Complexity Bounds for Ordinal-Based Termination

Sylvain Schmitz

with numerous colleagues: Sergio Abriola, Diego Figueira, Santiago Figueira, Jérôme Leroux, Philippe Schnoebelen

LSV, ENS Cachan & INRIA, France

RP 2014, September 22nd 2014, Oxford

Outline

termination proofs using ranking functions into ordinals

complexity bounds on termination times

- exact bounds
- "simple" case compared to well-quasi-orders

application: VAS Reachability

Termination Proofs

multiset path orderings for term rewriting systems

lexicographic path orderings for term rewriting systems

size-change abstractions for first-order programs

disjunctive termination arguments (aka Ramsey-based)

ranking functions

Complexity from Termination Proofs

multiset path orderings for term rewriting systems
Hofbauer (1992): primitive-recursive

lexicographic path orderings for term rewriting systems
Weiermann (1995): multiply-recursive

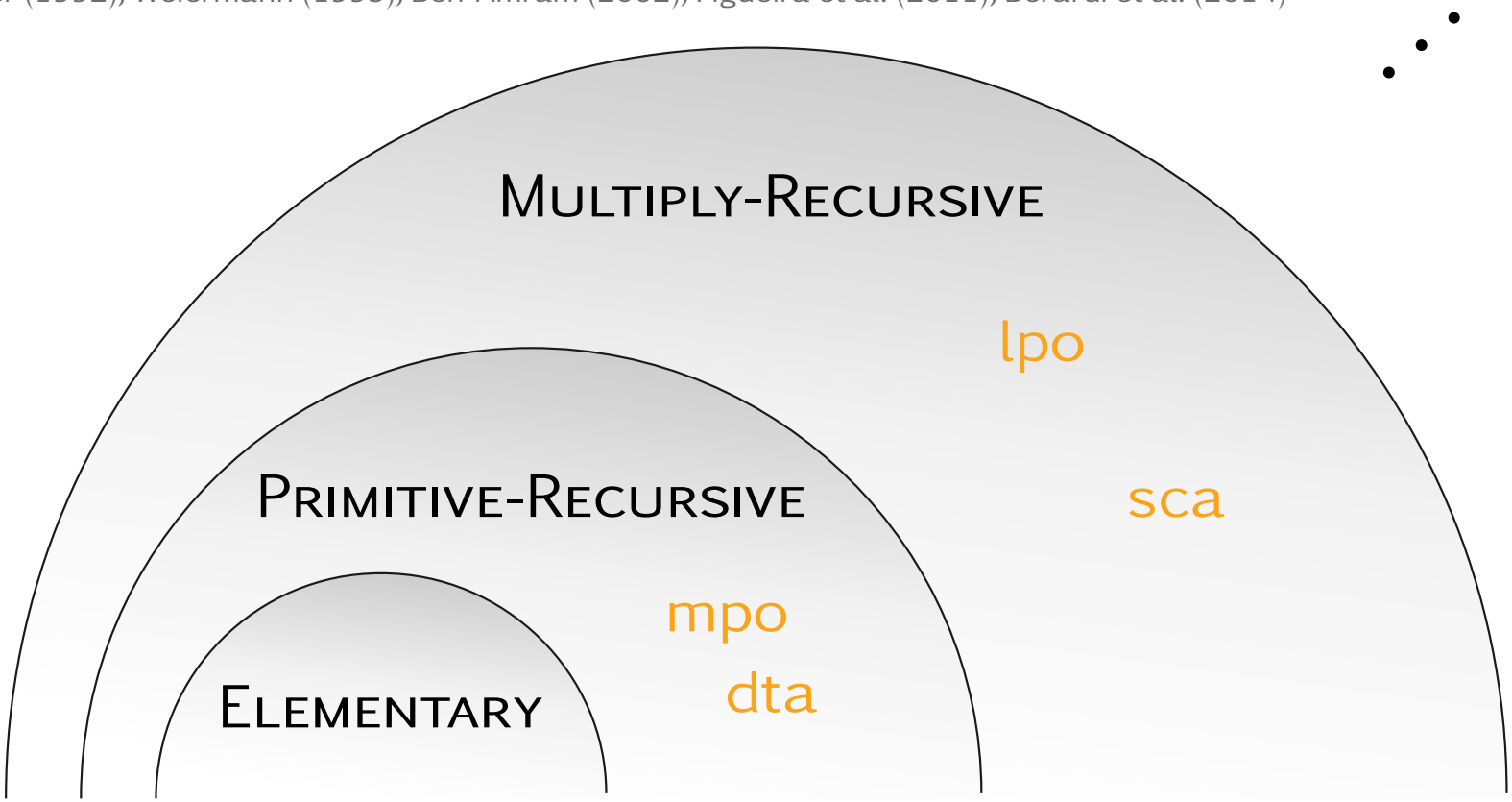
size-change abstractions for first-order programs
Ben-Amram (2002): multiply-recursive

disjunctive termination arguments (aka Ramsey-based)
Figueira et al. (2011): primitive-recursive
also Berardi, Oliva, and Steila (2014) tomorrow morning

ranking functions this talk

Complexity Classification

Hofbauer (1992); Weiermann (1995); Ben-Amram (2002); Figueira et al. (2011); Berardi et al. (2014)



Ranking Functions

Turing (1949)

"Finally the checker has to verify that the process comes to an end. Here again he should be assisted by the programmer giving a further definite assertion to be verified. This may take the form of a quantity which is asserted to decrease continually and vanish when the machine stops."

Source: Beryl Turing and King's College Library, Cambridge

Ranking Functions

operational semantics of programs using transition systems $\mathcal{S} = \langle Conf, \rightarrow_{\mathcal{S}} \rangle$

well-order (wo) $\langle A, \leqslant \rangle$, aka well-founded linear order: no infinite decreasing sequence $x_0 > x_1 > \cdots$ over A , e.g.

- $\langle \mathbb{N}, \leqslant \rangle$
- $\langle A \times B, \leqslant_{\text{lex}} \rangle$ for wo's $\langle A, \leqslant_A \rangle$ and $\langle B, \leqslant_B \rangle$, where $(n, m) \leqslant_{\text{lex}} (n', m')$ iff $n <_A n' \vee (n = n' \wedge m \leqslant_B m')$

Ranking Functions

operational semantics of programs using transition systems $\mathcal{S} = \langle Conf, \to_{\mathcal{S}} \rangle$

well-order (wo) $\langle A, \leqslant \rangle$, aka well-founded linear order: no infinite decreasing sequence $x_0 > x_1 > \cdots$ over A

ranking function $f : Conf \to A$: $c \to_{\mathcal{S}} c'$ implies $f(c) > f(c')$

Example

Program:

```
ℓ0: while x >= 0 and y > 0 do
  if x > 0 then
    a: x := x−1; n := 2n;
  else
    b: x := n; y := y−1; n := 2n;
done
```

Example

Program:

```
ℓ0: while x >= 0 and y > 0 do
      if x > 0 then
        a: x := x−1; n := 2n;
      else
        b: x := n; y := y−1; n := 2n;
done
```

Transition system:

a:

assume(x>0);
assume(y>0);
x := x−1;
n := 2n;

b:

assume(x=0);
assume(y>0);
x := n;
y := y−1;
n := 2n;

Example

Program:

```
ℓ0: while x >= 0 and y > 0 do
      if x > 0 then
        a: x := x−1; n := 2n;
      else
        b: x := n; y := y−1; n := 2n;
    done
```

Ranking function:

$$f{:}\{\ell_0\} \times \mathbb{Z}^3 \to \langle \mathbb{N}^2, \leqslant_{\text{lex}} \rangle$$
$$f(\ell_0, x, y, n) = (y, x)$$

Transition system:

a:
assume(x>0);
assume(y>0);
x := x−1;
n := 2n;

b:
assume(x=0);
assume(y>0);
x := n;
y := y−1;
n := 2n;

Example

Program:

```
ℓ0: while x >= 0 and y > 0 do
      if x > 0 then
        a: x := x−1; n := 2n;
      else
        b: x := n; y := y−1; n := 2n;
    done
```

Transition system:

a:
assume(x>0);
assume(y>0);
x := x−1;
n := 2n;

b:
assume(x=0);
assume(y>0);
x := n;
y := y−1;
n := 2n;

Ranking function:

$$f:\{\ell_0\}\times\mathbb{Z}^3\to\langle\mathbb{N}^2,\leqslant_{\text{lex}}\rangle$$
$$f(\ell_0,x,y,n)=(y,x)$$

