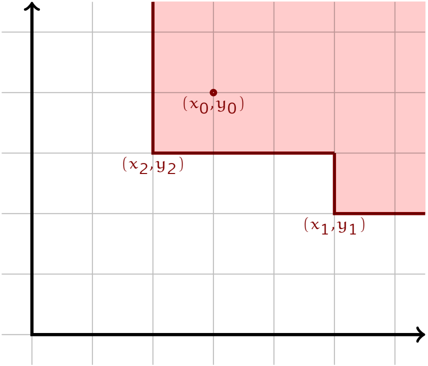
- over $\mathbb{Q}_{\geqslant 0} \times \mathbb{Q}_{\geqslant 0}$
- given initially (x_0, y_0)
- Eloise plays (x_j, y_j) s.t. $\forall 0 \leqslant i < j, x_i > x_j$ or $y_i > x_j$
- Can Eloise win, i.e. play indefinitely?
- If no, how long can she last?



If $(x_0, y_0) \neq (0,0)$, then choosing $(x_j, y_j) = (\frac{x_0}{2^j}, \frac{y_0}{2^j})$ wins.

A One-Player Game

- over $\mathbb{N} \times \mathbb{N}$
- given initially (x_0, y_0)
- Eloise plays (x_j, y_j) s.t. $\forall 0 \leqslant i < j, x_i > x_j$ or $y_i > x_j$



- Can Eloise win, i.e. play indefinitely?
- If no, how long can she last?

Assume there exists an infinite sequence $(x_j, y_j)_j$ of moves over $\mathbb{N}^2$. Consider the pairs of indices $i < j$: color (i, j)

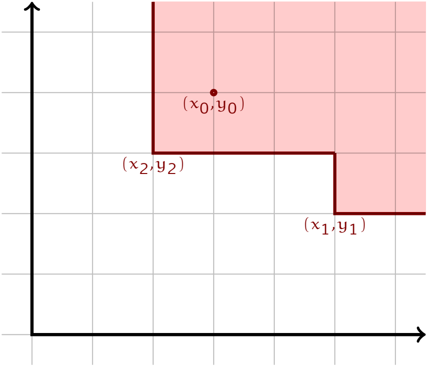
- purple if $x_i > x_j$ but $y_i \leqslant y_j$,
- red if $x_i > x_j$ and $y_i > y_j$,
- orange if $y_i > y_j$ but $x_i \leqslant x_j$.

$$(3,4) \qquad (5,2) \qquad (2,3) \qquad \cdots$$

By the infinite Ramsey Theorem, there exists an infinite monochromatic subset of indices. In all cases, it implies the existence of an infinite decreasing sequence in $\mathbb{N}$, a contradiction.

A One-Player Game

- over $\mathbb{N} \times \mathbb{N}$
- given initially (x_0, y_0)
- Eloise plays (x_j, y_j) s.t. $\forall 0 \leqslant i < j, x_i > x_j$ or $y_i > x_j$



- Can Eloise win, i.e. play indefinitely?
- If no, how long can she last?

- if $(x_0, y_0) = (0,0)$, 0 turns
- otherwise, an arbitrary number of turns N: if $x_0 > 0$:

$$(x_0, y_0), (0, N-1), (0, N-2), \ldots, (0,1), (0,0)$$

Well Quasi Orderings

Definition (wqo)

A wqo is a quasi-order $(A, \leqslant)$ s.t.

$$\forall \mathbf{x} = x_0, x_1, x_2, \ldots \in A^\omega, \exists i_1 < i_2, x_{i_1} \leqslant x_{i_2} .$$

Well Quasi Orderings

Definition (wqo)

A wqo is a quasi-order $(A, \leqslant)$ s.t.

$$\forall \mathbf{x} = x_0, x_1, x_2, \ldots \in A^\omega, \exists i_1 < i_2, x_{i_1} \leqslant x_{i_2} .$$

Example (Basic wqos)

- $(\mathbb{N}, \leqslant)$,
- $(\{0, 1, \ldots, k\}, \leqslant)$ for any $k \in \mathbb{N}$,
- $(\Gamma_p, =)$ for any finite set Γ_p with p elements.

Dickson's Lemma

Lemma

If $(A_1, \leqslant_{A_1})$ *and* $(A_2, \leqslant_{A_2})$ *are two wqos, then* $(A_1 \times A_2, \leqslant_\times)$ *is a wqo,*

where $\leqslant_\times$ *is the product ordering:*

$$\langle a_1, a_2 \rangle \leqslant_\times \langle b_1, b_2 \rangle \overset{\text{def}}{\Leftrightarrow} a_1 \leqslant_{A_1} b_1 \wedge a_2 \leqslant_{A_2} b_2 \,.$$

Example

$(\mathbb{N}^k, \leqslant)$ using the product ordering

Outline

well quasi orderings (wqo)

generic tools for termination arguments

but also

beyond termination: complexity bounds

contents

WQO Algorithms
Length Function Theorems
A Quick Survey

A Rich Theory

- multiple equivalent definitions
- algebraic constructions

A Rich Theory

- multiple equivalent definitions: $(A, \leqslant)$ wqo iff
  - $\leqslant$ is well-founded and has no infinite antichains,
  - every linearization of $\leqslant$ is well-founded,
  - $\leqslant$ has the Ascending Chain Condition,
  - if $\mathbf{x} = x_0 x_1 \cdots \in A^\omega$, then there exists an infinite sequence $i_0 < i_1 < \cdots$ with $x_{i_0} \leqslant x_{i_1} \leqslant \cdots$,
  - etc.
- algebraic constructions

A Rich Theory

- multiple equivalent definitions
- algebraic constructions
  - cartesian products (Dickson's Lemma),
  - finite sequences (Higman's Lemma),
  - disjoint sums,
  - finite sets,
  - finite trees (Kruskal's Tree Theorem),
  - graphs with minors (Robertson and Seymour's Theorem), etc.

