Ackermann-Hardness for Lossy Counter Machines (and Reset Petri Nets)

Philippe Schnoebelen

LSV, CNRS & ENS Cachan + Oxford 1-year visitor

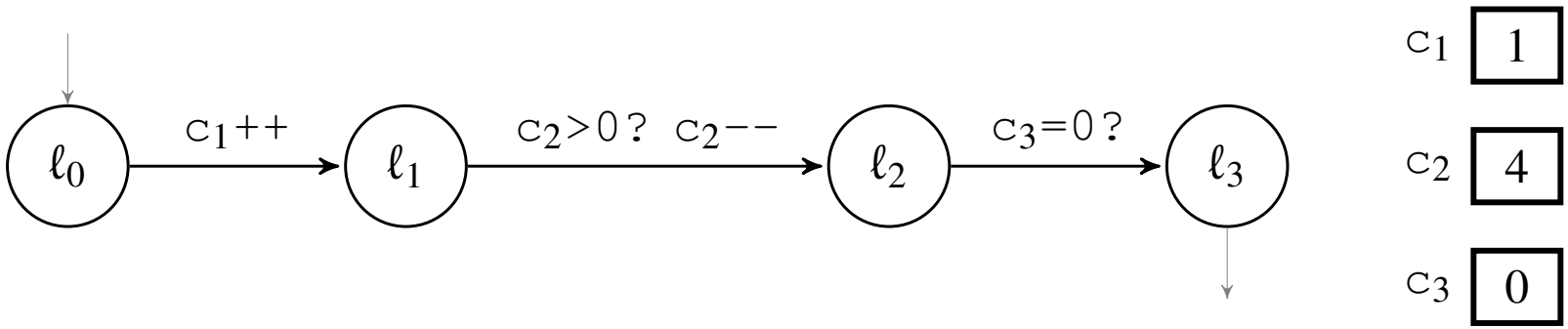
QM EECS–TCS Seminar, London, June 20th 2012

Part I:

Lossy counter machines

COUNTER MACHINES

Finite state control + finite number of "counters" (say m)
+ simple instructions and tests



Operational semantics:

– Configurations: $Conf \stackrel{\text{def}}{=} Loc \times \mathbb{N}^C = \{s,t,\ldots\}$, e.g., $s_0 = (\ell_0,1,4,0)$

– Steps: $(\ell_0,1,4,0) \to (\ell_1,2,4,0) \to (\ell_2,2,3,0) \to (\ell_3,2,3,0) \to \cdots$

A well-known model, Turing-powerful as soon as there are 2 counters

LCM = LOSSY COUNTER MACHINES

LCM = Counter machines with unreliability: "counters decrease nondeterministically" [R. Mayr, TCS 2003]

(Weaker) computational model useful, e.g., for logics like XPath or LTL+data. See decidability survey in [S., RP 2010].

Semantics. Reliable steps: $s \to_{\mathsf{rel}} t$ as above

Lossy steps: $s \to t \stackrel{\text{def}}{\Leftrightarrow} s \geqslant s' \to_{\mathsf{rel}} t' \geqslant t$ for some s' and t'
where
$s = (\ell, a_1, \ldots, a_m) \geqslant (\ell', b_1, \ldots, b_m) = s' \stackrel{\text{def}}{\Leftrightarrow} \ell = \ell' \wedge a_1 \geqslant b_1 \wedge \ldots \wedge a_m \geqslant b_m$

Prop. [Monotony] $s \stackrel{+}{\to} t$ implies $s' \stackrel{+}{\to} t'$ for all $s' \geqslant s$ and $t' \leqslant t$

NB. $(Conf, \leqslant)$ is a Well-Quasi-Ordering (a WQO) hence LCM's are well-structured

Part II:

Well-quasi-orderings and the length of bad sequences

WQO: WELL-QUASI-ORDERINGS

$(X, \leqslant)$ is a well-quasi-ordering (a WQO) if any infinite sequence $x_0, x_1, x_2 \ldots$ over X contains an increasing pair $x_i \leqslant x_j$ (for some $i < j$)

Examples.