Termination time:

steps	x	y	n
	3	3	3
aaa	0	3	$2^3\cdot 3$
b	$2^3\cdot 3$	2	$2^4\cdot 3=48$
$a^{2^3\cdot 3}$	0	2	$2^{2^3\cdot 3}\cdot 48$
b	$2^{2^3\cdot 3}\cdot 48$	1	$2^{2^3\cdot 3}\cdot 96$
$a^{2^{2^3\cdot 3}\cdot 48}$	0	1	…
b	…	0	…

Example

Program:

```
ℓ0: while x >= 0 and y > 0 do
      if x > 0 then
        a: x := x−1; n := 2n;
      else
        b: x := n; y := y−1; n := 2n;
    done
```

Transition system:

a:
assume(x>0);
assume(y>0);
x := x−1;
n := 2n;

b:
assume(x=0);
assume(y>0);
x := n;
y := y−1;
n := 2n;

Ranking function:

$$f:\{\ell_0\}\times\mathbb{Z}^3\to\langle\mathbb{N}^2,\leqslant_{\text{lex}}\rangle$$
$$f(\ell_0,x,y,n)=(y,x)$$

Termination time:

steps	x	y	n
	3	3	3
aaa	0	3	$2^3\cdot 3$
b	$2^3\cdot 3$	2	$2^4\cdot 3=48$
$a^{2^3\cdot 3}$	0	2	$2^{2^3\cdot 3}\cdot 48$
b	$2^{2^3\cdot 3}\cdot 48$	1	$2^{2^3\cdot 3}\cdot 96$
$a^{2^{2^3\cdot 3}\cdot 48}$	0	1	…
b	…	0	…

Example

Program:

```
ℓ0: while x >= 0 and y > 0 do
      if x > 0 then
        a: x := x−1; n := 2n;
      else
        b: x := n; y := y−1; n := 2n;
    done
```

Transition system:

a:
assume(x>0);
assume(y>0);
x := x−1;
n := 2n;

b:
assume(x=0);
assume(y>0);
x := n;
y := y−1;
n := 2n;

Ranking function:

$$f:\{\ell_0\}\times\mathbb{Z}^3\to\langle\mathbb{N}^2,\leqslant_{\text{lex}}\rangle$$
$$f(\ell_0,x,y,n)=(y,x)$$

Termination time:

steps	x	y	n
	3	3	3
aaa	0	3	$2^3\cdot 3$
b	$2^3\cdot 3$	2	$2^4\cdot 3=48$
$a^{2^3\cdot 3}$	0	2	$2^{2^3\cdot 3}\cdot 48$
b	$2^{2^3\cdot 3}\cdot 48$	1	$2^{2^3\cdot 3}\cdot 96$
$a^{2^{2^3\cdot 3}\cdot 48}$	0	1	…
b	…	0	…

Example

Program:

```
ℓ0: while x >= 0 and y > 0 do
      if x > 0 then
        a: x := x−1; n := 2n;
      else
        b: x := n; y := y−1; n := 2n;
done
```

Transition system:

a:
assume(x>0);
assume(y>0);
x := x−1;
n := 2n;

b:
assume(x=0);
assume(y>0);
x := n;
y := y−1;
n := 2n;

Ranking function:

$$f:\{\ell_0\}\times\mathbb{Z}^3\to\langle\mathbb{N}^2,\leqslant_{\text{lex}}\rangle$$
$$f(\ell_0,x,y,n)=(y,x)$$

Termination time:

steps	x	y	n
	3	3	3
aaa	0	3	$2^3\cdot 3$
b	$2^3\cdot 3$	2	$2^4\cdot 3=48$
$a^{2^3\cdot 3}$	0	2	$2^{2^3\cdot 3}\cdot 48$
b	$2^{2^3\cdot 3}\cdot 48$	1	$2^{2^3\cdot 3}\cdot 96$
$a^{2^{2^3\cdot 3}\cdot 48}$	0	1	…
b	…	0	…

Example

Program:

```
ℓ0: while x >= 0 and y > 0 do
      if x > 0 then
        a: x := x−1; n := 2n;
      else
        b: x := n; y := y−1; n := 2n;
done
```

Transition system:

a:
assume(x>0);
assume(y>0);
x := x−1;
n := 2n;

b:
assume(x=0);
assume(y>0);
x := n;
y := y−1;
n := 2n;

Ranking function:

$$f:\{\ell_0\}\times\mathbb{Z}^3\to\langle\mathbb{N}^2,\leqslant_{\text{lex}}\rangle$$
$$f(\ell_0,x,y,n)=(y,x)$$

Termination time:

steps	x	y	n
	3	3	3
aaa	0	3	$2^3\cdot 3$
b	$2^3\cdot 3$	2	$2^4\cdot 3=48$
$a^{2^3\cdot 3}$	0	2	$2^{2^3\cdot 3}\cdot 48$
b	$2^{2^3\cdot 3}\cdot 48$	1	$2^{2^3\cdot 3}\cdot 96$
$a^{2^{2^3\cdot 3}\cdot 48}$	0	1	…
b	…	0	…

Example

Program:

```
ℓ0: while x >= 0 and y > 0 do
    if x > 0 then
      a: x := x−1; n := 2n;
    else
      b: x := n; y := y−1; n := 2n;
done
```

Transition system:

a:
assume(x>0);
assume(y>0);
x := x−1;
n := 2n;

b:
assume(x=0);
assume(y>0);
x := n;
y := y−1;
n := 2n;

Ranking function:

$$f:\{\ell_0\}\times\mathbb{Z}^3\to\langle\mathbb{N}^2,\leqslant_{\text{lex}}\rangle$$
$$f(\ell_0,x,y,n)=(y,x)$$

Termination time:

non-elementary

steps	x	y	n
	3	3	3
aaa	0	3	$2^3\cdot 3$
b	$2^3\cdot 3$	2	$2^4\cdot 3=48$
$a^{2^3\cdot 3}$	0	2	$2^{2^3\cdot 3}\cdot 48$
b	$2^{2^3\cdot 3}\cdot 48$	1	$2^{2^3\cdot 3}\cdot 96$
$a^{2^{2^3\cdot 3}\cdot 48}$	0	1	…
b	…	0	…

Ranking Functions

Turing (1949)

"Finally the checker has to verify that the process comes to an end. Here again he should be assisted by the programmer giving a further definite assertion to be verified. This may take the form of a quantity which is asserted to decrease continually and vanish when the machine stops. To the pure mathematician it is natural to give an ordinal number."

Source: Beryl Turing and King's College Library, Cambridge

Ordinals

- isomorphism classes of wo's

$$o(\langle \mathbb{N}, \leqslant \rangle) = \omega$$

$$o(\langle A \times B, \leqslant_{\text{lex}} \rangle) = o(\langle A, \leqslant_A \rangle) \cdot o(\langle B, \leqslant_B \rangle)$$

