Part 1 Formal Grammars and Relations

What is a Formal Grammar?

Defining formal grammars

- We have our alphabet Σ, the elements of which we call terminal symbols (and often use $\{a, b, c, \dots\}$).
- We have a disjoint set of symbols N , the elements of which we call non-terminal symbols (and often use capital letters). $\{S, T, X, V, W\}$
- There is a special start symbol $S \in \Gamma$.
- Σ^* refers to the set of all possible strings that can be made with the alphabet
- $(\Sigma \cup \mathcal{N})^*$ refers to the set of all possible words that can be made
- A grammar then is a finite list of pairs (u, w) with $u, w \in (\Sigma \cup N)^*$ (called production rules, and often written $u \to w$).

A simple grammar example

Let $\Sigma = \{a, b\}, N = \{S, T\}$ and the production rules be

- $S \to \varepsilon$
- $S \to aT$
- $T \to bS$

  The resulting language is:

$$(ab)^n | n \in \mathbb{N} = \{\varepsilon, ab, abab, ababab, \dots\}$$

  ϵ is the empty word

How a formal grammar defines a language

Definition

A grammar G defines a language $L(G) \subseteq \Sigma^*$ by saying that $t \in L(G)$ if we can reach t with the following process:

1. Start with S.

2. Write our current word as v_0uv_1, and pick a rule (u, w). Then replace the current word by v_0wv_1.
3. If the current word is t, stop, else repeat Step 2.

Example

Let $\Sigma = \{a, b\}, N = \{S\}$ and the rules be $S \to \varepsilon, S \to aSa$ and $S \to bSb$.

- This grammar defines the even-length palindroms over $\{a, b\}$.

A “real” grammar

Example

Let $\Sigma = \{\text{the}, \text{dog}, \text{cat}, \text{eats}, \text{sleeps}\}$,

$$N = \{\text{S}, \text{NOUN}, \text{NP}, \text{VP}, \text{TRANS-VERB}, \text{INTRANS-VERB}\}$$

and the rules be S → NP VP, NP → the NOUN, NOUN → cat, NOUN → dog, VP → INTRANS-VERB, VP → TRANS-VERB NP, TRANS-VERB → eats, INTRANS-VERB → sleeps, INTRANS-VERB → eats.

Task

Find all “words” (ie sentences) belonging to the language of this grammar.

A complicated grammar example

Let $\Sigma = \{a, b, c\}, N = \{S, T, X, Y\}$ and the production rules be

- $S \to Tabc$
- $T \to \varepsilon$
- $T \to TXY$
- $XYa \to aXY$
- $Yb \to bY$
- $aXb \to aab$

- $bYc \to bbcc$
  This describes the language:

$$\{a^n b^n c^n | n \geq 1\} = \{abc, aabbcc, aaabbbccc, \ldots\}$$

Outlook

- Without restrictions on how rules might look like, it can be very time consuming to show that a word belongs to the language,
- and impossible(!!) to show that a word does not.
- We can formalize the derivation process a bit more, by introducing the transitive closure of a relation.

List of topics

- Formal Grammar definition

Links

- introduction to formal grammars

The Chomsky hierarchy

The hierarchy, overview

From simplest to most complicated:
3. Regular languages (having right-linear grammars)
2. Context-free languages (having context-free grammars)

1. Context-sensitive languages (having context-sensitive grammars)
2. Computably enumerable languages (having (unrestricted) grammars)
   -1. Arbitrary languages (not necessarily describable by a grammar at all)

Right-linear grammars

Definition

A grammar is *right-linear*, if all rules are of the form $T \to \varepsilon$ or $T \to aR$ for $T, R \in N$ and $a \in \Sigma$. I.e. the right hand side ends with non-terminal letters. We shall also allow the form $T \to a$ as abbreviation for $T \to aQ, Q \to \varepsilon$ for a fresh non-terminal Q.

Definition

A grammar is *left-linear*, if all rules are of the form $T \to \varepsilon$ or $T \to Ra$ for $T, R \in N$ and j. I.e. the right hand side starts with non-terminal letters

Theorem

Right-linear and *left-linear* grammars describe the same languages.

