type-two polynomial time and restriced lookahead

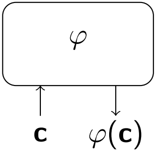
Bruce Kapron and **Florian Steinberg**

INRIA

July 5, 2018

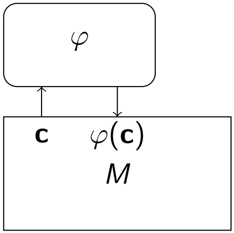
Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$



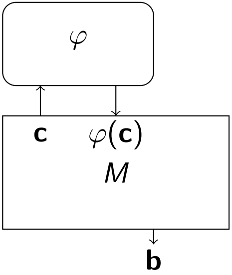
Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$



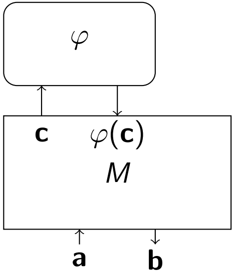
Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$



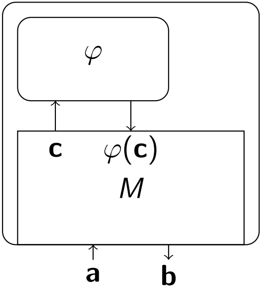
Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$



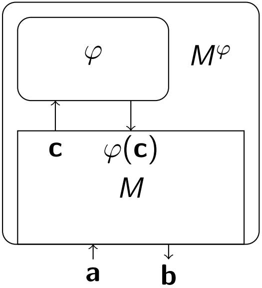
Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$



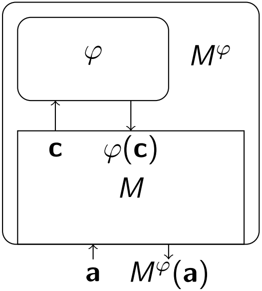
Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$



Oracle Turing machines

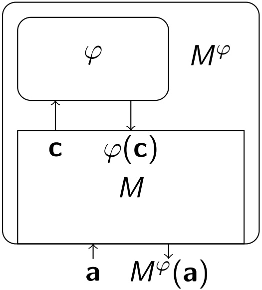
$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$



Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$

$$F : \subseteq \mathcal{B} \to \mathcal{B}, \quad \varphi \mapsto M^\varphi$$

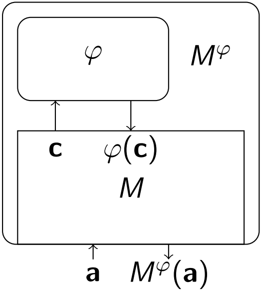


Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$

$$F :\subseteq \mathcal{B} \to \mathcal{B}, \quad \varphi \mapsto M^\varphi$$

$$\mathrm{time}_M(\varphi, \mathbf{a})$$

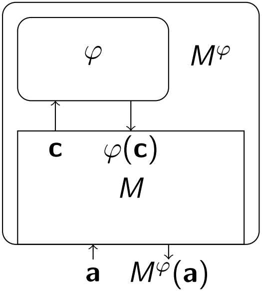


Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$

$$F :\subseteq \mathcal{B} \to \mathcal{B}, \quad \varphi \mapsto M^\varphi$$

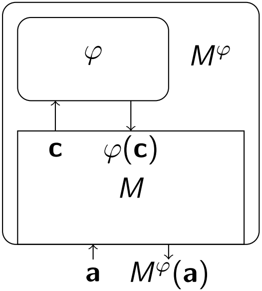
$$\mathrm{time}_M(\varphi, \mathbf{a}) \le T(\dots, |\mathbf{a}|)$$



Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$

$$F :\subseteq \mathcal{B} \to \mathcal{B}, \quad \varphi \mapsto M^\varphi$$



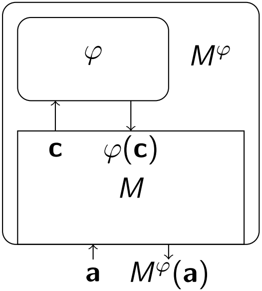
$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq T(\ldots, |\mathbf{a}|)$$

$$|\varphi|(n) := \max_{|\mathbf{a}| \leq n} |\varphi(\mathbf{a})|,$$

Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$

$$F :\subseteq \mathcal{B} \to \mathcal{B}, \quad \varphi \mapsto M^\varphi$$



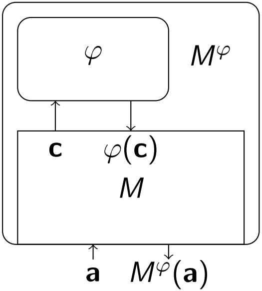
$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq T(|\varphi|, |\mathbf{a}|)$$