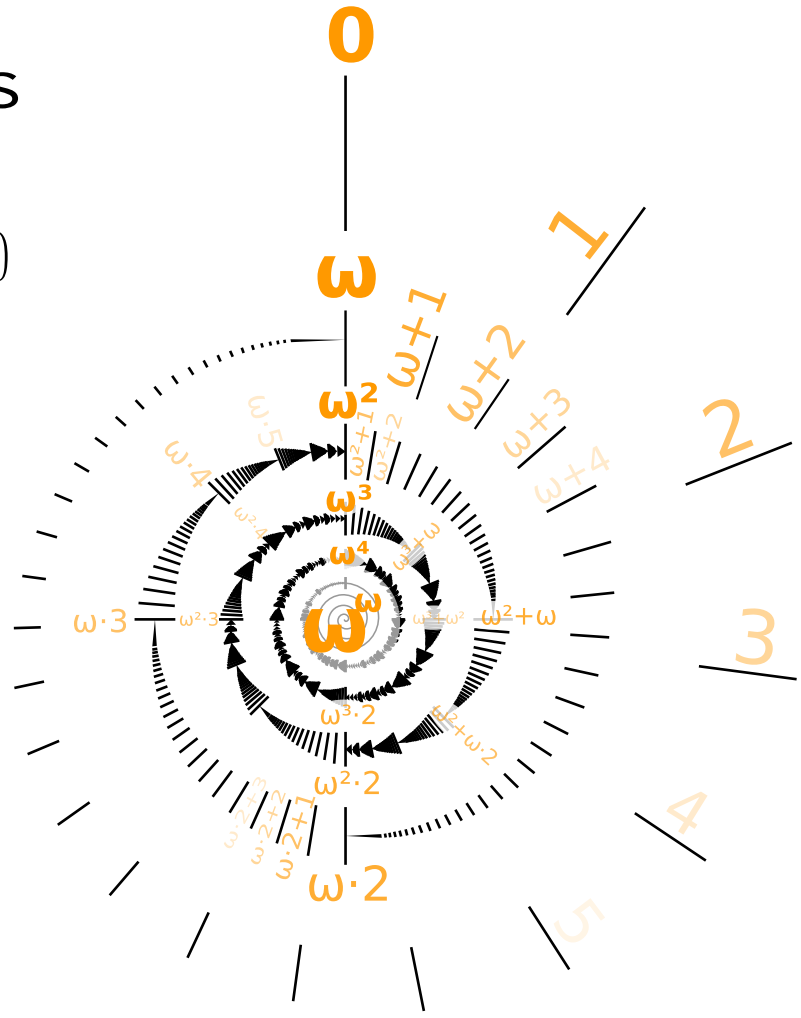
- set-theoretic construction

$$\alpha = \{\beta \mid \beta < \alpha\}$$

- Cantor normal form (CNF) below ε_0

$$\alpha = \omega^{\alpha_1} \cdot c_1 + \cdots + \omega^{\alpha_m} \cdot c_m$$

with $\alpha > \alpha_1 > \cdots > \alpha_m$ and $0 < c_1, \ldots, c_m < \omega$



Public domain image (source: Wikipedia)

Example

Program:

```
ℓ0: while x >= 0 and y > 0 do
  if x > 0 then
    a: x := x−1; n := 2n;
  else
    b: x := n; y := y−1; n := 2n;
done
```

Ranking function:

$$f{:}\{\ell_0\} \times \mathbb{Z}^3 \to \omega^2$$

$$f(\ell_0, x, y, n) = \omega \cdot y + x$$

Complexity Bounds

- the existence of a ranking function $f : Conf \to \alpha$ proves termination
- two main ingredients:
  - the ordinal α used as range
  - the function f itself
- goal: bound the length of program executions
- idea: bound the length of decreasing sequences in α

The Length of Decreasing Sequences

- no upper bound in general
  e.g. in $\langle \mathbb{N}, \leqslant \rangle$:

$$N > N-1 > N-2 > \cdots > 1 > 0$$

- even with a bound on the first element
  e.g. in $\langle \mathbb{N}^2, \leqslant_{\text{lex}} \rangle$:

$$(1,1) > (1,0) > (0,N) > \cdots > (0,1) > (0,0)$$

The Length of Decreasing Sequences

- no upper bound in general
  e.g. in $\langle \mathbb{N}, \leqslant \rangle$:

$$N > N-1 > N-2 > \cdots > 1 > 0$$

- even with a bound on the first element
  e.g. in $\langle \mathbb{N}^2, \leqslant_{\text{lex}} \rangle$:

$$(1,1) > (1,0) > (0,N) > \cdots > (0,1) > (0,0)$$

The Length of Decreasing Sequences

- no upper bound in general
  e.g. in $\langle \mathbb{N}, \leqslant \rangle$:

$$N > N-1 > N-2 > \cdots > 1 > 0$$

- even with a bound on the first element
  e.g. in $\langle \mathbb{N}^2, \leqslant_{\text{lex}} \rangle$:

$$\begin{array}{ccccccccc} (1,1) & > & (1,0) & > & (0,N) & > & \cdots > & (0,1) & > & (0,0) \\ \omega+1 & > & \omega & > & N & > & \cdots > & 1 & > & 0 \end{array}$$

Ordinal Norm

Buchholz et al. (1994)

$$N(\omega^{\alpha_1}\cdot c_1+\cdots+\omega^{\alpha_m}\cdot c_m) = \max_{1\leqslant i\leqslant m}(\max(N(\alpha_i),c_i))$$

In the case of ω^d, corresponds to infinite norm over $\mathbb{N}^d$:

$$\|(c_1,\ldots,c_d)\|_\infty = \max_{1\leqslant i\leqslant d} c_i$$

Ordinal Norm

Buchholz et al. (1994)

$$N(\omega^{\alpha_1}\cdot c_1+\cdots+\omega^{\alpha_m}\cdot c_m) = \max_{1\leqslant i\leqslant m}(\max(N(\alpha_i),c_i))$$

In the case of ω^d, corresponds to infinite norm over $\mathbb{N}^d$:

$$\|(c_1,\ldots,c_d)\|_\infty = \max_{1\leqslant i\leqslant d} c_i$$

Controlled Sequences

- $g : \mathbb{N} \to \mathbb{N}$ monotone a control function
- $n_0 \in \mathbb{N}$ an initial norm and $\iota : \mathit{Conf} \to \mathbb{N}$ a norm function on configurations
- a sequence $\beta_0, \beta_1, \ldots$ over α is (g, n_0)-controlled if

$$\forall i. N(\beta_i) \leqslant g^i(n_0)$$

- $f : \mathit{Conf} \to \alpha$ is (ι, g)-controlled if, for every execution $c_0 \to_S c_1 \to_S \cdots$ in $\langle \mathit{Conf}, \to_S \rangle$, the sequence $f(c_0), f(c_1), \ldots$ is $(\iota(c_0), g)$-controlled, i.e. if

$$\forall i. N(f(c_i)) \leqslant g^i(\iota(c_0))$$

Controlled Sequences

- $g : \mathbb{N} \to \mathbb{N}$ monotone a control function
- $n_0 \in \mathbb{N}$ an initial norm and $\iota : \mathit{Conf} \to \mathbb{N}$ a norm function on configurations
- a sequence $\beta_0, \beta_1, \ldots$ over α is (g, n_0)-controlled if

$$\forall i. N(\beta_i) \leqslant g^i(n_0)$$

- $f : \mathit{Conf} \to \alpha$ is (ι, g)-controlled if, for every execution $c_0 \to_S c_1 \to_S \cdots$ in $\langle \mathit{Conf}, \to_S \rangle$, the sequence $f(c_0), f(c_1), \ldots$ is $(\iota(c_0), g)$-controlled, i.e. if

$$\forall i. N(f(c_i)) \leqslant g^i(\iota(c_0))$$

Controlled Sequences

- $g : \mathbb{N} \to \mathbb{N}$ monotone a control function
- $n_0 \in \mathbb{N}$ an initial norm and $\iota : \mathit{Conf} \to \mathbb{N}$ a norm function on configurations
- a sequence $\beta_0, \beta_1, \ldots$ over α is (g, n_0)-controlled if

$$\forall i. N(\beta_i) \leqslant g^i(n_0)$$

- $f : \mathit{Conf} \to \alpha$ is (ι, g)-controlled if, for every execution $c_0 \to_S c_1 \to_S \cdots$ in $\langle \mathit{Conf}, \to_S \rangle$, the sequence $f(c_0), f(c_1), \ldots$ is $(\iota(c_0), g)$-controlled, i.e. if

$$\forall i. N(f(c_i)) \leqslant g^i(\iota(c_0))$$

Example

Program:

```
ℓ0: while x >= 0 and y > 0 do
  if x > 0 then
    a: x := x−1; n := 2n;
  else
    b: x := n; y := y−1; n := 2n;
done
```

Ranking function:

$$f{:}\{\ell_0\} \times \mathbb{Z}^3 \to \omega^2$$

$$f(\ell_0, x, y, n) = \omega \cdot y + x$$

Control:

$$\iota(\ell_0, x, y, n) = \max(x, y, n)$$

$$g(x) = 2x$$

Length Functions

- (g, n_0)-controlled decreasing sequences over α are bounded; we note their maximal length $L_{\alpha,g}(n_0)$
- Descent Equation

$$L_{\alpha,g}(n) = \max_{\beta<\alpha, N(\beta)\leqslant n} 1 + L_{\beta,g}\big(g(n)\big)$$

Proof idea.
Let $\beta_0 > \beta_1 > \beta_2 > \cdots$ be (g,n)-controlled over α. Then $\beta_1 > \beta_2 > \cdots$ is $(g, g(n))$-controlled over β_0. The longest sequence is thus obtained by maximising $\beta_0 < \alpha$ among those with $N(\beta_0) \leqslant n$. □

Length Functions

- (g, n_0)-controlled decreasing sequences over α are bounded; we note their maximal length $L_{\alpha,g}(n_0)$
- Descent Equation

$$L_{\alpha,g}(n) = \max_{\beta<\alpha, N(\beta)\leqslant n} 1 + L_{\beta,g}\big(g(n)\big)$$

Proof idea.

