Second-order complexity theory	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00

type-two polynomial time and restriced lookahead

Bruce Kapron and Florian Steinberg

INRIA

July 5, 2018

Second-order complexity theory	Oracle polynomial-time	Additional restrictions	Composition and decomposition	Thx

Second-order complexity theory

- Oracle polynomial-time
- 3 Additional restrictions
- 4 Composition and decomposition

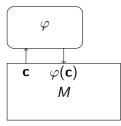
Second-order complexity theory ••	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Oracle Turing n	nachines			

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



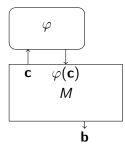
Second-order complexity theory ●○	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
\wedge \cdot $+$ \cdot				

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



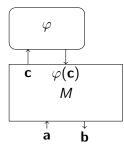
Second-order complexity theory ••	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Oracle Turing n	nachines			

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



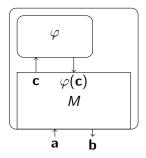
Second-order complexity theory ••	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Oracle Turing n	nachines			

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



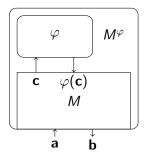
Second-order complexity theory ••	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Oracle Turing n	nachines			

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



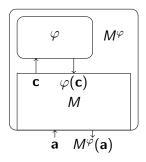
Second-order complexity theory •0	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Oracle Turing n	nachines			

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



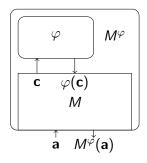
Second-order complexity theory • 0	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00

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



Second-order complexity theory •0	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Oracle Turing n	nachines			

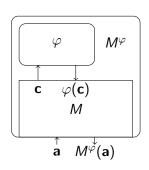
$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$
$$F : \subseteq \mathcal{B} \to \mathcal{B}, \quad \varphi \mapsto M^{\varphi}$$



Second-order complexity theory ••	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Oracle Turing n	nachines			

$$\begin{split} \varphi \in \mathcal{B} &:= \Sigma^* \to \Sigma^* \\ F &:\subseteq \mathcal{B} \to \mathcal{B}, \quad \varphi \mapsto M^{\varphi} \end{split}$$

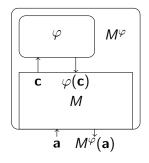
$$\operatorname{time}_{M}(\varphi, \mathbf{a})$$



Second-order complexity theory ●0	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Oracle Turing n	nachines			

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

$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq T(\ldots, |\mathbf{a}|)$$

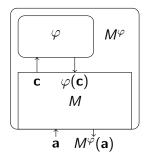


Second-order complexity theory ●○	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00

$$arphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$
 $F : \subseteq \mathcal{B} \to \mathcal{B}, \quad \varphi \mapsto M^{\varphi}$

$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq T(\ldots, |\mathbf{a}|)$$

$$|arphi|(n) := \max_{|\mathbf{a}| \le n} |arphi(\mathbf{a})|,$$

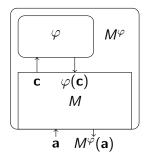


$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$

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

$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq T(|\varphi|, |\mathbf{a}|)$$

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

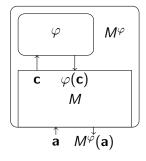


$$\varphi \in \mathcal{B} := \Sigma^* \to \Sigma^*$$

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

$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq T(|\varphi|, |\mathbf{a}|)$$

$$|arphi|(n) := \max_{|\mathbf{a}| \le n} |arphi(\mathbf{a})|, \quad |arphi| : \mathbb{N} o \mathbb{N}$$



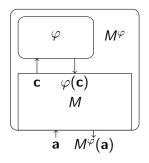
Second-order complexity theory $\bullet \circ$

Oracle polynomial-time

Additional restriction

Oracle Turing machines

$$arphi \in \mathcal{B} := \Sigma^* o \Sigma^*$$
 $F : \subseteq \mathcal{B} o \mathcal{B}, \quad \varphi \mapsto M^{\varphi}$



$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq T(|\varphi|, |\mathbf{a}|)$$

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

Second-order polynomials...

