The Power of Priority Channel Systems

C. Haase S. Schmitz Ph. Schnoebelen

LSV, ENS Cachan, France

Concur 2013, August 28th 2013

Outline

priority channel systems a model of computation

priority embedding a well quasi ordering

Contents

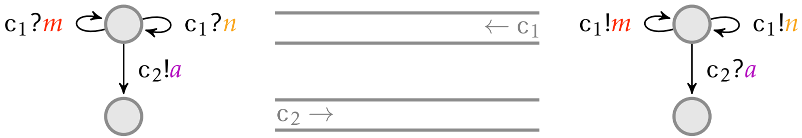
Outline

priority channel systems a model of computation

priority embedding a well quasi ordering

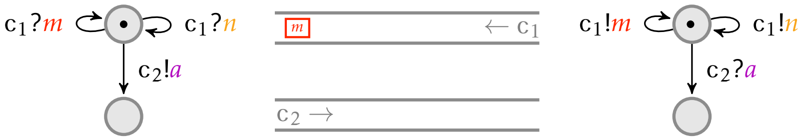
Contents

Channel Systems (CSs)



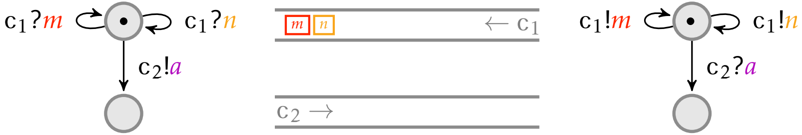
- modeling network communications
- Turing-powerful: Σ^0_1-complete

Channel Systems (CSs)



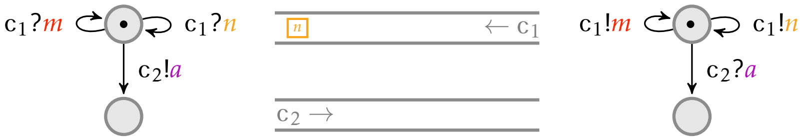
- modeling network communications
- Turing-powerful: Σ^0_1-complete

Channel Systems (CSs)



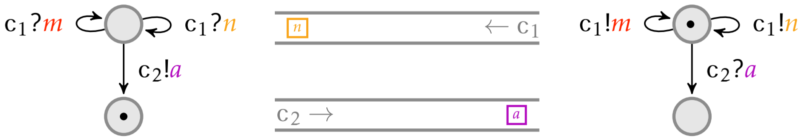
- modeling network communications
- Turing-powerful: Σ^0_1-complete

Channel Systems (CSs)



- modeling network communications
- Turing-powerful: Σ_1^0-complete

Channel Systems (CSs)



- modeling network communications
- Turing-powerful: Σ^0_1-complete

Channel Systems (CSs)

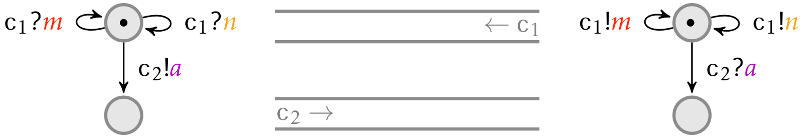
- modeling network communications
- Turing-powerful: Σ^0_1-complete

Channel Systems (CSs)

- modeling network communications
- Turing-powerful: Σ_1^0-complete

Lossy Channel Systems (LCSs)

(Abdulla and Jonsson, 1996; Cécé et al., 1996)



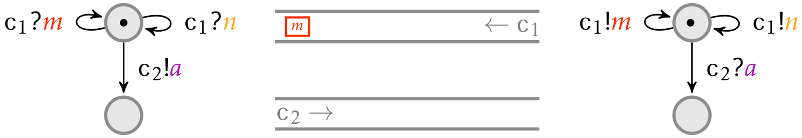
- apply losing rewriting rules to channel contents:

$$m \rightarrow_* \varepsilon \quad m \in M$$

- modeling imperfect communications, e.g. packet dropping policies against congestions
- decidable: $\mathbf{F}_{\omega^\omega}$-complete (Chambart and Schnoebelen, 2008)

Lossy Channel Systems (LCSs)

(Abdulla and Jonsson, 1996; Cécé et al., 1996)



