Problem-Set 05

Mostafa Touny

July 19, 2022

Contents

Lectures: Sipser (skimmed)
10: first half of Ch. 7.4 (and also Ch. 5.3 if it helps)
11: Ch. 9.3 and rest of Ch. 7.4.

Ex. 0

Naively we guess any transformation f of Boolean formulas preserves the property of satisfiability. Hence it would always be the case

$$w \notin UNSAT \leftrightarrow f(w) \in SAT$$

Ex. 1

Since the questions is about factor a we can ignore constants. $n^r + (n^l)^b = n^{r+lb}$. So $a = r + lb$.

Ex. 2

following NTM simulation by DTM, partition states into H1 and H2 subsets, and apply the same procedure on each. now the combination of delta1 and delta2 reaches all possible states of NTM.

First. *sipser-NTM* can be viewed as a sequence of states, Each of which, is a subset of a deterministic TM's states. A state of *binary-NTM* can be viewed as a subset of exactly two states from a deterministic TM. Since there are no restrictions on the number of elements of *sipser-NTM*'s subset, *binary-NTM* can be seen as a special case of it.

Second. Recall the proof idea of a deterministic TM simulating a non-deterministic TM, whereby a determinstic state encodes/resembles a non-determinstic subset of states. Following the same idea, partition $Q = Q_1 \cup Q_2$ and define $Q'_1 = P(Q_1)$ and $Q'_2 = P(Q_2)$. Here $P(Q_i)$ means the set of all subsets of states Q_i. Let δ_1 and δ_2 be responsible of Q'_1 and Q'_2, respectively. Observe any state of $P(Q)$ can be constructed by some $x \cup y$ where $x \in Q'_1$ and $y \in Q'_2$. Therefore, Any configuration sequence of *sipser-TM* can be encoded/represented by some configuration sequence of *binary-TM.*

Ex. 3

The proof is shown by constructing a *non-deterministic exponential-time* algorithm for solving *IMPLICIT-4COL.*

Given a circuit C, Construct its graph G_C by enumerating all possible 2^n inputs of i and all 2^n inputs of j, Computing $C(i, j)$, for $i \neq j$. The complexity is $2^n \times 2^n = 2^{n+1}$; Exponential.

Check whether the graph G_C is 4-colorable. For each vertex of the graph, non-determinstically brute-force all the possible 4 colours. Since there are 2^n vertices, The complexity is *NEXP*.

Clearly the total complexity of a subroutine of *EXP* followed by a subroutine of *NEXP*, is *NEXP*.

Ex. 4

We construct the reduction function through a polynomial algorithm. Let *C_colorable* and *C_uncolorable* be some fixed 4-colorable and 4-uncolorable graphs.

```
L-to-3COL(w)
  check whether w is in L by the given polynomial algorithm
  if w belongs to L output C_colorable
  otherwise output C_uncolorable
```

Observe our mapping necessarily satisfies

$$w \in L \leftrightarrow \textit{L-to-3COL(w)} \in 3COL \tag{1}$$