Formal Languages

Philippe de Groote

2020-2021

Outline

3. Regular expressions and regular languages
    - Definition
    - Some algebraic properties
    - From regular expressions to FSA
    - From FSA to type-3 grammars
    - From type-3 grammars to regular expressions

Definition

The *set of regular epressions* over an alphabet Σ is inductively defined as follows:

- 0 is a regular expression;
- 1 is a regular expression;
- every symbol $a \in \Sigma$ is a regular expression;
- if α is a regular expression so is α^*;
- if α and β are regular expressions so is $(\alpha \cdot \beta)$;
- if α and β are regular expressions so is $(\alpha + \beta)$;

Definition

The *set of regular epressions* over an alphabet Σ is inductively defined as follows:

- 0 is a regular expression;
- 1 is a regular expression;
- every symbol $a \in \Sigma$ is a regular expression;
- if α is a regular expression so is α^*;
- if α and β are regular expressions so is $(\alpha \cdot \beta)$;
- if α and β are regular expressions so is $(\alpha + \beta)$;

We write $\mathrm{rexp}(\Sigma)$ for the set of regular epressions over Σ.

Definition

Language defined by a regular epressions:

- $L(0) = \varnothing$;
- $L(1) = \{\epsilon\}$;
- $L(a) = \{a\}$ for every $a \in \Sigma$;
- $L(\alpha^*) = L(\alpha)^*$;
- $L(\alpha \cdot \beta) = L(\alpha) \cdot L(\beta)$;
- $L(\alpha + \beta) = L(\alpha) \cup L(\beta)$.

Some algebraic properties

$$(\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$$
$$\alpha + 0 = \alpha$$
$$0 + \alpha = \alpha$$
$$\alpha + \beta = \beta + \alpha$$
$$\alpha + \alpha = \alpha$$

$$(\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma)$$
$$\alpha \cdot 1 = \alpha$$
$$1 \cdot \alpha = \alpha$$
$$\alpha \cdot 0 = 0$$
$$0 \cdot \alpha = 0$$

$$\alpha \cdot (\beta + \gamma) = \alpha \cdot \beta + \alpha \cdot \gamma$$
$$(\alpha + \beta) \cdot \gamma = \alpha \cdot \gamma + \beta \cdot \gamma$$

Some algebraic properties

$$0^* = 1$$

$$1^* = 1$$

$$(\alpha^*)^* = \alpha^*$$

$$1 + \alpha \cdot (\alpha^*) = \alpha^*$$

$$1 + \alpha^* \cdot \alpha = \alpha^*$$

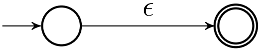
From regular expressions to FSA

Automaton accepting $L(0)$

From regular expressions to FSA

Automaton accepting $L(0)$

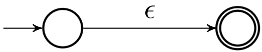
Automaton accepting $L(1)$:



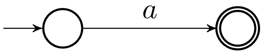
From regular expressions to FSA

Automaton accepting $L(0)$

Automaton accepting $L(1)$:



Automaton accepting $L(a)$:



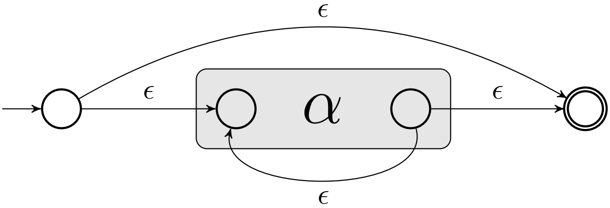
From regular expressions to FSA

Assuming we have an automaton accepting $L(\alpha)$:

From regular expressions to FSA

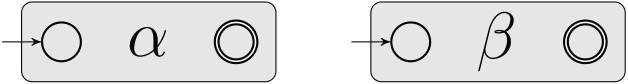
Assuming we have an automaton accepting $L(\alpha)$:

Automaton accepting $L(\alpha^*)$:



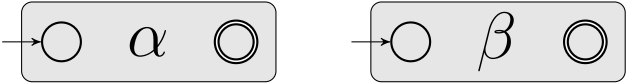
From regular expressions to FSA

Assuming we have automata accepting $L(\alpha)$ and $L(\beta)$:

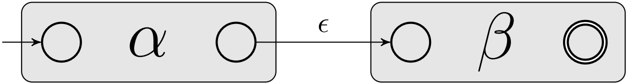


From regular expressions to FSA

Assuming we have automata accepting $L(\alpha)$ and $L(\beta)$:

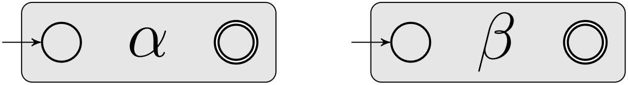


Automaton accepting $L(\alpha \cdot \beta)$:



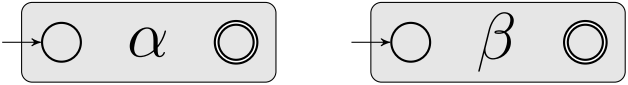
From regular expressions to FSA

Assuming we have automata accepting $L(\alpha)$ and $L(\beta)$:

