Automata Theory :: Context-Free Languages

Jörg Endrullis

Vrije Universiteit Amsterdam

Context-Free Languages

A grammar $G = (V, T, S, P)$ is called **context-free** if all production rules are of the form

$$A \to u$$

where $A \in V$ and $u \in (V \cup T)^*$.

That is, the left-hand side of all rules is a single variable.

Context-Free Languages

A grammar $G = (V, T, S, P)$ is called **context-free** if all production rules are of the form

$$A \to u$$

where $A \in V$ and $u \in (V \cup T)^*$.

That is, the left-hand side of all rules is a single variable.

A language L is **context-free** if there is a context-free grammar with $L = L(G)$.

Context-Free Languages

A grammar $G = (V, T, S, P)$ is called **context-free** if all production rules are of the form

$$A \to u$$

where $A \in V$ and $u \in (V \cup T)^*$.

That is, the left-hand side of all rules is a single variable.

A language L is **context-free** if there is a context-free grammar with $L = L(G)$.

Context-free grammars are used to

- describe the **syntax of programming languages**, and
- form the basis of **parsing algorithms**.

Exercises (1)

Show that the following language is context-free:

$$\{ a^n \, b^m \, c^{2n} \mid n \geq 0, \; m \geq 0 \}$$

Exercises (1)

Show that the following language is context-free:

$$\{ a^n\, b^m\, c^{2n} \mid n \geq 0,\; m \geq 0 \}$$

We give a context-free grammar for the language:

Exercises (1)

Show that the following language is context-free:

$$\{\, a^n\, b^m\, c^{2n} \mid n \geq 0,\ m \geq 0 \,\}$$

We give a context-free grammar for the language:

$$S \to aScc$$

The start variable is S.

Exercises (1)

Show that the following language is context-free:

$$\{\, a^n\, b^m\, c^{2n} \mid n \geq 0,\ m \geq 0 \,\}$$

We give a context-free grammar for the language:

$$S \to aScc \mid T$$

The start variable is S.

Exercises (1)

Show that the following language is context-free:

$$\{\, a^n\, b^m\, c^{2n} \mid n \geq 0,\ m \geq 0 \,\}$$

We give a context-free grammar for the language:

$$\begin{aligned} S &\rightarrow aScc \mid T \\ T &\rightarrow bT \mid \lambda \end{aligned}$$

The start variable is S.

Exercises (2)

Show that the following language is context-free:

$$L = \Sigma^* \setminus \{\, ww \mid w \in \Sigma^* \,\}$$

where $\Sigma = \{a, b\}$.

Exercises (2)

Show that the following language is context-free:

$$L = \Sigma^* \setminus \{ ww \mid w \in \Sigma^* \}$$

where $\Sigma = \{a, b\}$. What shape do words $w \in L$ have?

Exercises (2)

Show that the following language is context-free:

$$L = \Sigma^* \setminus \{\, ww \mid w \in \Sigma^* \,\}$$

where $\Sigma = \{a, b\}$. What shape do words $w \in L$ have?

- w has **odd length**;

Exercises (2)

Show that the following language is context-free:

$$L = \Sigma^* \setminus \{ ww \mid w \in \Sigma^* \}$$

where $\Sigma = \{a, b\}$. What shape do words $w \in L$ have?

- w has **odd length**;
- w has **even length** such that

Exercises (2)

Show that the following language is context-free:

$$L = \Sigma^* \setminus \{\, ww \mid w \in \Sigma^* \,\}$$

where $\Sigma = \{a, b\}$. What shape do words $w \in L$ have?

- w has **odd length**;
- w has **even length** such that $w = uv$ with $|u| = |v|$ and

Exercises (2)

Show that the following language is context-free:

$$L = \Sigma^* \setminus \{ ww \mid w \in \Sigma^* \}$$

where $\Sigma = \{a, b\}$. What shape do words $w \in L$ have?

