Automata and Grammars (BIE-AAG)

4. Regular expressions

Jan Holub

Department of Theoretical Computer Science
Faculty of Information Technology
Czech Technical University in Prague

Regular expressions

Definition

Regular expression V *over alphabet* Σ is defined as follows:

1. $\emptyset, \varepsilon, a$ are regular expressions for all $a \in \Sigma$.
2. If x, y are regular expressions over Σ, then:

   (a) $(x + y)$ (union, alternation),
   (b) $(x.y)$ (concatenation), and
   (c) $(x)^*$ (Kleene star)

   are regular expressions over Σ.

Note: If there is no ambiguity then parentheses and dots may be omitted.

Regular expressions

Only Contains Empty String

Definition

Value $h(x)$ *of regular expression* x is defined as follows:

1. $h(\emptyset) = \emptyset, h(\varepsilon) = \{\varepsilon\}, h(a) = \{a\}, a \in \Sigma$
2. $h(x + y) = h(x) \cup h(y)$,
   $h(x.y) = h(x).h(y)$, → Concatenation
   $h(x^*) = (h(x))^*$, where x, y are regular expressions.

Empty language

Equivalence of regular expressions

Definition (identical, equivalent, and similar r.e.)
Regular expressions x, y are called *identical* (denoted by $x \equiv y$) if x a y are two exactly same strings of symbols.
Regular expressions x, y are called *equivalent* (denoted by $x = y$) if they have the same value, $h(x) = h(y)$, that is, the regular sets described by these equations are identical.
Regular expressions x, y are called *similar* (denoted by $x \cong y$) if they can be converted into each other using the following identities:

$$\begin{aligned}
A_1 &: \quad x + x = x \\
A_2 &: \quad x + y = y + x \\
A_3 &: \quad (x + y) + z = x + (y + z) \\
A_4 &: \quad x + \emptyset = x \\
A_5 &: \quad x.\emptyset = \emptyset.x = \emptyset \\
A_6 &: \quad x.\varepsilon = \varepsilon.x = x
\end{aligned}$$

Conversions of regular expressions

Example

Is expression $(\varepsilon + \emptyset)(0+1)^*1 + (0+1)^*\emptyset$ similar to expression $(0+1)^*1$?

$$
\begin{aligned}
&(\varepsilon + \emptyset)(0+1)^*1 + (0+1)^*\emptyset = && x.\emptyset = \emptyset \\
&= (\varepsilon + \emptyset)(0+1)^*1 + \emptyset = && x + \emptyset = x \\
&= (\varepsilon + \emptyset)(0+1)^*1 = && x + \emptyset = x \\
&= \varepsilon(0+1)^*1 = && \varepsilon.x = x \\
&= (0+1)^*1.
\end{aligned}
$$

Regular expressions — properties

Observation

Let x, y, z be regular expressions. It holds:

$P_1:$	$x + (y + z) = (x + y) + z$	(associativity of union),
$P_2:$	$x + y = y + x$	(commutativity of union),
$P_3:$	$x + \emptyset = x$	($\emptyset$ is the identity element of union),
$P_4:$	$x + x = x$	(idempotence of union),
$P_5:$	$x.(y.z) = (x.y).z$	(associativity of concatenation),
$P_6:$	$\varepsilon x = x\varepsilon = x$	(ε is the identity element of concatenation),
$P_7:$	$\emptyset x = x\emptyset = \emptyset$	($\emptyset$ is the identity element of concatenation),
$P_8:$	$x.(y + z) = x.y + x.z$	(left distributivity),
$P_9:$	$(x + y).z = x.z + y.z$	(right distributivity),
$P_{10}:$	$x^* = \varepsilon + x^*x$,	
$P_{11}:$	$x^* = (\varepsilon + x)^*$.	

Theorem

Let x, α, β be regular expressions. It holds:

$V_1:$	$x = x\alpha + \beta \Rightarrow x = \beta\alpha^*$	(solution of left regular equation),
$V_2:$	$x = \alpha x + \beta \Rightarrow x = \alpha^*\beta$	(solution of right regular equation).