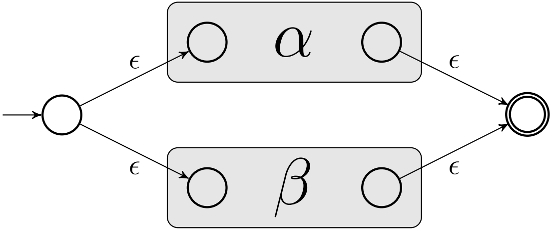


From regular expressions to FSA

Assuming we have automata accepting $L(\alpha)$ and $L(\beta)$:



Automaton accepting $L(\alpha + \beta)$:



From FSA to type-3 grammars

Let $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ be an DFSA. Define a type-3 grammar $G = \langle N, \Sigma_G, P, S \rangle$ as follows:

- $N = Q$
- $\Sigma_G = \Sigma$
- $P = \{A \rightarrow aB : \delta(A, a) = B\} \cup \{A \rightarrow \epsilon : A \in F\}$
- $S = q_0$

From FSA to type-3 grammars

Let $A = \langle Q, \Sigma, \delta, q_0, F\rangle$ be an DFSA. Define a type-3 grammar $G = \langle N, \Sigma_G, P, S\rangle$ as follows:

- $N = Q$
- $\Sigma_G = \Sigma$
- $P = \{A \rightarrow aB : \delta(A, a) = B\} \cup \{A \rightarrow \epsilon : A \in F\}$
- $S = q_0$

Proposition $L(A) = L(G)$.

From FSA to type-3 grammars

PROOF:

From FSA to type-3 grammars

PROOF:

We prove by induction on the length of α that $A \Rightarrow^* \alpha$ if and only if $\hat{\delta}(A, \alpha) \in F$.

From FSA to type-3 grammars

PROOF:

We prove by induction on the length of α that $A \Rightarrow^* \alpha$ if and only if $\hat{\delta}(A, \alpha) \in F$.

Basis:

From FSA to type-3 grammars

PROOF:

We prove by induction on the length of α that $A \Rightarrow^* \alpha$ if and only if $\hat{\delta}(A, \alpha) \in F$.

Basis:

$A \Rightarrow^* \epsilon$ iff $A \Rightarrow \epsilon$

From FSA to type-3 grammars

PROOF:

We prove by induction on the length of α that $A \Rightarrow^* \alpha$ if and only if $\hat{\delta}(A, \alpha) \in F$.

Basis:

$$\begin{aligned} A \Rightarrow^* \epsilon \text{ iff } & A \Rightarrow \epsilon \\ \text{iff } & (A \rightarrow \epsilon) \in P \end{aligned}$$

From FSA to type-3 grammars

PROOF:

We prove by induction on the length of α that $A \Rightarrow^* \alpha$ if and only if $\hat{\delta}(A, \alpha) \in F$.

Basis:

$$\begin{aligned} A \Rightarrow^* \epsilon \text{ iff } & A \Rightarrow \epsilon \\ \text{iff } & (A \rightarrow \epsilon) \in P \\ \text{iff } & A \in F \end{aligned}$$

From FSA to type-3 grammars

PROOF:

We prove by induction on the length of α that $A \Rightarrow^* \alpha$ if and only if $\hat{\delta}(A, \alpha) \in F$.

Basis:

$$\begin{aligned} A \Rightarrow^* \epsilon & \text{ iff } A \Rightarrow \epsilon \\ & \text{ iff } (A \rightarrow \epsilon) \in P \\ & \text{ iff } A \in F \\ & \text{ iff } \hat{\delta}(A, \epsilon) \in F \end{aligned}$$

From FSA to type-3 grammars

Induction:

From FSA to type-3 grammars

Induction:

$A \Rightarrow^* a\alpha'$ iff $A \Rightarrow aB \Rightarrow^* a\alpha',$ for some $(A \rightarrow aB) \in P$

From FSA to type-3 grammars

Induction:

$$\begin{aligned} A \Rightarrow^* a\alpha' \text{ iff } & A \Rightarrow aB \Rightarrow^* a\alpha', \text{ for some } (A \rightarrow aB) \in P \\ \text{iff } & (A \rightarrow aB) \in P \text{ and } B \Rightarrow^* \alpha' \end{aligned}$$

From FSA to type-3 grammars

Induction:

$$\begin{aligned} A \Rightarrow^* a\alpha' &\text{ iff } A \Rightarrow aB \Rightarrow^* a\alpha', \text{ for some } (A \rightarrow aB) \in P \\ &\text{ iff } (A \rightarrow aB) \in P \text{ and } B \Rightarrow^* \alpha' \\ &\text{ iff } \delta(A, a) = B \text{ and } B \Rightarrow^* \alpha' \end{aligned}$$

From FSA to type-3 grammars

Induction:

$$
\begin{aligned}
A \Rightarrow^* a\alpha' \text{ iff } & A \Rightarrow aB \Rightarrow^* a\alpha', \text{ for some } (A \rightarrow aB) \in P \\
\text{iff } & (A \rightarrow aB) \in P \text{ and } B \Rightarrow^* \alpha' \\
\text{iff } & \delta(A, a) = B \text{ and } B \Rightarrow^* \alpha' \\
\text{iff } & \delta(A, a) = B \text{ and } \hat{\delta}(B, \alpha') \in F \\
& \qquad\qquad \text{by induction hypothesis}
\end{aligned}
$$