Higman's Lemma

Lemma

If $(A, \leqslant)$ is a wqo, then $(A^*, \leqslant_*)$ is a wqo where $\leqslant_*$ is the subword embedding ordering:

$$a_1 \cdots a_m \leqslant_* b_1 \cdots b_n \overset{\text{def}}{\Leftrightarrow} \begin{cases} \exists 1 \leqslant i_1 < \cdots < i_m \leqslant n, \\ \bigwedge_{j=1}^{m} a_j \leqslant_A b_{i_j} \ . \end{cases}$$

Example

$$aba \leqslant_* baaacabbab$$

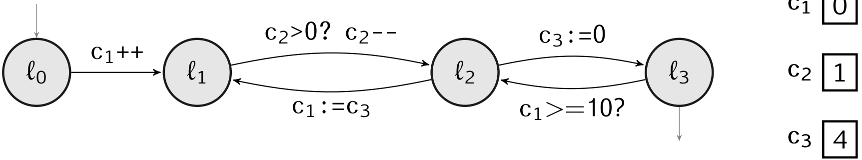
WQOs for Termination

Bad Sequences

- $\mathbf{x} = x_0, x_1, \ldots$ in A^∞ is a good sequence if $\exists i_1 < i_2, x_{i_1} \leqslant x_{i_2}$,
- a bad sequence otherwise,
- if $(A, \leqslant)$ is a wqo: every bad sequence is finite

Example

Monotonic Counter Systems



- A run of M:

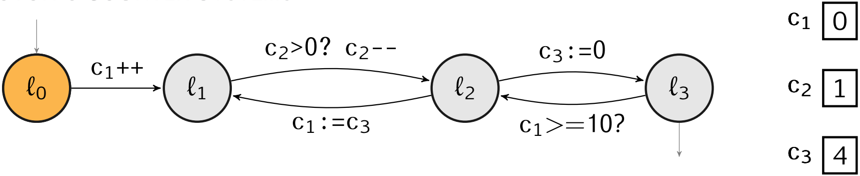
$$(\ell_0,0,1,4) \to (\ell_1,1,1,4) \to (\ell_2,1,0,4) \to (\ell_3,1,0,0)$$

- ordering configurations: $(\ell_1,0,0,0) \leqslant (\ell_1,0,1,2)$ but $(\ell_1,0,0,0) \not\leqslant (\ell_2,0,1,2)$

- a wqo as a product of wqos: $(Loc,=) \times (\mathbb{N}^3,\leqslant_\times)$

Example

Monotonic Counter Systems



- A run of M:

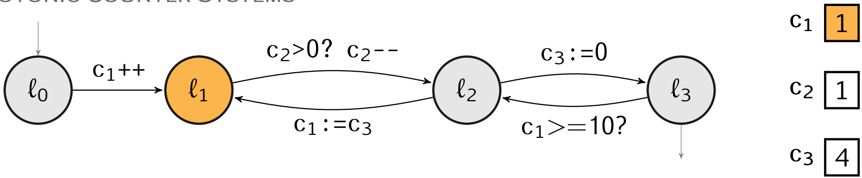
$$(\ell_0,0,1,4) \to (\ell_1,1,1,4) \to (\ell_2,1,0,4) \to (\ell_3,1,0,0)$$

- ordering configurations: $(\ell_1,0,0,0) \leqslant (\ell_1,0,1,2)$ but $(\ell_1,0,0,0) \not\leqslant (\ell_2,0,1,2)$

- a wqo as a product of wqos: $(Loc,=) \times (\mathbb{N}^3,\leqslant_\times)$

Example

Monotonic Counter Systems



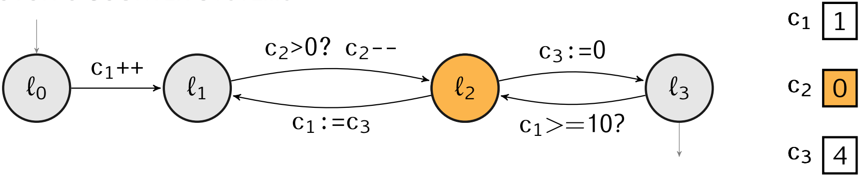
- A run of M:

$$(\ell_0,0,1,4) \to (\ell_1,1,1,4) \to (\ell_2,1,0,4) \to (\ell_3,1,0,0)$$

- ordering configurations: $(\ell_1,0,0,0) \leqslant (\ell_1,0,1,2)$ but $(\ell_1,0,0,0) \not\leqslant (\ell_2,0,1,2)$
- a wqo as a product of wqos: $(Loc,=) \times (\mathbb{N}^3,\leqslant_\times)$

Example

Monotonic Counter Systems



- A run of M:

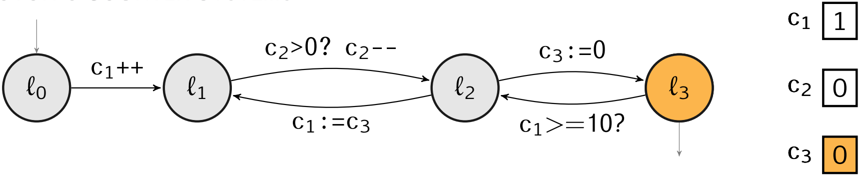
$$(\ell_0,0,1,4) \to (\ell_1,1,1,4) \to (\ell_2,1,0,4) \to (\ell_3,1,0,0)$$

- ordering configurations: $(\ell_1,0,0,0) \leqslant (\ell_1,0,1,2)$ but $(\ell_1,0,0,0) \not\leqslant (\ell_2,0,1,2)$

- a wqo as a product of wqos: $(Loc,=) \times (\mathbb{N}^3,\leqslant_\times)$

Example

Monotonic Counter Systems



- A run of M:

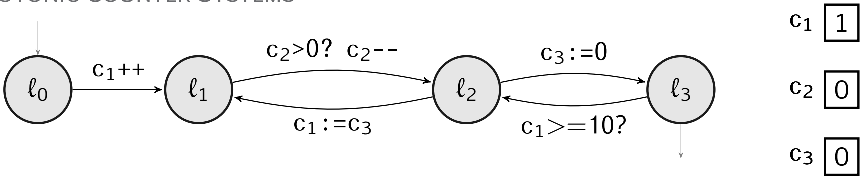
$$(\ell_0,0,1,4) \to (\ell_1,1,1,4) \to (\ell_2,1,0,4) \to (\ell_3,1,0,0)$$

- ordering configurations: $(\ell_1,0,0,0) \leqslant (\ell_1,0,1,2)$ but $(\ell_1,0,0,0) \not\leqslant (\ell_2,0,1,2)$

- a wqo as a product of wqos: $(Loc,=) \times (\mathbb{N}^3,\leqslant_\times)$

Example

Monotonic Counter Systems



- A run of M:

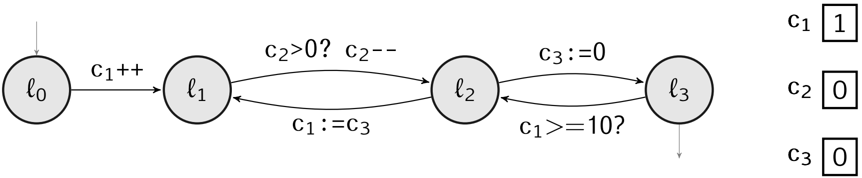
$$(\ell_0,0,1,4) \to (\ell_1,1,1,4) \to (\ell_2,1,0,4) \to (\ell_3,1,0,0)$$

- ordering configurations: $(\ell_1,0,0,0) \leqslant (\ell_1,0,1,2)$ but $(\ell_1,0,0,0) \not\leqslant (\ell_2,0,1,2)$

- a wqo as a product of wqos: $(Loc,=) \times (\mathbb{N}^3,\leqslant_\times)$

Example

Monotonic Counter Systems



- ordering configurations: $(\ell_1,0,0,0) \leqslant (\ell_1,0,1,2)$ but $(\ell_1,0,0,0) \not\leqslant (\ell_2,0,1,2)$
- a wqo as a product of wqos: $(Loc,=) \times (\mathbb{N}^3,\leqslant_\times)$
- compatibility between configurations: if $c \leqslant d$ and $c \to c'$, then $\exists d' \geqslant c'$, $d \to d'$

Example

Deciding whether a MCS Terminates

input: a MCS and an initial configuration c_0 in $Loc \times \mathbb{N}^k$

question: are all the runs starting from c_0 finite?

- termination is semi-decidable: explore all runs
- non-termination is semi-decidable: finite witness

$$
\begin{tikzcd}
c_0 \arrow[r, dotted] & c \arrow[r, dotted] \arrow[r, dashed, bend right, "\leqslant"'] & c'
\end{tikzcd}
$$

- generalizes to many classes of systems: well-structured transition systems (WSTS)

Example

Deciding whether a MCS Terminates

input	a MCS and an initial configuration c_0 in $Loc \times \mathbb{N}^k$
question	are all the runs starting from c_0 finite?

- termination is semi-decidable: explore all runs
- non-termination is semi-decidable: finite witness

$$\begin{tikzcd} c_0 \arrow[r, dotted] & c \arrow[r, dotted] \arrow[r, dashed, bend right, "\leqslant"'] & c' \end{tikzcd}$$

- generalizes to many classes of systems: well-structured transition systems (WSTS)

Example

Deciding whether a MCS Terminates

input a MCS and an initial configuration c_0 in $Loc \times \mathbb{N}^k$

question are all the runs starting from c_0 finite?

- termination is semi-decidable: explore all runs
- non-termination is semi-decidable: finite witness

$$
\begin{tikzcd}
c_0 \arrow[r, dotted] & c \arrow[r, dotted] \arrow[r, bend right, dashed, "\leqslant"'] & c'
\end{tikzcd}
$$

- generalizes to many classes of systems: well-structured transition systems (WSTS)

Example

Deciding whether a MCS Terminates

input a MCS and an initial configuration c_0 in $Loc \times \mathbb{N}^k$

question are all the runs starting from c_0 finite?

