

Theory of Computer Science

B2. Regular Grammars: ε -Rules

Gabriele Röger

University of Basel

March 3, 2025

1 / 22

Theory of Computer Science

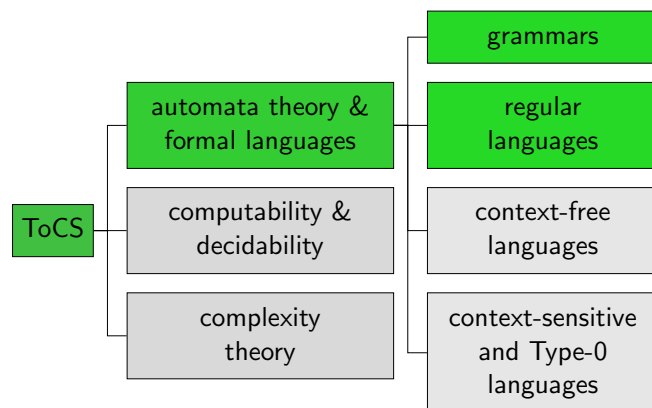
March 3, 2025 — B2. Regular Grammars: ε -Rules

B2.1 Recap

B2.2 Epsilon Rules

2 / 22

Content of the Course



3 / 22

B2. Regular Grammars: ε -Rules

Recap

B2.1 Recap

4 / 22

Recap: Regular Grammars

Definition (Regular Grammars)

A **regular grammar** is a 4-tuple $\langle V, \Sigma, R, S \rangle$ with

- ▶ V finite set of variables (nonterminal symbols)
- ▶ Σ finite alphabet of terminal symbols with $V \cap \Sigma = \emptyset$
- ▶ $R \subseteq (V \times (\Sigma \cup \Sigma V)) \cup \{\langle S, \epsilon \rangle\}$ finite set of rules
- ▶ if $S \rightarrow \epsilon \in R$, there is no $X \in V, y \in \Sigma$ with $X \rightarrow yS \in R$
- ▶ $S \in V$ start variable.

Rule $X \rightarrow \epsilon$ is only allowed if $X = S$ and S never occurs in the right-hand side of a rule.

How restrictive is this? If we don't restrict the usage of ϵ as right-hand side of a rule, what does this change?

Question (Slido)

With a regular grammar, how many steps does it take to derive a non-empty word (over Σ) from the start variable?



Recap: Regular Languages

A language is regular if it is generated by some regular grammar.

Definition (Regular Language)

A language $L \subseteq \Sigma^*$ is **regular** if there exists a regular grammar G with $\mathcal{L}(G) = L$.

B2.2 Epsilon Rules

Recap: Regular Grammars

Definition (Regular Grammars)

A **regular grammar** is a 4-tuple $\langle V, \Sigma, R, S \rangle$ with

- ▶ V finite set of variables (nonterminal symbols)
- ▶ Σ finite alphabet of terminal symbols with $V \cap \Sigma = \emptyset$
- ▶ $R \subseteq (V \times (\Sigma \cup \Sigma V)) \cup \{ \langle S, \varepsilon \rangle \}$ finite set of rules
- ▶ if $S \rightarrow \varepsilon \in R$, there is no $X \in V, y \in \Sigma$ with $X \rightarrow yS \in R$
- ▶ $S \in V$ start variable.

Rule $X \rightarrow \varepsilon$ is only allowed if $X = S$ and S never occurs in the right-hand side of a rule.

How restrictive is this? If we don't restrict the usage of ε as right-hand side of a rule, what does this change?

Recap: Regular Grammars

Definition (Regular Grammars)

A **regular grammar** is a 4-tuple $\langle V, \Sigma, R, S \rangle$ with

- ▶ V finite set of variables (nonterminal symbols)
- ▶ Σ finite alphabet of terminal symbols with $V \cap \Sigma = \emptyset$
- ▶ $R \subseteq (V \times (\Sigma \cup \Sigma V)) \cup \{ \langle S, \varepsilon \rangle \}$ finite set of rules
- ▶ **if $S \rightarrow \varepsilon \in R$, there is no $X \in V, y \in \Sigma$ with $X \rightarrow yS \in R$**
- ▶ $S \in V$ start variable.

Rule $X \rightarrow \varepsilon$ is only allowed if $X = S$ and S never occurs in the right-hand side of a rule.

How restrictive is this? If we don't restrict the usage of ε as right-hand side of a rule, what does this change?

Our Plan

We are going to show that every grammar with rules

$$R \subseteq V \times (\Sigma \cup \Sigma V \cup \{\varepsilon\})$$

generates a regular language.

Question



This is much simpler!
Why don't we define
regular languages
via such grammars?

Question

Both variants (restricting the occurrence of ε on the right-hand side of rules or not) characterize exactly the regular languages.



In the following situations, which variant would you prefer?

- ▶ You want to prove something for all regular languages.
- ▶ You want to specify a grammar to establish that a certain language is regular.
- ▶ You want to write an algorithm that takes a grammar for a regular language as input.

Our Plan

We are going to show that every grammar with rules

$$R \subseteq V \times (\Sigma \cup \Sigma V \cup \{\varepsilon\})$$

generates a regular language.

- ▶ The proof will be **constructive**, i. e. it will tell us how to construct a regular grammar for a language that is given by such a more general grammar.
- ▶ Two steps:
 - ① Eliminate the start variable from the right-hand side of rules.
 - ② Eliminate forbidden occurrences of ε .

Start Variable in Right-Hand Side of Rules

For every type-0 language L there is a grammar where the start variable does not occur on the right-hand side of any rule.