1. $(\mathbb{N}^k, \leqslant_{\text{prod}})$ is a WQO (Dickson's Lemma)
   where, e.g., $\langle 3,2,1 \rangle \leqslant \langle 5,2,2 \rangle$ but $\langle 1,2,3 \rangle \not\leqslant \langle 5,2,2 \rangle$
2. $(\Sigma^*, \sqsubseteq)$ is a WQO (Higman's Lemma)
   where, e.g., $abc \sqsubseteq bacbc$ but $cba \not\sqsubseteq bacbc$

Many other examples: $(Conf, \leqslant)$ for LCM's, finite trees with tree embedding (Kruskal's Theorem), graphs ordered as minors (Robertson-Seymour Theorem), ..

Systems where steps are monotonic wrt a WQO on configurations, called "well-structured systems" , enjoy generic decidability results [Finkel & S., TCS 2001]

My current research program: Algorithmic aspects of WQO-theory & Complexity of WQO-based algorithms

LENGTH OF BAD SEQUENCES

Def. A sequence $x_0, x_1, \ldots$ over X is bad $\overset{\text{def}}{\Leftrightarrow}$ there is no increasing pair $x_i \leqslant x_j$ with $i < j$

Now: Over a WQO, a bad sequence is necessarily finite. Complexity upper bounds $\simeq$ "how long can a bad sequence be?"

In general, bad sequences over a given WQO can be arbitrarily long. However, controlled bad sequences cannot:

Def. $x_0, x_1, \ldots$ is (g,n)-controlled $\overset{\text{def}}{\Leftrightarrow} |x_i| \leqslant g^i(n)$

Length Function Theorems are results of the form "Any (g,n)-controlled bad sequence $x_0, x_1, \ldots, x_l$ over X has length $l \leqslant \mathbf{L}_{X,g}(n)$" for some bounding functions $\mathbf{L}_{X,g}$.

The Fast-Growing Hierarchy

A.k.a. The (Extended) Grzegorczyk Hierarchy

For $\alpha = 0, 1, 2, \ldots$ define $F_\alpha : \mathbb{N} \to \mathbb{N}$ with:

$$F_0(n) \stackrel{\text{def}}{=} n+1 \tag{D1}$$

$$F_{\alpha+1}(n) \stackrel{\text{def}}{=} F_\alpha^{n+1}(n) = \overbrace{F_\alpha(F_\alpha(\ldots F_\alpha}^{n+1 \text{ times}}(n)\ldots)) \tag{D2}$$

$$F_\omega(n) \stackrel{\text{def}}{=} F_n(n) \simeq \mathit{Ackermann}(n) \tag{D3}$$

This yields: $F_1(n) = 2n+1$, $F_2(n) = (n+1)2^{n+1} - 1$ and $F_3(n) > \left.\begin{matrix} 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \end{matrix}\right\} n$ times

F_4 is ... impossible to grasp intuitively (at least for me)

Length Function Theorem for $\mathbb{N}^k$. [LICS 2011, ICALP 2011]
For primitive-recursive g, the length of (g,n)-controlled bad sequences over $(\mathbb{N}^k, \leqslant)$ is in $\mathcal{F}_{k+O(1)}$ (and in $\mathcal{F}_k$ for small g).

Part III:

Upper bounds for LCM's

Deciding Termination for LCM's

(Non-)Termination. There is an infinite run $s_{init} = s_0 \to s_1 \to s_2 \cdots$ iff there is a loop $s_{init} = s_0 \to \cdots \to s_k \to \cdots \to s_n = s_k$

Hence termination is co-r.e. for LCM's

Furthermore. There is a loop from s_{init} iff there is a loop that is a bad sequence (until s_{n-1})

Proof. Assume a length-n loop has an increasing pair $s_i \leqslant s_j$ for $i < j < n$. Then we obtain a shorter loop by replacing $s_{j-1} \to s_j$ by $s_{j-1} \to s'_j = s_i$. Thus the shortest loop has no increasing pair

Furthermore. Since necessarily $s \to t$ implies $|t| \leqslant |s| + 1$, any run is $Succ$-controlled

Hence $n \leqslant \mathbf{L}_{A,Succ}(|s_{init}|)$ for $A \equiv Loc \times \mathbb{N}^{|C|} \equiv \mathbb{N}^m \times |Loc|$.

Cor. Termination of LCM's can be decided with complexity in $\mathcal{F}_\omega$, and in $\mathcal{F}_m$ when we fix $|C| = m$

Deciding Reachability for LCM's

Same ideas work for reachability: "is there a run from s_{init} to s_{goal}?"