- termination is semi-decidable: explore all runs
- non-termination is semi-decidable: finite witness

$$\begin{tikzcd} c_0 \arrow[r, dotted] & c \arrow[r, dotted] \arrow[r, bend right, dashed, "\leqslant"'] & c' \end{tikzcd}$$

- generalizes to many classes of systems: well-structured transition systems (WSTS)

From Termination to Complexity

- wqos for termination: bad sequences are finite
- ... but how long can they be?

An Example

SIMPLE (a,b)
$c \longleftarrow 1$
while $a>0 \wedge b>0$
  $\langle a,b,c\rangle \longleftarrow \langle a-1,b,2c\rangle$
 or
  $\langle a,b,c\rangle \longleftarrow \langle 2c,b-1,1\rangle$
end

- in any run, $\langle a_0,b_0\rangle,\ldots,\langle a_n,b_n\rangle$ is a bad sequence over $(\mathbb{N}^2,\leqslant_\times)$,
- $(\mathbb{N}^2,\leqslant_\times)$ is a wqo: all the runs are finite
- How long can SIMPLE run?

An Example

$$
\begin{array}{l}
\textsc{simple}\,(a,b)\\
c \longleftarrow 1\\
\textbf{while } a>0 \wedge b>0\\
\qquad \langle a,b,c\rangle \longleftarrow \langle a-1,b,2c\rangle\\
\quad \textbf{or}\\
\qquad \langle a,b,c\rangle \longleftarrow \langle 2c,b-1,1\rangle\\
\textbf{end}
\end{array}
$$

- in any run, $\langle a_0,b_0\rangle,\ldots,\langle a_n,b_n\rangle$ is a bad sequence over $(\mathbb{N}^2,\leqslant_\times)$,
- $(\mathbb{N}^2,\leqslant_\times)$ is a wqo: all the runs are finite
- How long can SIMPLE run?

An Example

SIMPLE (a, b)
$c \longleftarrow 1$
while $a > 0 \wedge b > 0$
  $\langle a, b, c\rangle \longleftarrow \langle a-1, b, 2c\rangle$
 or
  $\langle a, b, c\rangle \longleftarrow \langle 2c, b-1, 1\rangle$
end

- in any run, $\langle a_0, b_0\rangle, \ldots, \langle a_n, b_n\rangle$ is a bad sequence over $(\mathbb{N}^2, \leqslant_\times)$,
- $(\mathbb{N}^2, \leqslant_\times)$ is a wqo: all the runs are finite
- How long can SIMPLE run?

An Example

$$
\begin{array}{l}
\textsc{simple}\,(a,b)\\
c \longleftarrow 1\\
\textbf{while } a>0 \wedge b>0\\
\qquad \langle a,b,c\rangle \longleftarrow \langle a-1,b,2c\rangle\\
\quad \textbf{or}\\
\qquad \langle a,b,c\rangle \longleftarrow \langle 2c,b-1,1\rangle\\
\textbf{end}
\end{array}
$$

- in any run, $\langle a_0,b_0\rangle,\ldots,\langle a_n,b_n\rangle$ is a bad sequence over $(\mathbb{N}^2,\leqslant_\times)$,
- $(\mathbb{N}^2,\leqslant_\times)$ is a wqo: all the runs are finite
- How long can SIMPLE run?

A Computation of simple$(2,3)$

SIMPLE (a,b)
$c \longleftarrow 1$
while $a > 0 \wedge b > 0$
$\quad \langle a,b,c\rangle \longleftarrow \langle a-1,b,2c\rangle$
or
$\quad \langle a,b,c\rangle \longleftarrow \langle 2c,b-1,1\rangle$
end

$\langle a,b,c\rangle$	loop iterations
$\langle 2,3,2^0\rangle$	0

A Computation of SIMPLE$(2,3)$

SIMPLE (a,b)
$c \longleftarrow 1$
while $a > 0 \wedge b > 0$
$\quad \langle a,b,c\rangle \longleftarrow \langle a-1,b,2c\rangle$
or
$\quad \langle a,b,c\rangle \longleftarrow \langle 2c,b-1,1\rangle$
end

$\langle a,b,c\rangle$	loop iterations
$\langle 2,3,2^0\rangle$	0
$\langle 1,3,2^1\rangle$	1

A Computation of simple$(2,3)$

SIMPLE (a,b)
$c \longleftarrow 1$
while $a > 0 \wedge b > 0$
$\quad \langle a,b,c\rangle \longleftarrow \langle a-1,b,2c\rangle$
or
$\quad \langle a,b,c\rangle \longleftarrow \langle 2c,b-1,1\rangle$
end

$\langle a,b,c\rangle$	loop iterations
$\langle 2,3,2^0\rangle$	0
$\langle 1,3,2^1\rangle$	1
$\langle 2^2,2,2^0\rangle$	2

A Computation of simple$(2,3)$

$\textsc{simple}\ (a,b)$
$c \longleftarrow 1$
while $a > 0 \wedge b > 0$
  $\langle a,b,c\rangle \longleftarrow \langle a-1,b,2c\rangle$
 or
  $\langle a,b,c\rangle \longleftarrow \langle 2c,b-1,1\rangle$
end

$\langle a,b,c\rangle$	loop iterations
$\vdots$	$\vdots$
$\langle 2^2,2,2^0\rangle$	2
$\vdots$	$\vdots$
$\langle 1,2,2^{2^2-1}\rangle$	$2+2^2-1$