From FSA to type-3 grammars

Induction:

$$
\begin{aligned}
A \Rightarrow^* a\alpha' \text{ iff } & A \Rightarrow aB \Rightarrow^* a\alpha', \text{ for some } (A \rightarrow aB) \in P \\
\text{iff } & (A \rightarrow aB) \in P \text{ and } B \Rightarrow^* \alpha' \\
\text{iff } & \delta(A, a) = B \text{ and } B \Rightarrow^* \alpha' \\
\text{iff } & \delta(A, a) = B \text{ and } \hat{\delta}(B, \alpha') \in F \\
& \qquad\qquad \text{by induction hypothesis} \\
\text{iff } & \hat{\delta}(\delta(A, a), \alpha') \in F
\end{aligned}
$$

From FSA to type-3 grammars

Induction:

$A \Rightarrow^* a\alpha'$ iff $A \Rightarrow aB \Rightarrow^* a\alpha'$, for some $(A \rightarrow aB) \in P$

iff $(A \rightarrow aB) \in P$ and $B \Rightarrow^* \alpha'$

iff $\delta(A, a) = B$ and $B \Rightarrow^* \alpha'$

iff $\delta(A, a) = B$ and $\hat{\delta}(B, \alpha') \in F$
by induction hypothesis

iff $\hat{\delta}(\delta(A, a), \alpha') \in F$

iff $\hat{\delta}(A, a\alpha') \in F$

From type-3 grammars to regular expressions

Consider $\alpha \in \mathrm{rexp}(\Sigma)^*$. One defines $L(\alpha)$ as follows:

- $L(\epsilon) = \{\epsilon\}$
- $L(e\alpha') = L(e) \cdot L(\alpha')$

We consider type-3 grammars whose set of terminal symbols is the set of regular expressions over some alphabet Σ:

$$G = \langle N, \mathrm{rexp}(\Sigma), P, S \rangle$$

From type-3 grammars to regular expressions

Consider $\alpha \in \mathrm{rexp}(\Sigma)^*$. One defines $L(\alpha)$ as follows:

- $L(\epsilon) = \{\epsilon\}$
- $L(e\alpha') = L(e) \cdot L(\alpha')$

We consider type-3 grammars whose set of terminal symbols is the set of regular expressions over some alphabet Σ:

$$G = \langle N, \mathrm{rexp}(\Sigma), P, S \rangle$$

For such grammars, one may define:

$$L_E(G) = \bigcup_{e \in L(G)} L(e)$$

From type-3 grammars to regular expressions

Example:

From type-3 grammars to regular expressions

Example:

$$G = \left\{ \begin{array}{l} S \rightarrow (a+b)\, S \\ S \rightarrow (c \cdot d) \end{array} \right.$$

From type-3 grammars to regular expressions

Example:

$$G = \left\{ \begin{array}{l} S \rightarrow (a{+}b)\, S \\ S \rightarrow (c{\cdot}d) \end{array} \right.$$

$$L(G) = \{(c{\cdot}d), (a{+}b)(c{\cdot}d), (a{+}b)(a{+}b)(c{\cdot}d), (a{+}b)(a{+}b)(a{+}b)(c{\cdot}d), \ldots\}$$

From type-3 grammars to regular expressions

Example:

$$G = \left\{ \begin{array}{l} S \rightarrow (a+b)\, S \\ S \rightarrow (c{\cdot}d) \end{array} \right.$$

$$L(G) = \{(c{\cdot}d), (a+b)(c{\cdot}d), (a+b)(a+b)(c{\cdot}d), (a+b)(a+b)(a+b)(c{\cdot}d), \ldots\}$$

$$L_E(G) = \{cd, acd, bcd, aacd, abcd, bacd, bbcd, aaacd, aabcd, abacd, \ldots\}$$

From type-3 grammars to regular expressions

Example:

$$G = \begin{cases} S \rightarrow (a{+}b)\, S \\ S \rightarrow (c{\cdot}d) \end{cases}$$

$$L(G) = \{(c{\cdot}d), (a{+}b)(c{\cdot}d), (a{+}b)(a{+}b)(c{\cdot}d), (a{+}b)(a{+}b)(a{+}b)(c{\cdot}d), \ldots\}$$

$$L_E(G) = \{cd, acd, bcd, aacd, abcd, bacd, bbcd, aaacd, aabcd, abacd, \ldots\}$$

Remark:

Since $\Sigma \subset \mathrm{rexp}(\Sigma)$, every grammar over Σ may be seen as a grammar over $\mathrm{rexp}(\Sigma)$, with $L_E(G) = L(G)$.

From type-3 grammars to regular expressions

Elimination of non-recursive non-terminal symbols

From type-3 grammars to regular expressions

Elimination of non-recursive non-terminal symbols

Given a type-3 grammar G, one says that a rule is recursive if it is of the form $A \to aA$. A non-terminal symbol A is said to be recursive in case there is at least one recursive rule whose lefthand side is A.

From type-3 grammars to regular expressions

Elimination of non-recursive non-terminal symbols

Given a type-3 grammar G, one says that a rule is recursive if it is of the form $A \to aA$. A non-terminal symbol A is said to be recursive in case there is at least one recursive rule whose lefthand side is A.

Let $G_1 = \langle N_1, \mathrm{rexp}(\Sigma), P_1, S_1 \rangle$ be a type-3 grammar, and let $A \in N_1$ be a non-recursive non-terminal symbol different from S.