Proof. if a run $s_{init} = s_0 \to s_1 \to \cdots \to s_n = s_{goal}$ has a decreasing pair $s_i \geqslant s_j$ for $0 < i < j$ it can be shortened as

$$s_0 \to \cdots \to s_{i-1} \to s_j \to \cdots \to s_n$$

Cor. If s_{goal} can be reached from s_{init}, this can be achieved via a run that is a (reversed) bad sequence

But. How is the reversed run g-controlled for some g?

Prop. In the smallest run, $|s_i| \leqslant |s_{i+1}| + 1$ for all $0 < i < n$

Cor. Reachability in LCM's can be decided with complexity in $\mathcal{F}_\omega$, or $\mathcal{F}_m$ (same as Termination)

Nb. generic technique extends to other problems/models

Part IV:

Lower Bounds

via Simulation of Fast-Growing Functions

Problem Statement

We have (rather disgusting) upper bounds on the complexity of verification for lossy counter machines.
Do we have matching lower bounds?

Answer. Unfortunately yes (see rest of this talk)

NB. We mean lower bounds on the decision problems, not just on the simple algorithms we just saw

Reduction stategy for proving lower bounds in lossy systems:

1. Compute unreliably fast-growing functions: Hardy hierarchy
2. Use this as an unreliable computational ressource
3. “Check” in the end that nothing was lost
4. Need computing unreliably the inverses of fast-growing functions

Fast-Growing vs. Hardy Hierarchy

$$F_0(n) \stackrel{\text{def}}{=} n+1 \qquad\qquad H_0(n) \stackrel{\text{def}}{=} n$$

$$F_{\alpha+1}(n) \stackrel{\text{def}}{=} F_\alpha^{n+1}(n) = \overbrace{F_\alpha(F_\alpha(\ldots F_\alpha}^{n+1 \text{ times}}(n)\ldots)) \qquad H_{\alpha+1}(n) \stackrel{\text{def}}{=} H_\alpha(n+1)$$

$$F_\lambda(n) \stackrel{\text{def}}{=} F_{\lambda_n}(n) \qquad\qquad H_\lambda(n) \stackrel{\text{def}}{=} H_{\lambda_n}(n)$$

Prop. $H_{\omega^\alpha}(n) = F_\alpha(n)$ for all α and n

Nb. $H_\alpha(n)$ can be evaluated by transforming a pair
$\alpha, n = \alpha_0, n_0 \xrightarrow{H} \alpha_1, n_1 \xrightarrow{H} \alpha_2, n_2 \xrightarrow{H} \cdots \xrightarrow{H} \alpha_k, n_k$ with $\alpha_0 > \alpha_1 > \alpha_2 > \cdots$
until eventually $\alpha_k = 0$ and $n_k = H_\alpha(n)$ % tail-recursion!!

Below we compute fast-growing functions and their inverses
by encoding $\alpha, n \xrightarrow{H} \alpha', n'$ and $\alpha', n' \xrightarrow{H^{-1}} \alpha, n$