Conversions of regular expressions

Theorems in the theory of regular expressions

$$
\begin{array}{ll}
V_1: & \emptyset^* = \varepsilon \\
V_2: & x^* + x = x^* \\
V_3: & (x^*)^* = x^* \\
V_4: & (x+y)^* = (x^* y^*)^* \\
V_5: & x^* y = y + x^* x y \\
V_6: & x^* y = y + x x^* y \\
V_7: & x^* y = (x^n)^* . (y + xy + x^2 y + \cdots + x^{n-1} y) \\
V_8: & x x^* = x^*, \text{if } \varepsilon \in h(x) \\
V_9: & (xy)^* x = x(yx)^* \\
V_{10}: & (x+y)^* = (x^* + y^*)^*
\end{array}
$$

Conversions of regular expressions

Example

Regular expressions

$x = \varepsilon + 1^*(011)^*(1^*(011)^*)^*$

$y = (1 + 011)^*$

Are they equivalent?

$$\begin{aligned} x &= \varepsilon + 1^*(011)^*(1^*(011)^*)^* \\ &= (1^*(011)^*)^* \\ &= (1 + 011)^* \\ &= y. \end{aligned}$$

$$\varepsilon + xx^* = x^*$$

$$(x^*y^*)^* = (x + y)^*$$

Conversions of regular expressions

Example

Equivalent REs that cannot be converted one to the other using the identities and theorems of this lecture:

- $B = (10^*1 + 0)^*(10^*1 + 0)$
- $B' = 0^*(1(0 + 10^*1)^*(100^* + 1) + 0)$

Regular equations

Example

Find a regular expression describing the language of all words containing even number of ones (no zeros) followed by string 010.

$$x = 11x + 010$$

solution: $x = (11)^*010$

$(11)^*010 = 11(11)^*010 + 010$ $\quad (x = \alpha x + \beta \Rightarrow x = \alpha^*\beta)$

$(11)^*010 = (11(11)^* + \varepsilon)010$ $\quad (xy + y = (x + \varepsilon)y)$

$(11)^*010 = (11)^*010$ $\quad (xx^* + \varepsilon = x^*)$

Regular equations

Definition
Standard system of regular equations has form:
$X_i = \alpha_{i0} + \alpha_{i1}X_1 + \alpha_{i2}X_2 + \cdots + \alpha_{in}X_n, 1 \leq i \leq n$, where $X_1, X_2, \ldots, X_n$ are variables and α_{ij} are regular expressions over alphabet Σ, which does not contain $X_1, X_2, \ldots, X_n$.

Regular equations

Example

$$A = 1A + 1B$$
$$B = 0A + 0B + 0$$

$A = 1^*1B$
$B = 01^*1B + 0B + 0$
$B = (01^*1 + 0)B + 0$
$B = (01^*1 + 0)^*0 = (0(1^*1 + \varepsilon))^*0 = (01^*)^*0$

Solution:

$$A = 1^*1(01^*)^*0$$
$$B = (01^*)^*0$$

Derivatives of regular expressions

Informal definition

Derivative $\frac{\mathrm{d}}{\mathrm{d}x}$ of regular expression V with respect to string $x \in \Sigma^*$:
$h(\frac{\mathrm{d}V}{\mathrm{d}x}) = \{y : xy \in h(V)\}$.

Derivatives of regular expressions

Definition

Derivative $\frac{\mathrm{d}}{\mathrm{d}x}$ of regular expression V with respect to string $x \in \Sigma^*$:

1. $\frac{\mathrm{d}V}{\mathrm{d}\varepsilon} = V$
2. For $a, b \in \Sigma$ it holds that:
   $\frac{\mathrm{d}\varepsilon}{\mathrm{d}a} = \emptyset$
   $\frac{\mathrm{d}\emptyset}{\mathrm{d}a} = \emptyset$
   $$\frac{\mathrm{d}b}{\mathrm{d}a} = \left\langle \begin{array}{l} \emptyset, \text{ if } a \neq b \\ \\ \varepsilon, \text{ if } a = b \end{array} \right.$$
   $\frac{\mathrm{d}(U+V)}{\mathrm{d}a} = \frac{\mathrm{d}U}{\mathrm{d}a} + \frac{\mathrm{d}V}{\mathrm{d}a}$
   $$\frac{\mathrm{d}(UV)}{\mathrm{d}a} = \left\langle \begin{array}{l} \frac{\mathrm{d}U}{\mathrm{d}a}V, \text{ if } \varepsilon \notin h(U), \\ \\ \frac{\mathrm{d}U}{\mathrm{d}a}V + \frac{\mathrm{d}V}{\mathrm{d}a}, \text{ if } \varepsilon \in h(U) \end{array} \right.$$
   $\frac{\mathrm{d}(V^*)}{\mathrm{d}a} = \frac{\mathrm{d}V}{\mathrm{d}a}.V^*$
3. For $x = a_1 a_2 \ldots a_n, a_i \in \Sigma$ it holds that
   $\frac{\mathrm{d}V}{\mathrm{d}x} = \frac{\mathrm{d}}{\mathrm{d}a_n}(\frac{\mathrm{d}}{\mathrm{d}a_{n-1}}(\cdots \frac{\mathrm{d}}{\mathrm{d}a_2}(\frac{\mathrm{d}V}{\mathrm{d}a_1}) \cdots))$

Derivatives of regular expressions

Example

Regular expression $y = (0+1)^*.1$.

$$
\begin{aligned}
\frac{\mathrm{d}y}{\mathrm{d}\varepsilon} &= (0+1)^*.1 \\
\frac{\mathrm{d}y}{\mathrm{d}1} &= \frac{\mathrm{d}(0+1)^*}{\mathrm{d}1}.1 + \frac{\mathrm{d}1}{\mathrm{d}1} \\
&= \frac{\mathrm{d}(0+1)}{\mathrm{d}1}.(0+1)^*.1 + \varepsilon \\
&= (\frac{\mathrm{d}0}{\mathrm{d}1} + \frac{\mathrm{d}1}{\mathrm{d}1})(0+1)^*.1 + \varepsilon \\
&= (\emptyset + \varepsilon).(0+1)^*.1 + \varepsilon \\
&= (0+1)^*.1 + \varepsilon \\
\frac{\mathrm{d}y}{\mathrm{d}0} &= \frac{\mathrm{d}(0+1)^*}{\mathrm{d}0}.1 + \frac{\mathrm{d}1}{\mathrm{d}0} \\
&= \frac{\mathrm{d}(0+1)}{\mathrm{d}0}.(0+1)^*.1 + \emptyset \\
&= (\varepsilon + \emptyset).(0+1)^*.1 + \emptyset \\
&= (0+1)^*.1
\end{aligned}
$$

Integral of regular expressions

Definition

Integral of regular expression V in respect to string $x \in \Sigma^*$ is defined as:

$$h(\int V \;\mathrm{d}x) = \{xy : y \in h(V)\}.$$

For the integration of regular expressions the following rules apply:

1. $\int V \mathrm{d}\varepsilon = V$
2. for $a \in \Sigma$ it holds:
   $\int \varepsilon \;\mathrm{d}a = a,$
   $\int \emptyset \;\mathrm{d}a = \emptyset,$
   $\int b \;\mathrm{d}a = ab,$
   $\int (U + V) \;\mathrm{d}a = \int U \;\mathrm{d}a + \int V \;\mathrm{d}a,$
   $\int (U.V) \;\mathrm{d}a = aUV,$
   $\int V^* \;\mathrm{d}a = aV^*.$
3. for $x = a_1 a_2 \cdots a_n \in \Sigma^*$ it hols:
   $\int V \;\mathrm{d}x = \int \cdots [\int (\int V \;\mathrm{d}a_n) \;\mathrm{d}a_{n-1}] \cdots \mathrm{d}a_1.$

Integral of regular expressions

$$\frac{\mathrm{d}}{\mathrm{d}x}\int V\ \mathrm{d}x = V,$$
$$\int \frac{\mathrm{d}V}{\mathrm{d}x}\ \mathrm{d}x = V.\ (?)$$