Let $\beta_0 > \beta_1 > \beta_2 > \cdots$ be (g,n)-controlled over α. Then $\beta_1 > \beta_2 > \cdots$ is $(g, g(n))$-controlled over β_0. The longest sequence is thus obtained by maximising $\beta_0 < \alpha$ among those with $N(\beta_0) \leqslant n$. □

Subrecursive Functions

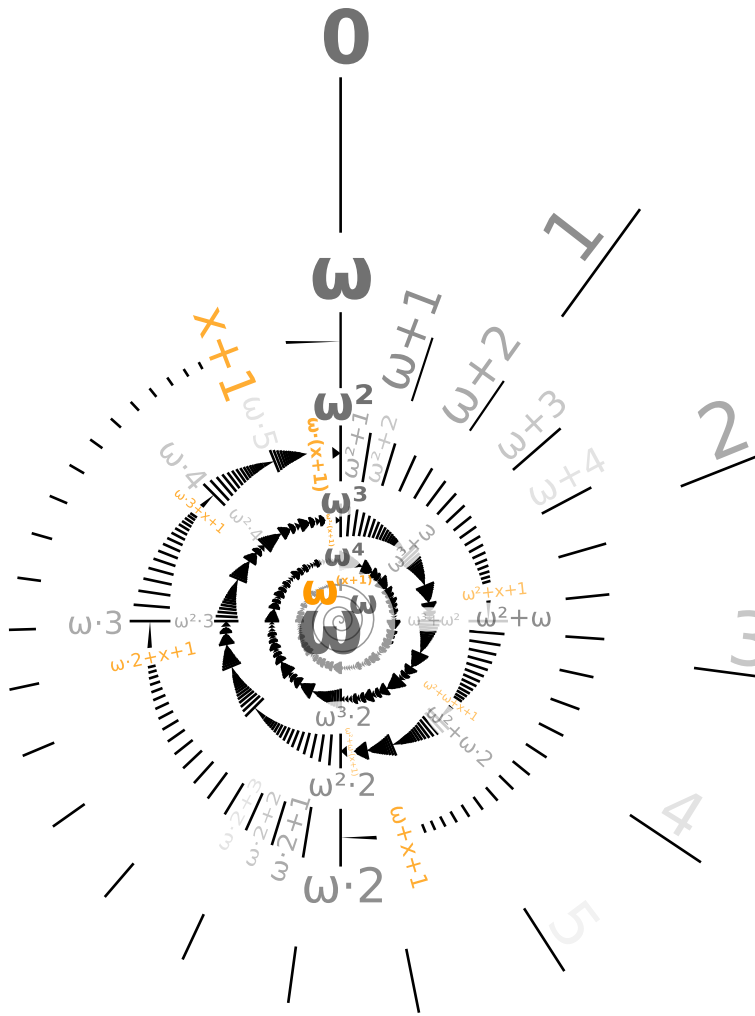
- goal: obtain explicit bounds for $L_{\alpha,g}$
- goal: classification in complexity hierarchies
- use subrecursive functions indexed with ordinal numbers

Fundamental Sequences

- for limit ordinals $\lambda = \gamma + \omega^\beta$ for $\beta > 0$.
- sequence $(\lambda(x))_{x<\omega}$ s.t. $\forall x.\lambda(x) < \lambda$ and $\sup_x \lambda(x) = \lambda$
- syntactic definition on Cantor normal forms:

$$(\gamma + \omega^{\beta+1})(x) = \gamma + \omega^\beta \cdot (x+1)$$

$$(\gamma + \omega^\lambda)(x) = \gamma + \omega^{\lambda(x)}$$



Hardy Hierarchy

- fix $h\colon \mathbb{N} \to \mathbb{N}$ monotone.
- Hardy hierarchy $(h^\alpha\colon \mathbb{N} \to \mathbb{N})_\alpha$

$$h^0(x) = x\,, \quad h^{\alpha+1}(x) = h^\alpha\big(h(x)\big)\,, \quad h^\lambda(x) = h^{\lambda(x)}(x)\,.$$

- "transfinite iteration"
- examples for $H(x) = x + 1$:

$$H^k(x) = x + k\,, \qquad H^\omega(x) = 2x + 1\,,$$

$$H^{\omega^2}(x) = 2^{x+1} \cdot (x+1) - 1\,,$$

$$H^{\omega^3} \approx \text{tower}\,, \qquad H^{\omega^\omega} \approx \text{ackermann}$$

Hardy Hierarchy

- fix $h\colon \mathbb{N} \to \mathbb{N}$ monotone.
- Hardy hierarchy $(h^\alpha\colon \mathbb{N} \to \mathbb{N})_\alpha$

  $h^0(x) = x\,, \quad h^{\alpha+1}(x) = h^\alpha\big(h(x)\big)\,, \quad h^\lambda(x) = h^{\lambda(x)}(x)\,.$
- "transfinite iteration"
- examples for $H(x) = x + 1$:

  $H^k(x) = x + k\,, \qquad H^\omega(x) = 2x + 1\,,$

  $H^{\omega^2}(x) = 2^{x+1} \cdot (x+1) - 1\,,$

  $H^{\omega^3} \approx \text{tower}\,, \qquad H^{\omega^\omega} \approx \text{ackermann}$

Hardy Hierarchy

- fix $h\colon \mathbb{N} \to \mathbb{N}$ monotone.
- Hardy hierarchy $(h^\alpha\colon \mathbb{N} \to \mathbb{N})_\alpha$

  $h^0(x) = x\,,\quad h^{\alpha+1}(x) = h^\alpha\big(h(x)\big)\,,\quad h^\lambda(x) = h^{\lambda(x)}(x)\,.$
- "transfinite iteration"
- examples for $H(x) = x + 1$:

  $H^k(x) = x + k\,,\qquad H^\omega(x) = 2x + 1\,,$

  $H^{\omega^2}(x) = 2^{x+1} \cdot (x+1) - 1\,,$

  $H^{\omega^3} \approx \text{tower}\,,\qquad H^{\omega^\omega} \approx \text{ackermann}$

Cichoń Hierarchy

Cichoń and Tahhan Bittar (1998)

- Cichoń hierarchy $(h_\alpha : \mathbb{N} \to \mathbb{N})_\alpha$

$$h_0(x) = 0\,, \quad h_{\alpha+1}(x) = 1 + h_\alpha\big(h(x)\big)\,, \quad h_\lambda(x) = h_{\lambda(x)}(x)$$

- "counts" iterations in Hardy functions:

$$h^\alpha(x) = h^{h_\alpha(x)}(x)$$

- for h strictly monotone,

$$h_\alpha(x) \leqslant h^\alpha(x) + x$$

Cichoń Hierarchy

Cichoń and Tahhan Bittar (1998)

- Cichoń hierarchy $(h_\alpha : \mathbb{N} \to \mathbb{N})_\alpha$

$$h_0(x) = 0\,, \quad h_{\alpha+1}(x) = 1 + h_\alpha\big(h(x)\big)\,, \quad h_\lambda(x) = h_{\lambda(x)}(x)$$

- "counts" iterations in Hardy functions:

$$h^\alpha(x) = h^{h_\alpha(x)}(x)$$

- for h strictly monotone,

$$h_\alpha(x) \leqslant h^\alpha(x) + x$$

Cichoń Hierarchy

Cichoń and Tahhan Bittar (1998)

- Cichoń hierarchy $(h_\alpha : \mathbb{N} \to \mathbb{N})_\alpha$

$$h_0(x) = 0\,, \quad h_{\alpha+1}(x) = 1 + h_\alpha\big(h(x)\big)\,, \quad h_\lambda(x) = h_{\lambda(x)}(x)$$

- "counts" iterations in Hardy functions:

$$h^\alpha(x) = h^{h_\alpha(x)}(x)$$

- for h strictly monotone,

$$h_\alpha(x) \leqslant h^\alpha(x) + x$$

Ordinal Norm

Buchholz et al. (1994)

$$N(\omega^{\alpha_1}\cdot c_1+\cdots+\omega^{\alpha_m}\cdot c_m) = \max_{1\leqslant i\leqslant m}(\max(N(\alpha_i),c_i))$$

