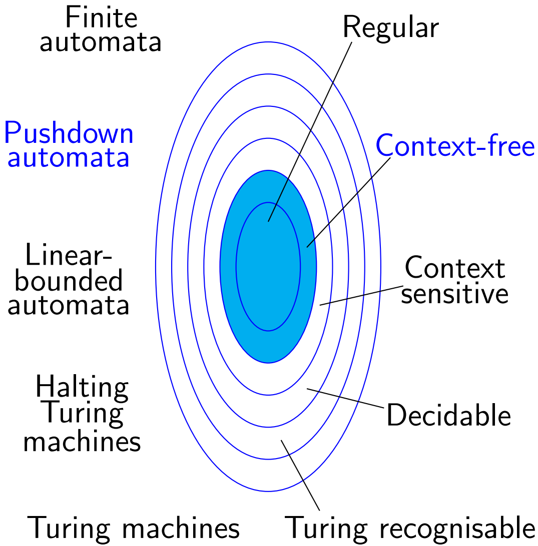
COMP30026 Models of Computation

Lecture 14: Context-Free Languages

Mak Nazecic-Andrlon and William Umboh

Semester 2, 2024

Machines vs Languages



Context-Free Grammars (CFGs)

Already used for wffs and regular expressions!

Main application: parsing programming languages.

Described as a set of substitution rules, or productions. Example:

$$
\begin{aligned}
R &\to 0 \\
R &\to 1 \\
R &\to \mathbf{eps} \\
R &\to \mathbf{empty} \\
R &\to R \cup R \\
R &\to R \circ R \\
R &\to R^*
\end{aligned}
$$

Shorthand: $R \to 0 \mid 1 \mid \mathbf{eps} \mid \mathbf{empty} \mid R \cup R \mid R\ R \mid R^*$

Generating with Grammars

Consider the grammar G:

$$A \rightarrow 0A0$$
$$A \rightarrow 1A1$$
$$A \rightarrow \epsilon$$

A is a variable. 0 and 1 are terminals.

Generation process:

1. Start with LHS of first rule.
2. Replace one instance of a variable with RHS of a matching rule.
3. Repeat step 2 until no variables remain.

Derivation Example

Given the grammar G:

$$A \to 0A0$$
$$A \to 1A1$$
$$A \to \epsilon$$

One possible derivation is

$$\begin{aligned} A &\Rightarrow 0A0 \\ &\Rightarrow 00A00 \\ &\Rightarrow 001A100 \\ &\Rightarrow 0010A0100 \\ &\Rightarrow 00100100 \end{aligned}$$

We say G generates 00100100.

The intermediate strings are called sentential forms.

Context-Free Languages

Definition

The language of a grammar is the set of strings it generates.

Definition

A language is context-free iff it is generated by some CFG.

Exercise: write a CFG for the non-regular language $\{0^n 1^n \mid n \geq 1\}$.

Context-Free Grammars Formally

Definition

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

1. V is a finite set of variables,
2. Σ is a finite set, disjoint from V, of terminals,
3. $R \subseteq V \times (V \cup \Sigma)^*$ is a finite set of rules,
4. $S \in V$ is the start variable.

Example

Our initial example has $V = \{A\}$, $\Sigma = \{0, 1\}$, $S = A$ and

$$\begin{aligned} R = \{&(A, 0A0), \\ &(A, 1A1), \\ &(A, \epsilon)\}. \end{aligned}$$

Derivation, Formally

Let $G = (V, \Sigma, R, S)$ be a CFG. Let $u, v, w \in (V \cup \Sigma)^*$ and $A \in V$.

Definition

uAw yields uvw iff $A \to v$ is a rule. We write $uAw \Rightarrow uvw$.

Definition

u derives v iff either $u = v$ or $u = u_1 \Rightarrow u_2 \Rightarrow \cdots \Rightarrow u_k = v$ for some sequence $u_1, \ldots, u_k \in (V \cup \Sigma)^*$. We write $u \overset{*}{\Rightarrow} v$.