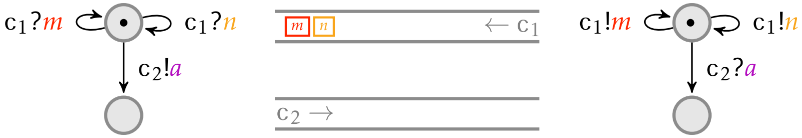
- apply losing rewriting rules to channel contents:

$$m \to_* \varepsilon \quad m \in M$$

- modeling imperfect communications, e.g. packet dropping policies against congestions
- decidable: $\mathbf{F}_{\omega^\omega}$-complete (Chambart and Schnoebelen, 2008)

Lossy Channel Systems (LCSs)

(Abdulla and Jonsson, 1996; Cécé et al., 1996)



- apply losing rewriting rules to channel contents:

$$m \to_* \varepsilon \qquad m \in M$$

- modeling imperfect communications, e.g. packet dropping policies against congestions

- decidable: $\mathbf{F}_{\omega^\omega}$-complete (Chambart and Schnoebelen, 2008)

Lossy Channel Systems (LCSs)

(Abdulla and Jonsson, 1996; Cécé et al., 1996)

- apply losing rewriting rules to channel contents:

$$m \rightarrow_* \varepsilon \qquad m \in M$$

- modeling imperfect communications, e.g. packet dropping policies against congestions
- decidable: $\mathbf{F}_{\omega^\omega}$-complete (Chambart and Schnoebelen, 2008)

Lossy Channel Systems (LCSs)

(Abdulla and Jonsson, 1996; Cécé et al., 1996)

- apply losing rewriting rules to channel contents:

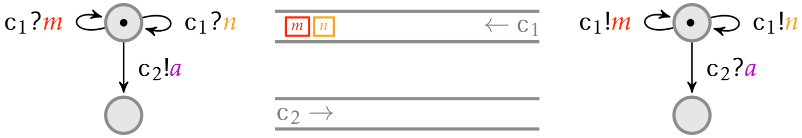
$$m \rightarrow_* \varepsilon \quad m \in M$$

- modeling imperfect communications, e.g. packet dropping policies against congestions

- decidable: $\mathbf{F}_{\omega^\omega}$-complete (Chambart and Schnoebelen, 2008)

LOSSY CHANNEL SYSTEMS (LCSs)

(ABDULLA AND JONSSON, 1996; CÉCÉ et al., 1996)

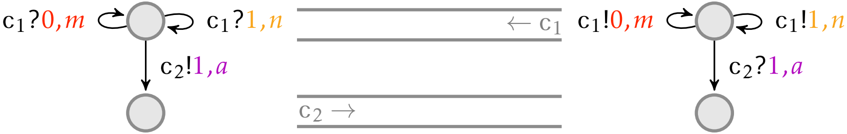


- apply losing rewriting rules to channel contents:

$$m \rightarrow_* \varepsilon \quad m \in M$$

- modeling imperfect communications, e.g. packet dropping policies against congestions
- decidable: $\mathbf{F}_{\omega^\omega}$-complete (Chambart and Schnoebelen, 2008)

Priority Channel Systems (PCSs)



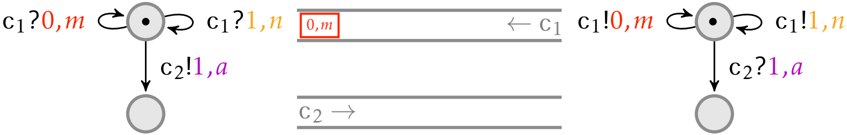
- apply superseding rewriting rules to channel contents:

$$(a,m)(b,n) \to_{\#} (b,n) \quad a \leqslant b \in \mathbb{N},\, m,n \in M$$

- modeling communications with QoS, e.g. differentiated services (RFC2475)

- decidable: $\mathbf{F}_{\varepsilon_0}$-complete

Priority Channel Systems (PCSs)

