Comp487/587: Construction of the configurations graph

Given a *TM* M that runs on space $s(n)$ and an input x of size n, we build a directed graph $G_{M,x}$, called the *configuration graph* as follows. Recall that we assume the space bound $s(n)$ is space-constructible (and so $s(n) \geq \log n$).

1 Vertices of $G_{M,x}$

The vertices of $G_{M,x}$ are the configurations of $M(x)$. As we've seen previously, the number of configurations is $2^{cs(n)}$, where c is some constant that depends on M. In particular, to specify a configuration requires specifying:

- The location of the input reading tape head. ($n+1$ possible options, so $O(\log n)$ space to store).
- The contents of each of the working tapes. (Each of the $s(n)$ squares on each of the k tapes has $|\Gamma|$ different possible values, plus an extra cell for the state, so $O(s(n)k \log |\Gamma|) = O(s(n))$ total space.
- The current state of M. ($|Q|$ different options, so $O(\log |Q|) = O(1)$ space).
- The contents of the writing tape. (One square with $|\Gamma|$ possible values, so $O(\log |\Gamma|) = O(1)$ space).

Thus specifying a configuration requires $O(\log n + s(n) + 1) = O(\log n + s(n)) = O(s(n))$ space (recall $s(n) \geq \log n$). It follows that there are $2^{O(s(n))}$ possible vertices and so $G_{M,x}$ has $2^{O(s(n))}$ vertices. Note that it takes $O(s(n))$ time to construct each vertex, and so it takes $2^{O(s(n))}$ time to construct the set of vertices of $G_{M,x}$.

Let c_0 denote the initial configuration of M and let c_{acc} be the accepting configuration of M (we may assume that there is a *unique* accepting configuration, since we can assume that the machine wipes the entire working tape before accepting).

2 Edges of $G_{M,x}$

Given two vertices of $G_{M,x}$ (i.e configurations) c_1 and c_2, we draw an edge from c_1 to c_2 iff $c_1 \vdash c_2$. Recall that $c_1 \vdash c_2$ means that there is a one-step transition in the transition function δ that leads from c_1 to c_2. Given c_1 and c_2, we can use the following computation to check if $c_1 \vdash c_2$ in $O(n + s(n))$ time and $O(s(n))$ space:

1. Extract from c_1: (1) the state q, (2) the location of the head on the input tape, and (3) the location of the head on all k working tapes.
2. Use the location of the head on the input tape, together with x, to extract the input character a that will be read by M from this configuration. This takes $O(n)$ time and $O(\log n)$ space.
3. Use the location of the head on each working tape to extract the character a_i at the head position on each working tape i. This takes $O(ks(n)) = O(s(n))$ time and $O(k) = O(1)$ space.
4. Find in δ the transition $\delta(q, a, a_1, \cdots, a_k)$ (i.e. the transition that occurs at state q, with input symbol a, and with working tape symbols a_i). This gives us possible values for $(q', R/L, (a', R/L)^k, h)$: the new machine state, R/L on the input tape, the symbol to write on each of the k working tapes, R/L on each of the k input tapes, and an output bit. (Note that the size of this set of possible $(q', R/L, (a', R/L)^k, h)$ is $O(1)$).
5. Check if c_2 is consistent with applying each transition in $\delta(q, a, a_1, \cdots, a_k)$. That is, check that (1) the reading head of c_2 matches the new location, (2) the working tape heads of c_2 each match their new location, (3) the contents of the working tape are correct (the new symbols are written, and all other symbols are unchanged), (4) the output is correct (that is, set to h). This can all be done in $O(s(n))$ time and space.
   - If so, $c_1 \vdash c_2$. If none of the transitions are consistent, $c_1 \not\vdash c_2$.

There are $2^{O(s(n))}$ vertices in the graph. We can therefore construct all edges in the graph in $2^{O(s(n))} \times 2^{O(s(n))} \times O(n + s(n)) = (2^{O(s(n))})^3 = 2^{O(s(n))}$ time and $2^{O(s(n))}$ space.