A Computation of simple$(2,3)$

```
SIMPLE (a, b)
c ⟵ 1
while a > 0 ∧ b > 0
        ⟨a,b,c⟩ ⟵ ⟨a−1, b, 2c⟩
    or
        ⟨a,b,c⟩ ⟵ ⟨2c, b−1, 1⟩
end
```

$\langle a,b,c\rangle$	loop iterations
$\vdots$	$\vdots$
$\langle 1,2,2^{2^2-1}\rangle$	$2+2^2-1$
$\langle 2^{2^2},1,1\rangle$	$2+2^2$

A COMPUTATION OF SIMPLE$(2,3)$

SIMPLE (a, b)
$c \longleftarrow 1$
while $a > 0 \wedge b > 0$
    $\langle a, b, c \rangle \longleftarrow \langle a-1, b, 2c \rangle$
  or
    $\langle a, b, c \rangle \longleftarrow \langle 2c, b-1, 1 \rangle$
end

$\langle a, b, c \rangle$	loop iterations
$\vdots$	$\vdots$
$\langle 2^{2^2}, 1, 1 \rangle$	$2 + 2^2$
$\vdots$	$\vdots$
$\langle 1, 1, 2^{2^{2^2}-1} \rangle$	$2 + 2^2 + 2^{2^2} - 1$

A COMPUTATION OF SIMPLE$(2,3)$

SIMPLE (a,b)
$c \longleftarrow 1$
while $a > 0 \wedge b > 0$
    $\langle a,b,c\rangle \longleftarrow \langle a-1,b,2c\rangle$
  or
    $\langle a,b,c\rangle \longleftarrow \langle 2c,b-1,1\rangle$
end

$\langle a,b,c\rangle$	loop iterations
$\vdots$	$\vdots$
$\langle 1,1,2^{2^{2^2}-1}\rangle$	$2+2^2+2^{2^2}-1$
$\langle 0,1,2^{2^{2^2}}\rangle$	$2+2^2+2^{2^2}$

A Computation of simple$(2,3)$

```
SIMPLE (a, b)
c ⟵ 1
while a > 0 ∧ b > 0
    ⟨a, b, c⟩ ⟵ ⟨a − 1, b, 2c⟩
  or
    ⟨a, b, c⟩ ⟵ ⟨2c, b − 1, 1⟩
end
```

$\langle a,b,c\rangle$	loop iterations
$\vdots$	$\vdots$
$\langle 0,1,2^{2^{2^2}}\rangle$	$2+2^2+2^{2^2}$

- non-elementary complexity
- derive (matching) upper bounds for termination arguments based on $(\mathbb{N}^2,\leqslant_\times)$ being a wqo

Controlled Sequences

- bound the length of bad sequences over $(A, \leqslant)$

Controlled Sequences

- bound the length of bad sequences over $(A, \leqslant)$
- recall our game: choose any N, and consider the bad sequence $\langle 3,3\rangle, \langle 3,2\rangle, \langle 3,1\rangle, \langle 3,0\rangle, \langle 2,N\rangle, \langle 2,N-1\rangle, \ldots$

Controlled Sequences

- bound the length of bad sequences over $(A, \leqslant\, ; |.|_A)$
- associate a norm function $|.|_A : A \to \mathbb{N}$ to each wqo $(A, \leqslant)$
- assume $|.|_A$ is proper $\overset{\text{def}}{\Leftrightarrow}$ for all n

$$A_{<n} \overset{\text{def}}{=} \{x \in A \mid |x|_A < n\} \text{ is finite}$$

Example (Normed wqos)

$$|k|_{\mathbb{N}} \overset{\text{def}}{=} k \qquad |a_i|_{\Gamma_p} \overset{\text{def}}{=} 0 \qquad |\langle a, b\rangle|_{A\times B} \overset{\text{def}}{=} \max(|a|_A, |b|_B)$$

Controlled Sequences

- bound the length of controlled bad sequences over $(A, \leqslant; |.|_A)$
- fix a control function $g : \mathbb{N} \to \mathbb{N}$ (strictly increasing)
- $\mathbf{x} = x_0, x_1, \ldots$ over A is (g, n)-controlled iff

$$\forall i,\ |x_i|_A < g^i(n)$$

Controlled Sequences

- bound the length of controlled bad sequences over $(A, \leqslant; |.|_A)$
- fix a control function $g : \mathbb{N} \to \mathbb{N}$
- $\mathbf{x} = x_0, x_1, \ldots$ over A is (g, n)-controlled iff

$$\forall i, |x_i|_A < g^i(n)$$

Example (SIMPLE(2,3))

$A = \mathbb{N}^2, n = 4, g(x) = 2x$

Controlled Sequences

- bound the length of controlled bad sequences over $(A, \leqslant; |.|_A)$
- for fixed A, g, n, there are finitely many bad (g, n)-controlled sequences over A
- maximal length function

$$L_{A,g}(n)$$

Length Function Theorems

Bound $L_{A,g}$ by some functions for various A.

Example

- for $(\Gamma_p, =; |.|_{\Gamma_p})$:
- for $(\mathbb{N}, \leqslant; |.|_{\mathbb{N}})$:
- recall SIMPLE: not every length function is that 'small'…

Length Function Theorems

Bound $L_{A,g}$ by some functions for various A.

Example

- for $(\Gamma_p, =; |.|_{\Gamma_p})$:
- for $(\mathbb{N}, \leqslant; |.|_{\mathbb{N}})$:
- recall SIMPLE: not every length function is that 'small'...

