Problem

Consider the problem of determining whether a PDA accepts some string of the form {wwl w ? $\{0,1\}^*$} . Use the computation history method to show that this problem is undecidable.

Step-by-step solution

Step 1 of 1

When M does not accept w , then M_{new} will not go to left of $\$$. The Turing machine M_{new} writes something on the original input when M accept w .

Consider a Turing machine *T* having input *x*. Create a PDA which accepts string in $h\$h\$$ form when $x \in L(T)$.

The description history of *T* for *x* is the configuration sequence $C_0, C_1, C_2 \ldots C_n$ where C_0 is the starting configuration of *T* in input *x*. The configuration C_n is the end or accepting configuration of *T* in input *x*.

Working of Turing machine:

1. At first PDA insert C_0 into stack and then confirms whether C_0, is the initial configuration in Turing machine T on input *w*.
2. At each odd configuration the PDA pops the last configuration from stack and then checks that the new configuration is the successor of last one or not.
3. When PDA read and first $ encountered, then it skips the C_0 configuration and insert the C_1 in the stack. At this time, it has confirmation that even configuration is the successor of odd configuration.
4. The PDA gets ACCEPTS when all check are true which means all new configuration is the successor of previous one and last configuration is the accepting configuration.

The PDA accepts the string ww if *x* is accepted by Turing machine *T*.

When *x* is not accepted by Turing machine *T* even then this PDA can accept the string in $y\$z\$$ form where $y \neq z$. This all happens because the construction depends on the different check whether input is in ww form or which does not happen in PDA to confirm that two passes check the same history.

Hence, PDA accepts string is un-decidable.

Comments (3)