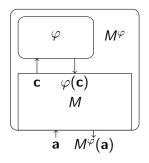
Second-order complexity theory $\bullet \circ$

Oracle polynomial-time

Additional restrictio

Oracle Turing machines

$$arphi \in \mathcal{B} := \Sigma^* o \Sigma^*$$
 $F : \subseteq \mathcal{B} o \mathcal{B}, \quad arphi \mapsto M^{arphi}$



$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq T(|\varphi|, |\mathbf{a}|)$$

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

Second-order polynomials...

Theorem

Closed under composition.

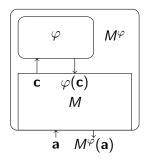
Second-order complexity theory $\bullet \circ$

Oracle polynomial-time

Additional restrictio

Oracle Turing machines

$$arphi \in \mathcal{B} := \Sigma^* o \Sigma^*$$
 $F : \subseteq \mathcal{B} o \mathcal{B}, \quad arphi \mapsto \mathcal{M}^{arphi}$



$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq T(|\varphi|, |\mathbf{a}|)$$

 $|arphi|(n) := \max_{|\mathbf{a}| \le n} |arphi(\mathbf{a})|, \quad |arphi| : \mathbb{N} o \mathbb{N}$

Second-order polynomials...

Theorem

Closed under composition.

Corollary

Preserves polynomial-time computability.

Second-order complexity theory ○●	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Comments				

Second-order complexity theory $o \bullet$	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Comments				

Theorem (Cook, Kapron, Urquart)

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

Second-order complexity theory $o \bullet$	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Comments				

Theorem (Cook, Kapron, Urquart)

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

Be carefull with resource bounded machines!

Second-order complexity theory o	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Comments				

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.

Second-order complexity theory $o \bullet$	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Comments				

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 runningtimes.

Second-order complexity theory $o \bullet$	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Comments				

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 runningtimes.
- Clockability.

Second-order complexity theory $\circ \bullet$	Oracle polynomial-time	Additional restrictions	Composition and decomposition	Thx 00
<u> </u>				

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 runningtimes.
- Clockability.

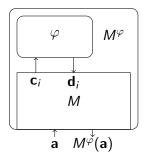
Theorem (Kawamura, S.)

Clocking is impossible.

Oracle polynomial-time

Additional restriction

Oracle polynomial-time



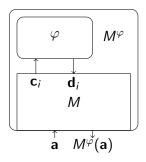
$t: \mathbb{N} \to \mathbb{N}$ is step-count

Second-order complexity theory $_{\rm OO}$

Oracle polynomial-time

Additional restrictio

Oracle polynomial-time



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

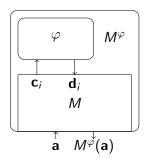
$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq t(m_{\varphi, \mathbf{a}}),$$

Second-order complexity theory $_{\rm OO}$

Oracle polynomial-time

Additional restriction

Oracle polynomial-time



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

$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq t(m_{\varphi, \mathbf{a}}),$$

where

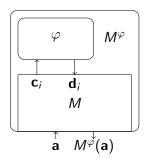
$$m_{arphi,\mathbf{a}} := \max\{|\mathbf{a}|,|\mathbf{d}_i|\}.$$

Second-order complexity theory $_{\rm OO}$

Oracle polynomial-time

Additional restriction

Oracle polynomial-time



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

$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq t(m_{\varphi, \mathbf{a}}),$$

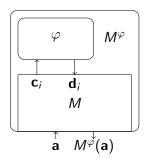
where

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

Oracle polynomial-time

Additional restriction

Oracle polynomial-time



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

$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq t(m_{\varphi, \mathbf{a}}),$$

where

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

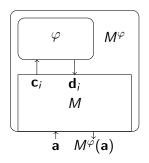
Lemma

Step-count condition holds in each step.

Oracle polynomial-time

Additional restriction

Oracle polynomial-time



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

$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq t(m_{\varphi, \mathbf{a}}),$$

where

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

Lemma

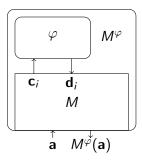
Step-count condition holds in each step.

because: total.