Context-free grammars

Definition

A grammar is *context-free*, if the left-hand side of every rule is a single non-terminal.

Let $\Sigma = \{a, b, c\}, N = \{S, T, X, Y\}$ and the production rules be

- $S \to Tabc$
- $S \to SA$
- $A \to bSc$
- $A \to ba$
- $T \to \varepsilon$
- $T \to TXY$
- $X \to XYab$

Context-sensitive grammars

Definition

A grammar is *context-sensitive*, if every rule is of the form $wAu \to wvu$ where $v \neq \varepsilon$; or is $S \to \varepsilon$, and where S never appears on the right-hand side of a rule.
Let $\Sigma = \{a, b, c\}$ and $N = \{A, B\}$

- $S \to abc$
- $S \to aAbc$
- $Ab \to bA$

- $Ac \to Bbcc$
- $bB \to Bb$
- $aB \to aa$
- $aB \to aaA$
- $bB \to \epsilon$

Definition

A grammar is monotonic, if for all rules $u \to w$ (except potentially $S \to \varepsilon$) it holds that $|u| \leq |w|$, ($|\cdot|$ being length) and S never appears on the right-hand side of a rule.

Theorem

A context sensitive grammar is monotonic
Every context-sensitve grammar is monotonic, or
or for every monotonic grammar there is an equivalent context-sensitive grammar
(link to proof)

Links

Relations and transitive closure

Relations

Definition

A relation between sets X, Y is a subset $R \subseteq X \times Y$. A relation on X is a subset $R \subseteq X \times X$.

Properties of relations

Reflexive if $\forall a \in X (a, a) \in R$
Symmetric if $\forall a, b \in X (a, b) \in R \Rightarrow (b, a) \in R$
Anti-reflexive if $\forall a \in X (a, a) \notin R$

Anti-symmetric if $\forall a, b \in X((a, b) \in R \wedge (b, a) \in R) \Rightarrow a = b$
Total if $\forall a, b \in X(a, b) \in R \vee (b, a) \in R$
Transitive if $\forall a, b, c \in X((a, b) \in R \wedge (b, c) \in R) \Rightarrow (a, c) \in R$

Example
Let $R \subseteq \{0, 1, 2, 3\} \times \{0, 1, 2, 3\}$ be defined as $R = \{(0, 1), (2, 3)\}$. Which properties does R have?

Example
Let $| \subseteq \mathbb{N} \times \mathbb{N}$ be defined as $(n, m) \in |$ iff n divides m. Which properties does $|$ have?

Definitions

A ***linear order*** is a anti-symmetric, total and transitive relation.
An ***equivalence relation*** is a reflexive, symmetric and transitive relation.

Composition of relations

Definition

Given $R \subseteq X \times Y$ and $Q \subseteq Y \times Z$, let $(Q \circ R) \subseteq X \times Z$ be defined as :

$$(Q \circ R) = (x, z) \in X \times Z | \exists y \in Y (x, y) \in R \wedge (y, z) \in Q$$

Transitive closure

Definition

Let R be a relation on X. We define $R^1 := R$, and $R^{n+1} := R^n \circ R$, and then $R^+ = \cup_{n \geq 1} R^n$.

The transitive closure is a mathematical operation that extends a binary relation between elements of a set to include all pairs of elements that are related by a path of one or more intermediate elements.

Formally, given a binary relation R on a set S, the transitive closure of R, denoted by R^+, is defined as the smallest transitive relation that contains R. In other words, R^+ includes all pairs (a, b) such that there exists a sequence of elements $a = x_1, x_2, \ldots, x_n = b$ in S, such that (x_i, x_{i+1}) is in R for all $1 \leq i < n$.

In other words, R^+ contains all the pairs of elements in A that can be related to each other by a finite sequence of steps using R. For example, if R represents the parent-child relationship in a family tree, then the transitive closure R^+ would include all the pairs of elements that are related as grandparents-grandchildren, great-grandparents-great-grandchildren, and so on.