Length Function Theorems

Bound $L_{A,g}$ by some functions for various A.

Example

- for $(\Gamma_p, =; |.|_{\Gamma_p})$: Pigeonhole principle: $L_{\Gamma_p,g}(n) = p$
- for $(\mathbb{N}, \leqslant; |.|_{\mathbb{N}})$:
- recall SIMPLE: not every length function is that 'small'…

Length Function Theorems

Bound $L_{A,g}$ by some functions for various A.

Example

- for $(\Gamma_p, =; |.|_{\Gamma_p})$: Pigeonhole principle: $L_{\Gamma_p,g}(n) = p$
- for $(\mathbb{N}, \leqslant; |.|_{\mathbb{N}})$:
- recall SIMPLE: not every length function is that 'small'...

Length Function Theorems

Bound $L_{A,g}$ by some functions for various A.

Example

- for $(\Gamma_p, =; |.|_{\Gamma_p})$: $L_{\Gamma_p,g}(n) = p$
- for $(\mathbb{N}, \leqslant; |.|_{\mathbb{N}})$: $L_{\mathbb{N},g}(n) = n$:

$$n-1, n-2, \ldots, 1, 0$$

- recall SIMPLE: not every length function is that 'small'…

Length Function Theorems

Bound $L_{A,g}$ by some functions for various A.

Example

- for $(\Gamma_p, =; |.|_{\Gamma_p})$: $L_{\Gamma_p,g}(n) = p$
- for $(\mathbb{N}, \leqslant; |.|_{\mathbb{N}})$: $L_{\mathbb{N},g}(n) = n$:

$$n-1, n-2, \ldots, 1, 0$$

- recall SIMPLE: not every length function is that 'small'...

Length Function Theorems

Bound $L_{A,g}$ by some functions for various A.

Example

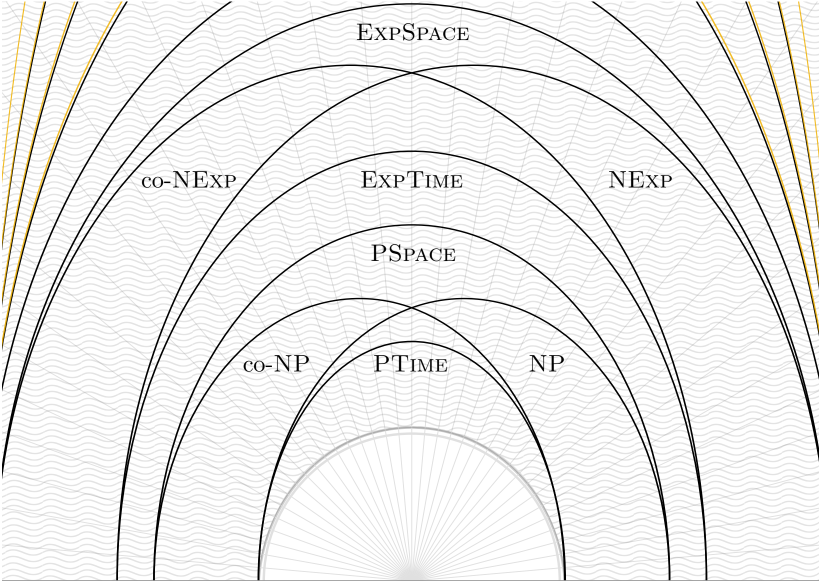
- for $(\mathbb{N}^k, \leqslant; |.|_{\mathbb{N}^k})$: let g be in $\mathfrak{F}\mathbf{F}_\gamma$ for $\gamma \geqslant 1$, then $L_{\mathbb{N}^k,g}(n)$ is bounded by a function in $\mathfrak{F}\mathbf{F}_{\gamma+k}$
- for $(\Gamma_p^*, \leqslant_*; |.|_{\Gamma_p^*})$: let g be primitive-recursive and $p \geqslant 2$, then $L_{\Gamma_p^*,g}(n)$ is bounded by a function in $\mathfrak{F}\mathbf{F}_{\omega^{p-1}}$
- more on $\mathfrak{F}\mathbf{F}_\alpha$ soon...

Using the Length Function

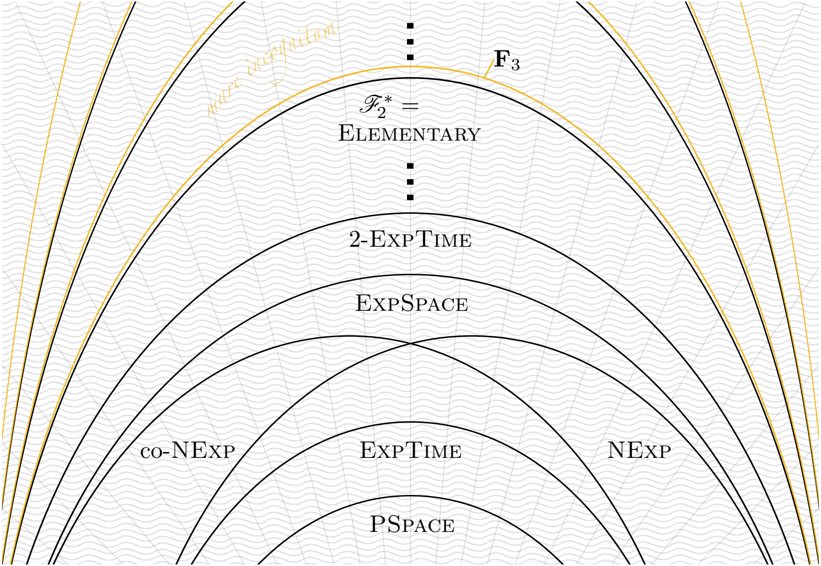
Example (MCS Termination)