Oracle polynomial-time

Additional restriction

Oracle polynomial-time



Theorem

Total operators have step-counts. $t:\mathbb{N}
ightarrow\mathbb{N}$ is step-count if

$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq t(m_{\varphi, \mathbf{a}}),$$

where

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

Lemma

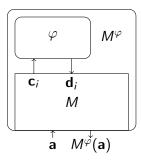
Step-count condition holds in each step.

because: total.

Oracle polynomial-time ●○

Additional restrictio

Oracle polynomial-time



Theorem

Total operators have step-counts. $t:\mathbb{N}
ightarrow\mathbb{N}$ is step-count if

$$\operatorname{time}_{M}(\varphi, \mathbf{a}) \leq t(m_{\varphi, \mathbf{a}}),$$

where

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

Lemma

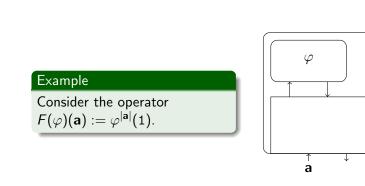
Step-count condition holds in each step.

because: total.

Theorem

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

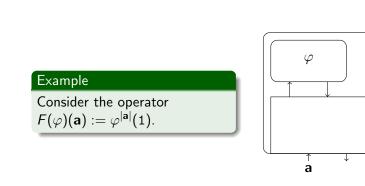
Second-order complexity theory	Oracle polynomial-time ⊙●	Additional restrictions	Composition and decomposition	Thx oo
Opt but not pt				



time(φ , **a**)

.

Second-order complexity theory	Oracle polynomial-time ⊙●	Additional restrictions	Composition and decomposition	Thx oo
Opt but not pt				

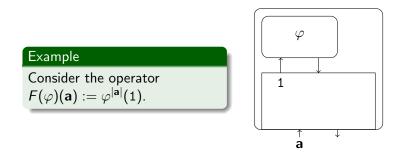


time(φ , **a**)

.

Second-order complexity theory	Oracle polynomial-time ○●	Additional restrictions	Composition and decomposition	Thx oo

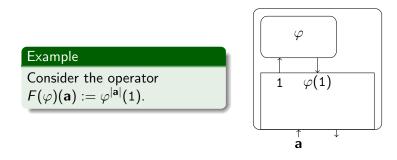
Opt but not pt



time(φ , **a**)

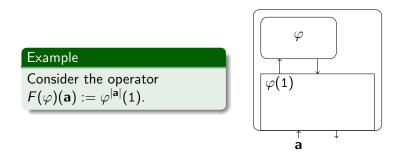
.

Second-order complexity theory	Oracle polynomial-time ○●	Additional restrictions	Composition and decomposition	Thx 00



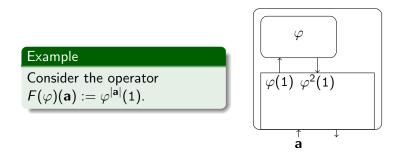
time(φ , **a**)

Second-order complexity theory	Oracle polynomial-time ○●	Additional restrictions	Composition and decomposition	Thx 00



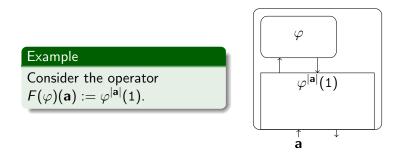
time(φ , **a**)

Second-order complexity theory	Oracle polynomial-time ⊙●	Additional restrictions	Composition and decomposition	Thx 00



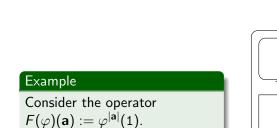
time(φ , **a**)

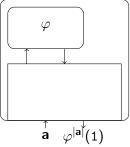
Second-order complexity theory	Oracle polynomial-time ⊙●	Additional restrictions	Composition and decomposition	Thx 00



time(φ , **a**)

Second-order complexity theory	Oracle polynomial-time ⊙●	Additional restrictions	Composition and decomposition	Thx oo
Opt but not pt				

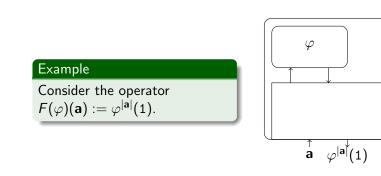




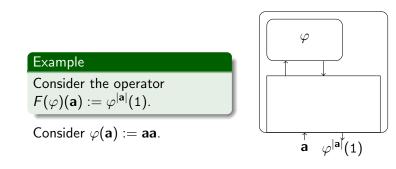
.