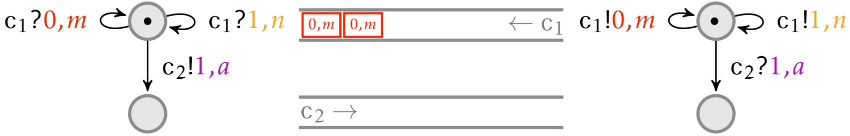


- apply superseding rewriting rules to channel contents:

$$(a,m)(b,n) \to_{\#} (b,n) \quad a \leqslant b \in \mathbb{N},\, m,n \in M$$

- modeling communications with QoS, e.g. differentiated services (RFC2475)
- decidable: $\mathbf{F}_{\varepsilon_0}$-complete

Priority Channel Systems (PCSs)

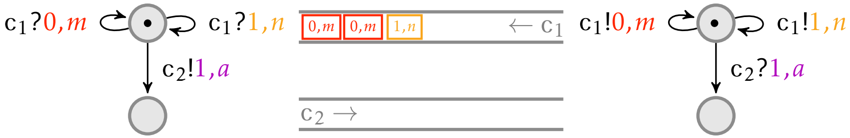


- apply superseding rewriting rules to channel contents:

$$(a,m)(b,n) \rightarrow_{\#} (b,n) \quad a \leqslant b \in \mathbb{N}, m,n \in M$$

- modeling communications with QoS, e.g. differentiated services (RFC2475)
- decidable: $\mathbf{F}_{\varepsilon_0}$-complete

Priority Channel Systems (PCSs)



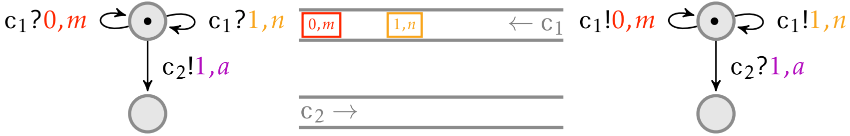
- apply superseding rewriting rules to channel contents:

$$(a,m)(b,n) \to_{\#} (b,n) \qquad a \leqslant b \in \mathbb{N},\, m,n \in M$$

- modeling communications with QoS, e.g. differentiated services (RFC2475)

- decidable: $\mathbf{F}_{\varepsilon_0}$-complete

Priority Channel Systems (PCSs)

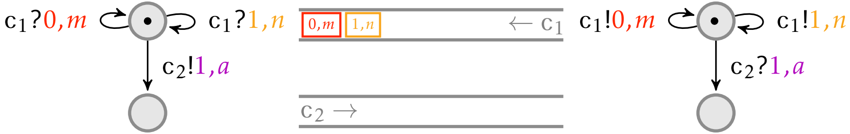


- apply superseding rewriting rules to channel contents:

$$(a,m)(b,n) \to_{\#} (b,n) \qquad a \leqslant b \in \mathbb{N},\ m,n \in M$$

- modeling communications with QoS, e.g. differentiated services (RFC2475)
- decidable: $\mathbf{F}_{\varepsilon_0}$-complete

Priority Channel Systems (PCSs)

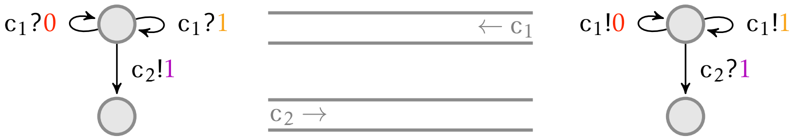


- apply superseding rewriting rules to channel contents:

$$(a,m)(b,n) \to_{\#} (b,n) \quad a \leqslant b \in \mathbb{N},\, m,n \in M$$

- modeling communications with QoS, e.g. differentiated services (RFC2475)
- decidable: $\mathbf{F}_{\varepsilon_0}$-complete

Priority Channel Systems (PCSs)



- apply superseding rewriting rules to channel contents:

$$a\,b \to_{\#} b \quad a \leqslant b \in \mathbb{N}$$

- modeling communications with QoS, e.g. differentiated services (RFC2475)

