Characterizing the Existence of Optimal Proof Systems and Complete Sets for Promise Classes

Olaf Beyersdorff
Institute of Theoretical Computer Science, Leibniz University Hanover, Germany

Zenon Sadowski
Institute of Mathematics, University of Białystok, Poland

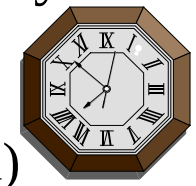
Preliminaries

Σ – a certain fixed finite alphabet

$x \in \Sigma^*$ – x is a string over the alphabet Σ

$|x|$ – the length of x

NPTM – nondeterministic polynomial-time clocked Turing machines

(our basic computational model) 

L(M) – the language accepted by M

$N_1, N_2, N_3, N_4, \ldots$ the standard enumeration of all polynomial-time clocked nondeterministic Turing machines

$D_1, D_2, D_3, D_4, \ldots$ the standard enumeration of all polynomial-time clocked deterministic Turing machines

1. Optimal proof systems

All propositional proof systems have three properties in common:

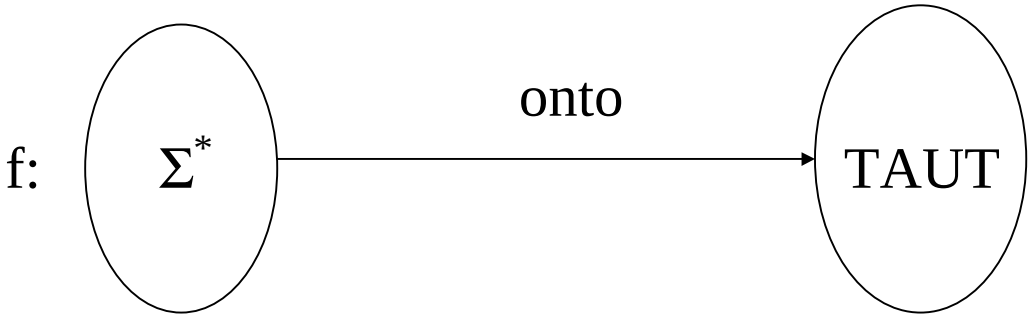
1. **Correctness (soundness)**: If there is a proof in the system, then the formula is indeed a tautology.

2. **Completeness**: Every tautology can be proved within the system.

3. **Verifiability:** The validity of a proof can be easily verified.

Definition (Cook, Reckhow)

An abstract propositional proof system (an abstract proof system for TAUT)...



f – computable by a deterministic Turing machine in time bounded by a polynomial in the length of the input

A string w such that f(w) = α we call a proof of a formula α.

Fact (Cook, Reckhow)

NP=co-**NP** if and only if, there exists a polynomially bounded propositional proof system.

How does one compare the efficiency of proof systems ?

h, h' – proof systems for TAUT

Definition

We say that h **p-simulates** h' iff, there exists a polynomial time computable function $\gamma : \Sigma^* \to \Sigma^*$ translating proofs in h' into proofs in h.

Definition

We say that h **simulates** h' iff, there exists a polynomial p such that for every tautology α, if α has a proof of length n in h' then α has a proof of length $\leq$ p(n) in h.

Definition (Krajíček, Pudlák)

A proof system for TAUT is p-optimal (optimal) if it p-simulates (simulates) any proof system for TAUT.

Open problems (Krajíček, Pudlák)

Does there exist a p-optimal proof system for TAUT ?

Does there exist an optimal proof system for TAUT ?

2. Semantic (promise) classes

UP, **NP**$\cap$co-**NP**, **BPP** – promise classes
Disjoint **NP** – pairs (**DNPP**) – also a promise class

These classes are defined using nondeterministic polynomial time clocked Turing machines which obey special conditions (promises).

UP-machine..., **NP**$\cap$co-**NP**-machine...,

Open problem

Do there exist complete languages for promise classes?

