MPRI 2-9-1 "Algorithmic Aspects of WQO Theory" Nov. 12th, 2020: Hardness of LCM verification

COUNTER MACHINES ON A BUDGET



Ensures:

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

COUNTER MACHINES ON A BUDGET



Ensures:

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

THE FAST-GROWING HIERARCHY

For $k \in {\rm I\!N},$ ${\rm F}_k: {\rm I\!N} \rightarrow {\rm I\!N}$ is defined by:

$$F_{0}(n) \stackrel{\text{def}}{=} n+1, \qquad n+1 \text{ times} \\ F_{k+1}(n) \stackrel{\text{def}}{=} F_{k}^{n+1}(n) = F_{k}(F_{k}(...,F_{k}(n)...)),$$
Yields $F_{1}(n) = 2n+1 \\ F_{2}(n) = (n+1)2^{n+1}-1 \quad \text{and} \quad F_{3}(n) > 2^{2^{2}}$ h times.

Every F_k is primitive-recursive. Every primitive-recursive function

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

THE FAST-GROWING HIERARCHY

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

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

$$\begin{aligned} & \text{Yields} \quad \begin{array}{l} F_1(n) &= 2n+1 \\ F_2(n) &= (n+1)2^{n+1}-1 \end{aligned} \quad \text{and} \quad \begin{array}{l} & \\ F_3(n) &> 2^2 \end{array} \stackrel{?}{\stackrel{?}{=}} \\ & \text{Further ensures } F_k(n+1) &> F_k(n) \text{ and } F_{k+1}(n) &\geq F_k(n). \end{aligned}$$

Every F_k is primitive-recursive. Every primitive-recursive function i 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 $k \in \mathbb{N},$ $F_k: \mathbb{N} \rightarrow \mathbb{N}$ is defined by:

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

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

Further ensures $\mathsf{F}_k(\mathfrak{n}+1) > \mathsf{F}_k(\mathfrak{n})$ and $\mathsf{F}_{k+1}(\mathfrak{n}) \geqslant \mathsf{F}_k(\mathfrak{n}).$

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

$$\begin{array}{ll} F_0(n) \stackrel{\text{def}}{=} n+1 & H_0(n) \stackrel{\text{def}}{=} n \\ F_{\alpha+1}(n) \stackrel{\text{def}}{=} F_{\alpha}^{n+1}(n) = \overbrace{F_{\alpha}(F_{\alpha}(\ldots F_{\alpha}(n)\ldots))}^{n+1 \text{ times}} & H_{\alpha+1}(n) \stackrel{\text{def}}{=} n \\ F_{\lambda}(n) \stackrel{\text{def}}{=} F_{\lambda_n}(n) & H_{\lambda}(n) \stackrel{\text{def}}{=} H_{\alpha}(n+1) \\ \end{array}$$

with
$$\lambda_n$$
 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

$$\begin{split} F_0(n) &\stackrel{\text{def}}{=} n+1 \\ F_{\alpha+1}(n) &\stackrel{\text{def}}{=} F_{\alpha}^{n+1}(n) = \overbrace{F_{\alpha}(F_{\alpha}(\ldots F_{\alpha}(n)\ldots))}^{n+1 \text{ times}} & H_0(n) \stackrel{\text{def}}{=} n \\ F_{\lambda}(n) &\stackrel{\text{def}}{=} F_{\lambda_n}(n) & H_{\alpha+1}(n) \stackrel{\text{def}}{=} H_{\alpha}(n+1) \\ H_{\lambda}(n) &\stackrel{\text{def}}{=} H_{\lambda_n}(n) \end{split}$$

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

$$\begin{array}{ll} F_0(n) \stackrel{\text{def}}{=} n+1 & H_0(n) \stackrel{\text{def}}{=} n \\ F_{\alpha+1}(n) \stackrel{\text{def}}{=} F_{\alpha}^{n+1}(n) = \overbrace{F_{\alpha}(F_{\alpha}(\ldots F_{\alpha}(n)\ldots))}^{n+1 \text{ times}} & H_{\alpha+1}(n) \stackrel{\text{def}}{=} n \\ F_{\lambda}(n) \stackrel{\text{def}}{=} F_{\lambda_n}(n) & H_{\lambda}(n) \stackrel{\text{def}}{=} H_{\lambda_n}(n) \end{array}$$

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 $\alpha = \omega^m . a_m + \omega^{m-1} . a_{m-1} + \dots + \omega^0 a_0$. Encoding of α is $[a_m, \dots, a_0] \in \mathbb{N}^{m+1}$.

$$[a_{m},\ldots,a_{0}+1], n \xrightarrow{H} [a_{m},\ldots,a_{0}], n+1 \qquad \qquad \% H_{\alpha+1}(n) = F$$