In the case of ω^d, corresponds to infinite norm over $\mathbb{N}^d$:

$$\|(c_1,\ldots,c_d)\|_\infty = \max_{1\leqslant i\leqslant d} c_i$$

Monotonicity: if $N(\beta)\leqslant x$ and $\beta<\beta'$, then

$$h_\beta(x)\leqslant h_{\beta'}(x) \qquad h^\beta(x)\leqslant h^{\beta'}(x)$$

Predecessors

- of ordinals > 0:

$$P_x(\alpha+1) = \alpha\,, \qquad P_x(\lambda) = P_x(\lambda(x))\,.$$

- for $\alpha > 0$:

$$h_\alpha(x) = 1 + h_{P_x(\alpha)}\big(h(x)\big)\,.$$

- if $N(\alpha) \leqslant x$, then

$$P_x(\alpha) = \max\{\beta < \alpha \mid N(\beta) \leqslant x\}$$

- if $N(\alpha) \leqslant x$, then

$$h_\alpha(x) = \max_{\beta<\alpha, N(\beta)\leqslant x} 1 + h_\beta\big(h(x)\big)$$

Predecessors

- of ordinals > 0:

$$P_x(\alpha+1) = \alpha\,, \qquad P_x(\lambda) = P_x(\lambda(x))\,.$$

- for $\alpha > 0$:

$$h_\alpha(x) = 1 + h_{P_x(\alpha)}\big(h(x)\big)\,.$$

- if $N(\alpha) \leqslant x$, then

$$P_x(\alpha) = \max\{\beta < \alpha \mid N(\beta) \leqslant x\}$$

- if $N(\alpha) \leqslant x$, then

$$h_\alpha(x) = \max_{\beta<\alpha, N(\beta)\leqslant x} 1 + h_\beta\big(h(x)\big)$$

Predecessors

- of ordinals > 0:

$$P_x(\alpha+1) = \alpha\,, \qquad P_x(\lambda) = P_x(\lambda(x))\,.$$

- for $\alpha > 0$:

$$h_\alpha(x) = 1 + h_{P_x(\alpha)}\big(h(x)\big)\,.$$

- if $N(\alpha) \leqslant x$, then

$$P_x(\alpha) = \max\{\beta < \alpha \mid N(\beta) \leqslant x\}$$

- if $N(\alpha) \leqslant x$, then

$$h_\alpha(x) = \max_{\beta<\alpha, N(\beta)\leqslant x} 1 + h_\beta\big(h(x)\big)$$

Predecessors

- of ordinals > 0:

$$P_x(\alpha+1) = \alpha\,, \qquad P_x(\lambda) = P_x(\lambda(x))\,.$$

- for $\alpha > 0$:

$$h_\alpha(x) = 1 + h_{P_x(\alpha)}\big(h(x)\big)\,.$$

- if $N(\alpha) \leqslant x$, then

$$P_x(\alpha) = \max\{\beta < \alpha \mid N(\beta) \leqslant x\}$$

- if $N(\alpha) \leqslant x$, then

$$h_\alpha(x) = \max_{\beta<\alpha, N(\beta)\leqslant x} 1 + h_\beta\big(h(x)\big)$$

Length Function Theorem

Theorem

If $N(\alpha) \leqslant n$, *then* $L_{\alpha,g}(n) = g_\alpha(n)$.

Corollary

If $f : Conf \to \alpha$ *is a* (ι, g)*-controlled ranking function for* $\mathcal{S} = \langle Conf, \to_{\mathcal{S}} \rangle$, *and* c_0 *is an initial configuration in* $Conf$, *then* $\mathcal{S}$ *terminates from* c_0 *in time* $O(g_\alpha(\iota(c_0)))$.

Example

Program:

```
ℓ0: while x >= 0 and y > 0 do
    if x > 0 then
      a: x := x−1; n := 2n;
    else
      b: x := n; y := y−1; n := 2n;
done
```

Ranking function:

$$f{:}\{\ell_0\} \times \mathbb{Z}^3 \to \omega^2$$

$$f(\ell_0, x, y, n) = \omega \cdot y + x$$

Control:

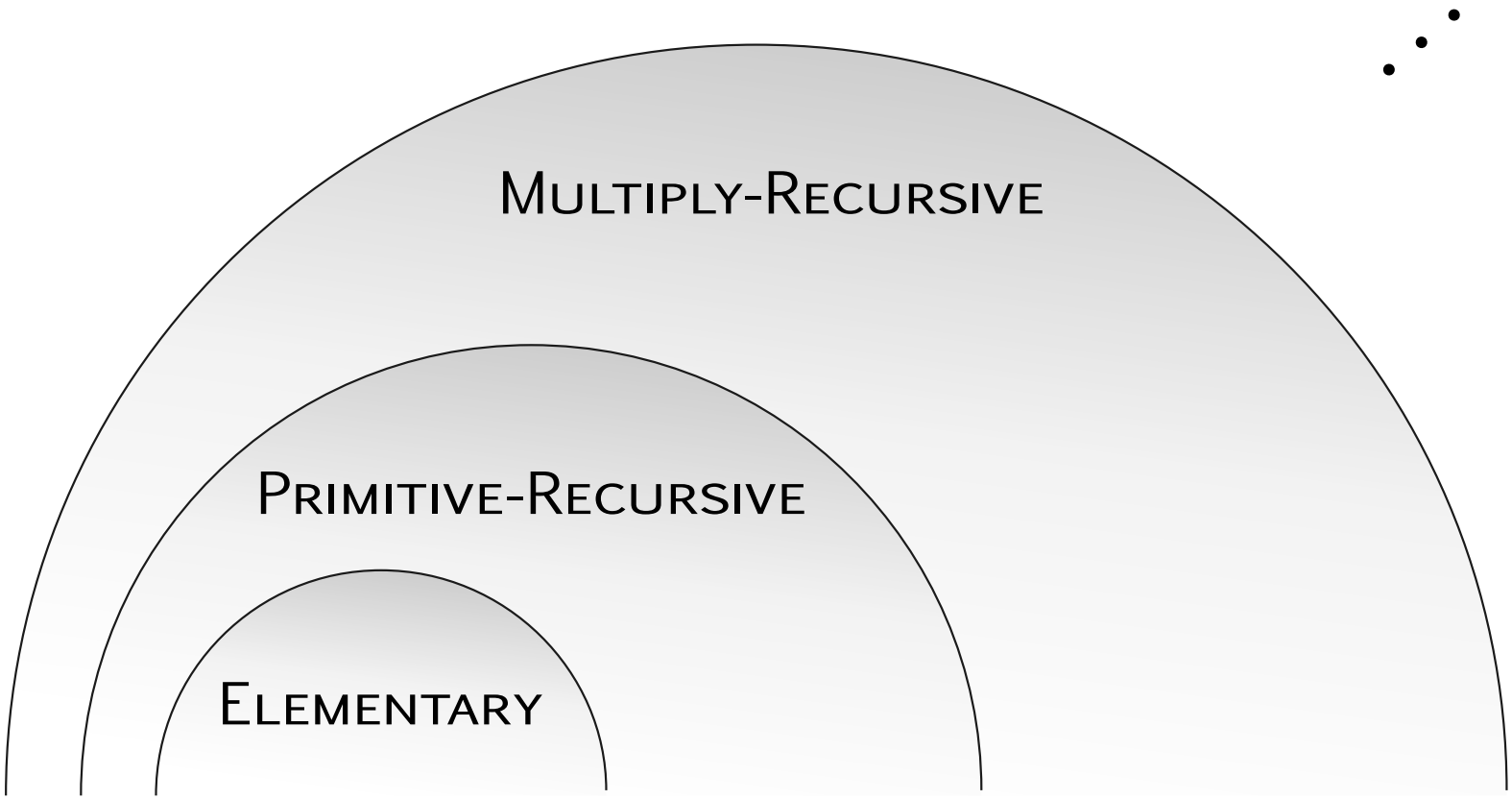
$$\iota(\ell_0, x, y, n) = \max(x, y, n)$$

$$g(x) = 2x$$

Actual complexity:

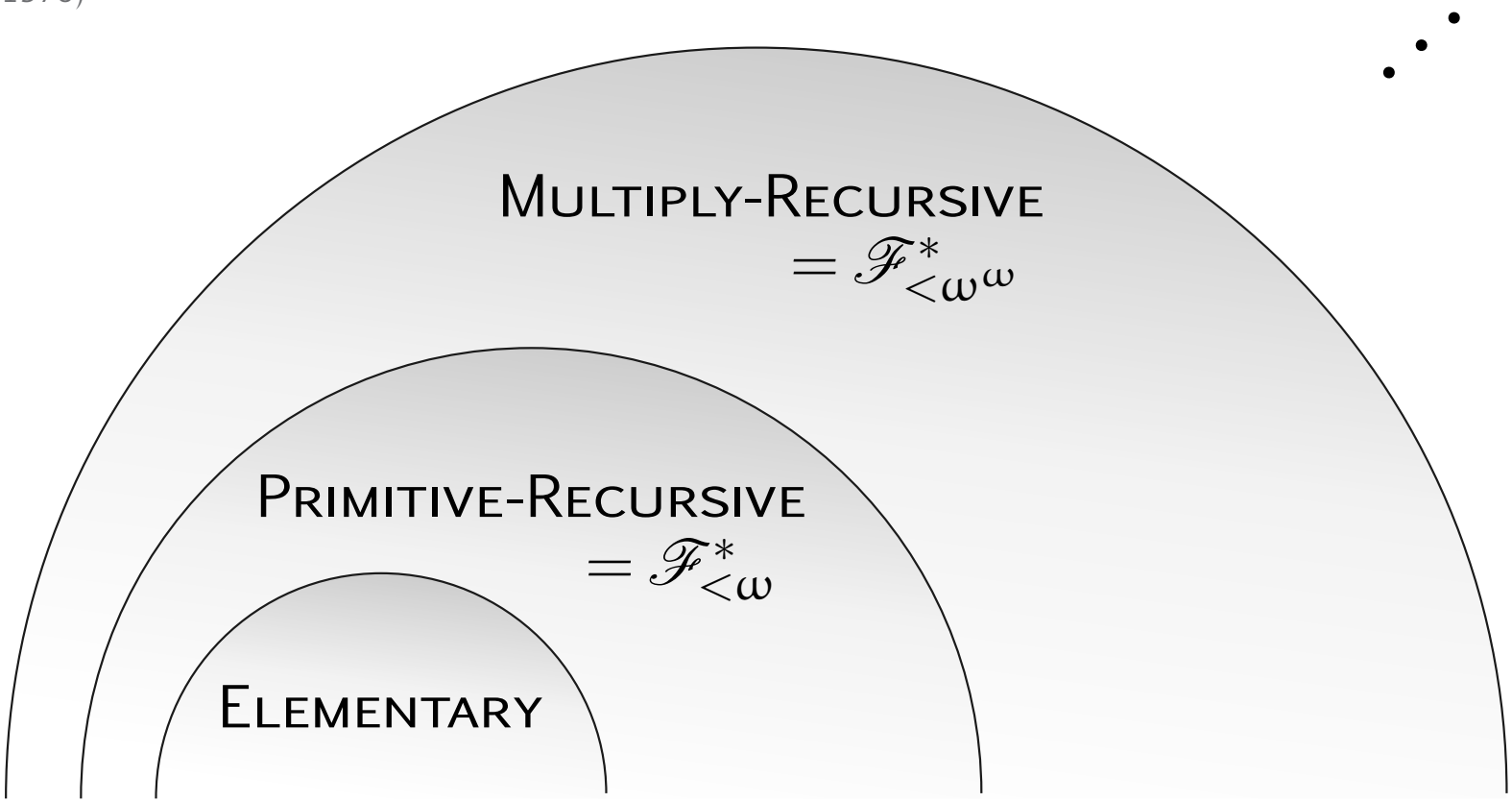
$$g_{\omega \cdot y + x}(n) \leqslant g_{\omega^2}(\iota(\ell_0, x, y, n))$$

Complexity Classification



Complexity Classification

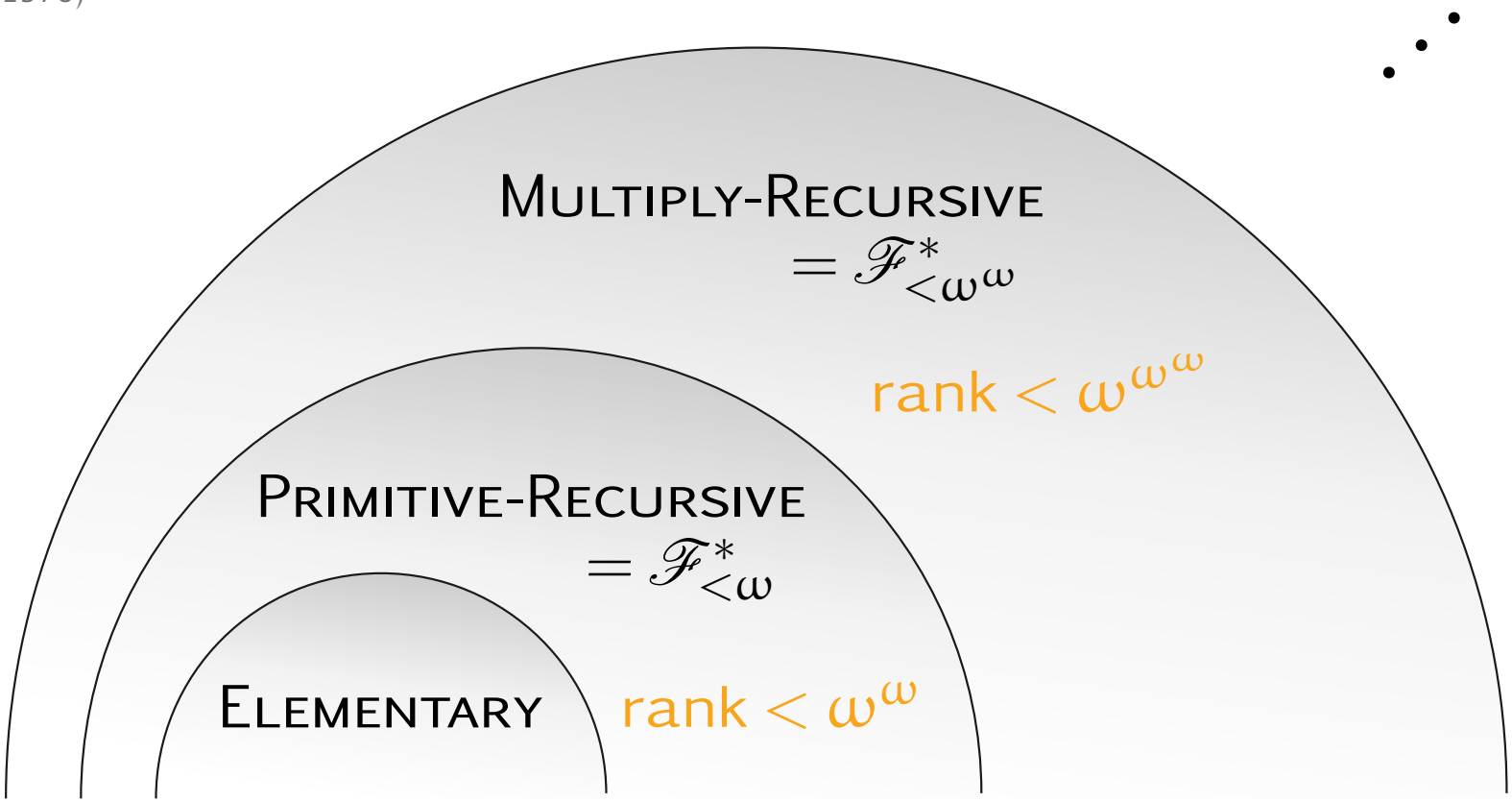
Wainer (1970)



$$\mathscr{F}^*_{<\alpha} \stackrel{\text{def}}{=} \bigcup_{\beta<\omega^\alpha} \textsc{DTime}\left(H^\beta(n)\right)$$

Complexity Classification

Wainer (1970)



$$\mathscr{F}_{<\alpha} \stackrel{\text{def}}{=} \bigcup_{\beta<\omega^\alpha} \textsc{DTime}\big(H^\beta(n)\big) \qquad \text{assuming prim. rec. } (\iota, g)$$

An Application

- very high bounds in general: interesting for complex algorithms
- provide the first known upper bound on VAS Reachability through the decomposition algorithms of Mayr (1981); Kosaraju (1982); Lambert (1992)
- work in progress with Jérôme Leroux