$$|\varphi|(n) := \max_{|\mathbf{a}| \leq n} |\varphi(\mathbf{a})|,$$

Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$

$$F :\subseteq \mathcal{B} \to \mathcal{B}, \quad \varphi \mapsto M^\varphi$$



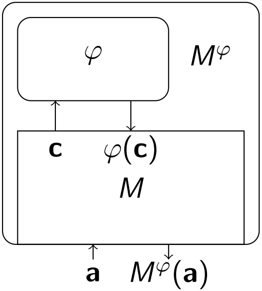
$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq T(|\varphi|, |\mathbf{a}|)$$

$$|\varphi|(n) := \max_{|\mathbf{a}| \leq n} |\varphi(\mathbf{a})|, \quad |\varphi| : \mathbb{N} \to \mathbb{N}$$

Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$

$$F :\subseteq \mathcal{B} \to \mathcal{B}, \quad \varphi \mapsto M^\varphi$$



$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq T(|\varphi|, |\mathbf{a}|)$$

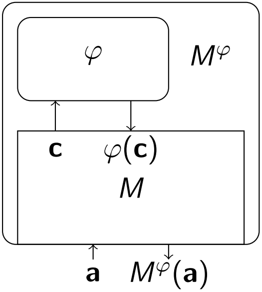
$$|\varphi|(n) := \max_{|\mathbf{a}| \leq n} |\varphi(\mathbf{a})|, \quad |\varphi| : \mathbb{N} \to \mathbb{N}$$

Second-order polynomials...

Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$

$$F : \subseteq \mathcal{B} \to \mathcal{B}, \quad \varphi \mapsto M^\varphi$$



$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq T(|\varphi|, |\mathbf{a}|)$$

$$|\varphi|(n) := \max_{|\mathbf{a}| \leq n} |\varphi(\mathbf{a})|, \quad |\varphi| : \mathbb{N} \to \mathbb{N}$$

Second-order polynomials...

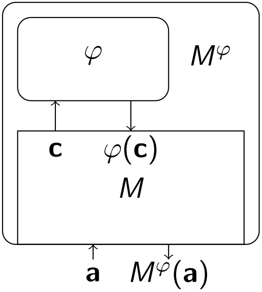
Theorem

Closed under composition.

Oracle Turing machines

$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$

$$F : \subseteq \mathcal{B} \to \mathcal{B}, \quad \varphi \mapsto M^\varphi$$



$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq T(|\varphi|, |\mathbf{a}|)$$

$$|\varphi|(n) := \max_{|\mathbf{a}| \leq n} |\varphi(\mathbf{a})|, \quad |\varphi| : \mathbb{N} \to \mathbb{N}$$

Second-order polynomials...

Theorem

Closed under composition.

Corollary

Preserves polynomial-time computability.

Comments

Original definition: bounded recursion scheme.

Comments

Original definition: bounded recursion scheme.

Theorem (Cook, Kapron, Urquart)

Coincides with the lambda closure of the polytime functions and a limited recursion operator.

Comments

Original definition: bounded recursion scheme.

Theorem (Cook, Kapron, Urquart)

Coincides with the lambda closure of the polytime functions and a limited recursion operator.

Be carefull with resource bounded machines!

Comments

Original definition: bounded recursion scheme.

Theorem (Cook, Kapron, Urquart)

Coincides with the lambda closure of the polytime functions and a limited recursion operator.

Be carefull with resource bounded machines!

Theorem

The length function is not polytime.

Comments

Original definition: bounded recursion scheme.

Theorem (Cook, Kapron, Urquart)

Coincides with the lambda closure of the polytime functions and a limited recursion operator.

Be carefull with resource bounded machines!

Theorem

The length function is not polytime.

- Can not evaluate running-times.

Comments

Original definition: bounded recursion scheme.

Theorem (Cook, Kapron, Urquart)

Coincides with the lambda closure of the polytime functions and a limited recursion operator.

Be carefull with resource bounded machines!

Theorem

The length function is not polytime.

- Can not evaluate running-times.
- Clockability.

Comments

Original definition: bounded recursion scheme.