From type-3 grammars to regular expressions

Elimination of non-recursive non-terminal symbols

Given a type-3 grammar G, one says that a rule is recursive if it is of the form $A \to aA$. A non-terminal symbol A is said to be recursive in case there is at least one recursive rule whose lefthand side is A.

Let $G_1 = \langle N_1, \mathrm{rexp}(\Sigma), P_1, S_1 \rangle$ be a type-3 grammar, and let $A \in N_1$ be a non-recursive non-terminal symbol different from S.

Let $P_A = \{A \to e_0 B_0, \ldots, A \to e_{m-1} B_{m-1}, A \to f_0, \ldots, A \to f_{n-1}\} \subset P_1$ be the set of all the production rules whose lefthand side is A.

From type-3 grammars to regular expressions

Elimination of non-recursive non-terminal symbols

Given a type-3 grammar G, one says that a rule is recursive if it is of the form $A \rightarrow aA$. A non-terminal symbol A is said to be recursive in case there is at least one recursive rule whose lefthand side is A.

Let $G_1 = \langle N_1, \mathrm{rexp}(\Sigma), P_1, S_1 \rangle$ be a type-3 grammar, and let $A \in N_1$ be a non-recursive non-terminal symbol different from S.

Let $P_A = \{A \rightarrow e_0 B_0, \ldots, A \rightarrow e_{m-1} B_{m-1}, A \rightarrow f_0, \ldots, A \rightarrow f_{n-1}\} \subset P_1$ be the set of all the production rules whose lefthand side is A.

Let $Q_A = \{C_0 \rightarrow a_0 A, \ldots, C_{l-1} \rightarrow a_{l-1} A\} \subset P_1$ be the set of all the production rules whose righthand side contains A.

From type-3 grammars to regular expressions

Elimination of non-recursive non-terminal symbols

Given a type-3 grammar G, one says that a rule is recursive if it is of the form $A \rightarrow aA$. A non-terminal symbol A is said to be recursive in case there is at least one recursive rule whose lefthand side is A.

Let $G_1 = \langle N_1, \mathrm{rexp}(\Sigma), P_1, S_1 \rangle$ be a type-3 grammar, and let $A \in N_1$ be a non-recursive non-terminal symbol different from S.

Let $P_A = \{A \rightarrow e_0 B_0, \ldots, A \rightarrow e_{m-1} B_{m-1}, A \rightarrow f_0, \ldots, A \rightarrow f_{n-1}\} \subset P_1$ be the set of all the production rules whose lefthand side is A.

Let $Q_A = \{C_0 \rightarrow a_0 A, \ldots, C_{l-1} \rightarrow a_{l-1} A\} \subset P_1$ be the set of all the production rules whose righthand side contains A.

Define $R_A = \bigcup_{i \in l}((\bigcup_{j \in m}\{C_i \rightarrow (a_i \cdot e_j) B_j\}) \cup (\bigcup_{j \in n}\{C_i \rightarrow (a_i \cdot f_j)\}))$

From type-3 grammars to regular expressions

One defines a new grammar $G_2 = \langle N_2, \mathrm{rexp}(\Sigma), P_2, S_2 \rangle$ as follows:

- $N_2 = N_1 \setminus \{A\}$
- $P_2 = (P \setminus (P_A \cup Q_A)) \cup R_A$
- $S_2 = S_1$

From type-3 grammars to regular expressions

One defines a new grammar $G_2 = \langle N_2, \mathrm{rexp}(\Sigma), P_2, S_2 \rangle$ as follows:

- $N_2 = N_1 \setminus \{A\}$
- $P_2 = (P \setminus (P_A \cup Q_A)) \cup R_A$
- $S_2 = S_1$

Proposition $L_E(G_1) = L_E(G_2)$.

From type-3 grammars to regular expressions

PROOF:

From type-3 grammars to regular expressions

PROOF:

We prove that $L_E(G_1) \subset L_E(G_2)$ and $L_E(G_2) \subset L_E(G_1)$.

From type-3 grammars to regular expressions

PROOF:

We prove that $L_E(G_1) \subset L_E(G_2)$ and $L_E(G_2) \subset L_E(G_1)$.

PART 1: $L_E(G_1) \subset L_E(G_2)$

From type-3 grammars to regular expressions

PROOF:

We prove that $L_E(G_1) \subset L_E(G_2)$ and $L_E(G_2) \subset L_E(G_1)$.

PART 1: $L_E(G_1) \subset L_E(G_2)$

Let us write $\Rightarrow_1$ and $\Rightarrow_2$ for the generation relations of G_1 and G_2, respectively. We prove that for every $B \in N_2$ and every $\alpha_1 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_1^* \alpha_1$, there exists $\alpha_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_2^* \alpha_2$ and $L(\alpha_1) = L(\alpha_2)$. The proof proceed by induction on the number of occurences of rules from Q_A that appear in the derivation $B \Rightarrow_1^* \alpha_1$.

From type-3 grammars to regular expressions

PROOF:

We prove that $L_E(G_1) \subset L_E(G_2)$ and $L_E(G_2) \subset L_E(G_1)$.

