Algorithmic Aspects of WQO (Well-Quasi-Ordering) Theory

Part III: Fast-growing complexity

Philippe Schnoebelen

LSV, CNRS & ENS Cachan

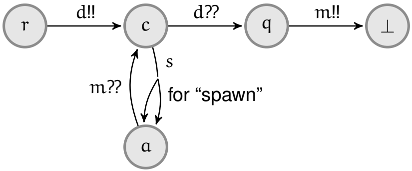
Chennai Mathematical Institute, Jan. 2017

Based on joint work with Sylvain Schmitz, Prateek Karandikar, K. Narayan Kumar, Alain Finkel, ..

Lecture notes & exercises available via `www.lsv.ens-cachan.fr/~phs`

Example of WSTS: Broadcast Protocols

Broadcast protocols (Esparza et al.'99) are dynamic & distributed collections of finite-state processes communicating via brodcasts and rendez-vous.



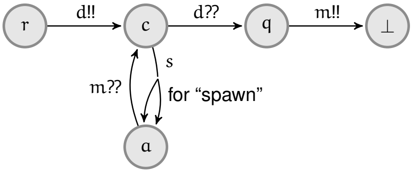
A configuration collects the local states of all processes. E.g., $s = \{c, r, c\}$, also denoted $\{c^2, r\}$.

Steps: $\{c^2, q, r\} \xrightarrow{s} \{a^2, c, q, r\} \xrightarrow{s} \{a^4, q, r\} \xrightarrow{m} \{c^4, r, \bot\} \xrightarrow{d} \{c, q^4, \bot\}$

We'll soon see: The above system does not have infinite runs

Example of WSTS: Broadcast Protocols

Broadcast protocols (Esparza et al.'99) are dynamic & distributed collections of finite-state processes communicating via brodcasts and rendez-vous.



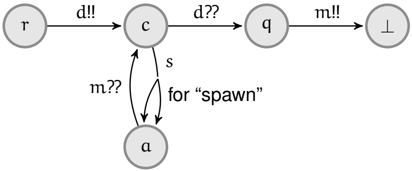
A configuration collects the local states of all processes. E.g., $s = \{c, r, c\}$, also denoted $\{c^2, r\}$.

Steps: $\{c^2, q, r\} \xrightarrow{s} \{a^2, c, q, r\} \xrightarrow{s} \{a^4, q, r\} \xrightarrow{m} \{c^4, r, \perp\} \xrightarrow{d} \{c, q^4, \perp\}$

We'll soon see: The above system does not have infinite runs

Example of WSTS: Broadcast Protocols

Broadcast protocols (Esparza et al.'99) are dynamic & distributed collections of finite-state processes communicating via brodcasts and rendez-vous.



A configuration collects the local states of all processes. E.g., $s = \{c, r, c\}$, also denoted $\{c^2, r\}$.

Steps: $\{c^2, q, r\} \xrightarrow{s} \{a^2, c, q, r\} \xrightarrow{s} \{a^4, q, r\} \xrightarrow{m} \{c^4, r, \perp\} \xrightarrow{d} \{c, q^4, \perp\}$

We'll soon see: The above system does not have infinite runs

Brodcast Protocols Are WSTS

Ordering of configurations is multiset inclusion, e.g., $\{c, q\} \subseteq \{c^2, r, q\}$

Fact. Configurations $(\mathbb{N}^{\{r,c,a,q,\perp\}}, \subseteq)$ is a wqo.

Proof: this is exactly $(\mathbb{N}^5, \leqslant_\times)$

Fact. Brodcast protocols are monotonic TS

Proof Idea: assume $s_1 \subseteq t_1$ and consider all cases for a step $s_1 \to s_2$. In each case we have to find some $t_1 \to t_2$ with $s_2 \subseteq t_2$.

Coro. Broadcast protocols are WSTS

BRODCAST PROTOCOLS ARE WSTS

Ordering of configurations is multiset inclusion, e.g., $\{c, q\} \subseteq \{c^2, r, q\}$

Fact. Configurations $(\mathbb{N}^{\{r,c,a,q,\perp\}}, \subseteq)$ is a wqo.

Proof: this is exactly $(\mathbb{N}^5, \leqslant_\times)$

Fact. Brodcast protocols are monotonic TS

Proof Idea: assume $s_1 \subseteq t_1$ and consider all cases for a step $s_1 \to s_2$. In each case we have to find some $t_1 \to t_2$ with $s_2 \subseteq t_2$.

Coro. Broadcast protocols are WSTS

Brodcast Protocols Are WSTS

Ordering of configurations is multiset inclusion, e.g., $\{c, q\} \subseteq \{c^2, r, q\}$

Fact. Configurations $(\mathbb{N}^{\{r,c,a,q,\perp\}}, \subseteq)$ is a wqo.

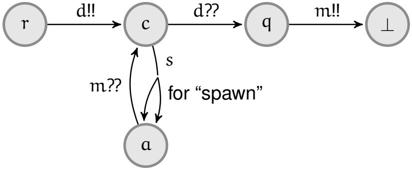
Proof: this is exactly $(\mathbb{N}^5, \leqslant_\times)$

Fact. Brodcast protocols are monotonic TS

Proof Idea: assume $s_1 \subseteq t_1$ and consider all cases for a step $s_1 \to s_2$. In each case we have to find some $t_1 \to t_2$ with $s_2 \subseteq t_2$.

