Linear Bounded Automata

Manoharan M

Department of Computer Science and Automation
Indian Institute of Science, Bangalore
Overview

• Definition

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

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

- Examples: \{a^n \mid n \text{ is a perfect square}\}

- Used as a model for actual computers rather than models for the computational process.
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
  \[ \alpha(n) = q \times n \times m^n \] 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
    - LBA 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'

• **Theorem 2:** The class of sets accepted by non-deterministic LBA is closed under complement.

  - idea for proof:
    1. Find out exactly how many configurations are reachable
    2. examine all of them and if any halting configurations are encountered, reject
    3. 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.

**Theorem 3.** The emptiness problem is unsolvable for linear bounded automata.

**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.
Thank You