CS525 Winter 2012 \ Class Assignment #4, 2/15/2012

Aylin Caliskan

Nicholas Milkovits

Ariel Stolerman

5.13)

Let $US_{TM} = \{\langle M \rangle \mid M\ is\ a\ TM\ with\ a\ useless\ state\}$. Following is a proof that US_{TM} is undecidable. Assume that US_{TM} is decidable, then there is a Turing machine R that decides it, so we can construct a TM S that will decide A_{TM} as follows.

We first define a Turing machine M' for any $\langle M, w \rangle$, with an additional symbol in the alphabet a and additional states are introduced, specifically $q_{useless}$ among them. In addition, any previous transition to q_{accept} shall be through $q_{useless}$ EXCEPT if the input symbol is the new a symbol. M' is defined as follows:

"On input x:

- If $x = a$, go through all states except $q_{useless}$ and then go to q_{accept} (accept).
- Otherwise, if $x \neq w$, go to q_{reject} (reject).
- Otherwise simulate M on $x = w$ (using the same states of M)."

Now we define the TM S as follows:

"On input $\langle M, w \rangle$:

- Construct M' from $\langle M, w \rangle$ as described above.
- Run R on M'. If it accepts, *reject*. Otherwise, *accept*.

First, note that M' will enter all states except for $q_{useless}$ in any case, as for the input a it goes through all but $q_{useless}, q_{reject}$ and for any input other than a, w it goes through q_{reject}. Now, for M' to be in US_{TM} it has to avoid going through $q_{useless}$ for the input w, and that will happen iff $\langle M, w \rangle \notin A_{TM}$. Therefore we can simulate R on the generated TM M' and be sure that if it accepts, it has a useless state $q_{useless}$, meaning M does not accept w (and the other way as well).

5.14)

Let

$A_{LEFT} =$

$\{\langle M, w \rangle \mid M \text{ is a Turing machine that for the input } w \text{ attempts to move its head left when it is on the leftmost tape cell}\}$.

Following is a proof that A_{LEFT} is undecidable. Assume A_{LEFT} is decidable, then there is a Turing machine R that decides it, so we can construct a TM S that decides A_{TM} as follows.

First we define a TM M' for any $\langle M \rangle$ as follows:

- Before doing anything, mark the first character. The first character in the tape will keep being marked every time it is changed by some state.

- Simulate M on the given input step by step, only whenever you land on a the marked character (which indicates the first position), and M dictates a left step, go right and left again instead, doing the state transition as originally dictated, and the character swap keeping the mark:
  In M: $\delta(q,a)=(r,b,L) \Rightarrow$ in M':
  - Not at first cell (unmarked character): $\delta'(q,a)=(r,b,L)$
  - At first cell (marked character): $\delta'(q,\hat{a})=(q',\hat{b},R), \forall c \in \Sigma: \delta'(q',c)=(r,c,L)$, where ˆ indicates the mark and q' is a new state in M'.
- For any original transition M has to its accept state, swap with a series of transitions that go to the leftmost cell of the tape, try to do one more left step disregarding any mark (the only states that will not do the right-step-left-step bypass as described previously), and then go to the accept state.

Now we define S as follows:

"On input $\langle M,w\rangle$:

- Construct M' as described above, with respect to the input M.
- Run R on $\langle M',w\rangle$. If it accepts, *accept*. Otherwise, *reject*."

The construction of M' for any given M makes sure that any attempts to go left from the leftmost tape cell are removed, by doing the original transition by going right and back left to the first cell again. Keeping the first cell marked makes sure there will be no attempts to go left from the leftmost cell during the simulation (until an accept state). Only when M accepts w, M' will attempt a left step from the leftmost tape cell. Therefore M accepts w if and only if M' will attempt to do a left step from the leftmost tape cell on the input w.

5.15)

Let $LEFT_{TM}=\{\langle M,w\rangle \mid M\ is\ a\ Turing\ machine\ that\ moves\ left\ during\ the\ computation\ over\ the\ input\ w\}$.

Following is a description of a TM R that decides $LEFT_{TM}$:

"On input $\langle M,w\rangle$:

- Simulate M for $|w|$ steps. If encountered a left move, *accept*.
- Simulate M for additional $|Q|+1$ steps. If encountered a left move, *accept*. Otherwise, *reject*.
- At any point if M halts (accepts or rejects w), and did not make a left move thus far, *reject*.

If M only moves to the right, after $|w|$ steps it will encounter only blanks. At this point, it is sufficient to simulate only additional $|Q|+1$ steps, as if there is no left move during that step sequence, we either halt, or detect a cycle, since all symbols read at this part of the tape are always blanks. During those $|Q|+1$ steps, if none of the states took a left move over a blank symbol, we will keep moving only right.