Coro. Broadcast protocols are WSTS

BROADCAST PROTOCOLS AND TERMINATION



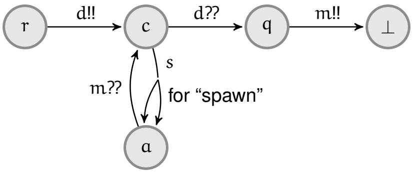
This broadcast protocol terminates: all its runs are bad sequences, hence are finite

Proof. Assume $s_0 \to s_1 \to \cdots \to s_n$ and pick two positions $i < j$.
Write $s_i = \{a^{n_a}, c^{n_c}, q^{n_q}, r^{n_r}, \perp^*\}$, and $s_j = \{a^{n'_a}, c^{n'_c}, q^{n'_q}, r^{n'_r}, \perp^*\}$.

– if a d has been broadcast during $s_i \xrightarrow{+} s_j$, then $n'_r < n_r$,
– if no d but a m have been broadcast, then $n'_q < n_q$,
– otherwise $s_i \xrightarrow{+} s_j$ uses only spawning steps, then $n'_c < n_c$.

In all cases, $s_i \not\subseteq s_j$. QED

Broadcast Protocols and Termination



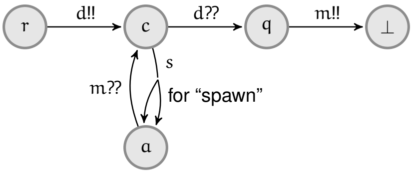
This broadcast protocol terminates: all its runs are bad sequences, hence are finite

Proof. Assume $s_0 \to s_1 \to \cdots \to s_n$ and pick two positions $i < j$.
Write $s_i = \{a^{n_a}, c^{n_c}, q^{n_q}, r^{n_r}, \perp^*\}$, and $s_j = \{a^{n'_a}, c^{n'_c}, q^{n'_q}, r^{n'_r}, \perp^*\}$.

– if a d has been broadcast during $s_i \xrightarrow{+} s_j$, then $n'_r < n_r$,
– if no d but a m have been broadcast, then $n'_q < n_q$,
– otherwise $s_i \xrightarrow{+} s_j$ uses only spawning steps, then $n'_c < n_c$.

In all cases, $s_i \not\subseteq s_j$. QED

Broadcast Protocols and Termination



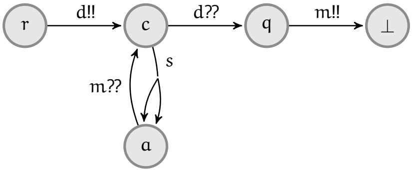
This broadcast protocol terminates: all its runs are bad sequences, hence are finite

Proof. Assume $s_0 \to s_1 \to \cdots \to s_n$ and pick two positions $i < j$.
Write $s_i = \{a^{n_a}, c^{n_c}, q^{n_q}, r^{n_r}, \perp^*\}$, and $s_j = \{a^{n'_a}, c^{n'_c}, q^{n'_q}, r^{n'_r}, \perp^*\}$.

– if a d has been broadcast during $s_i \xrightarrow{+} s_j$, then $n'_r < n_r$,
– if no d but a m have been broadcast, then $n'_q < n_q$,
– otherwise $s_i \xrightarrow{+} s_j$ uses only spawning steps, then $n'_c < n_c$.

In all cases, $s_i \not\subseteq s_j$. QED

Broadcast Protocols Take Their Time

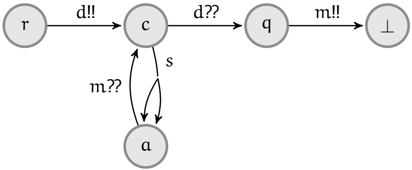


“Doubling” run: $\{c^{n}, q, \perp^{*}\} \xrightarrow{s^{n}} \{a^{2n}, q, \perp^{*}\} \xrightarrow{m} \{c^{2n}, \perp^{+}\}$

Building up: $\{c^{2^0}, q^{n}, r\} \xrightarrow{s^{2^0} m} \{c^{2^1}, q^{n-1}, r\} \xrightarrow{s^{2^1} m} \{c^{2^2}, q^{n-2}, r\} \to \cdots \to \{c^{2^{n-1}}, q, r\} \xrightarrow{s^{2^{n-1}} m} \{c^{2^n}, r\} \xrightarrow{d} \{c^{2^0}, q^{2^n}\}$

Then: $\{c, q, r^{n}\} \xrightarrow{*} \{c, q^{2^n}, r^{n-1}\} \xrightarrow{*} \{c, q^{\mathsf{tower}(n)}\}$

Broadcast Protocols Take Their Time

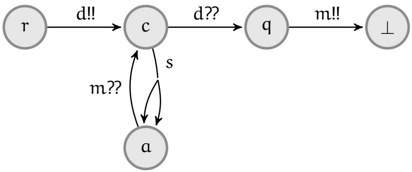


"Doubling" run: $\{c^{n}, q, \perp^{*}\} \xrightarrow{s^{n}} \{a^{2n}, q, \perp^{*}\} \xrightarrow{m} \{c^{2n}, \perp^{+}\}$

Building up: $\{c^{2^0}, q^{n}, r\} \xrightarrow{s^{2^0} m} \{c^{2^1}, q^{n-1}, r\} \xrightarrow{s^{2^1} m} \{c^{2^2}, q^{n-2}, r\} \rightarrow$
$\cdots \rightarrow \{c^{2^{n-1}}, q, r\} \xrightarrow{s^{2^{n-1}} m} \{c^{2^n}, r\} \xrightarrow{d} \{c^{2^0}, q^{2^n}\}$

Then: $\{c, q, r^{n}\} \xrightarrow{*} \{c, q^{2^n}, r^{n-1}\} \xrightarrow{*} \{c, q^{\text{tower}(n)}\}$