- w has **odd length**;
- w has **even length** such that $w = uv$ with $|u| = |v|$ and

  $$u(m) \neq v(m)$$

  for some $0 \leq m < |u|$.

Exercises (2)

Show that the following language is context-free:

$$L = \Sigma^* \setminus \{\, ww \mid w \in \Sigma^* \,\}$$

where $\Sigma = \{a, b\}$. What shape do words $w \in L$ have?

- w has **odd length**;
- w has **even length** such that $w = uv$ with $|u| = |v|$ and

  $$u(m) \neq v(m)$$

  for some $0 \leq m < |u|$. Let $n = |u| - m - 1$. Then w is in

  $$\begin{aligned} & \Sigma^m\{\,a\,\}\Sigma^n\ \Sigma^m\{\,b\,\}\Sigma^n \\ \cup\ & \underbrace{\Sigma^m\{\,b\,\}\Sigma^n}_{u}\ \underbrace{\Sigma^m\{\,a\,\}\Sigma^n}_{v} \end{aligned}$$

Exercises (2)

Show that the following language is context-free:

$$L = \Sigma^* \setminus \{ ww \mid w \in \Sigma^* \}$$

where $\Sigma = \{a, b\}$. What shape do words $w \in L$ have?

- w has **odd length**;
- w has **even length** such that $w = uv$ with $|u| = |v|$ and

  $$u(m) \neq v(m)$$

  for some $0 \leq m < |u|$. Let $n = |u| - m - 1$. Then w is in

$$\begin{array}{rl} & \Sigma^m\{a\}\Sigma^n \; \Sigma^m\{b\}\Sigma^n \\ \cup & \underbrace{\Sigma^m\{b\}\Sigma^n}_{u} \; \underbrace{\Sigma^m\{a\}\Sigma^n}_{v} \end{array} = \begin{array}{rl} & \Sigma^m\{a\}\Sigma^m \; \Sigma^n\{b\}\Sigma^n \\ \cup & \underbrace{\Sigma^m\{b\}\Sigma^m}_{u} \; \underbrace{\Sigma^n\{a\}\Sigma^n}_{v} \end{array}$$

Exercises (2)

Show that the following language is context-free:

$$L = \Sigma^* \setminus \{\, ww \mid w \in \Sigma^* \,\}$$

where $\Sigma = \{a, b\}$. What shape do words $w \in L$ have?

- w has **odd length**;
- w has **even length** such that $w = uv$ with $|u| = |v|$ and

  $$u(m) \neq v(m)$$

  for some $0 \leq m < |u|$. Let $n = |u| - m - 1$. Then w is in

$$\begin{array}{rl} & \Sigma^m\{\,a\,\}\Sigma^n\ \Sigma^m\{\,b\,\}\Sigma^n \\ \cup & \underbrace{\Sigma^m\{\,b\,\}\Sigma^n}_{u}\ \underbrace{\Sigma^m\{\,a\,\}\Sigma^n}_{v} \end{array} = \begin{array}{rl} & \Sigma^m\{\,a\,\}\Sigma^m\ \Sigma^n\{\,b\,\}\Sigma^n \\ \cup & \underbrace{\Sigma^m\{\,b\,\}\Sigma^m}_{u}\ \underbrace{\Sigma^n\{\,a\,\}\Sigma^n}_{v} \end{array}$$

We give a context-free grammar for the language:

$$S \to AB \mid BA \qquad A \to XAX \mid a \qquad X \to a \mid b$$
$$B \to XBX \mid b$$

Exercises (2)

Show that the following language is context-free:

$$L = \Sigma^* \setminus \{\, ww \mid w \in \Sigma^* \,\}$$

where $\Sigma = \{a, b\}$. What shape do words $w \in L$ have?

