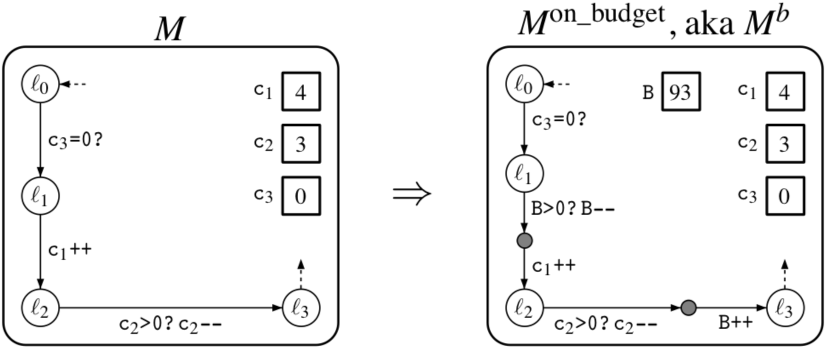
MPRI 2-9-1
"Algorithmic Aspects of WQO Theory"

Nov. 12th, 2020: Hardness of LCM verification

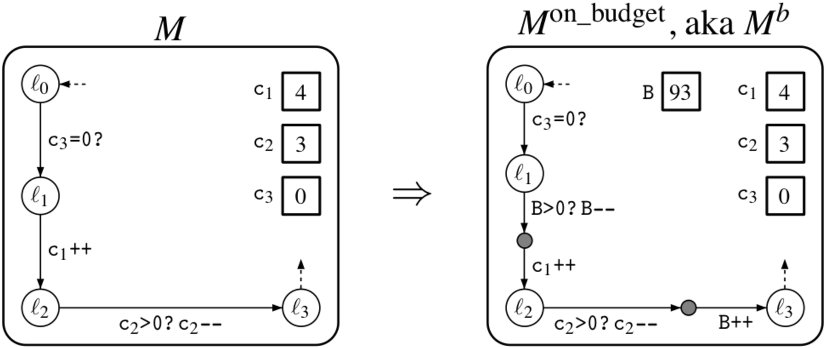
Counter machines on a budget



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 $B + |\mathbf{a}| = B' + |\mathbf{a}'|$

Counter machines on a budget



Ensures:

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

The Fast-Growing Hierarchy

For $k \in \mathbb{N}$, $F_k : \mathbb{N} \to \mathbb{N}$ is defined by:

$$F_0(n) \stackrel{\text{def}}{=} n+1,$$
$$F_{k+1}(n) \stackrel{\text{def}}{=} F_k^{n+1}(n) = \overbrace{F_k(F_k(\ldots F_k}^{n+1 \text{ times}}(n)\ldots)),$$

Yields $\begin{array}{l} F_1(n) = 2n+1 \\ F_2(n) = (n+1)2^{n+1} - 1 \end{array}$ and $F_3(n) > \left.2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\right\} n$ times.

Further ensures $F_k(n+1) > F_k(n)$ and $F_{k+1}(n) \geqslant F_k(n)$.

Every F_k is primitive-recursive. Every primitive-recursive function is dominated by some F_k.

Ackermann's function, $Ack(m) \stackrel{\text{def}}{=} F_m(m)$, is not primitive-recursive.

The Fast-Growing Hierarchy

For $k \in \mathbb{N}$, $F_k : \mathbb{N} \to \mathbb{N}$ is defined by:

$$F_0(n) \stackrel{\text{def}}{=} n+1,$$
$$F_{k+1}(n) \stackrel{\text{def}}{=} F_k^{n+1}(n) = \overbrace{F_k(F_k(\dots F_k}^{n+1 \text{ times}}(n)\dots)),$$

Yields $\begin{array}{l} F_1(n) = 2n+1 \\ F_2(n) = (n+1)2^{n+1}-1 \end{array}$ and $F_3(n) > \left.2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\right\} n$ times.

Further ensures $F_k(n+1) > F_k(n)$ and $F_{k+1}(n) \geqslant F_k(n)$.

Every F_k is primitive-recursive. Every primitive-recursive function is dominated by some F_k.

Ackermann's function, $Ack(m) \stackrel{\text{def}}{=} F_m(m)$, is not primitive-recursive.

The Fast-Growing Hierarchy

For $k \in \mathbb{N}$, $F_k : \mathbb{N} \to \mathbb{N}$ is defined by:

$$F_0(n) \stackrel{\text{def}}{=} n+1,$$
$$F_{k+1}(n) \stackrel{\text{def}}{=} F_k^{n+1}(n) = \overbrace{F_k(F_k(\ldots F_k}^{n+1 \text{ times}}(n)\ldots)),$$

Yields $\begin{array}{l} F_1(n) = 2n+1 \\ F_2(n) = (n+1)2^{n+1} - 1 \end{array}$ and $F_3(n) > \left.2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\right\} n$ times.

Further ensures $F_k(n+1) > F_k(n)$ and $F_{k+1}(n) \geqslant F_k(n)$.

Every F_k is primitive-recursive. Every primitive-recursive function is dominated by some F_k.

Ackermann's function, $Ack(m) \stackrel{\text{def}}{=} F_m(m)$, is not primitive-recursive.

Fast-Growing vs. Hardy Hierarchy

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

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

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

with λ_n given by $(\gamma+\omega^{k+1})_n \stackrel{\text{def}}{=} \gamma+\omega^k\cdot(n+1)$

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!!