Vector Addition Systems

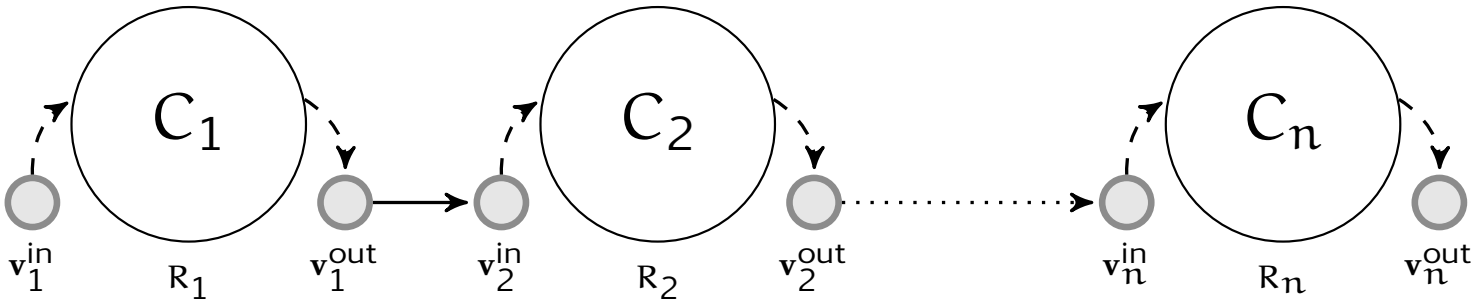
- a VAS $\mathcal{V} = \langle d, T, \mathbf{v}_0 \rangle$
  - $d \in \mathbb{N}$ dimension
  - $T \subseteq \mathbb{Z}^d$ finite set of transitions
  - $\mathbf{v}_0 \in \mathbb{N}^d$ initial vector
- operational semantics: $\langle \mathbb{N}^d, \rightarrow_{\mathcal{V}} \rangle$ where $\mathbf{v} \rightarrow \mathbf{v}'$ if $\mathbf{v}' - \mathbf{v} \in T$.
- reachability problem: given $\mathcal{V}$ and $\mathbf{v}_f \in \mathbb{N}^d$, does $\mathbf{v}_0 \rightarrow_{\mathcal{V}}^* \mathbf{v}_f$?

Vector Addition Systems

- a VAS $\mathcal{V} = \langle d, T, \mathbf{v}_0 \rangle$
  - $d \in \mathbb{N}$ dimension
  - $T \subseteq \mathbb{Z}^d$ finite set of transitions
  - $\mathbf{v}_0 \in \mathbb{N}^d$ initial vector
- operational semantics: $\langle \mathbb{N}^d, \rightarrow_{\mathcal{V}} \rangle$ where $\mathbf{v} \rightarrow \mathbf{v}'$ if $\mathbf{v}' - \mathbf{v} \in T$.
- reachability problem: given $\mathcal{V}$ and $\mathbf{v}_f \in \mathbb{N}^d$, does $\mathbf{v}_0 \rightarrow_{\mathcal{V}}^* \mathbf{v}_f$?

VAS Chains (informally)

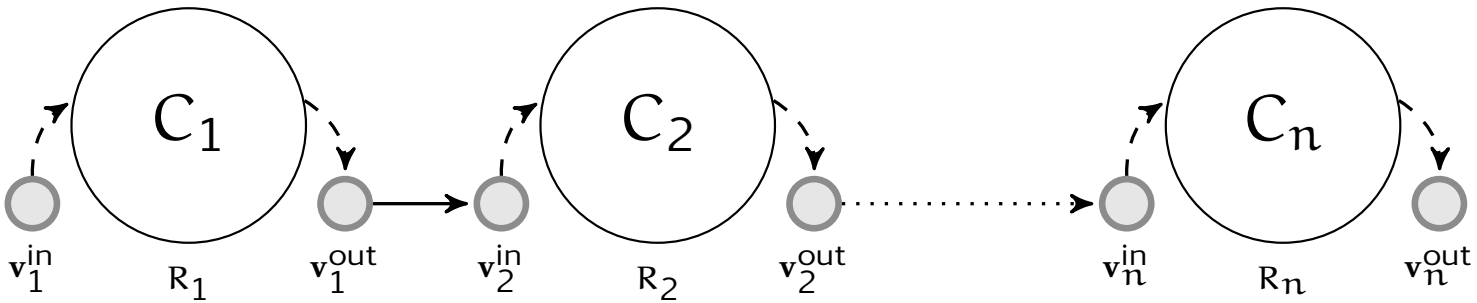
Kosaraju (1982)



- strongly connected components C_i
- input vector $\mathbf{v}_i^{\text{in}}$ and output vector $\mathbf{v}_i^{\text{out}}$ in $(\mathbb{N} \uplus \{\star\})^d$
- labelled using transitions from T
- rigid set R_i s.t. $\forall \mathbf{u}$ label in C_i and $\forall 1 \leqslant j \leqslant d$, $\mathbf{u}(j) = 0$

VAS Chains (informally)

Kosaraju (1982)

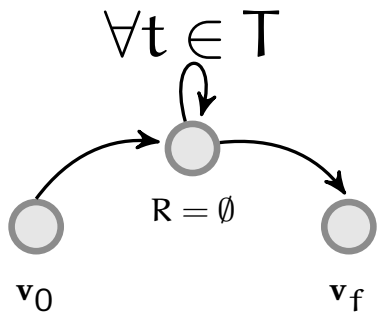


- θ conditions to ensure existence of a run, among which
  - transition counts unbounded
  - $\star$-components of input/output vectors unbounded
  - ...

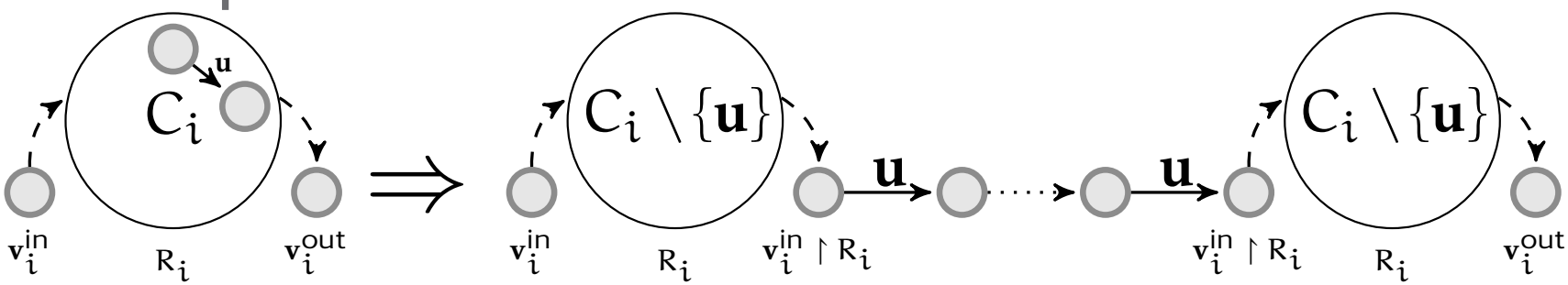
Decomposing VAS Chains

Kosaraju (1982)

Initial



Example: Bounded Transition



Termination

Kosaraju (1982)

- associate with each component C_i a triple in $[d] \times \mathbb{N} \times [2d]$

1. $d - |R_i|$ the number of non-rigid components
2. $|C_i|$ the number of arcs
3. total number of $\star$'s in $\mathbf{v}_i^{in}$ and $\mathbf{v}_i^{out}$

- associate with a VAS chain the corresponding multiset of triples
- ranking function into $\omega^{d \cdot \omega \cdot 2d}$

Termination

Kosaraju (1982)

- associate with each component C_i a triple in $[d] \times \mathbb{N} \times [2d]$

1. $d - |R_i|$ the number of non-rigid components
2. $|C_i|$ the number of arcs
3. total number of $\star$'s in $\mathbf{v}_i^{in}$ and $\mathbf{v}_i^{out}$

- associate with a VAS chain the corresponding multiset of triples
- ranking function into $\omega^{d \cdot \omega \cdot 2d}$

Controlling Decompositions

Figueira et al. (2011)

Use Length Function Theorem for Dickson's Lemma (Figueira et al., 2011) to control bounded values during decomposition:

$$h(x) = H^{\omega^{d+1}}(x)$$

Then overall bound on complexity

$$\left(H^{\omega^{d+1}}\right)^{\omega^{d\cdot\omega\cdot 2d}}(\|\mathbf{v}_0\| + \|\mathbf{v}_f\|) \leqslant H^{\omega^{\omega^2}}(e(n))$$

for some elementary function e and an instance of size n.