- decidable: $\mathbf{F}_{\varepsilon_0}$-complete

Remark: Alternative Models

strict superseding Turing-powerful

ordered channels with rules

$$a\,b \rightarrow_s b \quad a < b \in \mathbb{N}$$

overtaking Turing-powerful

ordered channels with rules

$$a\,b \rightarrow_o b\,a \quad a \leqslant b \in \mathbb{N}$$

priority queues decidable (VASS w. ordered 0-tests)

unordered channels, maximal priority messages read first

Remark: Alternative Models

strict superseding Turing-powerful

ordered channels with rules

$$a\,b \to_s b \quad a < b \in \mathbb{N}$$

overtaking Turing-powerful

ordered channels with rules

$$a\,b \to_o b\,a \quad a \leqslant b \in \mathbb{N}$$

priority queues decidable (VASS w. ordered 0-tests)

unordered channels, maximal priority messages read first

Losing as an Embedding

- losing rules define a quasi-ordering $\xleftarrow{*}_{*}$ over M^*
- can be restated as substring embedding:

$$x \sqsubseteq_* y \overset{\text{def}}{\Leftrightarrow} x = m_1 \cdots m_\ell, y = z_1 m_1 z_2 \cdots z_\ell m_\ell z_{\ell+1}, \forall i. z_i \in M^*$$

- examples:
  - $201 \sqsubseteq_* 22011$
  - $120 \sqsubseteq_* 10210$
  - $\forall y \in M^*. \varepsilon \sqsubseteq_* y$

SUPERSEDING AS AN EMBEDDING

- if $d \in \mathbb{N}$, write $\Sigma_d \stackrel{\text{def}}{=} \{0, \ldots, d\}$
- superseding rules define a quasi-ordering $\stackrel{*}{\leftarrow}_{\#}$ over Σ_d^*
- can be restated as priority embedding:

$$x \sqsubseteq_{\mathrm{p}} y \stackrel{\text{def}}{\Leftrightarrow} x = a_1 \cdots a_\ell, y = z_1 a_1 z_2 \cdots z_\ell a_\ell, \forall i. z_i \in \Sigma_{a_i}^*$$

- examples:
  - $201 \sqsubseteq_{\mathrm{p}} 22011$
  - $120 \not\sqsubseteq_{\mathrm{p}} 10210$
  - $\varepsilon \sqsubseteq_{\mathrm{p}} y$ iff $y = \varepsilon$

Priority Embedding is Well

c.f. related orderings of Schütte and Simpson (1985)

Definition (wqo)

A quasi-order $(A, \leqslant_A)$ is well $\overset{\text{def}}{\Leftrightarrow}$ in any infinite sequence $x_0, x_1, \ldots$ over A, there exist $i < j$ s.t. $x_i \leqslant_A x_j$.

Theorem

$(\Sigma_d^*, \sqsubseteq_p)$ *is a wqo.*

- proof by induction over d
- nested applications of Higman's Lemma

PCSs are Well-Structured

(Abdulla et al., 2000; Finkel and Schnoebelen, 2001)

For a PCS with state set Q and m channels:

transition system $(Q \times (\Sigma_d^*)^m, \rightarrow)$ with superseding steps or perfect steps

wqo $(Q \times (\Sigma_d^*)^m, \sqsubseteq_\mathrm{p})$ by Dickson's Lemma

monotonicity $\forall (p,\bar{x}),(q,\bar{x}'),(p,\bar{y}) \in Q \times (\Sigma_d^*)^m$,
if $(p,\bar{x}) \rightarrow (q,\bar{x}')$ and $\bar{x} \sqsubseteq_\mathrm{p} \bar{y}$,
then $\exists \bar{y}' \in (\Sigma_d^*)^m, \bar{y} \sqsubseteq_\mathrm{p} \bar{y}$ and $(p,\bar{y}) \rightarrow (q,\bar{y}')$.

Generic Algorithms

for Reachability, Inevitability, Simulation w. a finite-state system, etc.

PCSs are Well-Structured

(Abdulla et al., 2000; Finkel and Schnoebelen, 2001)

For a PCS with state set Q and m channels:

transition system $(Q \times (\Sigma_d^*)^m, \to)$ with superseding steps or perfect steps

wqo $(Q \times (\Sigma_d^*)^m, \sqsubseteq_\mathrm{p})$ by Dickson's Lemma

monotonicity $\forall (p, \bar{x}), (q, \bar{x}'), (p, \bar{y}) \in Q \times (\Sigma_d^*)^m$,
if $(p, \bar{x}) \to (q, \bar{x}')$ and $\bar{x} \sqsubseteq_\mathrm{p} \bar{y}$,
then $\exists \bar{y}' \in (\Sigma_d^*)^m, \bar{y} \sqsubseteq_\mathrm{p} \bar{y}$ and $(p, \bar{y}) \to (q, \bar{y}')$.