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

Fast-Growing vs. Hardy Hierarchy

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

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

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

with λ_n given by $(\gamma+\omega^{k+1})_n \stackrel{\text{def}}{=} \gamma+\omega^k\cdot(n+1)$

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!!

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

Fast-Growing vs. Hardy Hierarchy

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

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

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

with λ_n given by $(\gamma+\omega^{k+1})_n \stackrel{\text{def}}{=} \gamma+\omega^k\cdot(n+1)$

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!!

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

LCM weakly computing $\xrightarrow{H}$ for $\alpha < \omega^\omega$

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

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

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

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

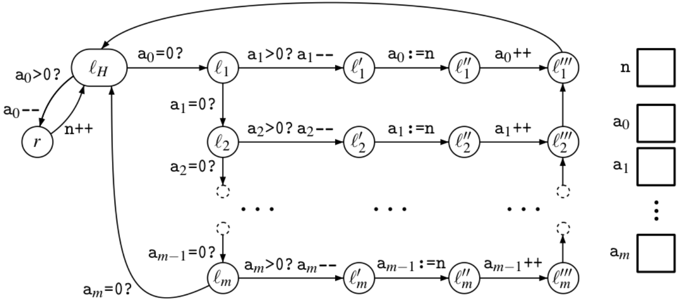
LCM WEAKLY COMPUTING $\xrightarrow{H}$ FOR $\alpha < \omega^\omega$

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

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

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

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



LCM WEAKLY COMPUTING $\overset{H}{\to}^{-1}$ FOR $\alpha < \omega^{\omega}$

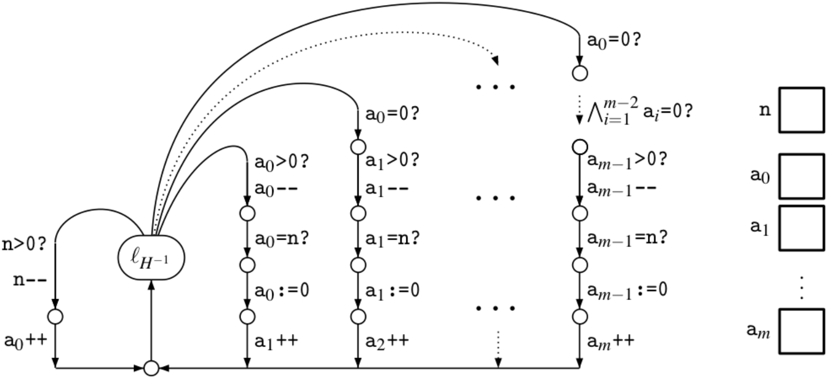
$$[a_m,\ldots,a_0],n+1 \overset{H}{\to}^{-1} [a_m,\ldots,a_0+1],n \qquad \%H_{\alpha+1}(n) = H_{\alpha}($$

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

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

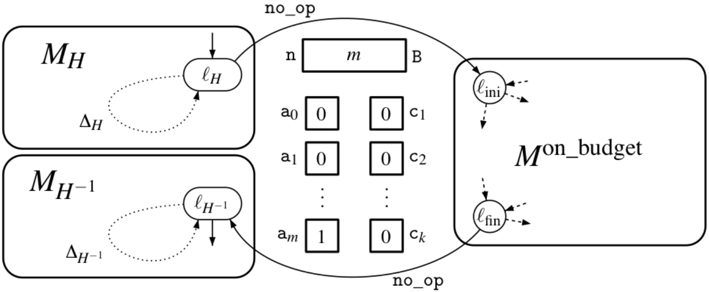
$$[a_m,\dots,a_0],n+1 \xrightarrow{H}{}^{-1} [a_m,\dots,a_0+1],n \qquad \%H_{\alpha+1}(n) = H_\alpha($$

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



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

$\mathrm{M}(\mathrm{m})$: WRAPPING IT UP



Prop. $\mathrm{M}(\mathrm{m})$ has a lossy run

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

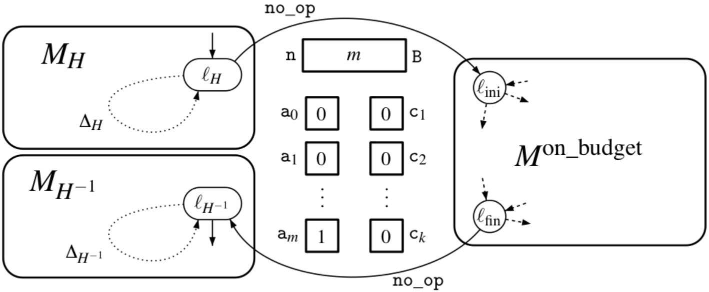
iff $\mathrm{M}(\mathrm{m})$ has a reliable run

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

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-hard

$M(m)$: WRAPPING IT UP



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

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

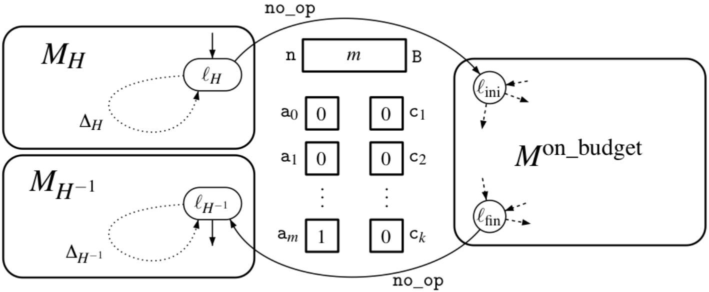
iff $M(m)$ has a reliable run

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

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-hard

$M(m)$: 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-hard