time(φ , **a**)

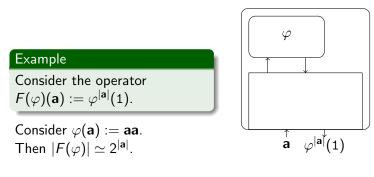
Second-order complexity theory	Oracle polynomial-time ⊙●	Additional restrictions	Composition and decomposition	Thx 00
Opt but not pt				



Second-order complexity theory	Oracle polynomial-time ⊙●	Additional restrictions	Composition and decomposition	Thx 00
Opt but not pt				



Second-order complexity theory	Oracle polynomial-time ⊙●	Additional restrictions	Composition and decomposition	Thx oo



Second-order complexity theory	Oracle polynomial-time ⊙●	Additional restrictions	Composition and decomposition	Thx oo

Example Consider the operator $F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1).$ Consider $\varphi(\mathbf{a}) := \mathbf{a}\mathbf{a}.$ Then $|F(\varphi)| \simeq 2^{|\mathbf{a}|}.$ $\Rightarrow F$ not polytime.

Second-order complexity theory	Oracle polynomial-time ⊙●	Additional restrictions	Composition and decomposition	Thx oo

Example Consider the operator $F(\varphi)(\mathbf{a}) := \varphi^{|\mathbf{a}|}(1).$ Consider $\varphi(\mathbf{a}) := \mathbf{a}\mathbf{a}.$ Then $|F(\varphi)| \simeq 2^{|\mathbf{a}|}.$ $\Rightarrow F$ not polytime. Example $\mathbf{a} = \varphi^{|\mathbf{a}|}(1)$

Essentially iteration operator.

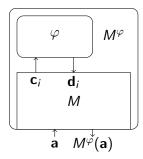
time
$$(\varphi, \mathbf{a}) = \mathcal{O}(m_{\varphi, \mathbf{a}}^2).$$

Oracle polynomial-time

Additional restrictions

Composition and decomposition

length and lookahead revisions

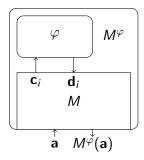


Oracle polynomial-time

Additional restrictions

Composition and decomposition

length and lookahead revisions



Definition

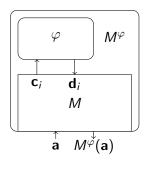
• finite length revision (spt).

Oracle polynomial-time

Additional restrictions

Composition and decomposition

length and lookahead revisions



Definition

• finite length revision (spt).

polynomial r.t.:

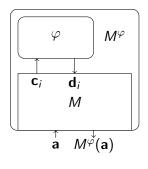
$$(l, n) \mapsto (p \circ l)^r (p(n)) + p(n).$$

Oracle polynomial-time

Additional restrictions

Composition and decomposition

length and lookahead revisions



Definition

• finite length revision (spt).

polynomial r.t.:

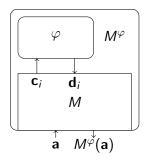
$$(l, n) \mapsto (p \circ l)^r (p(n)) + p(n).$$

Oracle polynomial-time

Additional restrictions

Composition and decomposition

length and lookahead revisions



Definition

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

polynomial r.t.:

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

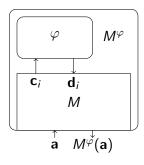
Second-order complexity theory 00

Oracle polynomial-time

Additional restrictions

Composition and decomposition

length and lookahead revisions



Definition

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

In both cases polynomial r.t.:

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

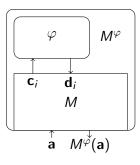
Second-order complexity theory 00

Oracle polynomial-time

Additional restrictions

Composition and decomposition

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 ~> finite lookahead revision.

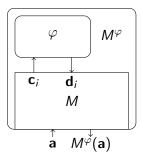
Second-order complexity theory 00

Oracle polynomial-time

Additional restrictions

Composition and decomposition

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 ~> finite lookahead revision.