Broadcast Protocols Take Their Time



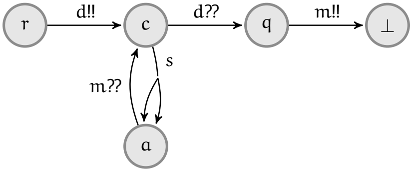
"Doubling" run: $\{c^{n}, q, \perp^*\} \xrightarrow{s^n} \{a^{2n}, q, \perp^*\} \xrightarrow{m} \{c^{2n}, \perp^+\}$

Building up: $\{c^{2^0}, q^{n}, r\} \xrightarrow{s^{2^0} m} \{c^{2^1}, q^{n-1}, r\} \xrightarrow{s^{2^1} m} \{c^{2^2}, q^{n-2}, r\} \to$
$\cdots \to \{c^{2^{n-1}}, q, r\} \xrightarrow{s^{2^{n-1}} m} \{c^{2^n}, r\} \xrightarrow{d} \{c^{2^0}, q^{2^n}\}$

Then: $\{c, q, r^{n}\} \xrightarrow{*} \{c, q^{2^n}, r^{n-1}\} \xrightarrow{*} \{c, q^{\text{tower}(n)}\}$

where $\text{tower}(n) \stackrel{\text{def}}{=} \left. 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \right\} n \text{ times}$

Broadcast Protocols Take Their Time



"Doubling" run: $\{c^{n}, q, \perp^*\} \xrightarrow{s^n} \{a^{2n}, q, \perp^*\} \xrightarrow{m} \{c^{2n}, \perp^+\}$

Building up: $\{c^{2^0}, q^{n}, r\} \xrightarrow{s^{2^0} m} \{c^{2^1}, q^{n-1}, r\} \xrightarrow{s^{2^1} m} \{c^{2^2}, q^{n-2}, r\} \rightarrow$
$\cdots \rightarrow \{c^{2^{n-1}}, q, r\} \xrightarrow{s^{2^{n-1}} m} \{c^{2^n}, r\} \xrightarrow{d} \{c^{2^0}, q^{2^n}\}$

Then: $\{c, q, r^{n}\} \xrightarrow{*} \{c, q^{2^n}, r^{n-1}\} \xrightarrow{*} \{c, q^{\text{tower}(n)}\}$

⇒ Runs of terminating systems may have nonelementary lengths

⇒ Running time of termination verification algorithm is not elementary (for broadcast protocols)

Ordinal Indexes for Complexity Classes

The complexity analysis for WQO-based algorithms use new complexity classes: $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \ldots$

Continues with transfinite indexes: $\mathbf{F}_4, \ldots, \mathbf{F}_\omega, \mathbf{F}_{\omega+1}, \mathbf{F}_{\omega+2}, \ldots, \mathbf{F}_{\omega\cdot 2}, \mathbf{F}_{\omega\cdot 2+1}, \ldots, \mathbf{F}_{\omega\cdot 3}, \ldots, \mathbf{F}_{\omega\cdot 4}, \ldots, \mathbf{F}_{\omega^2}, \mathbf{F}_{\omega^2+1}, \ldots, \mathbf{F}_{\omega^2+\omega}, \ldots, \mathbf{F}_{\omega^2+\omega\cdot 2}, \ldots, \mathbf{F}_{\omega^2\cdot 2}, \ldots, \mathbf{F}_{\omega^3}, \ldots, \mathbf{F}_{\omega^\omega}, \ldots, \mathbf{F}_{\omega^{\omega^\omega}}, \ldots, \mathbf{F}_{\omega^{\omega^{\omega^\omega}}}, \ldots,$

- We work with ordinals below ε_0 written in Cantor normal form:

$$\alpha = \omega^{\alpha_1} + \cdots + \omega^{\alpha_m} \quad \text{where } \alpha > \alpha_1 \geqslant \cdots \geqslant \alpha_m$$

NB: α is zero iff $m = 0$; it is a successor $\alpha = \beta + 1 = \beta + \omega^0$ iff $m > 0$ and $\alpha_m = 0$; otherwise it is a limit $\alpha = \lambda$

Alternative notation:

$$\alpha = \omega^{\alpha_1} \cdot c_1 + \cdots + \omega^{\alpha_m} \cdot c_m \quad \text{now with } \begin{array}{l} \alpha > \alpha_1 > \ldots > \alpha_m \\ c_1, \ldots, c_m \in \mathbb{N} \end{array}$$

ORDINAL INDEXES FOR COMPLEXITY CLASSES

The complexity analysis for WQO-based algorithms use new complexity classes: $F_1, F_2, F_3, \ldots$

Continues with transfinite indexes: $F_4, \ldots, F_\omega, F_{\omega+1}, F_{\omega+2}, \ldots, F_{\omega\cdot 2}, F_{\omega\cdot 2+1}, \ldots, F_{\omega\cdot 3}, \ldots, F_{\omega\cdot 4}, \ldots, F_{\omega^2}, F_{\omega^2+1}, \ldots, F_{\omega^2+\omega}, \ldots, F_{\omega^2+\omega\cdot 2}, \ldots, F_{\omega^2\cdot 2}, \ldots, F_{\omega^3}, \ldots, F_{\omega^\omega}, \ldots, F_{\omega^{\omega^\omega}}, \ldots, F_{\omega^{\omega^{\omega^\omega}}}, \ldots,$

- We work with ordinals below ε_0 written in Cantor normal form:

$$\alpha = \omega^{\alpha_1} + \cdots + \omega^{\alpha_m} \quad \text{where } \alpha > \alpha_1 \geqslant \cdots \geqslant \alpha_m$$

