Lecture 21. Unrestricted Grammars

November 24, 2009
University of Waterloo
Meng He
Definitions

- Both the left and right side of a production can be any string of variables and terminals
  - Left hand side must be nonempty
- More formally, a 4-tuple \((V, \Sigma, P, S)\) in which:
  - A production is of the form \(\alpha \rightarrow \beta\), with \(\alpha \in (V \cup \Sigma)^+\) and \(\beta \in (V \cup \Sigma)^*\)
- **Sentential form, derivation, direct derivation, \(L(G)\):** defined in the same way as CFG’s
An Example

- An unrestricted grammar for Language \( \{a_i^2: i \geq 1\} \)
  
  \[
  
  \begin{align*}
  S & \rightarrow \text{BRAE} \\
  B & \rightarrow \text{BRAA} \\
  \text{RA} & \rightarrow aAR \\
  \text{Ra} & \rightarrow aR \\
  \text{RE} & \rightarrow E \\
  B & \rightarrow X \\
  \text{XA} & \rightarrow X \\
  Xa & \rightarrow aX \\
  XE & \rightarrow \epsilon
  \end{align*}
  \]

- Key idea: \( 1 + 3 + 5 + \ldots + (2n-1) = n^2 \)
A Quick Review of Turing Machines

- A **TM** is a computing device equipped with an unbounded tape divided into individual cells
  - Cells are numbered 0, 1, 2, ...
  - Initially, input is on cell 1, 2, ..., n, and cells 0, n+1, n+2, ... have a distinguished blank symbol **B**
  - Based on the current state and symbol being scanned, a TM can change the state, rewrite the symbol, move left, right or stay stationary
  - A TM accepts its input if, when it starts by scanning cell 0, eventually enters the halting state (not necessary reading all its input)

- Formally, a TM is a 6-tuple \((Q, \Sigma, \Gamma, \delta, q_0, h)\)
  - \(\Sigma \subseteq \Gamma\), \(B \not\in \Sigma, B \in \Gamma\)
  - \(\delta\): A partial function from \(Q \times \Gamma\) to \(Q \times \Gamma \times \{L, R, S\}\)
  - Configuration: \(wqx\)

- Recursively enumerable languages and recursive languages
Let $G = (V, \Sigma, P, S)$ be an unrestricted grammar. Then $L(G)$ is recursively enumerable.

We prove this by constructing a non-deterministic four-tape TM accepting $L(G)$:

- Tape 1 holds the input $w$, and will never change.
- Tape 2 holds a sentential form.
- Tape 3 holds the left side of a production.
- Tape 4 holds the corresponding right side.
Let \( L \) be a recursively enumerable language. Then there exists an unrestricted grammar \( G \) such that \( L(G) = L \).

Proof sketch (Let \( M \) be the one-tape TM accepting \( L \))

- Modify \( M \) to get a nondeterministic “language generator” \( M' \)
  1. \( M' \) has two tracks on its tape which is initially blank
  2. Write a nondeterministically-chosen string \( w \in \Sigma \) on track 1.
  3. Copy \( w \) to track 2.
  4. Simulate \( M \) on track 1.
  5. If \( M \) accepts, erase track 1, copy track 2 back to the tape and enter the halting state \( h \). Otherwise, \( M' \) crashes (no next move).

- Create an unrestricted grammar mimicking the computations of \( M' = (Q, \Sigma, \Gamma, \delta, q_0, h) \)