PART 1: $L_E(G_1) \subset L_E(G_2)$

Let us write $\Rightarrow_1$ and $\Rightarrow_2$ for the generation relations of G_1 and G_2, respectively. We prove that for every $B \in N_2$ and every $\alpha_1 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_1^* \alpha_1$, there exists $\alpha_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_2^* \alpha_2$ and $L(\alpha_1) = L(\alpha_2)$. The proof proceed by induction on the number of occurences of rules from Q_A that appear in the derivation $B \Rightarrow_1^* \alpha_1$.

Basis:

From type-3 grammars to regular expressions

PROOF:

We prove that $L_E(G_1) \subset L_E(G_2)$ and $L_E(G_2) \subset L_E(G_1)$.

PART 1: $L_E(G_1) \subset L_E(G_2)$

Let us write $\Rightarrow_1$ and $\Rightarrow_2$ for the generation relations of G_1 and G_2, respectively. We prove that for every $B \in N_2$ and every $\alpha_1 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_1^* \alpha_1$, there exists $\alpha_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_2^* \alpha_2$ and $L(\alpha_1) = L(\alpha_2)$. The proof proceed by induction on the number of occurences of rules from Q_A that appear in the derivation $B \Rightarrow_1^* \alpha_1$.

Basis:

There is no occurence of any rule from Q_A in the derivation $B \Rightarrow_1^* \alpha_1$. Then, there is no occurrence of any rule from P_A either. Consequently, the derivation $B \Rightarrow_1^* \alpha_1$ is also a derivation of G_2, and we take $\alpha_2 = \alpha_1$.

From type-3 grammars to regular expressions

Induction:

From type-3 grammars to regular expressions

Induction:

If there is at least one occurence of a rule from Q_A in the derivation $B \Rightarrow_1^* \alpha_1$, it must obey one of the two following forms:

$$
\begin{array}{ll}
(1) & B \Rightarrow_1^* \beta C_i \Rightarrow_1 \beta a_i A \Rightarrow_1 \beta a_i e_j B_j \Rightarrow_1^* \beta a_i e_j \gamma_1 \\
(2) & B \Rightarrow_1^* \beta C_i \Rightarrow_1 \beta a_i A \Rightarrow_1 \beta a_i f_j
\end{array}
$$

where the occurrence of $(C_i \to a_i A) \in Q_A$ is the leftmost occurrence of a rule from Q_A. Consequently, $B \Rightarrow_1^* \beta C_i$ is also a derivation of G_2.

From type-3 grammars to regular expressions

Induction:

If there is at least one occurence of a rule from Q_A in the derivation $B \Rightarrow_1^* \alpha_1$, it must obey one of the two following forms:

$$\begin{array}{ll} (1) & B \Rightarrow_1^* \beta C_i \Rightarrow_1 \beta a_i A \Rightarrow_1 \beta a_i e_j B_j \Rightarrow_1^* \beta a_i e_j \gamma_1 \\ (2) & B \Rightarrow_1^* \beta C_i \Rightarrow_1 \beta a_i A \Rightarrow_1 \beta a_i f_j \end{array}$$

where the occurrence of $(C_i \rightarrow a_i A) \in Q_A$ is the leftmost occurrence of a rule from Q_A. Consequently, $B \Rightarrow_1^* \beta C_i$ is also a derivation of G_2.

In the first case, we have $\alpha_1 = \beta a_i e_j \gamma_1$ and $B_j \Rightarrow_1^* \gamma_1$. By induction hypothesis, there exists $\gamma_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B_j \Rightarrow_2^* \gamma_2$ and $L(\gamma_1) = L(\gamma_2)$. Hence:

$$B \Rightarrow_2^* \beta C_i \Rightarrow_2 \beta(a_i \cdot e_j) B_j \Rightarrow_2^* \beta(a_i \cdot e_j)\gamma_2$$

Then, we take $\alpha_2 = \beta(a_i \cdot e_j)\gamma_2$. Indeed $L(\alpha_1) = L(\beta a_i e_j \gamma_1) = L(\beta)L(a_i)L(e_j)L(\gamma_1) = L(\beta)L(a_i \cdot e_j)L(\gamma_2) = L(\beta(a_i \cdot e_j)\gamma_2) = L(\alpha_2)$

From type-3 grammars to regular expressions

In the second case, we have $\alpha_1 = \beta a_i f_j$. Then, we take $\alpha_2 = \beta(a_i \cdot f_j)$ Indeed, we have that

$$B \Rightarrow_2^* \beta C_i \Rightarrow_2 \beta(a_i \cdot f_j)$$

and that $L(\beta a_i f_j) = L(\beta(a_i \cdot f_j))$.

From type-3 grammars to regular expressions

In the second case, we have $\alpha_1 = \beta a_i f_j$. Then, we take $\alpha_2 = \beta(a_i \cdot f_j)$ Indeed, we have that

$$B \Rightarrow_2^* \beta C_i \Rightarrow_2 \beta(a_i \cdot f_j)$$

and that $L(\beta a_i f_j) = L(\beta(a_i \cdot f_j))$.

PART 2: $L_E(G_2) \subset L_E(G_1)$

From type-3 grammars to regular expressions

In the second case, we have $\alpha_1 = \beta a_i f_j$. Then, we take $\alpha_2 = \beta(a_i \cdot f_j)$ Indeed, we have that

$$B \Rightarrow_2^* \beta C_i \Rightarrow_2 \beta(a_i \cdot f_j)$$

and that $L(\beta a_i f_j) = L(\beta(a_i \cdot f_j))$.