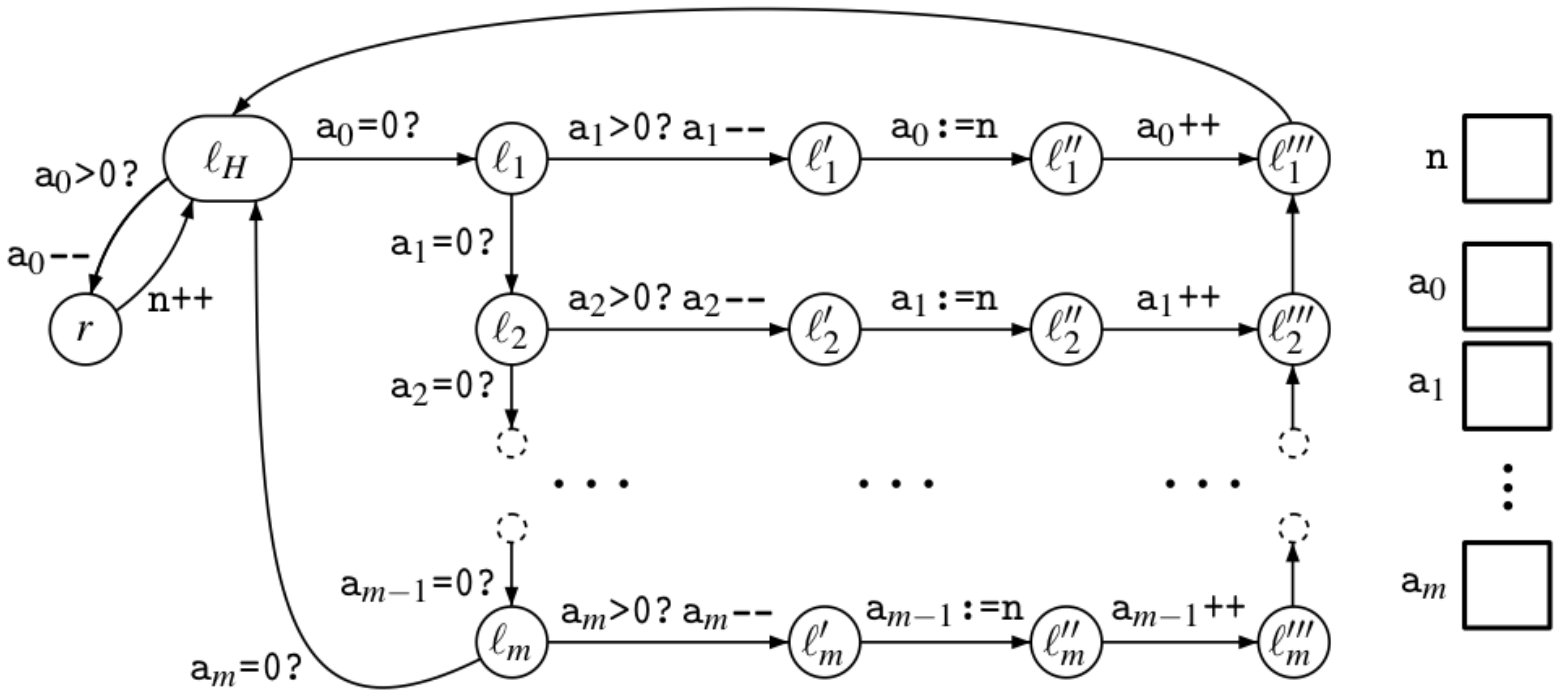
M_H: A LCM WEAKLY COMPUTING $\xrightarrow{H}$ FOR $\alpha < \omega^\omega$

Write α in CNF with coefficients $\alpha = \omega^m.a_m + \omega^{m-1}.a_{m-1} + \cdots + \omega^0 a_0$.
Encoding of α is $[a_m,\ldots,a_0] \in \mathbb{N}^{m+1}$.

$$[a_m,\ldots,a_0+1],n \xrightarrow{H} [a_m,\ldots,a_0],n+1 \qquad \%H_{\alpha+1}(n) = H_\alpha(n+1)$$

$$[a_m,\ldots,a_k+1,0,0,\ldots,0],n \xrightarrow{H} [a_m,\ldots,a_k,n+1,0,\ldots,0],n \qquad \%H_\lambda(n) = H_{\lambda_n}(n)$$