Definition

The language of G is $\{w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w\}$.

A Context-Free Grammar for Numeric Expressions

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

$$
\begin{array}{lcl}
E & \rightarrow & T \mid T + E \\
T & \rightarrow & F \mid F * T \\
F & \rightarrow & 0 \mid 1 \mid \cdots \mid 9 \mid (\ E\ )
\end{array}
$$

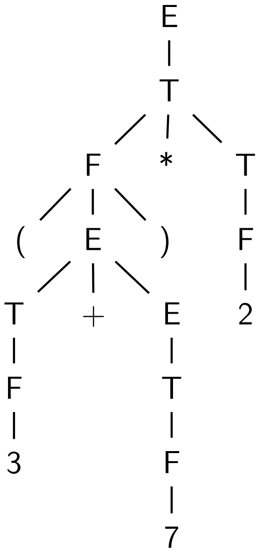
When the start variable is unspecified, it is assumed to be the variable of the first rule.

An example sentence in the language is

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

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

Parse Trees



A parse tree 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:

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

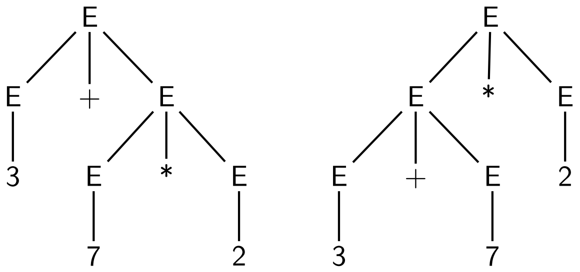
$$
\begin{aligned}
E &\rightarrow T \mid T + E \\
T &\rightarrow F \mid F * T \\
F &\rightarrow 0 \mid 1 \mid \ldots \mid 9 \mid (E)
\end{aligned}
$$

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 trees for `3 + 7 * 2`:



Accidental vs Inherent Ambiguity

Definition

A grammar is *ambiguous* iff it has two or more different parse trees for some string.

We can often find an equivalent but unambiguous grammar.

However, every CFG for $\{a^i b^j c^k \mid i = j \vee j = k\}$ is ambiguous.

Such CFLs are called inherently ambiguous.

Closure Properties for CFLs

Theorem

The class of context-free languages is closed under

- *union,*
- *concatenation,*
- *Kleene star,*
- *reversal.*

Exercise: figure out the constructions!

Closure Properties for CFLs

Not closed under intersection.

These two are CFLs (find CFGs!):

$$C = \{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^n \mid m, n \geq 0\},$$
$$D = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^m \mid m, n \geq 0\}.$$

But $C \cap D$ is *not* context-free.

How to prove? A new pumping lemma!

Pumping Lemma for CFLs

Lemma

If A is a context-free language over Σ, then there exists a length p such that, for all $s \in A$ with $|s| \geq p$, there exist $u, v, x, y, z \in \Sigma^*$ such that $s = uvxyz$ and

1. $uv^ixy^iz \in A$ *for all* $i \geq 0$,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Pumping Example

Theorem. $B = \{\mathrm{a}^n\mathrm{b}^n\mathrm{c}^n \mid n \geq 0\}$ is not context-free.

Proof. Assume it is, let p be the pumping length, and consider $s = \mathrm{a}^p\mathrm{b}^p\mathrm{c}^p \in B$. By the pumping lemma, $s = uvxyz$ for some strings u, v, x, y, z, with $uv^ixy^iz \in B$ for all $i \geq 0$, $|vy| > 0$ and $|vxy| \leq p$.

Since $|vxy| \leq p$, and the three blocks of symbols are of length p, the string vxy does not contain all three of a, b and c, and hence neither does v nor y.

Since $|vy| > 0$, at least one of v or y is nonempty. Thus uv^2xy^2z has strictly more than p occurrences of either one or two symbols, but at least one other symbol still only occurs p times.

Since every string in B has an equal number of a's, b's and c's, it follows that $uv^2xy^2z \notin B$. Contradiction!