Regarding VAS Reachability

cons

- huge gap with ExpSpace lower bound
- non primitive-recusive upper bound, even for fixed dimension d

pros

- very simple complexity argument: the algorithm and the ranking function were already provided by Kosaraju (1982)
- illustrates the power of length function theorems

Concluding Remarks

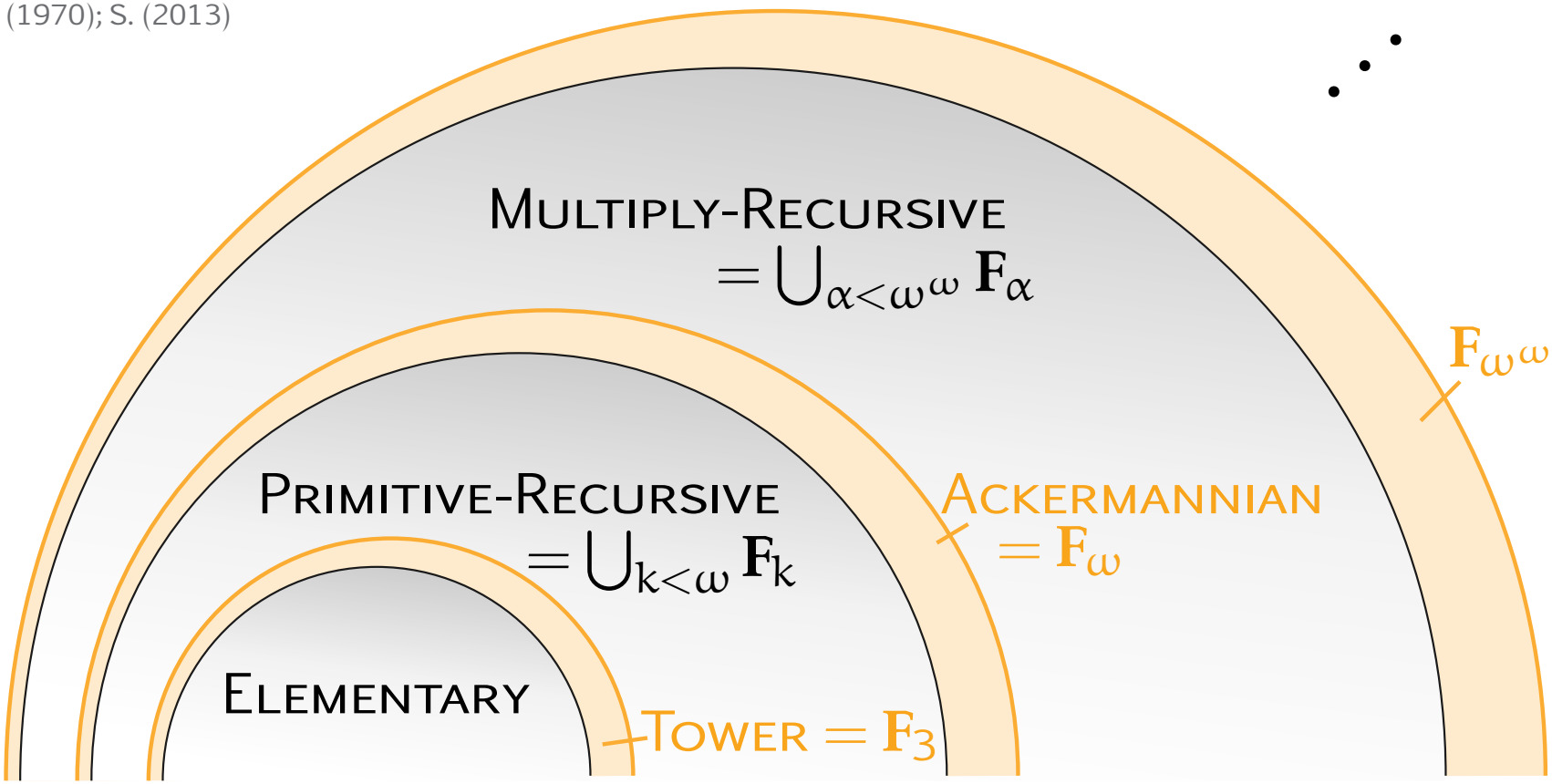
- Length Function Theorem for ordinals below ε_0
- crucial use of the Cichoń hierarchy and ordinal norms
- bounds on complexity of programs with ordinal ranking functions
- beyond ordinals: well-quasi-orders
  - many results already: McAloon (1984); Clote (1986); Weiermann (1994); Cichoń and Tahhan Bittar (1998); Figueira et al. (2011); S. and Schnoebelen (2011); Abriola et al. (2012)
  - how to relate maximal order types to length functions in general?

Concluding Remarks

- Length Function Theorem for ordinals below ε_0
- crucial use of the Cichoń hierarchy and ordinal norms
- bounds on complexity of programs with ordinal ranking functions
- beyond ordinals: well-quasi-orders
  - many results already: McAloon (1984); Clote (1986); Weiermann (1994); Cichoń and Tahhan Bittar (1998); Figueira et al. (2011); S. and Schnoebelen (2011); Abriola et al. (2012)
  - how to relate maximal order types to length functions in general?

Complexity Classification

Wainer (1970); S. (2013)



$$\mathscr{F}_{<\alpha} \stackrel{\text{def}}{=} \bigcup_{\beta<\omega^\alpha} \mathrm{FDTime}\big(H^\beta(n)\big)\,, \quad \mathbf{F}_\alpha \stackrel{\text{def}}{=} \bigcup_{p\in\mathscr{F}_{<\alpha}} \mathrm{DTime}\big(H^{\omega^\alpha}(p(n))\big)$$

Abriola, S., Figueira, S., and Senno, G., 2012. Linearizing bad sequences: upper bounds for the product and majoring well quasi-orders. In *WoLLIC 2012*, vol. 7456 of *LNCS*, pp. 110–126. Springer. doi:10.1007/978-3-642-32621-9_9.

Ben-Amram, A.M., 2002. General size-change termination and lexicographic descent. In *The Essence of Computation*, vol. 2566 of *LNCS*, pp. 3–17. Springer. doi:10.1007/3-540-36377-7_1.

Berardi, S., Oliva, P., and Steila, S., 2014. Proving termination with transition invariants of height omega. Preprint. arXiv:1407.4692[cs.LO].

Buchholz, W., Cichoń, E.A., and Weiermann, A., 1994. A uniform approach to fundamental sequences and hierarchies. *Math. Logic Quart.*, 40(2):273–286. doi:10.1002/malq.19940400212.

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.

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.

Cook, B., Podelski, A., and Rybalchenko, A., 2011. Proving program termination. *Communications of the ACM*, 54(5):88–98. doi:10.1145/1941487.1941509.

Figueira, D., Figueira, S., Schmitz, S., and Schnoebelen, Ph., 2011. Ackermannian and primitive-recursive bounds with Dickson's Lemma. In *LICS 2011*, pp. 269–278. IEEE. doi:10.1109/LICS.2011.39.

Hofbauer, D., 1992. Termination proofs by multiset path orderings imply primitive recursive derivation lengths. *Theor. Comput. Sci.*, 105(1):129–140. doi:10.1016/0304-3975(92)90289-R.

Kosaraju, S.R., 1982. Decidability of reachability in vector addition systems. In *STOC '82*, pp. 267–281. ACM. doi:10.1145/800070.802201.

Lambert, J.L., 1992. A structure to decide reachability in Petri nets. *Theor. Comput. Sci.*, 99(1):79–104. doi:10.1016/0304-3975(92)90173-D.

Mayr, E.W., 1981. An algorithm for the general Petri net reachability problem. In *STOC'81*, pp. 238–246. ACM. doi:10.1145/800076.802477.

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.

Schmitz, S. and Schnoebelen, Ph., 2011. Multiply-recursive upper bounds with Higman's Lemma. In *ICALP 2011*, vol. 6756 of *LNCS*, pp. 441–452. Springer. doi:10.1007/978-3-642-22012-8_35.

Schmitz, S., 2013. Complexity hierarchies beyond Elementary. Preprint. arXiv:1312.5686[cs.CC].

Turing, A.M., 1949. Checking a large routine. In *EDSAC 1949*, pp. 67–69.

Wainer, S.S., 1970. A classification of the ordinal recursive functions. *Arch. Math. Logic*, 13(3): 136–153. doi:10.1007/BF01973619.

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.

Weiermann, A., 1995. Termination proofs for term rewriting systems by lexicographic path orderings imply multiply recursive derivation lengths. *Theor. Comput. Sci.*, 139(1–2):355–362. doi:10.1016/0304-3975(94)00135-6.