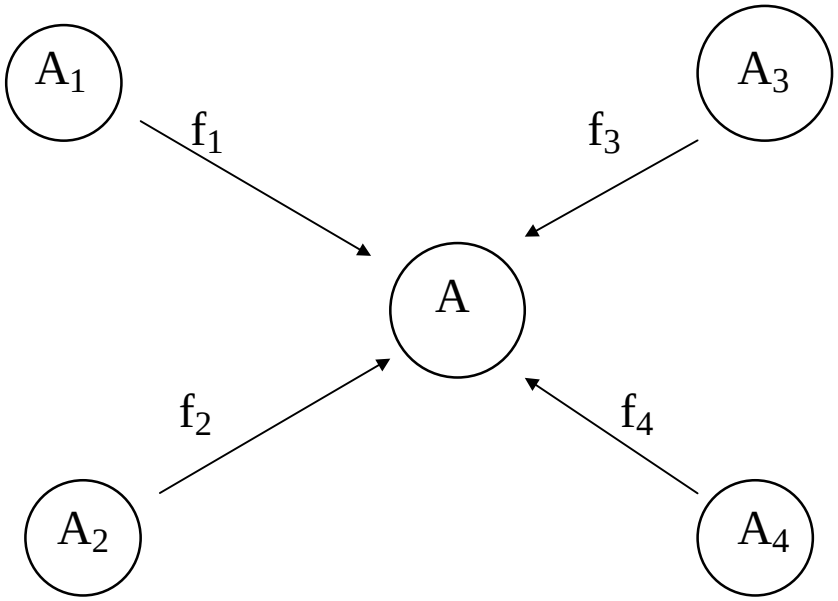
Syntactic vs. semantic classes

A class **C** is syntactically defined if for any polynomial-time clocked Turing machine N, we can decide whether or not it defines an element of **C** „simply by looking at it”.

Examples: **P** and **NP** are syntactic classes

- **P** = {L(M): M is a deterministic polynomial-time clocked Turing Machine}

- **NP** = {L(N): N is a nondeterministic polynomial-time clocked TM }



The rule of thumb:

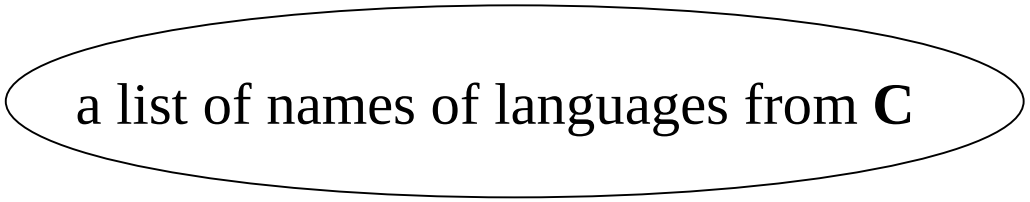
Syntactic classes possess complete languages, while semantic classes do not.

3. Uniform enumeration

Definition

A family of languages **C** has a recursive presentation if and only if there exists a recursively enumerable list of total Turing machines $\{M_{i_1}, M_{i_2}, M_{i_3},...\}$ such that

$$\mathbf{C} = \{(L(M_{i_k}): \quad k \geq 1\}$$

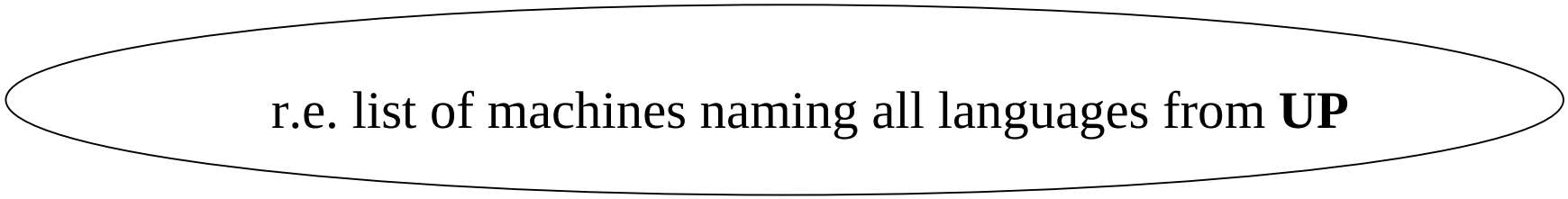


A recursive presentation by means of polynomial - time clocked Turing machines = a uniform enumeratiom

Theorem (Hartmanis, Hemachandra)
Statements (i) -- (ii) are equivalent:

(i) There exists a complete language for **UP**.

(ii) There exists a r. e. list of categorical nondeterministic polynomial-time clocked TMs $N_{i_1}, N_{i_2}, N_{i_3},...$ such that $\{(L(N_{i_k}): k \geq 1\} = \mathbf{UP}$.



a uniform enumeration of **UP**

NP$\cap$co-**NP** Kowalczyk 1981
BPP Hartmanis , Hemachandra 1988
DNPP Glasser, Selman,Sengupta 2004

easy = polynomial-time computable

NP-easy = acceptable by a nondeterministic polynomial-time Turing machine

Theorem

Statements (i) **---** (ii) are equivalent:

(i) There exists an optimal proof system for TAUT.