Theorem (Cook, Kapron, Urquart)

Coincides with the lambda closure of the polytime functions and a limited recursion operator.

Be carefull with resource bounded machines!

Theorem

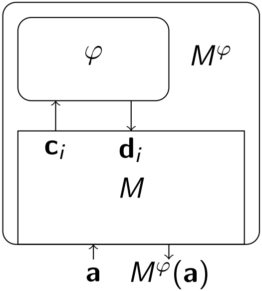
The length function is not polytime.

- Can not evaluate running-times.
- Clockability.

Theorem (Kawamura, S.)

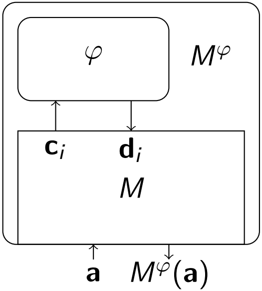
Clocking is impossible.

Oracle polynomial-time



$t : \mathbb{N} \to \mathbb{N}$ is step-count

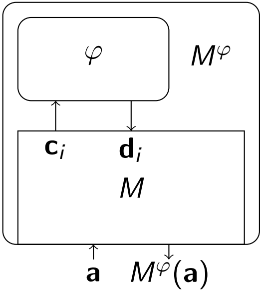
Oracle polynomial-time



$t : \mathbb{N} \to \mathbb{N}$ is step-count if

$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq t(m_{\varphi,\mathbf{a}}),$$

Oracle polynomial-time



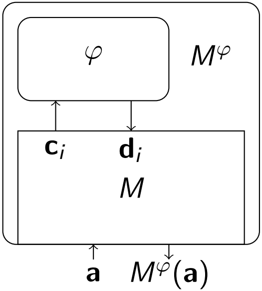
$t : \mathbb{N} \to \mathbb{N}$ is step-count if

$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq t(m_{\varphi,\mathbf{a}}),$$

where

$$m_{\varphi,\mathbf{a}} := \max\{|\mathbf{a}|, |\mathbf{d}_i|\}.$$

Oracle polynomial-time



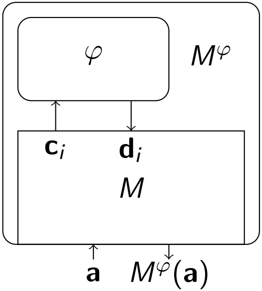
$t : \mathbb{N} \to \mathbb{N}$ is step-count if

$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq t(m_{\varphi,\mathbf{a}}),$$

where

$$m_{\varphi,\mathbf{a}} := \max\{|\mathbf{a}|, |\mathbf{d}_i|\}.$$

Oracle polynomial-time



$t : \mathbb{N} \to \mathbb{N}$ is step-count if

$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq t(m_{\varphi,\mathbf{a}}),$$

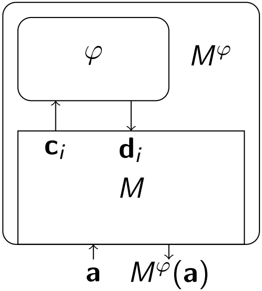
where

$$m_{\varphi,\mathbf{a}} := \max\{|\mathbf{a}|, |\mathbf{d}_i|\}.$$

Lemma

Step-count condition holds in each step.

Oracle polynomial-time



$t : \mathbb{N} \to \mathbb{N}$ is step-count if

$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq t(m_{\varphi,\mathbf{a}}),$$

where

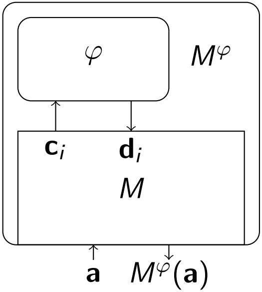
$$m_{\varphi,\mathbf{a}} := \max\{|\mathbf{a}|, |\mathbf{d}_i|\}.$$

Lemma

Step-count condition holds in each step.

because: total.

Oracle polynomial-time



$t : \mathbb{N} \to \mathbb{N}$ is step-count if

$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq t(m_{\varphi,\mathbf{a}}),$$

where

$$m_{\varphi,\mathbf{a}} := \max\{|\mathbf{a}|, |\mathbf{d}_i|\}.$$

Theorem

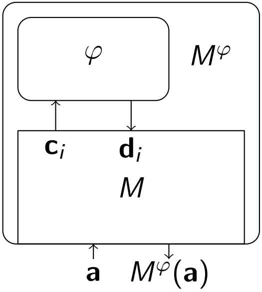
Total operators have step-counts.