- w has **odd length**;
- w has **even length** such that $w = uv$ with $|u| = |v|$ and

  $$u(m) \neq v(m)$$

  for some $0 \leq m < |u|$. Let $n = |u| - m - 1$. Then w is in

$$\begin{array}{rl} & \Sigma^m\{\,a\,\}\Sigma^n\ \Sigma^m\{\,b\,\}\Sigma^n \\ \cup & \underbrace{\Sigma^m\{\,b\,\}\Sigma^n}_{u}\ \underbrace{\Sigma^m\{\,a\,\}\Sigma^n}_{v} \end{array} \quad = \quad \begin{array}{rl} & \Sigma^m\{\,a\,\}\Sigma^m\ \Sigma^n\{\,b\,\}\Sigma^n \\ \cup & \underbrace{\Sigma^m\{\,b\,\}\Sigma^m}_{u}\ \underbrace{\Sigma^n\{\,a\,\}\Sigma^n}_{v} \end{array}$$

We give a context-free grammar for the language:

$$S \to AB \mid BA \mid A \mid B \qquad A \to XAX \mid a \qquad X \to a \mid b$$
$$B \to XBX \mid b$$

Exercises (3)

Is the following language context-free?

$$L = \{ ww \mid w \in \Sigma^* \}$$

where $\Sigma = \{a, b\}$.

On the previous slide, we have shown that $\overline{L}$ is context-free.

Exercises (3)

Is the following language context-free?

$$L = \{\, ww \mid w \in \Sigma^* \,\}$$

where $\Sigma = \{a, b\}$.

On the previous slide, we have shown that $\overline{L}$ is context-free.

This language is not context-free.

We will prove this using the context-free pumping lemma.

Exercises (3)

Is the following language context-free?

$$L = \{ ww \mid w \in \Sigma^* \}$$

where $\Sigma = \{a, b\}$.

On the previous slide, we have shown that $\overline{L}$ is context-free.

This language is not context-free.

We will prove this using the context-free pumping lemma.

The class of context-free languages is not **closed under complement**.

Derivation Trees and Ambiguity

Rightmost and Leftmost

Let G be a grammar, and consider a derivation $S \Rightarrow^* w$.

- If G is **(right) linear**, w contains **at most one variable**.
- If G is **context-free**, w can contain **multiple variables**.

Rightmost and Leftmost

Let G be a grammar, and consider a derivation $S \Rightarrow^* w$.

- If G is **(right) linear**, w contains **at most one variable.**
- If G is **context-free**, w can contain **multiple variables.**

Two strategies to choose which variable to expand:

- **leftmost**: always the leftmost variable
- **rightmost**: always the rightmost variable

$$S \to SaS \mid b$$

Two derivations of *bab*:

leftmost: $S \Rightarrow SaS \Rightarrow baS \Rightarrow bab$

rightmost: $S \Rightarrow SaS \Rightarrow Sab \Rightarrow bab$

Rightmost and Leftmost

Let G be a grammar, and consider a derivation $S \Rightarrow^* w$.

- If G is **(right) linear**, w contains **at most one variable**.
- If G is **context-free**, w can contain **multiple variables**.

Two strategies to choose which variable to expand:

- **leftmost**: always the leftmost variable
- **rightmost**: always the rightmost variable

$$S \to SaS \mid b$$

Two derivations of *bab*:

leftmost: $S \Rightarrow SaS \Rightarrow baS \Rightarrow bab$

rightmost: $S \Rightarrow SaS \Rightarrow Sab \Rightarrow bab$

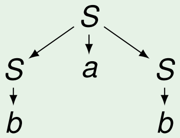
Result depends not on the strategy, but the choice of the rules.