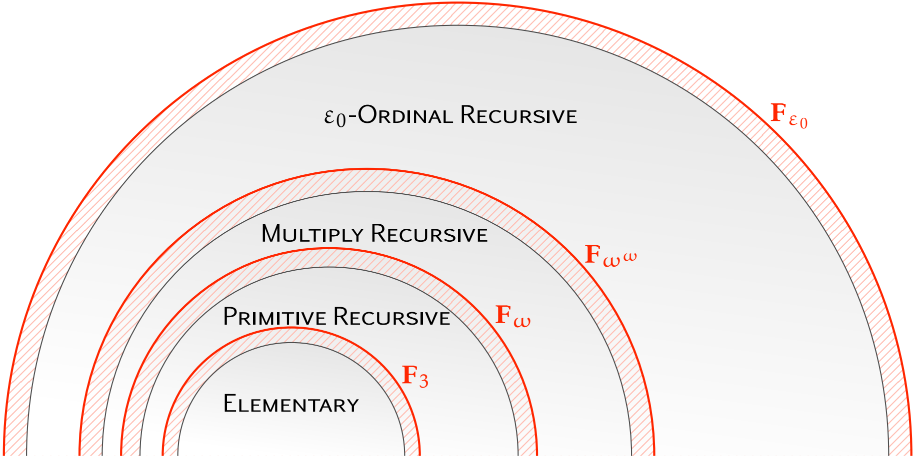
Generic Algorithms

for Reachability, Inevitability, Simulation w. a finite-state system, etc.

Fast-Growing Complexity Classes

(Schmitz and Schnoebelen, 2012)

Ordinal-indexed complexity hierarchy inside R:



Complexity of PCS Problems

Theorem

Reachability and Termination in PCSs are $\mathbf{F}_{\varepsilon_0}$-complete.

upper bound using length function theorems for applications of Higman's Lemma (Schmitz and Schnoebelen, 2011)

lower bound reduction from acceptance of a Turing machine working in $H^{\varepsilon_0}(n)$ space

Lower Bound: Hardy Functions

fundamental sequences $(\lambda(x))_x$
for limit ordinals λ in $\varepsilon_0 + 1$: $\lambda(x) < \lambda$
with $\lim_{x\to\omega} \lambda(x) = \lambda$

Example

$$\omega(x) = x + 1\,,$$
$$\omega^{\omega\cdot 2}(x) = \omega^{\omega+x+1}\,,$$
$$(\varepsilon_0)(x) = \Omega_{x+1} \stackrel{\text{def}}{=} \left.\omega^{\omega^{\cdots^{\omega}}}\right\}x+1 \text{ stacked } \omega\text{'s}$$

Lower Bound: Hardy Functions

Hardy functions $(H^\alpha)_{\alpha \leqslant \varepsilon_0}$

$$H^0(x) \stackrel{\text{def}}{=} x\,, \quad H^{\alpha+1}(x) \stackrel{\text{def}}{=} H^\alpha(x+1)\,, \quad H^\lambda(x) \stackrel{\text{def}}{=} H^{\lambda(x)}(x)\,.$$

Example

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

H^{ω^3} non elementary,

H^{ω^ω} Ackermannian,

H^{ε_0} not provably total in Peano arithmetic

Lower Bound: Hardy Computations

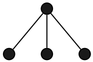
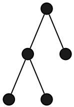
rewrite system over $(\varepsilon_0+1)\times\omega$:

$$\alpha+1, x \xrightarrow{H} \alpha, x+1$$
$$\lambda, x \xrightarrow{H} \lambda(x), x$$

computations $\alpha_0,x_0 \xrightarrow{H} \alpha_1,x_1 \xrightarrow{H} \cdots \xrightarrow{H} \alpha_n,x_n$

- preserve $H^{\alpha_i}(x_i)$
- in particular if $\alpha_n = 0$ then $x_n = H^{\alpha_0}(x_0)$

Lower Bound: Encoding Ordinals

$\alpha \in \Omega_{d+1}$	3	$\omega^2 + 1$
$t(\alpha) \in T_{d+1}$		
$s_d(\alpha) \in \Sigma_d^*$	222	1122

Proposition (Robustness)

If $s_d(\alpha) \sqsubseteq_p s_d(\beta)$, *then* $\forall x, H^\alpha(x) \leqslant H^\beta(x)$.