Lemma

Step-count condition holds in each step.

because: total.

Oracle polynomial-time



$t : \mathbb{N} \to \mathbb{N}$ is step-count if

$$\mathrm{time}_M(\varphi, \mathbf{a}) \leq t(m_{\varphi,\mathbf{a}}),$$

where

$$m_{\varphi,\mathbf{a}} := \max\{|\mathbf{a}|, |\mathbf{d}_i|\}.$$

Theorem

Total operators have step-counts.

Lemma

Step-count condition holds in each step.

because: total.

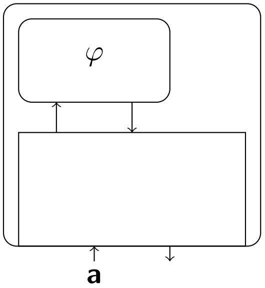
Theorem

M runs in time $P \rightsquigarrow n \mapsto P(l_n, n)$ is step-count.

Opt but not pt

Example

Consider the operator $F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.

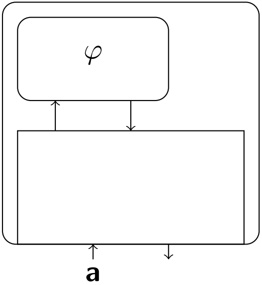


$$\mathrm{time}(\varphi, \mathbf{a}) \qquad\qquad .$$

Opt but not pt

Example

Consider the operator $F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.

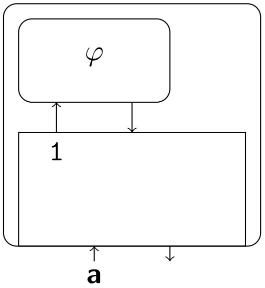


$\mathrm{time}(\varphi, \mathbf{a})$.

Opt but not pt

Example

Consider the operator
$F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.

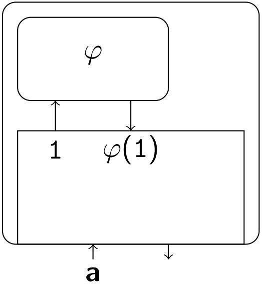


$\mathrm{time}(\varphi, \mathbf{a})$.

Opt but not pt

Example

Consider the operator $F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.

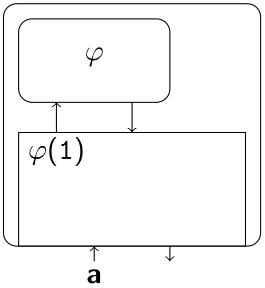


$\mathrm{time}(\varphi, \mathbf{a})$.

Opt but not pt

Example

Consider the operator $F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.

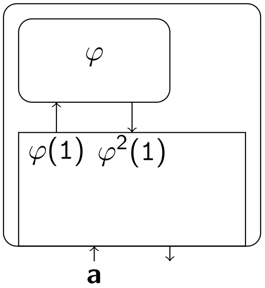


$$\mathrm{time}(\varphi, \mathbf{a}) \qquad .$$

Opt but not pt

Example

Consider the operator
$F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.

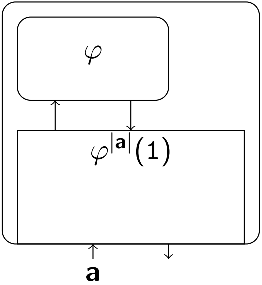


$\mathrm{time}(\varphi, \mathbf{a})$.

Opt but not pt

Example

Consider the operator
$F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.

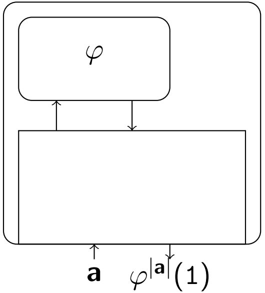


$\mathrm{time}(\varphi, \mathbf{a})$.

Opt but not pt

Example

Consider the operator $F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.

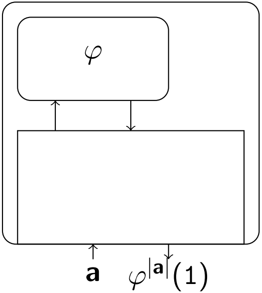


$$\mathrm{time}(\varphi, \mathbf{a}) \qquad .$$

Opt but not pt

Example

Consider the operator $F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.