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

Second-order complexity theory	Oracle polynomial-time	Additional restrictions	Composition and decomposition	Thx
00	00	○●		00
Examples				

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

Second-order complexity theory	Oracle polynomial-time	Additional restrictions	Composition and decomposition	Thx
00	00	○●		oo
Examples				

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

Second-order complexity theory 00	Oracle polynomial-time	Additional restrictions ○●	Composition and decomposition	Thx 00
Examples				

 F(φ)(a) := max{|φ(b)| | b ⊆ a} is polytime has finite lookahead revision

Second-order complexity theory 00	Oracle polynomial-time	Additional restrictions	Composition and decomposition	Thx 00
Examples				

 F(φ)(a) := max{|φ(b)| | b ⊆ a} is polytime has finite lookahead revision but no finite length revision.

Second-order complexity theory	Oracle polynomial-time	Additional restrictions	Composition and decomposition	Thx 00
Fxamples				

- F(φ)(a) := max{|φ(b)| | b ⊆ a} is polytime has finite lookahead revision but no finite length revision.
- Iteration: no finite lookahead revision

Second-order complexity theory	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Examples				

- F(φ)(a) := max{|φ(b)| | b ⊆ a} is polytime has finite lookahead revision but no finite length revision.
- Iteration: no finite lookahead revision but not polytime.

Second-order complexity theory	Oracle polynomial-time	Additional restrictions	Composition and decomposition	Thx 00
Examples				

- F(φ)(a) := max{|φ(b)| | b ⊆ 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.

Second-order complexity theory	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Examples				

- F(φ)(a) := max{|φ(b)| | b ⊆ 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

$$\mathsf{F}_0(arphi):=\epsilon$$
 and $\mathsf{F}_{n+1}(arphi):=(arphi\circarphi)(\mathsf{F}_n(arphi))^{\leq |arphi(\epsilon)|}$

Second-order complexity theory	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx 00
Examples				

- F(φ)(a) := max{|φ(b)| | b ⊆ 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

$$\mathsf{F}_0(arphi):=\epsilon$$
 and $\mathsf{F}_{n+1}(arphi):=(arphi\circarphi)(\mathsf{F}_n(arphi))^{\leq |arphi(\epsilon)|}$

Then set

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

Second-order complexity theory	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx oo
Examples				

- F(φ)(a) := max{|φ(b)| | b ⊆ 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

$$\mathsf{F}_0(arphi):=\epsilon$$
 and $\mathsf{F}_{n+1}(arphi):=(arphi\circarphi)(\mathsf{F}_n(arphi))^{\leq |arphi(\epsilon)|}$

Then set

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

This operator is polytime but not mpt.

Second-order complexity theory	Oracle polynomial-time	Additional restrictions	Composition and decomposition ••	Thx 00
The recursion c	perator			

Theorem (Cook, Urquart)

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

Second-order complexity theory	Oracle polynomial-time	Additional restrictions	Composition and decomposition ●0	Thx 00

I he 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 $\mathcal{R}(arphi, \mathbf{a}, \psi, \epsilon) := \mathbf{a}$ 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.

Second-order complexity theory	Oracle polynomial-time	Additional restrictions	Composition and decomposition ●0	Thx 00

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 $\mathcal R(arphi, \mathbf{a}, \psi, \epsilon) := \mathbf{a}$ 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.

Second-order complexity theory	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition ⊙●	Thx 00
A decompositio	n result			

Theorem (Kapron, S.)

Every mpt operator can be decomposed into two spt operators.

Second-order complexity theory	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition ⊙●	Thx 00
A decompositio				

Theorem (Kapron, S.)

Every mpt operator can be decomposed into two spt operators.

Example

Decomposition of the maximization operator.

Second-order complexity theory	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition ⊙●	Thx 00
A decompositio	n rocult			

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.

Second-order complexity theory	Oracle polynomial-time 00	Additional restrictions	Composition and decomposition	Thx ●0
Conclusion				

Conclusion.

Second-order complexity theory	Oracle polynomial-time	Additional restrictions	Composition and decomposition	Thx o●
Thanks!				

Thank you for listening!