NB: α is zero iff $m = 0$; it is a successor $\alpha = \beta + 1 = \beta + \omega^0$ iff $m > 0$ and $\alpha_m = 0$; otherwise it is a limit $\alpha = \lambda$

Alternative notation:

$$\alpha = \omega^{\alpha_1} \cdot c_1 + \cdots + \omega^{\alpha_m} \cdot c_m \quad \text{now with } \begin{array}{l} \alpha > \alpha_1 > \ldots > \alpha_m \\ c_1, \ldots, c_m \in \mathbb{N} \end{array}$$

Ordinal Indexes for Complexity Classes

The complexity analysis for WQO-based algorithms use new complexity classes: $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \ldots$

Continues with transfinite indexes: $\mathbf{F}_4, \ldots, \mathbf{F}_\omega, \mathbf{F}_{\omega+1}, \mathbf{F}_{\omega+2}, \ldots, \mathbf{F}_{\omega\cdot 2}, \mathbf{F}_{\omega\cdot 2+1}, \ldots, \mathbf{F}_{\omega\cdot 3}, \ldots, \mathbf{F}_{\omega\cdot 4}, \ldots, \mathbf{F}_{\omega^2}, \mathbf{F}_{\omega^2+1}, \ldots, \mathbf{F}_{\omega^2+\omega}, \ldots, \mathbf{F}_{\omega^2+\omega\cdot 2}, \ldots, \mathbf{F}_{\omega^2\cdot 2}, \ldots, \mathbf{F}_{\omega^3}, \ldots, \mathbf{F}_{\omega^\omega}, \ldots, \mathbf{F}_{\omega^{\omega^\omega}}, \ldots, \mathbf{F}_{\omega^{\omega^{\omega^\omega}}}, \ldots,$

- We work with ordinals below ε_0 written in Cantor normal form:

$$\alpha = \omega^{\alpha_1} + \cdots + \omega^{\alpha_m} \quad \text{where } \alpha > \alpha_1 \geqslant \cdots \geqslant \alpha_m$$

NB: α is zero iff $m = 0$; it is a successor $\alpha = \beta + 1 = \beta + \omega^0$ iff $m > 0$ and $\alpha_m = 0$; otherwise it is a limit $\alpha = \lambda$

Alternative notation:

$$\alpha = \omega^{\alpha_1} \cdot c_1 + \cdots + \omega^{\alpha_m} \cdot c_m \quad \text{now with } \begin{array}{l} \alpha > \alpha_1 > \ldots > \alpha_m \\ c_1, \ldots, c_m \in \mathbb{N} \end{array}$$

Fast-Growing Functions

$(F_\alpha)_{\alpha \in Ord}$: an ordinal-indexed family of functions $F_\alpha : \mathbb{N} \to \mathbb{N}$

$$F_0(x) \stackrel{\text{def}}{=} x+1 \qquad F_{\alpha+1}(x) \stackrel{\text{def}}{=} \overbrace{F_\alpha(F_\alpha(\dots F_\alpha}^{x+1}(x)\dots)) \qquad F_\omega(x) \stackrel{\text{def}}{=} F_{x+1}(x)$$

gives $F_1(x) = 2x+1 \approx 2x$, $F_2(x) = 2^{x+1}(x+1) - 1 \approx 2^x$, $F_3(x) \approx \text{tower}(x)$ and $F_\omega(x) \approx \text{ACKERMANN}(x)$, the first F_α that is not primitive recursive.

Generally $F_\lambda(x) \stackrel{\text{def}}{=} F_{\lambda_x}(x)$ with $\lambda_0 < \lambda_1 < \lambda_2 < \cdots < \lambda$ a fundamental sequence for λ, given by

$$(\gamma + \omega^{\beta+1})_x \stackrel{\text{def}}{=} \gamma + \omega^\beta \cdot (x+1) \qquad (\gamma + \omega^\lambda)_x \stackrel{\text{def}}{=} \gamma + \omega^{\lambda_x}$$

E.g. $F_{\omega^2}(7) = F_{\omega\cdot 8}(7) = F_{\omega\cdot 7+8}(7) = \overbrace{F_{\omega\cdot 7+7}(F_{\omega\cdot 7+7}(\cdots(F_{\omega\cdot 7+7}}^{8}(7))\cdots))$

Fast-Growing Functions

$(F_\alpha)_{\alpha \in Ord}$: an ordinal-indexed family of functions $F_\alpha : \mathbb{N} \to \mathbb{N}$

$$F_0(x) \stackrel{\text{def}}{=} x+1 \qquad F_{\alpha+1}(x) \stackrel{\text{def}}{=} \overbrace{F_\alpha(F_\alpha(\ldots F_\alpha}^{x+1}(x)\ldots)) \qquad F_\omega(x) \stackrel{\text{def}}{=} F_{x+1}(x)$$

gives $F_1(x) = 2x+1 \approx 2x$, $F_2(x) = 2^{x+1}(x+1) - 1 \approx 2^x$, $F_3(x) \approx \text{tower}(x)$ and $F_\omega(x) \approx \textsc{Ackermann}(x)$, the first F_α that is not primitive recursive.