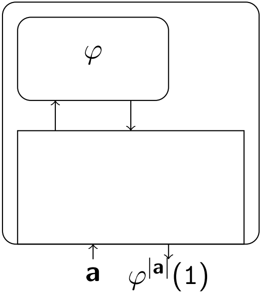
$$\mathrm{time}(\varphi, \mathbf{a}) = \mathcal{O}(m^2_{\varphi,\mathbf{a}}).$$

Opt but not pt

Example

Consider the operator $F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.

Consider $\varphi(\mathbf{a}) := \mathbf{aa}$.



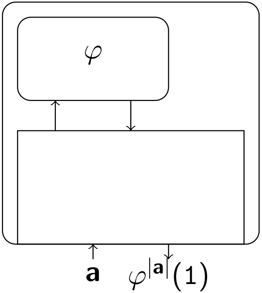
$$\mathrm{time}(\varphi, \mathbf{a}) = \mathcal{O}(m_{\varphi,\mathbf{a}}^2).$$

Opt but not pt

Example

Consider the operator $F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.

Consider $\varphi(\mathbf{a}) := \mathbf{aa}$.
Then $|F(\varphi)| \simeq 2^{|\mathbf{a}|}$.



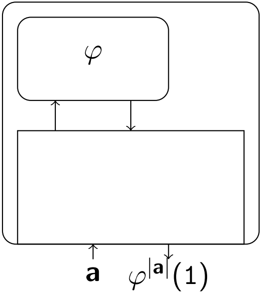
$$\mathrm{time}(\varphi, \mathbf{a}) = \mathcal{O}(m_{\varphi,\mathbf{a}}^2).$$

Opt but not pt

Example

Consider the operator $F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.

Consider $\varphi(\mathbf{a}) := \mathbf{aa}$.
Then $|F(\varphi)| \simeq 2^{|\mathbf{a}|}$.
$\rightsquigarrow$ F not polytime.



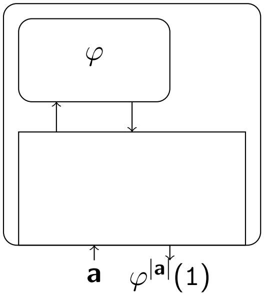
$$\mathrm{time}(\varphi, \mathbf{a}) = \mathcal{O}(m^2_{\varphi,\mathbf{a}}).$$

Opt but not pt

Example

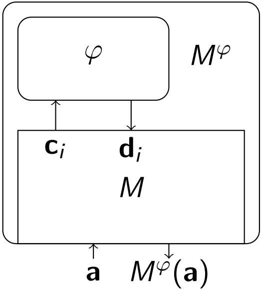
Consider the operator $F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1)$.

Consider $\varphi(\mathbf{a}) := \mathbf{aa}$.
Then $|F(\varphi)| \simeq 2^{|\mathbf{a}|}$.
$\rightsquigarrow$ F not polytime.
Essentially iteration operator.

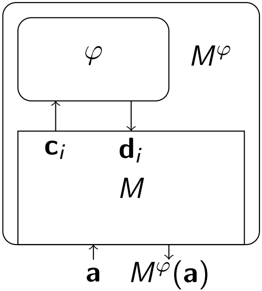


$$\mathrm{time}(\varphi, \mathbf{a}) = \mathcal{O}(m^2_{\varphi,\mathbf{a}}).$$

length and lookahead revisions



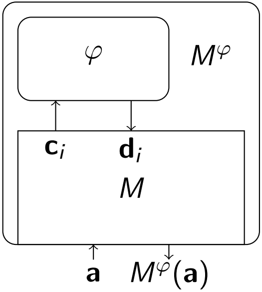
length and lookahead revisions



Definition

- finite length revision (spt).

length and lookahead revisions



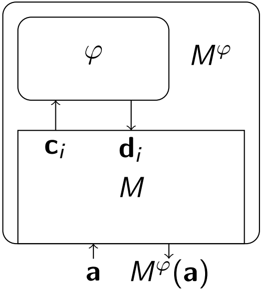
Definition

- finite length revision (spt).

