Syntax and grammars

NTNU

TDT4165 fall 2018

Formal grammars

A formal grammar, G, can be defined as a set G = N,T,R,S where N, T, R are finite sets and S∈N.

Terminal symbols T is a set that contains our alphabet. These are any characters that are legal in our language.
Example: T = {a,b,c,d,4,$}

Non-terminal symbols N contains of the non-terminal symbols of a language. They are also called syntactical variables. Non-terminal symbols are symbols that can be replaced with other symbols. It is common to denote non-terminal symbols using uppercase alphabetic characters or to surround them with angular brackets ⟨A⟩.
Example: N = {⟨Z⟩,⟨Y⟩,⟨S⟩}

Starting point ⟨S⟩ is our starting point and is in our set of non-terminal symbols.

Production rules The set R contains our production rules. Production rules define how we can perform symbol substitution in G.
Example: R = {⟨Z⟩ → a | a⟨Y⟩b,⟨Y⟩ → c | ϵ,⟨S⟩ → ⟨Z⟩ | ⟨Y⟩} To determine legal strings in our language, we recursively replace expressions on the left hand side, with expressions on the right hand side, starting at ⟨S⟩.

```
<S> ::= <Z> | <Y>
<Z> ::= a | a <Y> b
<Y> ::= c | ϵ
```

To get the string **ab**, we start at ⟨S⟩ which can be ⟨Z⟩ or ⟨Y⟩. Choosing ⟨Z⟩, we have the choices **a** and **a** ⟨**Y**⟩ **b**. For ⟨Y⟩, we also have two choices: **c** or the empty string. We have:

```
<S> → <Z> → a <Y> b → aϵb → ab
```

This is an example of a finite language. We can also create infinite languages, like the one below.

```
<S> ::= <Z>
<Z> ::= a <Z> b | c
```

This language contains all strings $a_1...a_n c b_1..b_n, n >= 0$:

```
n = 0: c
n = 1: acb
n = 2: aacbb
...
```

Chomsky hierarchy

The Chomsky hierarchy is a way to categorize formal grammars, by their properties. Each higher level category, inherits the properties of the lower categories.

Unrestricted grammar Production rules have the form $\alpha \rightarrow \beta$, where α and β are any combination of terminal and non-terminal symbols. The left hand side (α) of a production rule cannot be empty (ϵ) and contains at least one non-terminal.

Context-sensitive grammar Production rules have the form $\alpha\langle \mathrm{A} \rangle\beta \rightarrow \alpha\gamma\beta$, where α, β are any combination of terminal and non-terminal symbols or ϵ, γ is any combination of terminal and non-terminal symbols, but cannot be ϵ and A is a non-terminal.

Context-free grammar Every production rule is on the form $\langle \mathrm{A} \rangle \rightarrow \alpha$, where A is a non-terminal and α is any combination of terminal and non-terminal symbols.

Regular languages Production rules have the form A $\rightarrow$ aB, or A $\rightarrow$ a, where A and B are non-terminals and a is a terminal. This example is a right regular grammar. Grammars with production rules on the form A $\rightarrow$ Ba, are left regular grammars.