(ii) The class of all **NP**-easy subsets of TAUT is uniformly enumerable.

Theorem

Statements (i) **---** (ii) are equivalent:

(i) There exists a p-optimal proof system for TAUT.

(ii) The class of all easy subsets of TAUT is uniformly enumerable.

4. Two concrete examples

UP = {L(N): N is a nondeterministic polynomial-time Turing machine which on every input has at most one accepting path}

DNPP = {(A, B): A, B $\in$ NP are nonempty and A $\cap$ B = $\varnothing$ }

UP, **DNPP** ——— semantic classes with propositionally expressible promises

Expressing

For any nondeterministic poly-time clocked N we can construct propositional formulas ....
α^{N}_{1} , α^{N}_{2} , α^{N}_{3} , α^{N}_{4} , ...

Correctness

α^{N}_{n} is a tautology if and only if, N obeys **UP** – promise for any input of the length n.

More precisely

α^{N}_{n} is a tautology if and only if N on any input of the length n has at most one accepting path.

Representing (capturing)

Let A be a language such that A $\in$ **UP**
Let *f* be a proof system for TAUT

We say that A is **p-representable in f** if and only if there exists a polynomial-time clocked UP machine N such that:

1. A = L(N)

2. we have short *f*-proofs of the tautologies
α^{N}_{1}, α^{N}_{2}, α^{N}_{3}, α^{N}_{4}, ...

3. these proofs can be constructed in polynomial time

If every language A $\in$ **UP** is p-representable in *f* then we say that the class **UP is p-representable in *f***.

Characterization

Theorem 1

Statements (i) – (iii) are equivalent:

(i) There exists a complete language for **UP**

(ii) UP has a uniform enumeration

(iii) There exists a propositional proof system *h* such that **UP** is p-representable in *h*

uniform/nonuniform

DNPP

Expressing

(N, M) $\longrightarrow$ propositional formulas:

$\alpha^{N,M}_{1}$, $\alpha^{N,M}_{2}$, $\alpha^{N,M}_{3}$, $\alpha^{N,M}_{4}$, ...

Correctness

$\alpha^{N,M}_{n}$ is a tautology if and only if there does not exist a word x of the length n such that $x \in L(N)$ and $x \in L(M)$.

Representing (capturing)

Let *f* be a proof system for TAUT

We say that (A, B) is **representable in *f*** if and only if there exist polynomial-time clocked nondeterministic Turing machines N and M such that:

1. A = L(N) and B = L(M)

2. we have short *f*-proofs of the tautologies
$\alpha^{N,M}_1$, $\alpha^{N,M}_2$, $\alpha^{N,M}_3$, $\alpha^{N,M}_4$, ...

3. ———————

If every disjoint **NP** pair (A, B) is representable in *f* then we say that the class **DNPP is representable in *f***.

Characterization

Theorem

Statements (i) – (iv) are equivalent:

(i) There exists a complete disjoint **NP** pair

(ii) **DNPP** has a uniform enumeration

(iii) There exists a proof system *f* for TAUT such that **DNPP** is p-representable in *f*

(iv) There exists a proof system *f* for TAUT such that **DNPP** is representable in *f*

The class **DNPP** „can use nondeterminisn”

DNPP machines (pairs of machines) can perform nondeterministic computations without violating the promise

5. The generalized approach to promise (semantic) classes

How does one formalize the general notion of a promise class?

Promise = binary relation between nondeterministic poly-time machines and strings

R(N, x) **-----** N obeys promise R on input x

A machine N is called an R-machine if N obeys R on any input x

Given a promise relation R we define the promise class generated by R as

$$\mathbf{C_R} = \{L(N) : \text{ N is an R-machine}\}$$

Example

The promise for **UP**:

R(N,x) holds iff N(x) has at most one accepting path.

Then **UP** = $\mathbf{C_R}$

Let L be a language (e. g.) L = TAUT, L = SAT, L = QTAUT (the set of all tautological quantified propositional formulas)

The promise of **NP**$\cap$co-**NP** is expressible in QTAUT

Expressibility

Definition

A promise R is **expressible in a language L** if there exists a polynomial-time computable function corr: $\Sigma^* \times \Sigma^* \times 0^* \rightarrow \Sigma^*$ such that the following conditions hold:

(1) Correctness: For every Turing machine N, for every $x \in \Sigma^*$ and $m \in \mathbf{N}$ if $corr(x, N, 0^m) \in L$, then N obeys promise R on input x.

