CS 750 Slides 3: Relativization

André Nies

University of Auckland

October 6, 2010

ORACLES

DEFINITION

An oracle for a language A is a device capable of reporting whether any string w is a member of A.

TM M with oracle A writes a string w on the oracle tape. It is then informed in a single step whether $w \in A$.

Let P^A be the class of languages decidable by a polynomial time oracle TM with oracle A.

Similarly, define NP^A, PSPACE^A, etc.

Relativization is the process of passing from a class C to the class C^A.

Many proofs can be "relativized" to an oracle A.

For instance , the proof that $NP \subseteq PSPACE$ relativizes to a proof of $NP^A \subseteq PSPACE^A$ for each oracle A.

FACT

$NP \subseteq P^{SAT}$.

Proof: given $A \in \text{NP}$, there is a polynomial time computable function g such that

$$w \in A \Leftrightarrow g(w) \in \text{SAT}.$$

The oracle TM M, on input w
1. computes $g(w)$ and puts it on the oracle tape.
2. queries the oracle whether $g(w) \in \text{SAT}$.

If so, ACCEPT, otherwise, REJECT □

Remarks: The argument actually shows that if C is a class and S is C-complete (for $\leq_P$), then $C \subseteq P^S$.
Since $P^{\text{SAT}} = coP^{\text{SAT}}$, we also have $co\text{NP} \subseteq P^{\text{SAT}}$.

We give an example of a language in $\mathrm{NP}^{\mathrm{SAT}}$ that "probably" is not in NP.

- We say that Boolean formulas ϕ, ψ are *equivalent* if they have the same value for each truth assigment. In other words, the fmla $\neg(\phi \leftrightarrow \psi)$ is not satisfiable.
- fmla ϕ is *minimal* if there is no shorter fmla equivalent to it.

$$\text{NM-fmla} = \{\langle \phi \rangle : \ \phi \text{ is not a minimal fmla}\}.$$

FACT

NM-fmla is in NP^{SAT}.

Proof: On input $\langle \phi \rangle$:
1. Guess a shorter fmla ψ.
2. Using SAT as an oracle, check whether ϕ is equivalent to ψ. If so ACCEPT, else REJECT.

EXERCISE.

(can be added as Bonus to A3. Worth 3 marks).

EXAMPLE

Sketch a proof of the following:

$A \in \mathrm{NP} \cap co\mathrm{NP}$ if and only if $\mathrm{NP}^A = \mathrm{NP}$.

THE P= NP PROBLEM AND RELATIVIZATION

THEOREM

(i) There is an oracle A such that $P^A \neq NP^A$.
(Baker, Gill, Solovay, 1975)
(ii) There is an oracle B such that $P^B = NP^B$.

Conclusion from (ii): we cannot prove $P \neq \text{NP}$ via a diagonalization argument: otherwise, this argument could be carried out relative to B. Instead, we have to analyze computations, using local means.

Conclusion from (i): There is no discrete, step-by-step simulation showing that $P = \text{NP}$: such a simulation could be carried out relative to A.

PROOF OF THE EASIER PART (II)

Let $B = \text{TQBF}$.

$$\begin{aligned} \text{NP}^{\text{TQBF}} &\subseteq \text{NP}SPACE \\ &\subseteq PSPACE \\ &\subseteq P^{\text{TQBF}}. \end{aligned}$$

.

- First inclusion: because we can simulate a polynomial time NTM M with oracle for TQBF by NPSPACE machine N which actually computes the answers to all the oracle queries. Note that the length of the queries is polynomial in the length of the input because M works in polynomial time.
- Second inclusion: by Savitch's theorem
- Third inclusion: because TQBF is PSPACE complete.

PROOF OF (I)

We want an oracle A such that $P^A \neq \mathrm{NP}^A$.
For each oracle A consider the "lengths set"

$$L_A = \{1^n : \exists x \in A \, |x| = n\}.$$

Then $L_A \in \mathrm{NP}^A$ via the following oracle NTM:

On input w of the form 1^n:
1. Guess a string x of length n.
2. Ask the oracle whether $x \in A$. If so ACCEPT, otherwise REJECT.

To ensure $L_A \notin P^A$, we will design A in such a way that no polynomial time oracle TM M with oracle A decides L_A.

Let $M_1, M_2, \ldots$ be an effective listing of all polynomial time oracle TM. We can arrange that M_i runs in time $n^i + i$ (this will simplify the notation).

We construct A in stages i. At stage i we determine enough of A so that M_i^A does not decide L_A.

CONSTRUCTION

Stage i (ensure that M_i^A does not decide L_A):

1. So far, finitely many strings have been declared in A or out of A. Choose n larger than the length of any such string, and also large enough to ensure that $2^n > n^i + i$.
2. Simulate M_i on input 1^n. Respond to its oracle queries as follows.
   1. If M_i queries "$x \in A$?" for a string x where membership in A has already been determined, answer accordingly.
   2. Otherwise, respond NO and declare x to be out of A.
3. If M_i rejects 1^n (within it allotted time $n^i + i$) then pick a string w of length n that has not been queried and put w into A (since $n^i + i < 2^n$, such a string exists). Thus $1^n \in L_A$, while still M_i rejects 1^n with oracle A.
4. Any string of length $\leq n$ whose membership has not been decided yet is declared to be out of A.

VERIFICATION

In Step 3, we succeed in showing that M_i^A does not decide L_A:

- If M_i rejects 1^n, we ensure $1^n \in L_A$ while preserving the M_i computation.
- If M_i accepts 1^n, we put nothing of length n into A, so that $1^n \notin L_A$.

□

Remarks:
A is a very sparse language: for each n it has at most one string of length n.
A is decidable. Actually we can ensure that $A \in \text{TIME}(2^n)$.

FOOD FOR THOUGHT

EXERCISES

(i) Find an oracle B such that $\text{PSPACE}^B = P^B$.

(ii) (Sipser problem 9.20)
Extend Baker-Gill-Solovay to show that there is an oracle C with $\text{NP}^C \neq co\text{NP}^C$.

(iii) Show that $\text{NTIME}^B(n) = \text{TIME}^B(n)$ for $B = TQBF$. Thus, the Paul, Pippenger, Szemeredi and Trotter result does not relativize!