Theorem

For every grammar $G = \langle V, \Sigma, R, S \rangle$ there is a grammar $G' = \langle V', \Sigma, R', S \rangle$ with rules $R' \subseteq (V' \cup \Sigma)^* V' (V' \cup \Sigma)^* \times (V' \setminus \{S\} \cup \Sigma)^*$ such that $\mathcal{L}(G) = \mathcal{L}(G')$.

Note: this theorem is true for **all** grammars.

Start Variable in Right-Hand Side of Rules: Example

Before we prove the theorem, let's illustrate its idea.

Consider $G = \langle \{S, X\}, \{a, b\}, R, S \rangle$ with the following rules in R :

$$bS \rightarrow \varepsilon \quad S \rightarrow XabS \quad bX \rightarrow aSa \quad X \rightarrow abc$$

The new grammar has all original rules except that S is replaced with a new variable S' (allowing to derive everything from S' that could originally be derived from the start variable S):

$$bS' \rightarrow \varepsilon \quad S' \rightarrow XabS' \quad bX \rightarrow aS'a \quad X \rightarrow abc$$

In addition, it has rules that allow to start from the original start variable but switch to S' after the first rule application:

$$S \rightarrow XabS'$$

Start Variable in Right-Hand Side of Rules: Proof

Proof.

Let $G = \langle V, \Sigma, R, S \rangle$ be a grammar and $S' \notin V$ be a new variable. Construct rule set R' from R as follows:

- ▶ for every rule $r \in R$, add a rule r' to R' , where r' is the result of replacing all occurrences of S in r with S' .
- ▶ for every rule $S \rightarrow w \in R$, add a rule $S \rightarrow w'$ to R' , where w' is the result of replacing all occurrences of S in w with S' .

Then $\mathcal{L}(G) = \mathcal{L}(\langle V \cup \{S'\}, \Sigma, R', S \rangle)$. \square

Note that the rules in R' are not fundamentally different from the rules in R . In particular:

- ▶ If $R \subseteq V \times (\Sigma \cup \Sigma V \cup \{\varepsilon\})$ then $R' \subseteq V' \times (\Sigma \cup \Sigma V' \cup \{\varepsilon\})$.
- ▶ If $R \subseteq V \times (V \cup \Sigma)^*$ then $R' \subseteq V' \times (V' \cup \Sigma)^*$.

Epsilon Rules

Theorem

For every grammar G with rules $R \subseteq V \times (\Sigma \cup \Sigma V \cup \{\varepsilon\})$ there is a regular grammar G' with $\mathcal{L}(G) = \mathcal{L}(G')$.

Epsilon Rules: Example

Let's again first illustrate the idea. Consider

$G = \langle \{S, X, Y\}, \{a, b\}, R, S \rangle$ with the following rules in R :

$S \rightarrow \varepsilon \quad S \rightarrow aX \quad X \rightarrow aX \quad X \rightarrow aY \quad Y \rightarrow bY \quad Y \rightarrow \varepsilon$

- 1 The start variable does not occur on a right-hand side. ✓
- 2 Determine the set of variables that can be replaced with the empty word: $V_\varepsilon = \{S, Y\}$.
- 3 Eliminate forbidden rules: ~~$Y \rightarrow aY$~~ ~~$Y \rightarrow bY$~~ ~~$Y \rightarrow \varepsilon$~~
- 4 If a variable from V_ε occurs in the right-hand side, add another rule that directly emulates a subsequent replacement with the empty word: $X \rightarrow a$ and $Y \rightarrow b$

Epsilon Rules

Theorem

For every grammar G with rules $R \subseteq V \times (\Sigma \cup \Sigma V \cup \{\varepsilon\})$ there is a regular grammar G' with $\mathcal{L}(G) = \mathcal{L}(G')$.

Proof.

Let $G = \langle V, \Sigma, R, S \rangle$ be a grammar s.t. $R \subseteq V \times (\Sigma \cup \Sigma V \cup \{\varepsilon\})$. Use the previous proof to construct grammar $G' = \langle V', \Sigma, R', S \rangle$ s.t. $R' \subseteq V' \times (\Sigma \cup \Sigma(V' \setminus \{S\}) \cup \{\varepsilon\})$ and $\mathcal{L}(G') = \mathcal{L}(G)$. Let $V_\varepsilon = \{A \mid A \rightarrow \varepsilon \in R'\}$.

Let R'' be the rule set that is created from R' by removing all rules of the form $A \rightarrow \varepsilon$ ($A \neq S$). Additionally, for every rule of the form $B \rightarrow xA$ with $A \in V_\varepsilon$, $B \in V'$, $x \in \Sigma$ we add a rule $B \rightarrow x$ to R'' .

Then $G'' = \langle V', \Sigma, R'', S \rangle$ is regular and $\mathcal{L}(G) = \mathcal{L}(G'')$. \square

Exercise (Slido)

Consider $G = (\{S, X, Y\}, \{a, b\}, R, S)$ with the following rules in R :

$$\begin{array}{ll} S \rightarrow \varepsilon & S \rightarrow aX \\ X \rightarrow aX & X \rightarrow aY \\ Y \rightarrow bY & Y \rightarrow \varepsilon \end{array}$$


- ▶ Is G a regular grammar?
- ▶ Is $\mathcal{L}(G)$ regular?

Summary

- ▶ Regular grammars restrict the usage of ε in rules.
- ▶ This restriction is **not necessary for the characterization of regular languages** but convenient if we want to prove something for all regular languages.