(2) Completeness: For every R-machine N with polynomial time bound p the set $Correct(N) = \{corr(x, N, 0^{p(|x|)}): x \in \Sigma^* \}$ is a subset of L.

(3) Local recognizability: For every Turing machine N, the set Correct(N) is polynomial-time decidable.

Definition (Cook, Reckhow)

A proof system for a language L is a polynomial-time computable function *f* with range L.

Representations

Let **C** be a promise class which is expressible in a language L. Let further $A \in \mathbf{C}$ and *f* be a proof system for L.

Definition

We say that A is **representable in *f*** if there exists a **C**-machine N for A with running time p such that for every $x \in \Sigma^*$ we have short *f*-proofs of $corr(x, N, 0^{p(|x|)})$. If these *f*-proofs can even be constructed in polynomial time, then we say that A is **p-representable in *f***.

Questions

Q1: Given a language L, does L have an optimal proof system ?

Q2: Given a promise class **C**, do there exist complete languages in **C** ?

Theorem
Statements (i) – (iii) are equivalent:

(i) L has a p-optimal proof system for L.
(ii) The class of all easy subsets of L is uniformly enumerable.
(iii) There exists a proof system *f* for L such that the class of all easy subsets of L is p-representable in *f*.

6. Q2 – Complete Languages for Promise Classes

Let **C** be a promise language (or function) class and L be a language such that **C** is expressible in L.

Theorem

The following conditions are equivalent:

(i) **C** has a complete language (or function).
(ii) **C** has a uniform enumeration.
(iii) There exists a proof system for L in which **C** is p-representable.

uniform/nonuniform

Let **C** be a promise language (or function) class which can use nondeterminism and let L be a language such that **C** is expressible in L.

Theorem

The following conditions are equivalent:

(i) **C** has a complete language (or function).
(ii) **C** has a uniform enumeration.
(iii) There exists a proof system for L in which **C** is p-representable.
(iv) There exists a proof system for L in which **C** is representable.

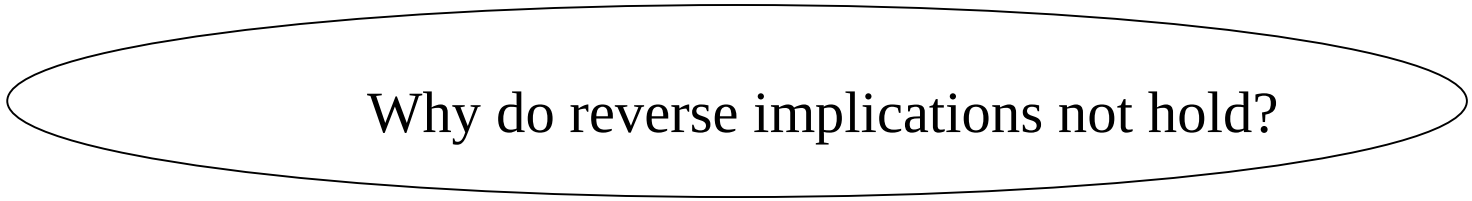
7. Q1 and Q2 -- Optimal Proof Systems and Complete Sets

Theorem (A. Razborov)

If there exists an optimal proof system for TAUT then there exists a complete disjoint **NP**-pair.

Theorem (J. Messner, J. Torán)

If there exists a p-optimal proof system for TAUT then there exists a complete language for **UP**.



Why, for some classes, does the existence of an optimal proof system imply the existence of complete languages but for other classes the existence of a p-optimal proof system implies the existence of complete languages?

Theorem

The following conditions are equivalent:

(i) There exists a p-optimal proof system for L.
(ii) There exists a proof system for L in which **any** promise class which is expressible in L is p-representable.
(iii) **Every** promise language and function class which is expressible in L has a complete language or function.

The promise that a Turing machine computes a proof system for L is the hardest one among those promises which are expressible in L.

Theorem

The following conditions are equivalent:

(i) There exists an optimal proof system for L.

(ii) L has a proof system *f* such that **every** promise class which is expressible in L is representable in *f*.

Corollary

If L has an optimal proof system, then **any** promise or function class C which is expressible in L and which can use nondeterminism has a complete language or function.

Conclusions

1. The class of all proof systems for L is the most semantic in the category of promise classes expressible in L

2. The phenomenon of uniformity versus nonuniformity, known from proof complexity, also appears in the context of the problems of the existence of complete languages for promise classes.

"The links between propositional proof systems and bounded arithmetic theories have many facets but informally one can view them as two sides of the same thing: The former is a non-uniform version of the latter."

J. Krajíček