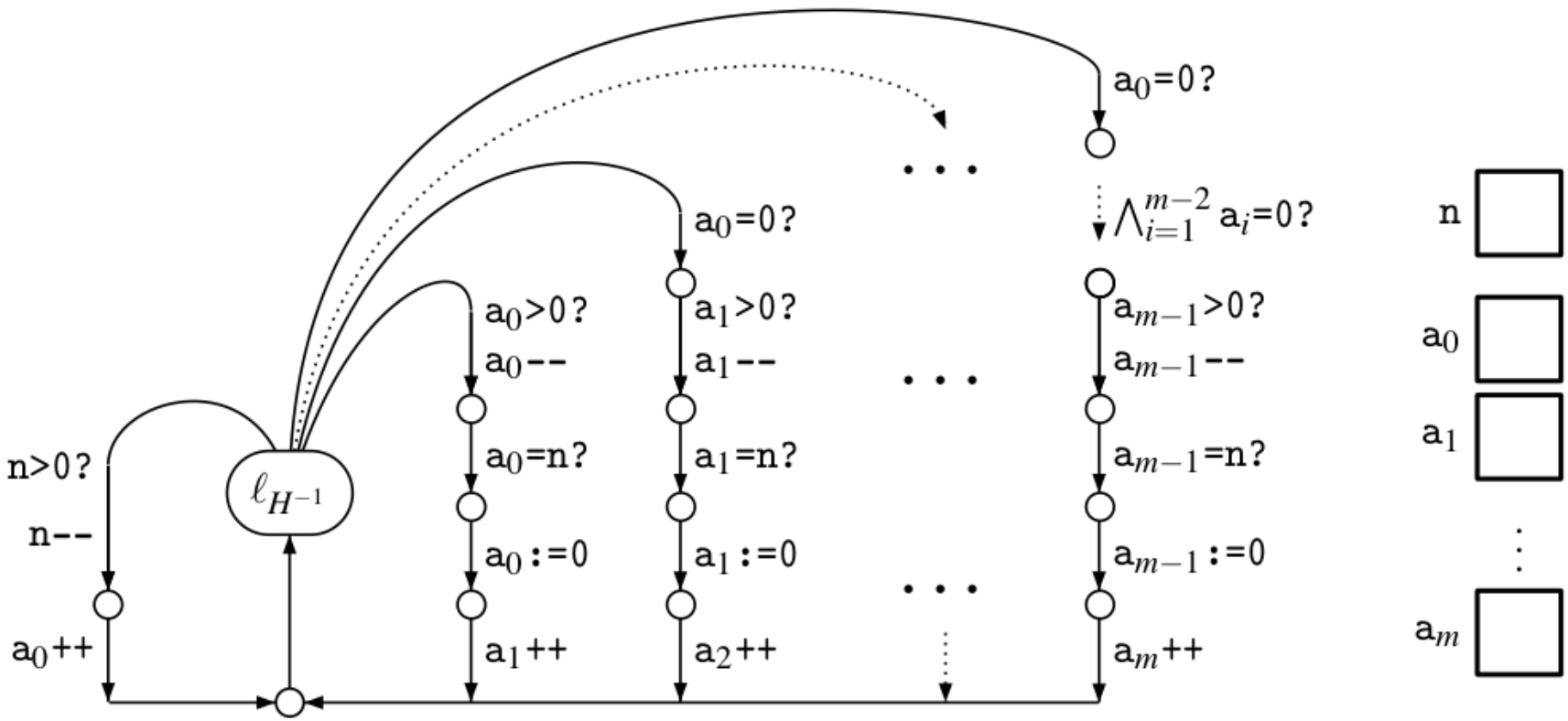
Recall $(\gamma + \omega^{k+1})_n = \gamma + \omega^k \cdot (n+1)$



$M_{H^{-1}}$: A LCM WEAKLY COMPUTING $\xrightarrow{H}{}^{-1}$ FOR $\alpha < \omega^\omega$