Lower Bound: Weak Hardy Computations

Implement Hardy steps $\alpha, n \to \beta, m$ as a PCS:

- work on string encodings: $s_d(\alpha), n \xrightarrow{H}_{\#} s_d(\beta'), m'$
- weak: $s_d(\beta') \sqsubseteq_{\mathrm{p}} s_d(\beta)$ and $m' \leqslant m$, but the perfect behaviour is possible
- also for inverse steps $s_d(\beta), m \xrightarrow{H^{-1}}_{\#} s_d(\alpha), n'$ with $s_d(\alpha') \sqsubseteq_{\mathrm{p}} s_d(\alpha)$ and $n' \leqslant n$

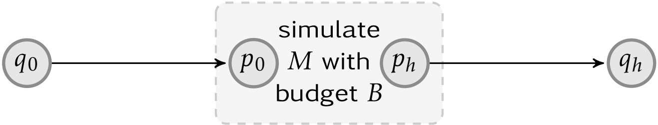
Lower Bound: Weak Hardy Computations

Implement Hardy steps $\alpha, n \to \beta, m$ as a PCS:

- work on string encodings: $s_d(\alpha), n \xrightarrow{H}_{\#} s_d(\beta'), m'$
- weak: $s_d(\beta') \sqsubseteq_{\mathrm{p}} s_d(\beta)$ and $m' \leqslant m$, but the perfect behaviour is possible
- also for inverse steps $s_d(\beta), m \xrightarrow{H^{-1}}_{\#} s_d(\alpha), n'$ with $s_d(\alpha') \sqsubseteq_{\mathrm{p}} s_d(\alpha)$ and $n' \leqslant n$

Lower Bound: Wrapping Up

reduction from a Turing machine M working in space $H^{\varepsilon_0}(n) = H^{\Omega_d}(n)$ for $d = n+1$.

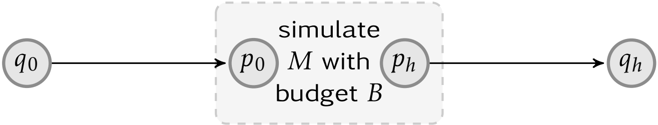


$\Omega_d, n \xrightarrow{H}_{\#} \cdots \xrightarrow{H}_{\#} 0, B \qquad 0, B' \xrightarrow{H^{-1}}_{\#} \cdots \xrightarrow{H^{-1}}_{\#} \alpha, n'$

- robustness: $H^{\Omega_d}(n) \geqslant B \geqslant B' \geqslant H^{\alpha}(n')$
- coverability: $\alpha = \Omega_d \wedge n = n'$
- implies perfect simulation

Lower Bound: Wrapping Up

reduction from a Turing machine M working in space $H^{\varepsilon_0}(n) = H^{\Omega_d}(n)$ for $d = n+1$.

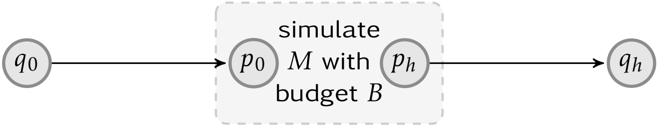


$$\Omega_d, n \xrightarrow{H}_{\#} \cdots \xrightarrow{H}_{\#} 0, B \qquad 0, B' \xrightarrow{H^{-1}}_{\#} \cdots \xrightarrow{H^{-1}}_{\#} \alpha, n'$$

- robustness: $H^{\Omega_d}(n) \geqslant B \geqslant B' \geqslant H^{\alpha}(n')$
- coverability: $\alpha = \Omega_d \wedge n = n'$
- implies perfect simulation

Lower Bound: Wrapping Up

reduction from a Turing machine M working in space $H^{\varepsilon_0}(n) = H^{\Omega_d}(n)$ for $d = n+1$.

