Formal Languages and Computability 11

Ragnar Björn Ingvarsson, rbi3

12. nóvember 2024

1 You have implemented an algorithm using a 3-tape Turing machine X that runs in time $O(n^2)$. What is the time complexity when X is simulated with a single-tape Turing machine.

We observe that no matter the number of tapes, if a Turing machine exists that runs on X in polynomial time, all other Turing machines also run on X in at most polynomial time.

We also see that every $t(n)$ time multitape Turing machine has an equivalent $O(t^2(n))$ time single-tape Turing machine which means that when X is simulated with a single-tape Turing machine, the time complexity is $O((n^2)^2) = O(n^4)$.

2 Determine which of the following formulae are satisfiable

a. $(x_1 \vee x_2) \wedge (x_1 \vee \overline{x}_2)$

This is satisfiable since we can say that $x_1 = T$, and then x_2 is irrelevant.

b. $(x_1 \vee x_2) \wedge (x_1 \vee \overline{x}_2) \wedge (\overline{x}_1 \vee x_2) \wedge (\overline{x}_1 \vee \overline{x}_2)$

This is not satisfiable because the first clause states that either x_1 or x_2 are true. Then the last states that one of them is false. Therefore we conclude that they are either true and false or false and true. But the middle clauses state that both cases must be true meaning that this cannot be satisfied.

c. $(\overline{x}_1 \vee \overline{x}_2) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_3)$

This is satisfiable since we can simply state that x_1 is false and x_2 is true. Then, the first and last clause are true and the middle also. The value of x_3 is then irrelevant.

3 A presidential election takes place with two parties, A and B. There are n polling stations and both parties have nominated volunteers to act as election monitors. Each volunteer can attend only one station from some subset of stations. Both parties demand to have at least one own election monitor in each station.

a. Formulate the corresponding language $ELECTION$.

We can say that the language takes in the tuple (A, B, n) where A is a set of subsets which contain the polling stations each representative from A can monitor and B is the same but for party B, and n is the number of polling stations.

The language is then defined as such

For each $i \in \{1, 2, .., n\}$ there exists a subset in A and B which contains i, where we can express this link as the tuple (i, a_j, b_k), where $\forall a \in A \setminus \{a_j\} : a_j \neq a$ and
$\forall b \in B \setminus \{b_k\} : b_k \neq b$.

b. Give a polynomial time reduction to SAT.

We can express this in two parts. First we will create the satisfiable expression for ensuring that all polling stations are monitored, then we will ensure that no volunteer monitors two stations or more.

We simply express the first part as

$$\bigwedge_{s \in S} \left(\bigvee_{a \in A} x_{sa} \right) \wedge \left(\bigvee_{b \in B} x_{sb} \right) \tag{1}$$

Where x_{ij} is true if a volunteer j monitors station i.

The second part is quite tricky. We will consider the fact that for every volunteer that can monitor a station, all pairs of volunteers must contain a volunteer that does not monitor the station. Therefore we can express this as

$$\bigwedge_{a \in A, \alpha \in A \setminus \{a\}} (\overline{x}_a \vee \overline{x}_\alpha) \tag{2}$$

We can then repeat this for B and add them both to our expression to get the answer:

$$\bigwedge_{s \in S} \left(\left(\bigvee_{a \in A} x_{sa} \right) \wedge \left(\bigvee_{b \in B} x_{sb} \right) \bigwedge_{a \in A, \alpha \in A \setminus \{a\}} (\overline{x}_{sa} \vee \overline{x}_{s\alpha}) \bigwedge_{b \in B, \beta \in B \setminus \{b\}} (\overline{x}_{sb} \vee \overline{x}_{s\beta}) \right) \tag{3}$$

4 Let us consider a board game called the Geothermal Park. This game is played on an $n \times m$ board. The game master first blocks some squares out, then the player is given a set of k construction blocks with sizes $1 \times h_i$ that can be rotated. The task is to build a continuous path using non-overlapping blocks through the area. The path must start at the bottom and end at the top.

a. Formulate the corresponding language $PARK$.

The language takes in a tuple (n, m, C, B) where n is the length and m is the height of the game board, C is the set of cells that are not blocked and B is the set of numbers that describe the block lengths.

To formulate the language, we need a few conditions.

i. There exists a path $P = \{p_1, p_2, ..., p_l\}$ where each $p_i = (x_i, y_i)$ is a cell on the board such that $y_1 = 1$ and $y_l = m$, and each consecutive cell in P is adjacent to the last, $|x_i - x_{i+1}| + |y_i - y_{i+1}| = 1$, and lastly, $P \subseteq C$.

ii. There exists a set of blocks T where T_i is the set of cells occupied by block b_i. Then $T_i \subseteq P$ and $T_i \cap T_j = \varnothing$ for all $i \neq j$ and lastly $P = \bigcup_i T_i$

iii. For each block with coordinates $(x_{ij}, y_{ij}) \in T_i$ we require that $(\forall j \in T_i : |x_{ij} - x_{i,j+1}| = 1) \vee (\forall j \in T_i : |y_{ij} - y_{i,j+1}| = 1)$. By doing this we ensure that every block is either connected by cells horizontally or vertically.

From these conditions we can create the language as such

$$PARK = \{(n, m, C, B) \mid \text{Conditions i., ii. and iii. are satisfied}\} \tag{4}$$

b. Show that $PARK$ is in NP.

To show this we create a verifier V described like such:

On an input (n, m, C, K, P) we first say that if $P \nsubseteq C$ we reject. Then we create a set S which contains all possible sets of splits of P, where every cell is accounted for, into sets of lengths. These lengths represent chains of cells in P that are adjacent either vertically or horizontally.

If $(L \subseteq K) \in S$ we accept, otherwise reject.

c. Show that $PARK$ is NP-complete.

We have already shown that $PARK$ is in NP, so we will show that $PARK$ is NP-hard by showing that it is polynomial time reducible to some NP-complete problem.

Let's reduce from SAT, and I will only give high level descriptions here. Let's set the conditions:

i. There exists a tile in the bottom row and top row that is included in the path.

ii. For every block length, there exist such many tiles that we can express as being in a "block".

iii. For every block in the path, every tile in each block is distinct from any other tile in any other block.

iv. Each block is either vertical or horizontal, meaning each tile is either adjacent vertically or horizontally to its neighboring tile.

v. Any tile of every block must be adjacent to a tile of some other block.

vi. Every tile is in C.

We notice here that this forces all blocks to be used, but we can simply run this on the powerset of B to circumvent the trouble. Every one of these conditions can be expressed as a SAT which means the whole problem can be expressed as a SAT, meaning it is polynomial time reducible to SAT, a known NP-complete problem, so $PARK$ is NP-hard, as well as being in NP, which shows that it is NP-complete.