COMP30026 Models of Computation

Context-Free Languages

Lecture Week 8 Part

Semester 2, 2021

A Bit of History

Finite-state machines go back to McCulloch and Pitts (1943), who wanted to model the working of neurons and synapses.

The formalism that we use today is from Moore (1956).

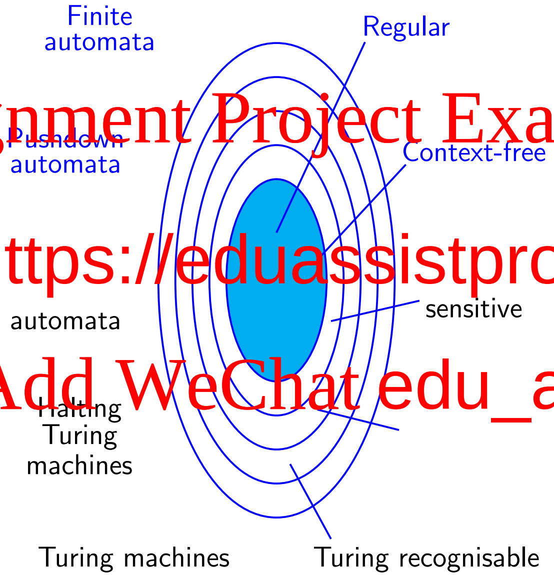
Kleene (1
expressi

We now turn to context-free grammars

These go back to Post's "productions" and Chom
formalism (1956).

Chomsky, a linguist, proposed a range of formalisms in grammar form for the description of natural language syntax.

Machines vs Languages



Context-Free Grammars in Computer Science

In the 60's, computer scientists started adopting context-free free grammars to describe syntax of programming languages.

They are fr NF).

Standar

indirectl

It is extensively used to specify syntax of programmi es, data formats (XML, JSON), etc.

Pushdown automata are to context-free grammars what finite-state automata are to regular languages.

We have already used the formalism of context-free grammars. To specify the syntax of regular expressions we gave a grammar, much like

$$R \rightarrow 0$$
$$R$$

$$R \rightarrow R\ R$$
$$R \rightarrow R^*$$

Hence a grammar is a set of substitutio e have the shorthand notation

$$R \rightarrow 0 \mid 1 \mid \text{eps} \mid \text{empty} \mid R \cup R \mid R\ R \mid R^*$$

Derivations, Sentences and Sentential Forms

A simpler example is this grammar G:

$$
\begin{array}{rcl}
A & \rightarrow & 0\ A\ 0 \\
A & \rightarrow & 1\ A\ 1 \\
A & \rightarrow & \epsilon
\end{array}
$$

Using the t s such as

$$
\begin{array}{rl}
\Rightarrow & 00A0 \\
\Rightarrow & 001A \\
\Rightarrow & 0010
\end{array}
$$

A is called a variable. Other symbols (here 0 and 1) are terminals. We refer to a valid string of terminals (such as 00100100) as a sentence. The intermediate strings that mix variables and terminals are sentential forms.

Clearly a grammar determines a formal language.

The language of G is written $L(G)$.

A langua [illegible] ar is a context-free language (CFL).

It should be clear that some of the languages that we fo [illegible] to be regular are context-free, for example

$$\{0^n 1^n \mid n \geq 1\}$$

Context-Free Grammars Formally

A context-free grammar (CFG) G is a 4-tuple (V, Σ, R, S), where

1. V is a finite set of variables,
2. Σ is a finite set of terminals,
3. R i
   left side),
4. S i

The binary relation $\Rightarrow$ on $(V \cup \Sigma)^*$ is de

Let $u, v, w \in (V \cup \Sigma)^*$. Then $uAw \Rightarrow$ R.
That is, $\Rightarrow$ captures a single derivation step.

Let $\overset{*}{\Rightarrow}$ be the reflexive transitive closure of $\Rightarrow$.

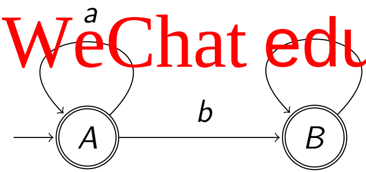
$$L(G) = \{s \in \Sigma^* \mid S \overset{*}{\Rightarrow} s\}$$

Right regular grammar:

$$\begin{aligned} A &\to a\ A \\ A &\to \epsilon \\ A &\to b\ B \end{aligned}$$

Left regular grammar:

$$\begin{aligned} A &\to A\ b \\ A &\to \epsilon \\ A &\to B\ a \end{aligned}$$



A Context-Free Grammar for Numeric Expressions

Here is a grammar with three variables, 14 terminals, and 15 rules:

$$
\begin{array}{lll}
E & \rightarrow & T \mid T + E \\
T & & F \quad F \quad T
\end{array}
$$

When th

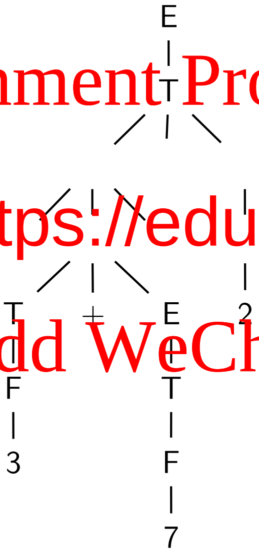
variable of the first rule.

An example sentence in the language is


```
(3 + 7) * 2
```

The grammar ensures that $*$ binds tighter than $+$.

Parse Trees



for

(3 + 7) * 2

Parse Trees

There are different derivations leading to the sentence `(3 + 7) * 2`, all corresponding to the parse tree above. They differ in the order in which we choose to replace variables. Here is the leftmost derivation:

$E \quad T$

$$
\begin{aligned}
&\Rightarrow (F + E) * T \\
&\Rightarrow (3 + E) * T \\
&\Rightarrow (3 + T) * T \\
&\Rightarrow (3 + F) * T \\
&\Rightarrow (3 + 7) * T \\
&\Rightarrow (3 + 7) * F \\
&\Rightarrow (3 + 7) * 2
\end{aligned}
$$

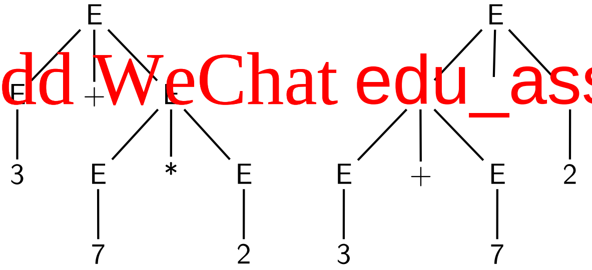
$E\ )$

Ambiguity

Consider the grammar

$$E \rightarrow E + E \mid E * E \mid (E) \mid 0 \mid 1 \mid \ldots \mid 9$$

This grammar allows not only different derivations, but different parse tre



A gramm

ambigu

Someti h

is not ambiguous, and which generates the same lan