polynomial r.t.:

$$(l, n) \mapsto (p \circ l)^r(p(n)) + p(n).$$

length and lookahead revisions



Definition

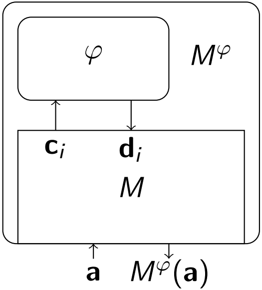
- finite length revision (spt).

polynomial r.t.:

$$(l, n) \mapsto (p \circ l)^r(p(n)) + p(n).$$

- First query can not be bigger than $p(|\mathbf{a}|)$... etc.

length and lookahead revisions



Definition

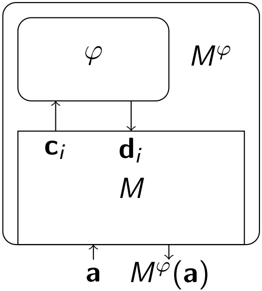
- finite length revision (spt).
- finite lookahead revision (mpt).

polynomial r.t.:

$$(l, n) \mapsto (p \circ l)^r(p(n)) + p(n).$$

- First query can not be bigger than $p(|\mathbf{a}|)$... etc.

length and lookahead revisions



Definition

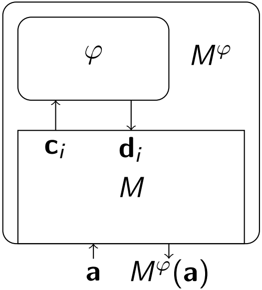
- finite length revision (spt).
- finite lookahead revision (mpt).

In both cases polynomial r.t.:

$$(l, n) \mapsto (p \circ l)^r(p(n)) + p(n).$$

- First query can not be bigger than $p(|\mathbf{a}|)$... etc.

length and lookahead revisions



Definition

- finite length revision (spt).
- finite lookahead revision (mpt).

In both cases polynomial r.t.:

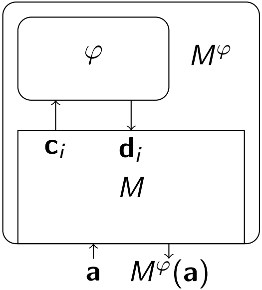
$$(l, n) \mapsto (p \circ l)^r(p(n)) + p(n).$$

Lemma

Finite length revision $\rightsquigarrow$ finite lookahead revision.

- First query can not be bigger than $p(|\mathbf{a}|)$... etc.

length and lookahead revisions



Definition

- finite length revision (spt).
- finite lookahead revision (mpt).

In both cases polynomial r.t.:

$$(l, n) \mapsto (p \circ l)^r(p(n)) + p(n).$$

Lemma

Finite length revision $\rightsquigarrow$ finite lookahead revision.

- First query can not be bigger than $p(|\mathbf{a}|)$... etc.
- Have to modify machines.

Examples

Examples

- $F(\varphi)(\mathbf{a}) := \max\{|\varphi(\mathbf{b})| \mid \mathbf{b} \subseteq \mathbf{a}\}$

Examples

Examples

- $F(\varphi)(\mathbf{a}) := \max\{|\varphi(\mathbf{b})| \mid \mathbf{b} \subseteq \mathbf{a}\}$ is polytime

Examples

Examples

- $F(\varphi)(\mathbf{a}) := \max\{|\varphi(\mathbf{b})| \mid \mathbf{b} \subseteq \mathbf{a}\}$ is polytime has finite lookahead revision

Examples

Examples

- $F(\varphi)(\mathbf{a}) := \max\{|\varphi(\mathbf{b})| \mid \mathbf{b} \subseteq \mathbf{a}\}$ is polytime has finite lookahead revision but no finite length revision.

Examples

Examples

- $F(\varphi)(\mathbf{a}) := \max\{|\varphi(\mathbf{b})| \mid \mathbf{b} \subseteq \mathbf{a}\}$ is polytime has finite lookahead revision but no finite length revision.
- Iteration: no finite lookahead revision

Examples

Examples

- $F(\varphi)(\mathbf{a}) := \max\{|\varphi(\mathbf{b})| \mid \mathbf{b} \subseteq \mathbf{a}\}$ is polytime has finite lookahead revision but no finite length revision.
- Iteration: no finite lookahead revision but not polytime.

Examples

Examples

- $F(\varphi)(\mathbf{a}) := \max\{|\varphi(\mathbf{b})| \mid \mathbf{b} \subseteq \mathbf{a}\}$ is polytime has finite lookahead revision but no finite length revision.
- Iteration: no finite lookahead revision but not polytime.
- There is an operator F that is polytime but not mpt.