PART 2: $L_E(G_2) \subset L_E(G_1)$

We prove that for every $B \in N_2$ and every $\alpha_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_2^* \alpha_2$, there exists $\alpha_1 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_1^* \alpha_1$ and $L(\alpha_2) = L(\alpha_1)$. The proof, which proceed by induction on the number of occurences of rules from R_A that appear in the derivation $B \Rightarrow_2^* \alpha_2$, is similar to the proof of Part 1.

From type-3 grammars to regular expressions

Elimination of recursive rules

From type-3 grammars to regular expressions

Elimination of recursive rules

Let $G_1 = \langle N_1, \mathrm{rexp}(\Sigma), P_1, S_1\rangle$ be a type-3 grammar, and let $A \in N_1$ be a recursive non-terminal symbol different from S.

From type-3 grammars to regular expressions

Elimination of recursive rules

Let $G_1 = \langle N_1, \mathrm{rexp}(\Sigma), P_1, S_1 \rangle$ be a type-3 grammar, and let $A \in N_1$ be a recursive non-terminal symbol different from S.

Let $P_A = \{A \to e_0 B_0, \ldots, A \to e_{l-1} B_{l-1}, A \to f_0, \ldots, A \to f_{m-1}\} \subset P_1$ be the set of all the non-recursive production rules whose lefthand side is A.

From type-3 grammars to regular expressions

Elimination of recursive rules

Let $G_1 = \langle N_1, \mathrm{rexp}(\Sigma), P_1, S_1 \rangle$ be a type-3 grammar, and let $A \in N_1$ be a recursive non-terminal symbol different from S.

Let $P_A = \{A \rightarrow e_0 B_0, \ldots, A \rightarrow e_{l-1} B_{l-1}, A \rightarrow f_0, \ldots, A \rightarrow f_{m-1}\} \subset P_1$ be the set of all the non-recursive production rules whose lefthand side is A.

Let $Q_A = \{A \rightarrow g_0 A, \ldots, A \rightarrow g_{n-1} A\} \subset P_1$ be the set of all the recursive production rules whose lefthand side is A.

From type-3 grammars to regular expressions

Elimination of recursive rules

Let $G_1 = \langle N_1, \mathrm{rexp}(\Sigma), P_1, S_1 \rangle$ be a type-3 grammar, and let $A \in N_1$ be a recursive non-terminal symbol different from S.

Let $P_A = \{A \rightarrow e_0 B_0, \ldots, A \rightarrow e_{l-1} B_{l-1}, A \rightarrow f_0, \ldots, A \rightarrow f_{m-1}\} \subset P_1$ be the set of all the non-recursive production rules whose lefthand side is A.

Let $Q_A = \{A \rightarrow g_0 A, \ldots, A \rightarrow g_{n-1} A\} \subset P_1$ be the set of all the recursive production rules whose lefthand side is A.

Define $R_A = (\bigcup_{i \in l} \{A \rightarrow ((g_0 + \cdots + g_{n-1})^* \cdot e_i) B_i\}) \cup$
$(\bigcup_{i \in m} \{A \rightarrow ((g_0 + \cdots + g_{n-1})^* \cdot f_i)\})$

From type-3 grammars to regular expressions

One defines a new grammar $G_2 = \langle N_2, \mathrm{rexp}(\Sigma), P_2, S_2 \rangle$ as follows:

- $N_2 = N_1$
- $P_2 = (P \setminus (P_A \cup Q_A)) \cup R_A$
- $S_2 = S_1$

From type-3 grammars to regular expressions

One defines a new grammar $G_2 = \langle N_2, \mathrm{rexp}(\Sigma), P_2, S_2 \rangle$ as follows:

- $N_2 = N_1$
- $P_2 = (P \setminus (P_A \cup Q_A)) \cup R_A$
- $S_2 = S_1$

Proposition $L_E(G_1) = L_E(G_2)$.

From type-3 grammars to regular expressions

PROOF:

From type-3 grammars to regular expressions

PROOF:

PART 1: $L_E(G_1) \subset L_E(G_2)$

From type-3 grammars to regular expressions

PROOF:

PART 1: $L_E(G_1) \subset L_E(G_2)$

We prove that for every $B \in N_1$ and every $\alpha_1 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_1^* \alpha_1$, there exists $\alpha_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_2^* \alpha_2$ and $L(\alpha_1) \subset L(\alpha_2)$. The proof proceed by induction on the number of occurences of rules from P_A that appear in the derivation $B \Rightarrow_1^* \alpha_1$.

From type-3 grammars to regular expressions

PROOF:

PART 1: $L_E(G_1) \subset L_E(G_2)$

We prove that for every $B \in N_1$ and every $\alpha_1 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_1^* \alpha_1$, there exists $\alpha_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_2^* \alpha_2$ and $L(\alpha_1) \subset L(\alpha_2)$. The proof proceed by induction on the number of occurences of rules from P_A that appear in the derivation $B \Rightarrow_1^* \alpha_1$.

Basis:

From type-3 grammars to regular expressions

PROOF:

PART 1: $L_E(G_1) \subset L_E(G_2)$

We prove that for every $B \in N_1$ and every $\alpha_1 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_1^* \alpha_1$, there exists $\alpha_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_2^* \alpha_2$ and $L(\alpha_1) \subset L(\alpha_2)$. The proof proceed by induction on the number of occurences of rules from P_A that appear in the derivation $B \Rightarrow_1^* \alpha_1$.