Generally $F_\lambda(x) \stackrel{\text{def}}{=} F_{\lambda_x}(x)$ with $\lambda_0 < \lambda_1 < \lambda_2 < \cdots < \lambda$ a fundamental sequence for λ, given by

$$(\gamma + \omega^{\beta+1})_x \stackrel{\text{def}}{=} \gamma + \omega^\beta \cdot (x+1) \qquad (\gamma + \omega^\lambda)_x \stackrel{\text{def}}{=} \gamma + \omega^{\lambda_x}$$

E.g. $F_{\omega^2}(7) = F_{\omega \cdot 8}(7) = F_{\omega\cdot 7+8}(7) = \overbrace{F_{\omega\cdot 7+7}(F_{\omega\cdot 7+7}(\cdots(F_{\omega\cdot 7+7}}^{8}(7))\cdots))$

FAST-GROWING FUNCTIONS

$(F_\alpha)_{\alpha \in Ord}$: an ordinal-indexed family of functions $F_\alpha : \mathbb{N} \to \mathbb{N}$

$$F_0(x) \stackrel{\text{def}}{=} x+1 \qquad F_{\alpha+1}(x) \stackrel{\text{def}}{=} \overbrace{F_\alpha(F_\alpha(\ldots F_\alpha}^{x+1}(x)\ldots)) \qquad F_\omega(x) \stackrel{\text{def}}{=} F_{x+1}(x)$$

gives $F_1(x) = 2x+1 \approx 2x$, $F_2(x) = 2^{x+1}(x+1) - 1 \approx 2^x$,
$F_3(x) \approx \text{tower}(x)$ and $F_\omega(x) \approx \text{ACKERMANN}(x)$, the first F_α that is not primitive recursive.

Generally $F_\lambda(x) \stackrel{\text{def}}{=} F_{\lambda_x}(x)$ with $\lambda_0 < \lambda_1 < \lambda_2 < \cdots < \lambda$ a fundamental sequence for λ, given by

$$(\gamma + \omega^{\beta+1})_x \stackrel{\text{def}}{=} \gamma + \omega^\beta \cdot (x+1) \qquad (\gamma + \omega^\lambda)_x \stackrel{\text{def}}{=} \gamma + \omega^{\lambda_x}$$

E.g. $F_{\omega^2}(7) = F_{\omega\cdot 8}(7) = F_{\omega\cdot 7+8}(7) = \overbrace{F_{\omega\cdot 7+7}(F_{\omega\cdot 7+7}(\cdots(F_{\omega\cdot 7+7}}^{8}(7))\cdots))$

The Fast-Growing Hierarchy

By Schmitz (2013), after Wainer & Löb (1970), Grzegorczyk (1953)

$$\mathbf{F}_\alpha \stackrel{\text{def}}{=} \bigcup_{p \in \mathscr{F}_{<\alpha}} \mathrm{FDT{\small IME}}(F_\alpha(p(n)))$$, ie all functions in time $F_\alpha(\textit{negligible}(n))$

$$\mathscr{F}_{<\alpha} \stackrel{\text{def}}{=} \bigcup_{\beta<\alpha} \mathscr{F}_\beta \qquad \mathscr{F}_\alpha \stackrel{\text{def}}{=} \bigcup_{c\in\mathbb{N}} \mathbf{F}^c_\alpha \qquad \mathbf{F}^c_\alpha \stackrel{\text{def}}{=} \bigcup_{p\in\mathscr{F}_{<\alpha}} \mathrm{FDT{\small IME}}(F^c_\alpha(p(n)))$$

1. These classes admit many other characterizations and capture some well-known cases:

$\mathbf{F}_2 = \mathrm{E} = \mathrm{DT{\small IME}}(2^{O(n)})$, $\mathscr{F}_{<3} = \mathrm{FE{\small LEM}}$, $\mathscr{F}_{<\omega} = \mathrm{PR}$, $\mathscr{F}_{<\omega^\omega} = \mathrm{MPR}$

2. A strict hierarchy: $\mathbf{F}_\beta \subsetneq \mathbf{F}^{c+1}_\beta \subsetneq \mathbf{F}_\alpha$ for all $\beta < \alpha$ and $c > 0$.

3. There exist $\mathbf{F}_\alpha$-complete problems for each $\alpha \geqslant 2$

THE FAST-GROWING HIERARCHY

By Schmitz (2013), after Wainer & Löb (1970), Grzegorczyk (1953)

$\mathbf{F}_\alpha \stackrel{\text{def}}{=} \bigcup_{p \in \mathscr{F}_{<\alpha}} \text{FDTIME}(F_\alpha(p(n)))$, ie all functions in time $F_\alpha($*negligible*$(n))$

$$\mathscr{F}_{<\alpha} \stackrel{\text{def}}{=} \bigcup_{\beta<\alpha} \mathscr{F}_\beta \qquad \mathscr{F}_\alpha \stackrel{\text{def}}{=} \bigcup_{c \in \mathbb{N}} \mathbf{F}_\alpha^c \qquad \mathbf{F}_\alpha^c \stackrel{\text{def}}{=} \bigcup_{p \in \mathscr{F}_{<\alpha}} \text{FDTIME}(F_\alpha^c(p(n)))$$

1. These classes admit many other characterizations and capture some well-known cases:

$\mathbf{F}_2 = \text{E} = \text{DTIME}(2^{O(n)})$, $\mathscr{F}_{<3} = \text{FELEM}$, $\mathscr{F}_{<\omega} = \text{PR}$, $\mathscr{F}_{<\omega^\omega} = \text{MPR}$

2. A strict hierarchy: $\mathbf{F}_\beta \subsetneq \mathbf{F}_\beta^{c+1} \subsetneq \mathbf{F}_\alpha$ for all $\beta < \alpha$ and $c > 0$.