- control function $g(n) = n + 1$ in $\mathfrak{F}\mathbf{F}_1$
- any run of length $L_{Loc \times \mathbb{N}^k, g}(n_0) + 1$ is good
- new algorithm for termination: try to find such a run
- complexity in $\mathbf{F}_{k+2}$ when k is fixed, $\mathbf{F}_\omega$ when not

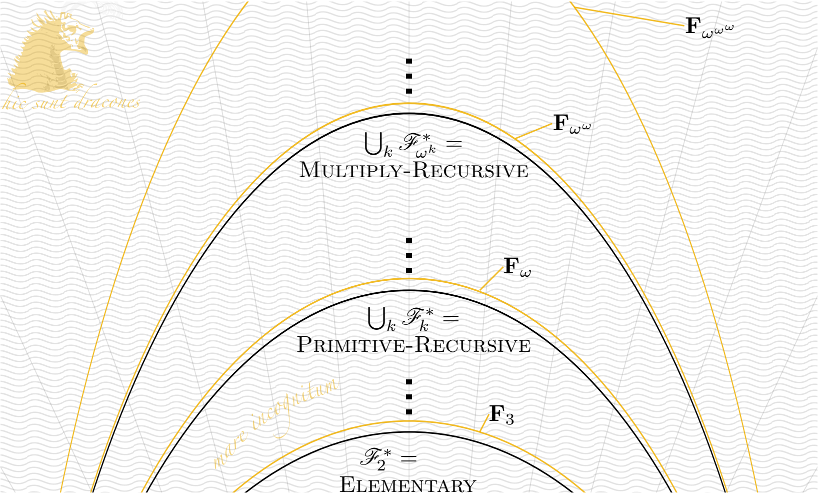
Beyond Elementary



Beyond Elementary



Beyond Elementary



Fast-Growing Functions

(Löb and Wainer, 1970)

$$F_0(x) \stackrel{\text{def}}{=} x+1, \quad F_{\alpha+1}(x) \stackrel{\text{def}}{=} F_\alpha^x(x), \quad F_\lambda \stackrel{\text{def}}{=} F_{\lambda_x}(x).$$

where λ_x is the xth element of a fundamental sequence $(\lambda_x)_x$ for the limit ordinal λ

Example

$$F_1(x) = 2x$$

$$F_2(x) = x \cdot 2^x$$

F_3 is non elementary

F_ω is non primitive-recursive

F_{ω^ω} is non multiply-recursive

Fast-Growing Complexities

$$\alpha \geqslant 2: \qquad \mathscr{F}_\alpha \stackrel{\text{def}}{=} \bigcup_{c<\omega} \mathrm{FSpace}\big(F_\alpha^c(n)\big)$$

$$\alpha \geqslant 3: \qquad \mathbf{F}_\alpha \stackrel{\text{def}}{=} \bigcup_{p \in \bigcup_{\beta<\alpha} \mathscr{F}_\beta} \mathrm{Space}\big(F_\alpha(p(n))\big)$$

Example

$\mathbf{F}_3$ complexity bounded by a tower of exponentials of elementary height,

$\mathbf{F}_\omega$ Ackermannian complexity of some primitive-recursive function,

Some $\mathbf{F}_\omega$-Complete Problems

Decision of problems on

- monotonic counter systems (Finkel and Schnoebelen, 2001), e.g.
  - finite VASS containment (Mayr and Meyer, 1981; Jančar, 2001)
  - lossy counter systems termination (Schnoebelen, 2010),
- relevance logics (Urquhart, 1999),
- data logics (Demri and Lazić, 2009; Figueira and Segoufin, 2009), ...

Some $\mathbf{F}_{\omega^\omega}$-Complete Problems

Decision of problems on

- lossy channel systems (Chambart and Schnoebelen, 2008),
- Post embedding problem RatEP (Chambart and Schnoebelen, 2007),
- 1-clock alternating timed automata (Lasota and Walukiewicz, 2008),
- Metric temporal logic (Ouaknine and Worrell, 2007),
- finite concurrent programs under weak (TSO/PSO) memory models (Atig et al., 2010)
- alternating register automata over ordered domains (Figueira et al., 2010),...

Summary

- wqos for termination of algorithms
- length function theorems: out-of-the-box complexity upper bounds
- matching lower bounds for many problems

Course material at

http://www.lsv.ens-cachan.fr/~schmitz/teach/2012_esslli/

Perspectives

Some big challenges ahead, for instance:

- complexity of VASS reachability
- decidability of BVASS reachability

Also easier problems: internships within ANR ReacHard:

http://www.lsv.ens-cachan.fr/projects/anr-reachard

References

Upper Bounds for Dickson's Lemma

McAloon, K., 1984. Petri nets and large finite sets. *Theor. Comput. Sci.*, 32(1–2): 173–183. doi:10.1016/0304-3975(84)90029-X.

Clote, P., 1986. On the finite containment problem for Petri nets. *Theor. Comput. Sci.*, 43:99–105. doi:10.1016/0304-3975(86)90169-6.