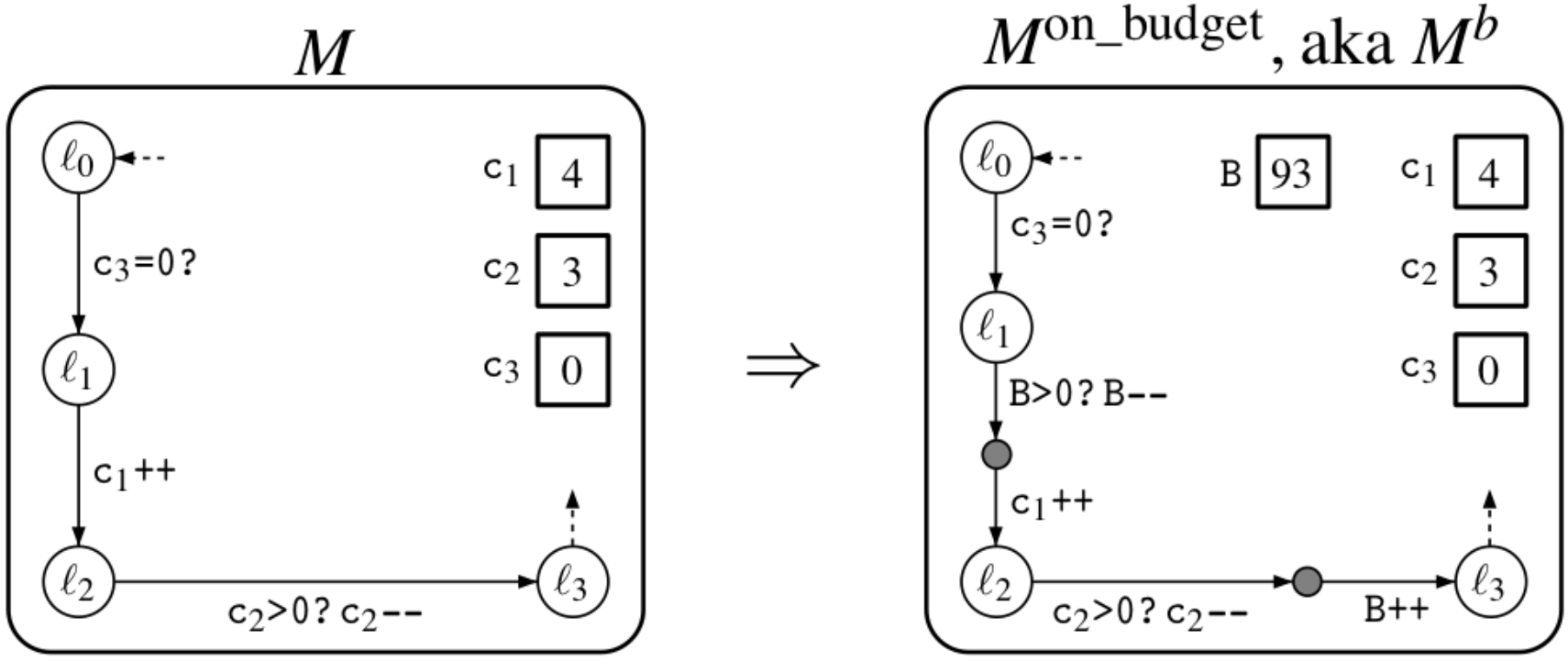
$$[a_m,\ldots,a_0],n+1 \xrightarrow{H}{}^{-1} [a_m,\ldots,a_0+1],n \qquad \%H_{\alpha+1}(n) = H_\alpha(n+1)$$