 $[\mathfrak{a}_{\mathfrak{m}},\ldots,\mathfrak{a}_{k}+1,0,0,\ldots,0],\mathfrak{n} \xrightarrow{H} [\mathfrak{a}_{\mathfrak{m}},\ldots,\mathfrak{a}_{k},\mathfrak{n}+1,0,\ldots,0],\mathfrak{n} \quad \%H_{\lambda}(\mathfrak{n})=H_{\lambda_{\mathfrak{n}}}$

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 $\alpha = \omega^m . a_m + \omega^{m-1} . a_{m-1} + \dots + \omega^0 a_0$. Encoding of α is $[a_m, \dots, a_0] \in \mathbb{N}^{m+1}$.

$$[a_{\mathfrak{m}},\ldots,a_{0}+1],\mathfrak{n}\xrightarrow{H}[a_{\mathfrak{m}},\ldots,a_{0}],\mathfrak{n}+1 \qquad \qquad \% H_{\alpha+1}(\mathfrak{n})=H_{\alpha+1}(\mathfrak{n})$$

 $[a_{\mathfrak{m}},\ldots,a_{k}+1,0,0,\ldots,0], \mathfrak{n} \xrightarrow{H} [a_{\mathfrak{m}},\ldots,a_{k},\mathfrak{n}+1,0,\ldots,0], \mathfrak{n} \quad \% H_{\lambda}(\mathfrak{n}) = H_{\lambda_{\mathfrak{n}}}$

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



LCM weakly computing $\xrightarrow{H}{\rightarrow}{}^{-1}$ for $\alpha < \omega^{\omega}$

$$\begin{split} [a_{m},\ldots,a_{0}],n+1 \xrightarrow{H} {}^{-1} [a_{m},\ldots,a_{0}+1],n & & & \\ & & \\ [a_{m},\ldots,a_{k},n+1,\ldots,0],n \xrightarrow{H} {}^{-1} [a_{m},\ldots,a_{k}+1,0,\ldots,0],n & & & \\ & & & \\ & &$$

LCM weakly computing $\xrightarrow{H}{-1}$ for $\alpha < \omega^{\omega}$

$$[a_{m},...,a_{0}], n+1 \xrightarrow{H} {}^{-1} [a_{m},...,a_{0}+1], n \qquad \qquad \% H_{\alpha+1}(n) = H_{\alpha}(n)$$
$$[a_{m},...,a_{k}, n+1,...,0], n \xrightarrow{H} {}^{-1} [a_{m},...,a_{k}+1,0,...,0], n \qquad \% H_{\lambda}(n) = H_{\lambda_{n}}(n)$$



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



Prop. M(m) has a lossy run

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

iff M(m) has a reliable run

 $(\ell_H, \mathfrak{a}_{\mathfrak{m}}: 1, 0, \dots, \mathfrak{n}: \mathfrak{m}, 0, \dots) \xrightarrow{*}_{\mathsf{rel}} (\ell_{H^{-1}}, \mathfrak{a}_{\mathfrak{m}}: 1, 0, \dots, \mathfrak{n}: \mathfrak{m}, 0, \dots)$

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



Prop. M(m) has a lossy run

$$(\ell_H, \mathfrak{a}_{\mathfrak{m}} : 1, 0, \dots, \mathfrak{n} : \mathfrak{m}, 0, \dots) \xrightarrow{*} (\ell_{H^{-1}}, 1, 0, \dots, \mathfrak{m}, 0, \dots)$$

iff M(m) has a reliable run

 $(\ell_H, \mathfrak{a}_{\mathfrak{m}}: 1, 0, \dots, \mathfrak{n}: \mathfrak{m}, 0, \dots) \xrightarrow{*}_{\mathsf{rel}} (\ell_{H^{-1}}, \mathfrak{a}_{\mathfrak{m}}: 1, 0, \dots, \mathfrak{n}: \mathfrak{m}, 0, \dots)$

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



Prop. M(m) has a lossy run

$$(\ell_H, \mathfrak{a}_{\mathfrak{m}} : 1, 0, \dots, \mathfrak{n} : \mathfrak{m}, 0, \dots) \xrightarrow{*} (\ell_{H^{-1}}, 1, 0, \dots, \mathfrak{m}, 0, \dots)$$

iff M(m) has a reliable run

$$(\ell_H, \mathfrak{a}_\mathfrak{m}: 1, 0, \dots, \mathfrak{n}: \mathfrak{m}, 0, \dots) \xrightarrow{*}_{\mathsf{rel}} (\ell_{H^{-1}}, \mathfrak{a}_\mathfrak{m}: 1, 0, \dots, \mathfrak{n}: \mathfrak{m}, 0, \dots)$$

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