Derivation Trees

A **derivation tree** for a context-free grammar $G = (V, T, S, P)$:

- Nodes have labels from $V \cup T \cup \{\lambda\}$. The **root** has label S.
- A node with label $A \in V$ can have children labelled
  - $x_1, \ldots, x_n$ if there is a rule $A \to x_1 \cdots x_n$ with $n \geq 1$, or
  - λ (single child) if there is a rule $A \to \lambda$.
- Every node with a label from $T \cup \{\lambda\}$ is a **leave.**

In the grammar $S \to SaS \mid b$, a **derivation tree** for *bab* is:

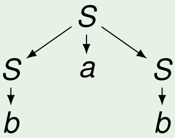


Derivation Trees

A **derivation tree** for a context-free grammar $G = (V, T, S, P)$:

- Nodes have labels from $V \cup T \cup \{\lambda\}$. The **root** has label S.
- A node with label $A \in V$ can have children labelled
  - $x_1, \ldots, x_n$ if there is a rule $A \to x_1 \cdots x_n$ with $n \geq 1$, or
  - λ (single child) if there is a rule $A \to \lambda$.
- Every node with a label from $T \cup \{\lambda\}$ is a **leave.**

In the grammar $S \to SaS \mid b$, a **derivation tree** for *bab* is:



The labels of the leaves of a derivation tree, read from left to right (skipping λ), form a word in $L(G)$.

Ambiguity

Ambiguous Grammars

A context-free grammar G is **ambiguous** if there exists a word $w \in L(G)$ for which there are multiple derivation trees.

Ambiguous Grammars

A context-free grammar G is **ambiguous** if there exists a word $w \in L(G)$ for which there are multiple derivation trees.

Is the following grammar ambiguous?

$$S \to S + S \mid 0$$

Ambiguous Grammars

A context-free grammar G is **ambiguous** if there exists a word $w \in L(G)$ for which there are multiple derivation trees.

Is the following grammar ambiguous?

$$S \to S + S \mid 0$$

Yes, there are two derivations trees for $0 + 0 + 0$:

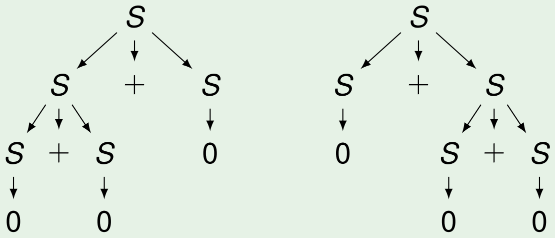
Ambiguous Grammars

A context-free grammar G is **ambiguous** if there exists a word $w \in L(G)$ for which there are multiple derivation trees.

Is the following grammar ambiguous?

$$S \to S + S \mid 0$$

Yes, there are two derivations trees for $0 + 0 + 0$:



Ambiguous Grammars

Is the following grammar ambiguous?

$$S \to S + 0 \mid 0$$

Ambiguous Grammars

Is the following grammar ambiguous?

$$S \to S + 0 \mid 0$$

No, every word has precisely one derivation tree.

Ambiguous Grammars

Is the following grammar ambiguous?

$$S \to S + 0 \mid 0$$

No, every word has precisely one derivation tree.

Note that both grammars

$$S \to S + S \mid 0 \qquad\qquad S \to S + 0 \mid 0$$

generate the same language:

$$\{\, 0(+0)^n \mid n \geq 0 \,\}$$

Ambiguity in Programming Languages

The following production rules (from ALGOL 60) became known as **dangling else problem**:

$$
\begin{array}{lcl}
\mathit{Statement} & \rightarrow & \texttt{if}\ \mathit{Condition}\ \texttt{then}\ \mathit{Statement} \quad | \\
& & \texttt{if}\ \mathit{Condition}\ \texttt{then}\ \mathit{Statement}\ \texttt{else}\ \mathit{Statement} \quad | \\
& & \ldots \\
\mathit{Condition} & \rightarrow & \cdots
\end{array}
$$

Ambiguity in Programming Languages

The following production rules (from ALGOL 60) became known as **dangling else problem**:

$$
\begin{aligned}
\mathit{Statement} \quad &\rightarrow \quad \texttt{if}\ \mathit{Condition}\ \texttt{then}\ \mathit{Statement} \quad | \\
& \quad \texttt{if}\ \mathit{Condition}\ \texttt{then}\ \mathit{Statement}\ \texttt{else}\ \mathit{Statement} \quad | \\
& \quad \ldots \\
\mathit{Condition} \quad &\rightarrow \quad \cdots
\end{aligned}
$$