Examples

Examples

- $F(\varphi)(\mathbf{a}) := \max\{|\varphi(\mathbf{b})| \mid \mathbf{b} \subseteq \mathbf{a}\}$ is polytime has finite lookahead revision but no finite length revision.
- Iteration: no finite lookahead revision but not polytime.
- There is an operator F that is polytime but not mpt. First define functionals F_i by

$$F_0(\varphi) := \epsilon \quad \text{and} \quad F_{n+1}(\varphi) := (\varphi \circ \varphi)(F_n(\varphi))^{\leq|\varphi(\epsilon)|}.$$

Examples

Examples

- $F(\varphi)(\mathbf{a}) := \max\{|\varphi(\mathbf{b})| \mid \mathbf{b} \subseteq \mathbf{a}\}$ is polytime has finite lookahead revision but no finite length revision.
- Iteration: no finite lookahead revision but not polytime.
- There is an operator F that is polytime but not mpt. First define functionals F_i by

$$F_0(\varphi) := \epsilon \quad \text{and} \quad F_{n+1}(\varphi) := (\varphi \circ \varphi)(F_n(\varphi))^{\leq |\varphi(\epsilon)|}.$$

Then set

$$F(\varphi)(\mathbf{a}) := F_{|\mathbf{a}|}(\varphi).$$

Examples

Examples

- $F(\varphi)(\mathbf{a}) := \max\{|\varphi(\mathbf{b})| \mid \mathbf{b} \subseteq \mathbf{a}\}$ is polytime has finite lookahead revision but no finite length revision.
- Iteration: no finite lookahead revision but not polytime.
- There is an operator F that is polytime but not mpt. First define functionals F_i by

$$F_0(\varphi) := \epsilon \quad \text{and} \quad F_{n+1}(\varphi) := (\varphi \circ \varphi)(F_n(\varphi))^{\leq |\varphi(\epsilon)|}.$$

  Then set

$$F(\varphi)(\mathbf{a}) := F_{|\mathbf{a}|}(\varphi).$$

  This operator is polytime but not mpt.

The recursion operator

Theorem (Cook, Urquart)

The feasible functionals of type-two are exactly those realized by lambda terms with polytime functions and $\mathcal{R}$ as constants.

The recursion operator

Theorem (Cook, Urquart)

The feasible functionals of type-two are exactly those realized by lambda terms with polytime functions and $\mathcal{R}$ as constants.

Let $\mathcal{R}$ be defined by $\mathcal{R}(\varphi, \mathbf{a}, \psi, \epsilon) := \mathbf{a}$ and

$$\mathcal{R}(\varphi, \mathbf{a}, \psi, \mathbf{c}i) := \varphi(\mathbf{c}i, \mathcal{R}(\varphi, \mathbf{a}, \psi, \mathbf{c})) \text{ if smaller than } |\psi(\mathbf{c}i)|.$$

Lemma (Kapron, S.)

$\mathcal{R}$ is mpt.

The recursion operator

Theorem (Cook, Urquart)

The feasible functionals of type-two are exactly those realized by lambda terms with polytime functions and $\mathcal{R}$ as constants.

Let $\mathcal{R}$ be defined by $\mathcal{R}(\varphi, \mathbf{a}, \psi, \epsilon) := \mathbf{a}$ and

$$\mathcal{R}(\varphi, \mathbf{a}, \psi, \mathbf{c}i) := \varphi(\mathbf{c}i, \mathcal{R}(\varphi, \mathbf{a}, \psi, \mathbf{c})) \text{ if smaller than } |\psi(\mathbf{c}i)|.$$

Lemma (Kapron, S.)

$\mathcal{R}$ is mpt.

Theorem

The lambda closure of mpt are the feasible functionals.

A decomposition result

Theorem (Kapron, S.)

Every mpt operator can be decomposed into two spt operators.

A decomposition result

Theorem (Kapron, S.)

Every mpt operator can be decomposed into two spt operators.

Example

Decomposition of the maximization operator.

A decomposition result

Theorem (Kapron, S.)

Every mpt operator can be decomposed into two spt operators.

Example

Decomposition of the maximization operator.

Corollary

The lambda closure of spt are the feasible functionals.

Conclusion

Conclusion.

Thanks!

Thank you for listening!