3. There exist $\mathbf{F}_\alpha$-complete problems for each $\alpha \geqslant 2$

The Fast-Growing Hierarchy

By Schmitz (2013), after Wainer & Löb (1970), Grzegorczyk (1953)

$$\mathbf{F}_\alpha \stackrel{\text{def}}{=} \bigcup_{p \in \mathscr{F}_{<\alpha}} \mathrm{FDT{\small IME}}(F_\alpha(p(n))), \text{ ie all functions in time } F_\alpha(\textit{negligible}(n))$$

$$\mathscr{F}_{<\alpha} \stackrel{\text{def}}{=} \bigcup_{\beta<\alpha} \mathscr{F}_\beta \qquad \mathscr{F}_\alpha \stackrel{\text{def}}{=} \bigcup_{c \in \mathbb{N}} \mathbf{F}^c_\alpha \qquad \mathbf{F}^c_\alpha \stackrel{\text{def}}{=} \bigcup_{p \in \mathscr{F}_{<\alpha}} \mathrm{FDT{\small IME}}(F^c_\alpha(p(n)))$$

1. These classes admit many other characterizations and capture some well-known cases:

$$\mathbf{F}_2 = \mathrm{E} = \mathrm{DT{\small IME}}(2^{O(n)}),\ \mathscr{F}_{<3} = \mathrm{FE{\small LEM}},\ \mathscr{F}_{<\omega} = \mathrm{PR},\ \mathscr{F}_{<\omega^\omega} = \mathrm{MPR}$$

2. A strict hierarchy: $\mathbf{F}_\beta \subsetneq \mathbf{F}^{c+1}_\beta \subsetneq \mathbf{F}_\alpha$ for all $\beta < \alpha$ and $c > 0$.

3. There exist $\mathbf{F}_\alpha$-complete problems for each $\alpha \geqslant 2$

Complexity Analysis?

When analyzing the termination algorithm, the main question is "how long can a bad sequence be?"

WQO-theory only says that a bad sequence is finite

Over $(\mathbb{N}^k, \leqslant_\times)$, one can find arbitrarily long bad sequences:

— 999, 998, ..., 1, 0

— $(2,2)$, $(2,1)$, $(2,0)$, $(1,999)$, ..., $(1,0)$, $(0,999999999)$, ...

Same over $(A^*, \leqslant_{\text{subword}})$ for $A = \{a, b, c\}$:
— aa, $bbabb$, $bbbab$, $bbbbbbba$,
$ccccbcccbccccccccbcccccccccbccccccccccbcccccccbcccccacccc$, ...

Two tricks: unbounded start element, or unbounded increase in a step

The runs of a broadcast protocol don't play these tricks!

Complexity Analysis?

When analyzing the termination algorithm, the main question is "how long can a bad sequence be?"

WQO-theory only says that a bad sequence is finite

Over $(\mathbb{N}^k, \leqslant_\times)$, one can find arbitrarily long bad sequences:

— 999, 998, . . . , 1, 0

— (2,2), (2,1), (2,0), (1,999), . . . , (1,0), (0,999999999), . . .

Same over $(A^*, \leqslant_{\text{subword}})$ for $A = \{a, b, c\}$:

— aa, $bbabb$, $bbbab$, $bbbbbbba$, $ccccbcccbcccccccbcccccccccbccccccccccbcccccccbcccccacccc$, . . .

Two tricks: unbounded start element, or unbounded increase in a step

The runs of a broadcast protocol don't play these tricks!

Complexity Analysis?

When analyzing the termination algorithm, the main question is "how long can a bad sequence be?"

WQO-theory only says that a bad sequence is finite

Over $(\mathbb{N}^k, \leqslant_\times)$, one can find arbitrarily long bad sequences:

— 999, 998, ..., 1, 0

— (2,2), (2,1), (2,0), (1,999), ..., (1,0), (0,999999999), ...

Same over $(A^*, \leqslant_{\text{subword}})$ for $A = \{a, b, c\}$:
— aa, $bbabb$, $bbbab$, $bbbbbbba$,
$ccccbcccbcccccccbcccccccccbccccccccccbcccccccbcccccacccc$, ...

Two tricks: unbounded start element, or unbounded increase in a step

The runs of a broadcast protocol don't play these tricks!

Complexity Analysis?

When analyzing the termination algorithm, the main question is "how long can a bad sequence be?"

WQO-theory only says that a bad sequence is finite

Over $(\mathbb{N}^k, \leqslant_\times)$, one can find arbitrarily long bad sequences:

— 999, 998, . . . , 1, 0

— $(2,2)$, $(2,1)$, $(2,0)$, $(1,999)$, . . . , $(1,0)$, $(0,999999999)$, . . .

Same over $(A^*, \leqslant_{\text{subword}})$ for $A = \{a,b,c\}$:
— aa, $bbabb$, $bbbab$, $bbbbbbba$, $ccccbcccbccccccccbccccccccbcccccccccbccccccccbcccccacccc$, . . .

Two tricks: unbounded start element, or unbounded increase in a step

The runs of a broadcast protocol don't play these tricks!

COMPLEXITY ANALYSIS?

When analyzing the termination algorithm, the main question is "how long can a bad sequence be?"

WQO-theory only says that a bad sequence is finite

Over $(\mathbb{N}^k, \leqslant_\times)$, one can find arbitrarily long bad sequences:

— 999, 998, ..., 1, 0

— $(2,2)$, $(2,1)$, $(2,0)$, $(1,999)$, ..., $(1,0)$, $(0,999999999)$, ...

Same over $(A^*, \leqslant_{\text{subword}})$ for $A = \{a, b, c\}$:
— aa, $bbabb$, $bbbab$, $bbbbbbba$, $ccccbcccbccccccccbcccccccccbccccccccccbcccccccbcccccacccc$, ...

Two tricks: unbounded start element, or unbounded increase in a step

The runs of a broadcast protocol don't play these tricks!

Complexity Analysis?

When analyzing the termination algorithm, the main question is "how long can a bad sequence be?"