There are two derivation trees for

```
if ... then if ... then ... else ...
```

Ambiguity in Programming Languages

The following production rules (from ALGOL 60) became known as **dangling else problem**:

$$
\begin{aligned}
\mathit{Statement} \quad &\rightarrow \quad \texttt{if}\ \mathit{Condition}\ \texttt{then}\ \mathit{Statement} \quad | \\
&\qquad\ \texttt{if}\ \mathit{Condition}\ \texttt{then}\ \mathit{Statement}\ \texttt{else}\ \mathit{Statement} \quad | \\
&\qquad\ \ldots \\
\mathit{Condition} \quad &\rightarrow \quad \cdots
\end{aligned}
$$

There are two derivation trees for

```
if ... then if ... then ... else ...
```

The production rules are ambiguous.

Donald Knuth, *The Remaining Trouble Spots in ALGOL 60*, CACM, 1967

Ambiguity Typically Unwanted

Ambiguity is typically unwanted:

- derivation trees often used to assign meaning to words,
- multiple derivation tree may result in double meaning.

Ambiguity Typically Unwanted

Ambiguity is typically unwanted:

- derivation trees often used to assign meaning to words,
- multiple derivation tree may result in double meaning.

In practice, ambiguity is often resolved outside of the grammar.

For example, by a precedence on the rules:

- For example, $0 + 2 * 1$ is parsed as $0 + (2 * 1)$.

Ambiguity Typically Unwanted

Ambiguity is typically unwanted:

- derivation trees often used to assign meaning to words,
- multiple derivation tree may result in double meaning.

In practice, ambiguity is often resolved outside of the grammar.

For example, by a precedence on the rules:

- For example, $0 + 2 * 1$ is parsed as $0 + (2 * 1)$.

Ambiguity is undecidable. That is, there exists no algorithm that decides whether a context-free grammar is ambiguous.

Inherently Ambiguous Languages

A language L is **inherently ambiguous** if

- L is context-free, and
- **every** grammar G with $L(G) = L$ is ambiguous.

Inherently Ambiguous Languages

A language L is **inherently ambiguous** if

- L is context-free, and
- **every** grammar G with $L(G) = L$ is ambiguous.

The following context-free language is inherently ambiguous

$$\{ a^n b^m c^\ell \mid n = m \ \vee \ m = \ell \}$$

It is generated by the following (ambiguous) grammar

$$S \to A \mid B \qquad A \to Ac \mid C \qquad B \to aB \mid D$$
$$C \to aCb \mid \lambda \qquad D \to bDc \mid \lambda$$

Inherently Ambiguous Languages

A language L is **inherently ambiguous** if

- L is context-free, and
- **every** grammar G with $L(G) = L$ is ambiguous.

The following context-free language is inherently ambiguous

$$\{ a^n b^m c^\ell \mid n = m \ \vee \ m = \ell \}$$

It is generated by the following (ambiguous) grammar

$$S \to A \mid B \qquad A \to Ac \mid C \qquad B \to aB \mid D$$
$$C \to aCb \mid \lambda \qquad D \to bDc \mid \lambda$$

Exercise

Find two derivation trees for the word *abc*.

Parsing

Parsing

Parsing is the search for a derivation tree for a given word.

Parsing

Parsing is the search for a derivation tree for a given word.

Theorem

For context-free grammars, parsing is possible in $O(|w|^3)$ time. (Here $|w|$ is the length of the input word.)

Parsing

Parsing is the search for a derivation tree for a given word.

Theorem

For context-free grammars, parsing is possible in $O(|w|^3)$ time.
(Here $|w|$ is the length of the input word.)

For every context-free grammar G:

- there exists an algorithm and $C \geq 0$ such that
- for every word w, the algorithm determines in at most

$$C \cdot |w|^3 \text{ steps}$$

whether $w \in L(G)$ and a derivation tree if $w \in L(G)$.