Basis:

There is no occurence of any rule from P_A in the derivation $B \Rightarrow_1^* \alpha_1$. Then, there is no occurrence of any rule from Q_A either. Consequently, the derivation $B \Rightarrow_1^* \alpha_1$ is also a derivation of G_2, and we take $\alpha_2 = \alpha_1$.

From type-3 grammars to regular expressions

Induction:

From type-3 grammars to regular expressions

Induction:

If there is at least one occurence of a rule from P_A in the derivation $B \Rightarrow_1^* \alpha_1$, it must obey one of the two following forms:

$$\begin{array}{ll}(1) & B \Rightarrow_1^* \beta A \Rightarrow_1 \beta g_{i_0} A \Rightarrow_1 \cdots \Rightarrow_1 \beta g_{i_0} \ldots g_{i_{k-1}} A \\ & \quad \Rightarrow_1 \beta g_{i_0} \ldots g_{i_{k-1}} e_i B_i \Rightarrow_1 \beta g_{i_0} \ldots g_{i_{k-1}} e_i \gamma_1 \\ (2) & B \Rightarrow_1^* \beta A \Rightarrow_1 \beta g_{i_0} A \Rightarrow_1 \cdots \Rightarrow_1 \beta g_{i_0} \ldots g_{i_{k-1}} A \\ & \quad \Rightarrow_1 \beta g_{i_0} \ldots g_{i_{k-1}} f_i \end{array}$$

where the occurrence of $(A \rightarrow e_i B_i) \in P_A$ (respectively, $(A \rightarrow f_i) \in P_A$) is the leftmost occurrence of a rule from P_A, and the occurrence of $(A \rightarrow g_{i_0} A) \in Q_A$ is the leftmost occurrence of a rule from Q_A. Consequently, $B \Rightarrow_1^* \beta A$ is also a derivation of G_2.

From type-3 grammars to regular expressions

In the first case, we have $\alpha_1 = \beta g_{i_0} \ldots g_{i_{k-1}} e_i \gamma_1$ and $B_i \Rightarrow_1^* \gamma_1$. By induction hypothesis, there exists $\gamma_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B_i \Rightarrow_2^* \gamma_2$ and $L(\gamma_1) \subset L(\gamma_2)$. Hence:

$$B \Rightarrow_2^* \beta A \Rightarrow_2 \beta((g_0 + \cdots + g_{n-1})^* \cdot e_i) B_i \Rightarrow_2 \beta((g_0 + \cdots + g_{n-1})^* \cdot e_i)\gamma_2$$

Then, we take

$$\alpha_2 = \beta((g_0 + \cdots + g_{n-1})^* \cdot e_i)\gamma_2$$

Indeed

$$L(\alpha_1) \subset L(\alpha_2)$$

because

$$L(g_{i_0} \ldots g_{i_{k-1}}) \subset L((g_0 + \cdots + g_{n-1})^*) \text{ and } L(\alpha_1) \subset L(\alpha_2)$$

From type-3 grammars to regular expressions

In the first case, we have $\alpha_1 = \beta g_{i_0} \ldots g_{i_{k-1}} e_i \gamma_1$ and $B_i \Rightarrow_1^* \gamma_1$. By induction hypothesis, there exists $\gamma_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B_i \Rightarrow_2^* \gamma_2$ and $L(\gamma_1) \subset L(\gamma_2)$. Hence:

$$B \Rightarrow_2^* \beta A \Rightarrow_2 \beta((g_0 + \cdots + g_{n-1})^* \cdot e_i) B_i \Rightarrow_2 \beta((g_0 + \cdots + g_{n-1})^* \cdot e_i) \gamma_2$$

Then, we take

$$\alpha_2 = \beta((g_0 + \cdots + g_{n-1})^* \cdot e_i) \gamma_2$$

Indeed

$$L(\alpha_1) \subset L(\alpha_2)$$

because

$$L(g_{i_0} \ldots g_{i_{k-1}}) \subset L((g_0 + \cdots + g_{n-1})^*) \text{ and } L(\alpha_1) \subset L(\alpha_2)$$

Similarly, in the second case, we have

$$B \Rightarrow_2^* \beta A \Rightarrow_2 \beta((g_0 + \cdots + g_{n-1})^* \cdot f_i)$$

And, we take

$$\alpha_2 = \beta((g_0 + \cdots + g_{n-1})^* \cdot f_i)$$

From type-3 grammars to regular expressions

PART 2: $L_E(G_2) \subset L_E(G_1)$

From type-3 grammars to regular expressions

PART 2: $L_E(G_2) \subset L_E(G_1)$

We prove that for every $B \in N_2$, every $\alpha_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_2^* \alpha_2$, and every $\omega \in L(\alpha_2)$, there exists $\alpha_1 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_1^* \alpha_1$ and $\omega \in L(\alpha_1)$. The proof proceed by induction on the number of occurences of rules from R_A that appear in the derivation $B \Rightarrow_2^* \alpha_2$.