WQO-theory only says that a bad sequence is finite

Over $(\mathbb{N}^k, \leqslant_\times)$, one can find arbitrarily long bad sequences:

— 999, 998, ..., 1, 0

— $(2,2)$, $(2,1)$, $(2,0)$, $(1,999)$, ..., $(1,0)$, $(0,999999999)$, ...

Same over $(A^*, \leqslant_{\text{subword}})$ for $A = \{a, b, c\}$:
— aa, $bbabb$, $bbbab$, $bbbbbbba$,
$ccccbcccbcccccccccbccccccccccbccccccccccbcccccccbcccccacccc$, ...

Two tricks: unbounded start element, or unbounded increase in a step

The runs of a broadcast protocol don't play these tricks!

Controlled Bad Sequences

Def. A sequence $x_0, x_1, \ldots$ is controlled $\overset{\text{def}}{\Leftrightarrow}$ $|x_i| \leqslant g^i(n_0)$ for all $i = 0, 1, \ldots$

Here the control is the pair (n_0, g) of $n_0 \in \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$.

Fact. For a fixed wqo $(A, \leqslant, |.|)$ and control (n_0, g), there is max length on controlled bad sequences (Kőnig's Lemma again)

Write $L_{g,A}(n_0)$ for this maximum length.

Satisfies well-founded recurrence:

$$L_{g,A}(n) = \max_{|x| \leqslant n} 1 + L_{g,A \smallsetminus \uparrow x}(g(n))$$

Length Function Theorems for $(\mathbb{N}^k, \leqslant_\times)$:

- If g is in $\mathscr{F}_\gamma$ for $\gamma > 0$ then $L_{g,\mathbb{N}^k}$ is in $\mathscr{F}_{\gamma+k}$
- If g is in $g \in \mathscr{F}_1$ then $L_{g,Q\times\mathbb{N}^k}$ is in $\mathbb{F}_k^{|Q|}$

CONTROLLED BAD SEQUENCES

Def. A sequence $x_0, x_1, \ldots$ is controlled $\overset{\text{def}}{\Leftrightarrow}$ $|x_i| \leqslant g^i(n_0)$ for all $i = 0, 1, \ldots$

Here the control is the pair (n_0, g) of $n_0 \in \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$.

Fact. For a fixed wqo $(A, \leqslant, |.|)$ and control (n_0, g), there is max length on controlled bad sequences (Kőnig's Lemma again)

Write $L_{g,A}(n_0)$ for this maximum length.

Satisfies well-founded recurrence:

$$L_{g,A}(n) = \max_{|x| \leqslant n} 1 + L_{g,A \smallsetminus \uparrow x}(g(n))$$

Length Function Theorems for $(\mathbb{N}^k, \leqslant_\times)$:

- If g is in $\mathscr{F}_\gamma$ for $\gamma > 0$ then $L_{g,\mathbb{N}^k}$ is in $\mathscr{F}_{\gamma+k}$
- If g is in $g \in \mathscr{F}_1$ then $L_{g,Q \times \mathbb{N}^k}$ is in $\mathbb{F}_k^{|Q|}$

Controlled Bad Sequences

Def. A sequence $x_0, x_1, \ldots$ is controlled $\overset{\text{def}}{\Leftrightarrow}$ $|x_i| \leqslant g^i(n_0)$ for all $i = 0, 1, \ldots$

Here the control is the pair (n_0, g) of $n_0 \in \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$.

Fact. For a fixed wqo $(A, \leqslant, |.|)$ and control (n_0, g), there is max length on controlled bad sequences (Kőnig's Lemma again)

Write $L_{g,A}(n_0)$ for this maximum length.

Satisfies well-founded recurrence:

$$L_{g,A}(n) = \max_{|x| \leqslant n} 1 + L_{g,A \smallsetminus \uparrow x}(g(n))$$

Length Function Theorems for $(\mathbb{N}^k, \leqslant_\times)$:

- If g is in $\mathscr{F}_\gamma$ for $\gamma > 0$ then $L_{g,\mathbb{N}^k}$ is in $\mathscr{F}_{\gamma+k}$
- If g is in $g \in \mathscr{F}_1$ then $L_{g,Q \times \mathbb{N}^k}$ is in $\mathbb{F}_k^{|Q|}$

Controlled Bad Sequences

Def. A sequence $x_0, x_1, \ldots$ is controlled $\overset{\text{def}}{\Leftrightarrow}$ $|x_i| \leqslant g^i(n_0)$ for all $i = 0, 1, \ldots$

Fact. For a fixed wqo $(A, \leqslant, |.|)$ and control (n_0, g), there is max length on controlled bad sequences (Kőnig's Lemma again)

Write $L_{g,A}(n_0)$ for this maximum length.

Satisfies well-founded recurrence:

$$L_{g,A}(n) = \max_{|x| \leqslant n} 1 + L_{g, A \smallsetminus \uparrow x}(g(n))$$

Length Function Theorems for $(\mathbb{N}^k, \leqslant_\times)$:

- If g is in $\mathscr{F}_\gamma$ for $\gamma > 0$ then $L_{g,\mathbb{N}^k}$ is in $\mathscr{F}_{\gamma+k}$
- If g is in $g \in \mathscr{F}_1$ then $L_{g,Q\times\mathbb{N}^k}$ is in $\mathbb{F}_k^{|Q|}$

Applying to Broadcast Protocols

Fact. The runs explored by the Termination algorithm are controlled with $|s_{\text{init}}|$ and $Succ : \mathbb{N} \to \mathbb{N}$.

$\Rightarrow$ Time/space bound in $\mathbf{F}_k$ for broadcast protocols with k states, and in $\mathbf{F}_\omega$ when k is not fixed.