Figueira, D., Figueira, S., Schmitz, S., and Schnoebelen, Ph., 2011. Ackermannian and primitive-recursive bounds with Dickson's Lemma. In *LICS 2011*. IEEE Press. arXiv:1007.2989[cs.LO].

Upper Bounds for Higman's Lemma

Cichoń, E.A. and Tahhan Bittar, E., 1998. Ordinal recursive bounds for Higman's Theorem. *Theor. Comput. Sci.*, 201(1–2):63–84. doi:10.1016/S0304-3975(97)00009-1.

Schmitz, S. and Schnoebelen, Ph., 2011. Multiply recursive upper bounds with Higman's Lemma. In *ICALP 2011* volume 6756 of *LNCS*, pages 441–452. Springer. arXiv:1103/4399[cs.LO].

Upper Bounds for Kruskal's Theorem

Weiermann, A., 1994. Complexity bounds for some finite forms of Kruskal's Theorem. *J. Symb. Comput.*, 18(5):463–488. doi:10.1006/jsco.1994.1059.

Fast Growing Hierarchy

Löb, M. and Wainer, S., 1970. Hierarchies of number theoretic functions, I. *Arch. Math. Logic*, 13:39–51. doi:10.1007/BF01967649.

WSTS

Abdulla, P.A., Čerāns, K., Jonsson, B., and Tsay, Y.K., 2000. Algorithmic analysis of programs with well quasi-ordered domains. *Inform. and Comput.*, 160(1–2): 109–127. doi:10.1006/inco.1999.2843.

Finkel, A. and Schnoebelen, Ph., 2001. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 256(1–2):63–92. doi:10.1016/S0304-3975(00)00102-X.

Lower Bounds

Schnoebelen, Ph., 2010. Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. In *MFCS 2010*, volume 6281 of *LNCS*, pages 616–628. Springer. doi:10.1007/978-3-642-15155-2_54.

Chambart, P. and Schnoebelen, Ph., 2008. The ordinal recursive complexity of lossy channel systems. In *LICS 2008*, pages 205–216. IEEE. doi:10.1109/LICS.2008.47.

Haddad, S., Schmitz, S., and Schnoebelen, Ph., 2012. The ordinal-recursive complexity of timed-arc Petri nets, data nets, and other enriched nets. In *LICS 2012*, pages 355–364. IEEE. doi:10.1109/LICS.2012.46

Applications of Higman's Lemma

Chambart, P. and Schnoebelen, Ph., 2007. Post embedding problem is not primitive recursive, with applications to channel systems. In Arvind, V. and Prasad, S., editors, *FSTTCS 2007*, volume 4855 of *LNCS*, pages 265–276. Springer. doi:10.1007/978-3-540-77050-3_22.

Ouaknine, J.O. and Worrell, J.B., 2007. On the decidability and complexity of Metric Temporal Logic over finite words. *Logic. Meth. in Comput. Sci.*, 3(1):8. doi:10.2168/LMCS-3(1:8)2007.

Lasota, S. and Walukiewicz, I., 2008. Alternating timed automata. *ACM Trans. Comput. Logic*, 9(2):10. doi:10.1145/1342991.1342994.

Atig, M.F., Bouajjani, A., Burckhardt, S., and Musuvathi, M., 2010. On the verification problem for weak memory models. In *POPL 2010*, pages 7–18. ACM Press. doi:10.1145/1706299.1706303.

Figueira, D., Hofman, P., and Lasota, S., 2010. Relating timed and register automata. In Fröschle, S. and Valencia, F., editors, *EXPRESS 2010*, volume 41 of *EPTCS*, pages 61–75. doi:10.4204/EPTCS.41.5.

Applications of Dickson's Lemma

Demri, S., 2006. Linear-time temporal logics with Presburger constraints: An overview. *J. Appl. Non-Classical Log.*, 16(3–4):311–347. doi:10.3166/jancl.16.311-347.

Demri, S. and Lazić, R., 2009. LTL with the freeze quantifier and register automata. *ACM Trans. Comput. Logic*, 10(3). doi:10.1145/1507244.1507246.

Figueira, D. and Segoufin, L., 2009. Future-looking logics on data words and trees. In *MFCS 2009*, volume 5734 of *LNCS*, pages 331–343. Springer. doi:10.1007/978-3-642-03816-7_29.

Gallo, G. and Mishra, B., 1994. A solution to Kronecker's Problem. *Appl. Algebr. Eng. Comm.*, 5(6):343–370.

Jančar, P., 2011. Nonprimitive recursive complexity and undecidability for Petri net equivalences. *Theor. Comput. Sci.*, 256(1–2):23–30. 10.1016/S0304-3975(00)00100-6.

Mayr, E.W. and Meyer, A.R., 1981. The complexity of the finite containment problem for Petri nets. *J. ACM*, 28(3):561–576. 10.1145/322261.322271.

Podelski, A. and Rybalchenko, A., 2004. Transition invariants. In *LICS 2004*, pages 32–41. IEEE. doi:10.1109/LICS.2004.1319598.

Revesz, P.Z., 1993. A closed-form evaluation for Datalog queries with integer (gap)-order constraints. *Theor. Comput. Sci.*, 116(1):117–149. doi:10.1016/0304-3975(93)90222-F.

Urquhart, A., 1999. The complexity of decision procedures in relevance logic II. *J. Symb. Log.*, 64(4):1774–1802. doi:10.2307/2586811.