From type-3 grammars to regular expressions

PART 2: $L_E(G_2) \subset L_E(G_1)$

We prove that for every $B \in N_2$, every $\alpha_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_2^* \alpha_2$, and every $\omega \in L(\alpha_2)$, there exists $\alpha_1 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_1^* \alpha_1$ and $\omega \in L(\alpha_1)$. The proof proceed by induction on the number of occurences of rules from R_A that appear in the derivation $B \Rightarrow_2^* \alpha_2$.

Basis:

From type-3 grammars to regular expressions

PART 2: $L_E(G_2) \subset L_E(G_1)$

We prove that for every $B \in N_2$, every $\alpha_2 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_2^* \alpha_2$, and every $\omega \in L(\alpha_2)$, there exists $\alpha_1 \in \mathrm{rexp}(\Sigma)^*$ such that $B \Rightarrow_1^* \alpha_1$ and $\omega \in L(\alpha_1)$. The proof proceed by induction on the number of occurences of rules from R_A that appear in the derivation $B \Rightarrow_2^* \alpha_2$.

Basis:

There is no occurence of any rule from R_A in the derivation $B \Rightarrow_2^* \alpha_2$. Consequently, the derivation $B \Rightarrow_2^* \alpha_2$ is also a derivation of G_1, and we take $\alpha_1 = \alpha_2$.

From type-3 grammars to regular expressions

Induction:

From type-3 grammars to regular expressions

Induction:

If there is at least one occurence of a rule from R_A in the derivation $B \Rightarrow_2^* \alpha_2$, it must obey one of the two following forms:

$$
\begin{array}{ll}
(1) & B \Rightarrow_2^* \beta A \Rightarrow_2 \beta((g_0+\cdots+g_{n-1})^* \cdot e_i)B_i \Rightarrow_2 \beta((g_0+\cdots+g_{n-1})^* \cdot e_i)\gamma_2 \\
(2) & B \Rightarrow_2^* \beta A \Rightarrow_2 \beta((g_0+\cdots+g_{n-1})^* \cdot f_i)
\end{array}
$$

where the occurrence of $(A \rightarrow ((g_0+\cdots+g_{n-1})^* \cdot e_i)B_i) \in R_A$ (respectively, $(A \rightarrow ((g_0+\cdots+g_{n-1})^* \cdot f_i)) \in R_A$) is the leftmost occurrence of a rule from R_A. Consequently, $B \Rightarrow_2^* \beta A$ is also a derivation of G_1.

From type-3 grammars to regular expressions

In the first case, we have $\alpha_2 = \beta((g_0 + \cdots + g_{n-1})^* \cdot e_i)\gamma_2$ and $B_i \Rightarrow_2^* \gamma_2$. Now, let $\omega \in L(\alpha_2)$. It must obey the following form:

$$\omega = \omega_1 g_{i_0} \ldots g_{i_{k-1}} \omega_2 \omega_3$$

where:

- $\omega_1 \in L(\beta)$;
- the sequence of g_i's is possibly empty;
- $\omega_2 \in L(e_i)$;
- $\omega_3 \in L(\gamma_2)$.

By induction hypothesis, there exists $\gamma_1 \in \mathrm{rexp}(\Sigma)^*$ such that $B_i \Rightarrow_1^* \gamma_1$ and $\omega_3 \in L(\gamma_1)$. Then, we take

$$\alpha_1 = \beta g_{i_0} \ldots g_{i_{k-1}} e_i \gamma_1$$

Indeed

$$\begin{aligned} B &\Rightarrow_1^* \beta A \Rightarrow_1 \beta g_{i_0} A \Rightarrow_1 \cdots \Rightarrow_1 \beta g_{i_0} \ldots g_{i_{k-1}} A \\ &\Rightarrow_1 \beta g_{i_0} \ldots g_{i_{k-1}} e_i B_i \Rightarrow_1 \beta g_{i_0} \ldots g_{i_{k-1}} e_i \gamma_1 \end{aligned}$$

From type-3 grammars to regular expressions

In the first case, we have $\alpha_2 = \beta((g_0 + \cdots + g_{n-1})^* \cdot e_i)\gamma_2$ and $B_i \Rightarrow_2^* \gamma_2$. Now, let $\omega \in L(\alpha_2)$. It must obey the following form:

$$\omega = \omega_1 g_{i_0} \dots g_{i_{k-1}} \omega_2 \omega_3$$

where:

- $\omega_1 \in L(\beta)$;
- the sequence of g_i's is possibly empty;
- $\omega_2 \in L(e_i)$;
- $\omega_3 \in L(\gamma_2)$.

By induction hypothesis, there exists $\gamma_1 \in \mathrm{rexp}(\Sigma)^*$ such that $B_i \Rightarrow_1^* \gamma_1$ and $\omega_3 \in L(\gamma_1)$. Then, we take

$$\alpha_1 = \beta g_{i_0} \dots g_{i_{k-1}} e_i \gamma_1$$

Indeed

$$\begin{aligned} B &\Rightarrow_1^* \beta A \Rightarrow_1 \beta g_{i_0} A \Rightarrow_1 \cdots \Rightarrow_1 \beta g_{i_0} \dots g_{i_{k-1}} A \\ &\Rightarrow_1 \beta g_{i_0} \dots g_{i_{k-1}} e_i B_i \Rightarrow_1 \beta g_{i_0} \dots g_{i_{k-1}} e_i \gamma_1 \end{aligned}$$

The second case, is similar.