Intuitively, we can think of the transitive closure as the "closure" of the relation R under the transitive property. That is, if R relates a to b, and b to c, then R^+ includes the pair (a, c) as well. In other words, the transitive closure adds all the pairs of elements that are related through a sequence of intermediate steps in R.

Theorem

The relation R^+ is the smallest transitive relation extending R, and we thus call it the transitive closure of R.

Example

Let $S = \{(n, n+1) | n \in \mathbb{N}\}$. Then $S^+ =<$.

The derivation relation

Definition

Consider a grammar G specified by terminals Σ, non-terminals $\mathcal{N}$, start symbol S and set rules R. The one-step derivation relation on $(\Sigma \cup \mathcal{N})^*$ is defined as:

$$\hookrightarrow := \{(\text{uvw}, \text{uv}\prime\text{w}) | u, v, w, v\prime \in (\Sigma \cup \mathcal{N})^* \quad (v, v\prime) \in R\}$$

Explanation

In the context of formal languages and automata theory, the one-step derivation relation is a binary relation between strings, denoted by the symbol $\hookrightarrow$. It represents the ability of a formal grammar to generate a string by replacing a single nonterminal symbol with a string of terminal and/or nonterminal symbols.

More formally, given a context-free grammar $G = (V, \Sigma, R, S)$, where V is a set of nonterminal symbols, Σ is a set of terminal symbols, R is a set of production rules, and S is the start symbol, the one-step derivation relation is defined as follows:

For any strings α, β, and $\gamma \in (V \cup \Sigma)*$ and any nonterminal symbol $A \in V, \alpha A \beta \to \alpha \gamma \beta$ if and only if there exists a production rule $A \to \gamma \in R$.

This means that if the nonterminal symbol A appears in the middle of a string $\alpha A\beta$, it can be replaced with the string γ to obtain a new string $\alpha\gamma\beta$. This new string can then be used as input to the one-step derivation relation again, possibly leading to further string expansions.

The one-step derivation relation is an important concept in the theory of formal languages, since it provides a way to formally define the notion of a context-free grammar generating a language. By repeatedly applying the one-step derivation relation, it is possible to generate all the strings in the language generated by a context-free grammar.

Definition

The derivation relation

$$\hookrightarrow\!\!\!\!\rightarrow := \hookrightarrow^+$$

is defined as the transitive closure of the one-step derivation relation.

Infix notation

We typically use infix notation for the (one-step) derivation relation, i.e. we write $u \hookrightarrow w$ for $(u, w) \in \hookrightarrow$ and $u, \hookrightarrow\!\!\!\!\rightarrow w$ for $(u, w) \in, \hookrightarrow\!\!\!\!\rightarrow$.
(Just as we write $n|m$ for n divides m, and $x = y$ rather than $(x, y) \in =, etc.$).

The language defined by a grammar

Definition

The language defined by a grammar G is:

$$L(G) := \{w \in \Sigma^* | S \hookrightarrow\!\!\!\!\rightarrow w\}$$

A grammar is a formal system that specifies a language in terms of its rules for generating strings in that language. A grammar consists of a set of nonterminal symbols, a set of terminal symbols, a start symbol, and a set of production rules.

The language defined by a grammar G is the set of all strings that can be generated by the grammar. This is denoted by $L(G)$ and is defined as follows:

$$L(G) := \{w \in \Sigma^* \mid S \hookrightarrow\!\!\!\!\rightarrow w\}$$

where Σ is the set of terminal symbols, S is the start symbol, and the relation $\hookrightarrow\!\!\!\!\!\twoheadrightarrow$ denotes a sequence of productions that can be used to generate a string from the start symbol. In other words, $L(G)$ consists of all strings that can be derived from the start symbol S by applying a sequence of production rules.

To summarize, the definition of $L(G)$ in automata theory specifies the language generated by a grammar G, which is the set of all strings that can be derived from the start symbol S by applying a sequence of production rules.