Complete problems in the polynomial hierarchy

November 4, 2014

There are three possible definition of the classes Σ_k and Π_k. We will first present all three definitions and then we will show that they are equivalent and define the same classes of problems. When M is a deterministic Turing machine that always halts and gives two possible outputs "accept" and "reject", then we will write $M(w) = 1$ iff M accepts the string w.

1 Definition 1

Let σ_k be the problem of deciding whether a given quantified formula with at most $k-1$ quantifier changes, beginning with $\exists$, is satisfied.
Let π_k be the problem of deciding whether a given quantified formula with at most $k-1$ quantifier changes, beginning with $\forall$, is satisfied.

Σ_k is the class of all problems that can be reduced deterministically in polynomial time to σ_k, and
Π_k is the class of all problems that can be reduced deterministically in polynomial time to π_k.

This definition immediately implies that σ_k, π_k, resp. are complete problems in the classes Σ_k, Π_k, resp.

2 Definition 2

Σ_k is the class of all problems L such that there is a deterministic polynomial time bounded Turing machine M such that for some non-negative integers $m_1, \ldots, m_k$ such that $x \in L$ if and only if

$$\exists u_1 \in \{0,1\}^{m_1} \forall u_2 \in \{0,1\}^{m_2} \ldots Q u_k \in \{0,1\}^{m_k} \quad M(x\#u_1\#\ldots\#u_k) = 1,$$

where Q is $\exists$ if k is odd and $\forall$ if k is even.

3 Definition 3

$$\Delta_0 = \Sigma_0 = \Pi_0 = \mathbf{P}$$

$$\Delta_k = \mathbf{P}^{\mathbf{\Sigma_{k-1}}} \qquad \Sigma_k = \mathbf{NP}^{\mathbf{\Sigma_{k-1}}} \qquad \Pi_k = \mathbf{coNP}^{\mathbf{\Sigma_{k-1}}}$$

for $k = 1, 2, \ldots$.
Lemma

$$\Delta_k = \mathbf{P}^{\mathbf{\Pi_{k-1}}} \qquad \Sigma_k = \mathbf{NP}^{\mathbf{\Pi_{k-1}}} \qquad \Pi_k = \mathbf{coNP}^{\mathbf{\Pi_{k-1}}}$$

for $k = 1, 2, \ldots$.

4 Proof $1 \rightarrow 2$

Let L be reducible to σ_k using a Turing machine R. The machine M for the definition 2 is a machine that first reduces an input to a quantified formula and then checks the validity of the formula for given $u_1, \ldots, u_k$.

5 Proof $2 \rightarrow 1$

Similarly as in the proof of Cook's theorem, we can find a formula Φ s.t. $\Phi(x, u_1, \ldots, u_k)$ is satisfied iff $M(x\#u_1\# \ldots \#u_k) = 1$.

6 Proof $2 \rightarrow 3$

Let us suppose that $x \in L$ iff

$$\exists u_1 \in \{0,1\}^{m_1} \forall u_2 \in \{0,1\}^{m_2} \ldots Q u_k \in \{0,1\}^{m_k} \quad M(x\#u_1\# \ldots \#u_k) = 1.$$

A non-deterministic machine with oracle from Π_{k-1} to solve the problem L can be constructed as follows: it first guesses the string u_1, inserts the string into the statement and then asks the oracle whether

$$\forall u_2 \in \{0,1\}^{m_2} \ldots Q u_k \in \{0,1\}^{m_k} \quad M(x\#u_1\# \ldots \#u_k) = 1.$$

7 Proof $3 \rightarrow 2$

This implication is a bit more complicated. Assume that L is in Σ_k according to the third definition. Then there is a non-deterministic Turing machine N which, using an oracle $O \in \Sigma_{k-1}$, accepts a string x if and only if $x \in L$. This means that there are non-deterministic choices of N that can be ordered in a string u_1, queries $q_1, \ldots, q_m$ of N to the oracle O and the corresponding answers $a_1, \ldots, a_m$ of the oracle O.

Thus, N accepts x iff there is a string u_1 and there are strings $q_1, \ldots, q_m$ and $a_1, \ldots, a_m$ such that
(1) a_i is the correct answer to q_i for all $i = 1, \ldots, m$, and
(2) if N gets an input x and makes non-deterministic choices described by the

string u_1, then it makes queries $q_1, \ldots, q_m$, and if it gets the answers $a_1, \ldots, a_m$, then it accepts.

The condition (2) can be tested by N transformed into a deterministic algorithm N' in the following way:
instead of making non-deterministic steps, it checks the corresponding bit of the string u_1, and
instead of asking the oracle O, it takes a_i as the (correct) answer to the question q_i.

It is easy to check that this machine asks the correct questions and that accepts.

So the machine M of the definition 2 first does the above computation. Then we have to check that answers of the oracle were correct. Since the oracle answers questions from Σ_{k-1}, we know that there is a deterministic Turing machine P such that
the oracle answers "yes" for the question q if an only if

$$\exists u_2 \forall u_3 \ldots Q u_k P(q, u_2, u_3, \ldots, u_k) = 1.$$

Thus, if the answer to q_i is correct, we have to check

$$\exists u_{2,i} \forall u_{3,i} \ldots Q u_{k,i} P(q_i, u_{2,i}, u_{3,i}, \ldots, u_{k,i}) = 1,$$

otherwise we have to check

$$\nexists u_{2,i} \forall u_{3,i} \ldots Q u_{k,i} P(q_i, u_{2,i}, u_{3,i}, \ldots, u_{k,i}) = 1,$$

i.e.

$$\forall u_{2,i} \exists u_{3,i} \ldots Q' u_{k,i} P(q_i, u_{2,i}, u_{3,i}, \ldots, u_{k,i}) = 0,$$

where Q' is the other quantifier that Q.

Combining all this, we can see that the language L belong also to Σ_k according to Definition 2.