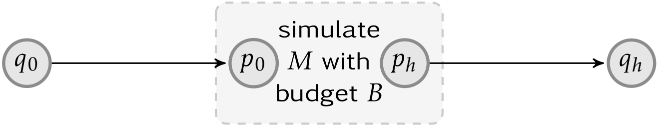


$$\Omega_d, n \xrightarrow{H}_{\#} \cdots \xrightarrow{H}_{\#} 0, B \qquad 0, B' \xrightarrow{H^{-1}}_{\#} \cdots \xrightarrow{H^{-1}}_{\#} \alpha, n'$$

- robustness: $H^{\Omega_d}(n) \geqslant B \geqslant B' \geqslant H^{\alpha}(n')$
- coverability: $\alpha = \Omega_d \wedge n = n'$
- implies perfect simulation

Lower Bound: Wrapping Up

reduction from a Turing machine M working in space $H^{\varepsilon_0}(n) = H^{\Omega_d}(n)$ for $d = n + 1$.



$$\Omega_d, n \xrightarrow{H}_{\#} \cdots \xrightarrow{H}_{\#} 0, B \qquad 0, B' \xrightarrow{H^{-1}}_{\#} \cdots \xrightarrow{H^{-1}}_{\#} \alpha, n'$$

- robustness: $H^{\Omega_d}(n) \geqslant B \geqslant B' \geqslant H^{\alpha}(n')$
- coverability: $\alpha = \Omega_d \wedge n = n'$
- implies perfect simulation

Concluding Remarks

model priority channel systems

ordering priority embedding

Perspectives

verifying PCSs regular model checking and acceleration

using PCSs reducing problems about other models (e.g. manipulating bounded depth trees and graphs)

Concluding Remarks

model priority channel systems

ordering priority embedding

Perspectives

verifying PCSs regular model checking and acceleration

using PCSs reducing problems about other models (e.g. manipulating bounded depth trees and graphs)

References

Abdulla, P.A. and Jonsson, B., 1996. Verifying programs with unreliable channels. *Inform. and Comput.*, 127(2): 91–101. doi:10.1006/inco.1996.0053.

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.

Cécé, G., Finkel, A., and Purushothaman Iyer, S., 1996. Unreliable channels are easier to verify than perfect channels. *Inform. and Comput.*, 124(1):20–31. doi:10.1006/inco.1996.0003.

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

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.

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. doi:10.1007/978-3-642-22012-8_35.

Schmitz, S. and Schnoebelen, Ph., 2012. Algorithmic aspects of WQO theory. Lecture notes. http://cel.archives-ouvertes.fr/cel-00727025.

Schütte, K. and Simpson, S.G., 1985. Ein in der reinen Zahlentheorie unbeweisbarer Satz über endliche Folgen von natürlichen Zahlen. *Arch. Math. Logic*, 25(1):75–89. doi:10.1007/BF02007558.

Fundamental Sequences

Definition (Fundamental Sequences)

For limit ordinals in $\varepsilon_0 + 1$:

$$
\begin{aligned}
(\gamma + \omega^{\beta+1})(x) &\stackrel{\text{def}}{=} \gamma + \omega^{\beta} \cdot (x+1) \\
(\gamma + \omega^{\lambda})(x) &\stackrel{\text{def}}{=} \gamma + \omega^{\lambda(x)} \\
(\varepsilon_0)(x) &\stackrel{\text{def}}{=} \Omega_{x+1} \stackrel{\text{def}}{=} \omega^{\omega^{\cdots^{\omega}}} \Big\}\, x+1 \text{ stacked } \omega\text{'s}
\end{aligned}
$$

Encodings

$s_d\colon T_{d+1} \to \Sigma_d^*$ by induction on d:

$$s_d(\bullet(t_1 \cdots t_n)) \stackrel{\text{def}}{=} \begin{cases} \varepsilon & \text{if } n = 0, \\ s_{d-1}(t_1)d \cdots s_{d-1}(t_n)d & \text{otherwise.} \end{cases}$$

$s_d\colon \Omega_{d+1} \to \Sigma_d^*$ by induction on d:

$$s_d\left(\sum_{i=1}^{n} \gamma_i\right) \stackrel{\text{def}}{=} s_d(\gamma_1) \cdots s_d(\gamma_n)\,, \quad s_d(\omega^\alpha) \stackrel{\text{def}}{=} s_{d-1}(\alpha)\, d\,.$$