Fact. The minimal pseudo-runs explored by the backward-chaining Coverability algorithm are controlled by $|s_{target}|$ and $Succ$.

$\Rightarrow \cdots$ *same upper bounds* $\cdots$

This is a general situation:
— WSTS model (or WQO-based algorithm) provides g
— Length function theorem provides bounds for $L_{A,g}$
— Translates as complexity upper bounds for WQO-based algorithm

APPLYING TO BROADCAST PROTOCOLS

Fact. The runs explored by the Termination algorithm are controlled with $|s_{\mathsf{init}}|$ and $Succ : \mathbb{N} \to \mathbb{N}$.

$\Rightarrow$ Time/space bound in $\mathbb{F}_k$ for broadcast protocols with k states, and in $\mathbb{F}_\omega$ when k is not fixed.

Fact. The minimal pseudo-runs explored by the backward-chaining Coverability algorithm are controlled by $|s_{\mathit{target}}|$ and $Succ$.

$\Rightarrow \cdots$ *same upper bounds* $\cdots$

This is a general situation:
— WSTS model (or WQO-based algorithm) provides g
— Length function theorem provides bounds for $L_{A,g}$
— Translates as complexity upper bounds for WQO-based algorithm

Applying to Broadcast Protocols

Fact. The runs explored by the Termination algorithm are controlled with $|s_{\text{init}}|$ and $Succ : \mathbb{N} \to \mathbb{N}$.

$\Rightarrow$ Time/space bound in $\mathbb{F}_k$ for broadcast protocols with k states, and in $\mathbb{F}_\omega$ when k is not fixed.

Fact. The minimal pseudo-runs explored by the backward-chaining Coverability algorithm are controlled by $|s_{\textit{target}}|$ and $Succ$.

$\Rightarrow \cdots$ *same upper bounds* $\cdots$

This is a general situation:
— WSTS model (or WQO-based algorithm) provides g
— Length function theorem provides bounds for $L_{A,g}$
— Translates as complexity upper bounds for WQO-based algorithm

More Length Function Theorems

For finite words with $\leqslant_{\text{subword}}$, L_{A^*} is in $\mathbb{F}_{\omega^{|A|-1}}$, and in $\mathbb{F}_{\omega^\omega}$ when alphabet is not fixed. Applies e.g. to lossy channel systems.

For sequences over $\mathbb{N}^k$ with embedding, $L_{(\mathbb{N}^k)^*}$ is in $\mathbb{F}_{\omega^{\omega^k}}$, and in $\mathbb{F}_{\omega^{\omega^\omega}}$ when k is not fixed. Applies e.g. to timed-arc Petri nets.

For finite words with priority ordering, L_{A^*} is in $\mathbb{F}_{\varepsilon_0}$ Applies e.g. to priority channel systems and higher-order LCS.

Bottom line: we have definite complexity upper bounds for WQO-based algorithms

Next course: are these tight upper bounds? how does one prove fast-growing lower bounds?

More Length Function Theorems

For finite words with $\leqslant_{\text{subword}}$, L_{A^*} is in $\mathbb{F}_{\omega^{|A|-1}}$, and in $\mathbb{F}_{\omega^\omega}$ when alphabet is not fixed. Applies e.g. to lossy channel systems.

For sequences over $\mathbb{N}^k$ with embedding, $L_{(\mathbb{N}^k)^*}$ is in $\mathbb{F}_{\omega^{\omega^k}}$, and in $\mathbb{F}_{\omega^{\omega^\omega}}$ when k is not fixed. Applies e.g. to timed-arc Petri nets.

For finite words with priority ordering, L_{A^*} is in $\mathbb{F}_{\varepsilon_0}$ Applies e.g. to priority channel systems and higher-order LCS.

Bottom line: we have definite complexity upper bounds for WQO-based algorithms

Next course: are these tight upper bounds? how does one prove fast-growing lower bounds?

More Length Function Theorems

For finite words with $\leqslant_{\text{subword}}$, L_{A^*} is in $\mathbb{F}_{\omega^{|A|-1}}$, and in $\mathbb{F}_{\omega^\omega}$ when alphabet is not fixed. Applies e.g. to lossy channel systems.

For sequences over $\mathbb{N}^k$ with embedding, $L_{(\mathbb{N}^k)^*}$ is in $\mathbb{F}_{\omega^{\omega^k}}$, and in $\mathbb{F}_{\omega^{\omega^\omega}}$ when k is not fixed. Applies e.g. to timed-arc Petri nets.

For finite words with priority ordering, L_{A^*} is in $\mathbb{F}_{\varepsilon_0}$ Applies e.g. to priority channel systems and higher-order LCS.

Bottom line: we have definite complexity upper bounds for WQO-based algorithms

Next course: are these tight upper bounds? how does one prove fast-growing lower bounds?

More Length Function Theorems

For finite words with $\leqslant_{\text{subword}}$, L_{A^*} is in $\mathbb{F}_{\omega^{|A|-1}}$, and in $\mathbb{F}_{\omega^\omega}$ when alphabet is not fixed. Applies e.g. to lossy channel systems.

For sequences over $\mathbb{N}^k$ with embedding, $L_{(\mathbb{N}^k)^*}$ is in $\mathbb{F}_{\omega^{\omega^k}}$, and in $\mathbb{F}_{\omega^{\omega^\omega}}$ when k is not fixed. Applies e.g. to timed-arc Petri nets.

For finite words with priority ordering, L_{A^*} is in $\mathbb{F}_{\varepsilon_0}$ Applies e.g. to priority channel systems and higher-order LCS.

Bottom line: we have definite complexity upper bounds for WQO-based algorithms

Next course: are these tight upper bounds? how does one prove fast-growing lower bounds?