Linear Bounded Automata

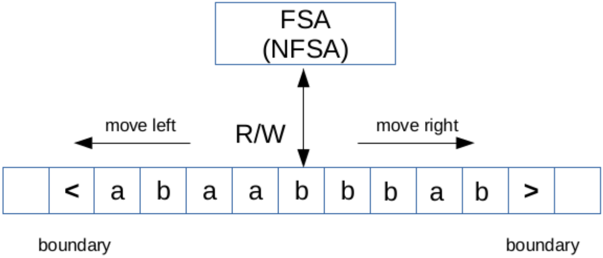
Alvin George and Habeeb .P

November 30, 2018

Overview

- Definition
- Results about LBA

Definition



A Turing machine that uses only the tape space occupied by the input is called a linear-bounded automaton (LBA).

Formal Definition

A linear bounded automaton is a non-deterministic Turing machine $M = (Q, \Sigma, \Gamma, \delta, s, t, r)$ such that:

- There are two special tape symbols $<$ and $>$ (the left end marker and right end marker).
- The TM begins in the configuration $(s, < x >, 0)$.
- The TM cannot replace $<$ or $>$ with anything else, nor move the tape head left of $<$ or right of $>$.

LBA

- An equivalent definition of an LBA is that it uses only k times the amount of space occupied by the input string, where k is a constant fixed for the particular machine.
- Possible to simulate k tape cells with a single tape cell, by increasing the size of the tape alphabet.
- Used as a model for actual computers rather than models for the computational process.

Examples

$\{a^n \mid n \text{ is a perfect square}\}$

- repeat
  - clear the 3 rd and 4 th tracks
  - add another a to the 2 nd track
  - copy the 2 nd track to the 3 rd track
  - while there are as written on the 3 rd track
    - delete an a from the 3 rd track
    - add the 2 nd track's a's to those on 4 th track
- until overflow takes place or 4 th track = input
- if there was no overflow then accept

Number of configurations

- Suppose that a given LBA M has
  - q states,
  - m characters in the tape alphabet ,
  - the input length is n
- Then M can be in at most
  - $\alpha(n) = q * n * m^n$ configurations
  - i.e. With m symbols and a tape which is n cells long, we can have only m^n different tapes.
  - The tape head can be on any of the n cells and we can be executing any of the q states

Results about LBA

Theorem 1

The halting problem is solvable for LBA.

- Idea for proof
  - The number of possible configurations for an LBA
  - An LBA that stops on input w must stop in at most $\alpha(|w|)$ steps
- Corollary:
  The membership problems for sets accepted by linear bounded automata are solvable

Results about LBA

Lemma

For any non-deterministic linear bounded automaton there is another which can compute the number of configurations reachable from an input.

Idea for proof:

- Enumerate all possible configuration
- Check whether the NLBA can get to them for a given input '*w*'

- $x = 0$
- for $i = 1 to k * n * s^n$
  - generate C_i
  - guess a path from C_0 to C_i
  - verify that it is a proper path
  - if C_i is reachable then $x = x + 1$
- verify that $x \leqslant m$ (otherwise reject)

Results about LBA

Theorem 2

The class of sets accepted by non-deterministic LBA is closed under complement.

Idea for proof:

- Find out exactly how many configurations are reachable
- Examine all of them and if any halting configurations are encountered, reject
- Otherwise accept

Results about LBA

Lemma

For every Turing machine there is a linear bounded automaton which accepts the set of strings which are valid halting computations for the Turing machine.

Results about LBA

Theorem 3

The emptiness problem is unsolvable for linear bounded Proof.

- If a Turing machine accepts no inputs then it does not have any valid halting computations.
- Thus the linear bounded automaton which accepts the Turing machine's valid halting computations accepts nothing.
- This means that if we could solve the emptiness problem for linear bounded automata then we could solve it for Turing machines.

References

Previous years {2011, 2013, 2016} slides on LBA.
Andrew Cumming, notes on "Linear Bounded Automata".

Thank You