Algorithms 2018/2019

as. Mia Filić

April 18, 2019
Exercises 6 - E6
A Turing machine is a 7-tuple, \((Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})\), where \(Q, \Sigma, \Gamma\) are all finite sets and

1. \(Q\) is the set of states,
2. \(\Sigma\) is the input alphabet not containing the blank symbol \(\sqcup\),
3. \(\Gamma\) is the tape alphabet, where \(\sqcup \in \Gamma\) and \(\Sigma \subseteq \Gamma\),
4. \(\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}\) is the transition function,
5. \(q_0 \in Q\) is the start state,
6. \(q_{\text{accept}} \in Q\) is the accept state, and
7. \(q_{\text{reject}} \in Q\) is the reject state, where \(q_{\text{reject}} \neq q_{\text{accept}}\).
Let $M$ be the TM that decides\(^8\) language $L = \{w\#w | w \in \Sigma^*\}$.

Sequence of configurations (computation) for a TM $M$ on the input string $011000\#011000$.

---

\(^8\)Decides means that it always halts, e.g. reaches the reject or accept state on every input. Iff an input is from the language it decides, it halts in accepting state. Note that recognisers are similar to deciders, but does not have to stop.
Let $t : \mathbb{N} \rightarrow \mathbb{R}^+$ be a function. Define the *time complexity class*, \( \text{TIME}(t(n)) \), to be the collection of all languages that are decidable by an \( O(t(n)) \) time Turing machine.

Time complexity of the TM $M$ is basically number of moves that its head makes before getting to some final state.
Moreover, for a formal definition of the complexity of an arbitrary problem, we only observe the time complexity of the algorithm for some TM $M^9$ that solves it.
Theorem 5.1

*SAT is NP-complete problem.*

SAT or satisfiability problem searches for a TM machine that:

1. Given a set of variables denoted by $U$, and a collection of clauses $C$,
2. Can say if there is some interpretation on $U$ that KNF of $C$ yields true.
THE COOK-LEVIN THEOREM

Proof.
We have to 2 major steps:

1. SAT is NP,
2. every NP (polynomialy) reduces to SAT\(^{10}\)

\(^{10}\)Thus, all NP problems are at least as hard as SAT.
One instance of the SAT problem consists of the following:

- **INSTANCE:**
  A set of variables denoted by $U$, a collection of clauses $C$.

- **QUESTION:**
  Is there some assignment for $C$ so that CNF described by $C$ yields $TRUE$ (is satisfied)?

Thus, every time we are creating an instance of the SAT problem we need to define $U$ and $C$.\(^{11}\)

---

\(^{11}\) And in our case, on the input string $w$, we simulate the behavior of the TM $M_A$ that decides language $A$ to defined to support an arbitrary NP problem.
### THE COOK-LEVIN THEOREM

<table>
<thead>
<tr>
<th>Clause group</th>
<th>Restriction imposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1$</td>
<td>At each time $i$, $M$ is in exactly one state.</td>
</tr>
<tr>
<td>$G_2$</td>
<td>At each time $i$, the read-write head is scanning exactly one tape square.</td>
</tr>
<tr>
<td>$G_3$</td>
<td>At each time $i$, each tape square contains exactly one symbol from $\Gamma$.</td>
</tr>
<tr>
<td>$G_4$</td>
<td>At time 0, the computation is in the initial configuration of its checking stage for input $x$.</td>
</tr>
<tr>
<td>$G_5$</td>
<td>By time $p(n)$, $M$ has entered state $q_Y$ and hence has accepted $x$.</td>
</tr>
<tr>
<td>$G_6$</td>
<td>For each time $i$, $0 \leq i &lt; p(n)$, the configuration of $M$ at time $i+1$ follows by a single application of the transition function $\delta$ from the configuration at time $i$.</td>
</tr>
</tbody>
</table>

\[ r = |Q| - 1, \; v = |\Gamma| - 1, \; p(n) \text{ is polynomial bound on time compl. for } M, \; n = |w| \text{ input size for } M. \]
## THE COOK-LEVIN THEOREM

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
<th>Intended meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q[i,k]$</td>
<td>$0 \leq i \leq p(n)$&lt;br&gt;$0 \leq k \leq r$</td>
<td>At time $i$, $M$ is in state $q_k$.</td>
</tr>
<tr>
<td>$H[i,j]$</td>
<td>$0 \leq i \leq p(n)$&lt;br&gt;$-p(n) \leq j \leq p(n)+1$</td>
<td>At time $i$, the read-write head is scanning tape square $j$.</td>
</tr>
<tr>
<td>$S[i,j,k]$</td>
<td>$0 \leq i \leq p(n)$&lt;br&gt;$-p(n) \leq j \leq p(n)+1$&lt;br&gt;$0 \leq k \leq v$</td>
<td>At time $i$, the contents of tape square $j$ is symbol $s_k$.</td>
</tr>
</tbody>
</table>
A function $f : \Sigma^* \rightarrow \Sigma^*$ is a *computable function* if some Turing machine $M$, on every input $w$, halts with just $f(w)$ on its tape.