$$[a_m,\ldots,a_k,n+1,\ldots,0],n \xrightarrow{H}{}^{-1} [a_m,\ldots,a_k+1,0,\ldots,0],n \qquad \%H_\lambda(n) = H_{\lambda_n}(n)$$



Prop. [Robustness] $\mathbf{a} \leqslant \mathbf{a}'$ and $n \leqslant n'$ imply $H_{[\mathbf{a}]}(n) \leqslant H_{[\mathbf{a}']}(n')$

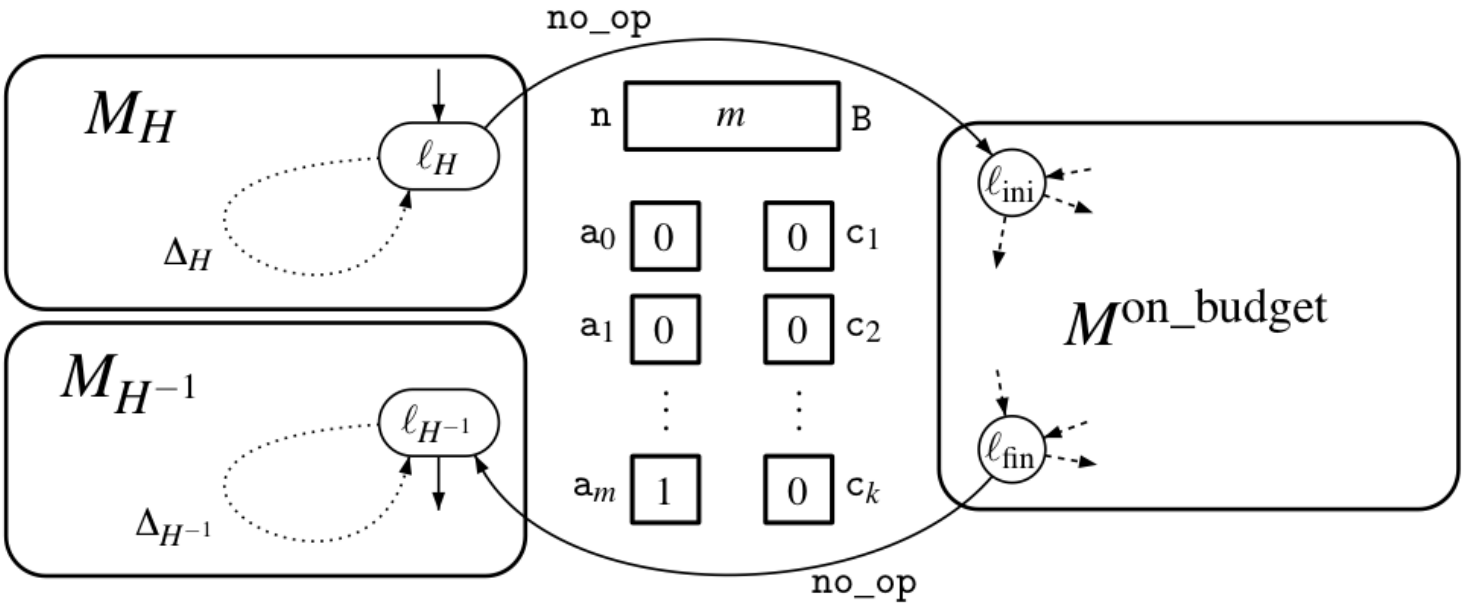
COUNTER MACHINES ON A BUDGET



Ensures:

1. $M^b \vdash (\ell, B, \mathbf{a}) \xrightarrow{*}_{\text{rel}} (\ell, B', \mathbf{a}')$ implies $B + |\mathbf{a}| = B' + |\mathbf{a}'|$
2. $M^b \vdash (\ell, B, \mathbf{a}) \xrightarrow{*}_{\text{rel}} (\ell, B', \mathbf{a}')$ implies $M \vdash (\ell, \mathbf{a}) \xrightarrow{*}_{\text{rel}} (\ell', \mathbf{a}')$
3. If $M \vdash (\ell, \mathbf{a}) \xrightarrow{*}_{\text{rel}} (\ell, \mathbf{a}')$ then $\exists B, B'$: $M^b \vdash (\ell, B, \mathbf{a}) \xrightarrow{*}_{\text{rel}} (\ell', B', \mathbf{a}')$
4. If $M^b \vdash (\ell, B, \mathbf{a}) \xrightarrow{*} (\ell, B', \mathbf{a}')$
   then $M^b \vdash (\ell, B, \mathbf{a}) \xrightarrow{*}_{\text{rel}} (\ell, B', \mathbf{a}')$ iff $B + |\mathbf{a}| = B' + |\mathbf{a}'|$

$M(m)$: Wrapping it up



Prop. $M(m)$ has a lossy run

$$(\ell_H, a_m : 1, 0, \ldots, n : m, 0, \ldots) \xrightarrow{*} (\ell_{H^{-1}}, 1, 0, \ldots, m, 0, \ldots)$$

iff $M(m)$ has a reliable run

$$(\ell_H, a_m : 1, 0, \ldots, n : m, 0, \ldots) \xrightarrow{*}_{\text{rel}} (\ell_{H^{-1}}, a_m : 1, 0, \ldots, n : m, 0, \ldots)$$

iff M has a reliable run from ℓ_{ini} to ℓ_{fin} that is bounded by $H_{\omega^m}(m)$, i.e., by *Ackermann*(m)

Cor. LCM verification is Ackermann-complete

Conclusion

Length of bad sequences is key to bounding the complexity of WQO-based algorithms

Here verification people have a lot to learn from proof-theory and combinatorics

Proving matching lower bounds is not necessarily tricky (and is easy for LCM's or Reset Petri nets) but we still lack:

— a collection of hard problems: Post Embedding Problem, ...

— a tutorial/textbook on subrecursive hierarchies (like fast-growing and Hardy hierarchies)

— a toolkit of coding tricks and lemmas for ordinals

The approach seems workable: recently we could characterize the complexity of Timed-Arc Petri nets and Data Petri Nets at $\mathcal{F}_{\omega^{\omega^\omega}}$

Bibliographical Pointers

Finkel & S., Theor.Comp.Sci. 2001: well-structured transition systems

Baier, Bertrand & S., LPAR 2006: more on well-structured transition systems (games, probabilities, ..)

Figueira, Figueira, Schmitz & S., LICS 2011: length of bad sequences over $\mathbb{N}^k$

Schmitz & S., ICALP 2011: compositional length of bad sequences

S., MFCS 2010: hardness for LCM's and related models

S., RP 2010: decidability for LCM's

Chambart & S., LICS 2008: hardness for LCS's (lossy fifo channels)

Haddad, Schmitz & S., LICS 2012: hardness for Data nets and Timed-arc Petri nets

Chambart & S., ICALP 2010: Post Embedding Problem