Integral with an integration constant Z:

$$\int V\ \mathrm{d}x = xV + Z$$
$$\frac{\mathrm{d}Z}{\mathrm{d}x} = \emptyset$$

Integral of regular expressions

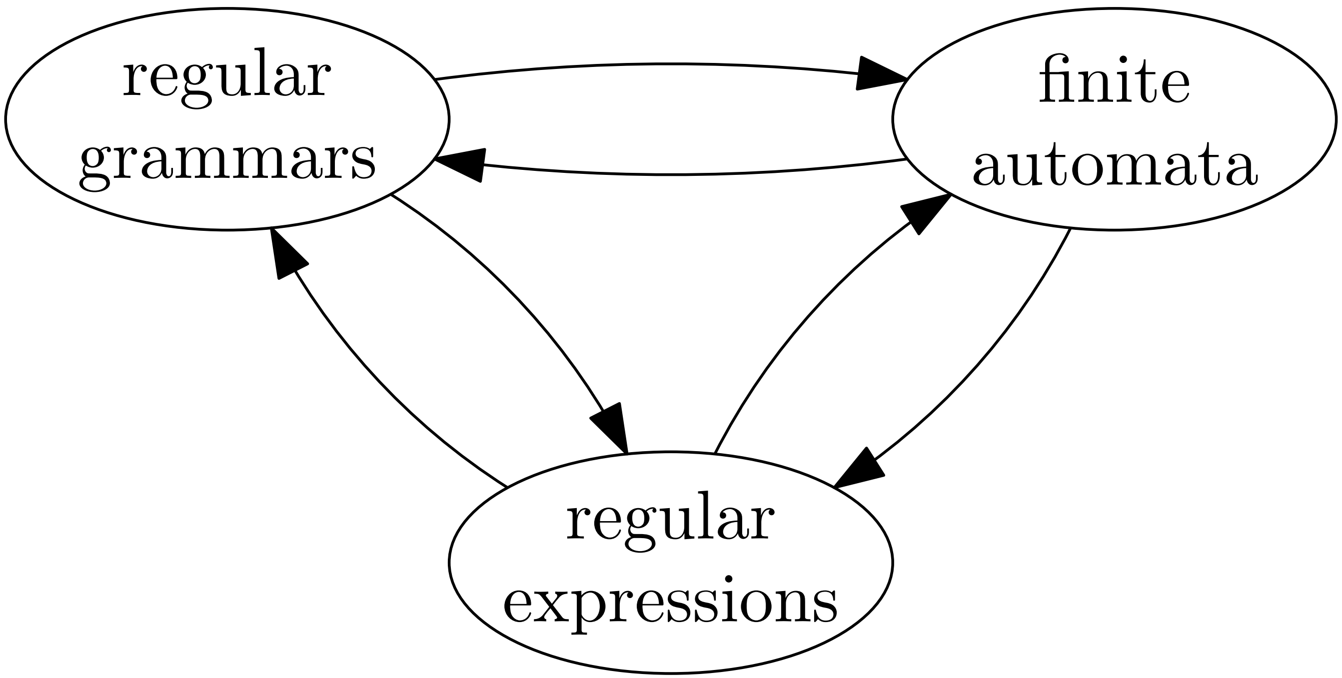
Example

Regular expression $(0+1)^*.1$.

$$\int (0+1)^*.1\ d1 = 1.(0+1)^*.1 + Z_1,$$
$$\int (0+1)^*.1\ d0 = 0.(0+1)^*.1 + Z_0.$$

Relations between formal systems of RL

Relations between formal systems for description of RL



Kleene's Theorem

Theorem (Kleene)
A language over an alphabet is regular iff it can be accepted by a finite automaton.

Relationship between RG and FA

Algorithm NFA for a given regular grammar

Input: Regular grammar $G = (N, \Sigma, P, S)$.
Output: NFA M such that $L(G) = L(M)$.

1: $Q \leftarrow N \cup \{A\}, A \notin N$
2: $\delta(B, a) \leftarrow \{C : (B \to aC) \in P\}$, $\forall a \in \Sigma, \forall B \in N$
3: $\delta(B, a) \leftarrow \delta(B, a) \cup \{A : (B \to a) \in P\}$, $\forall a \in \Sigma, \forall B \in N$
4: $q_0 \leftarrow S$
5: $F \leftarrow \{S, A\}$, if $(S \to \varepsilon) \in P$,
$F \leftarrow \{A\}$, otherwise
6: $M \leftarrow (Q, \Sigma, \delta, q_0, F)$
7: **return** M

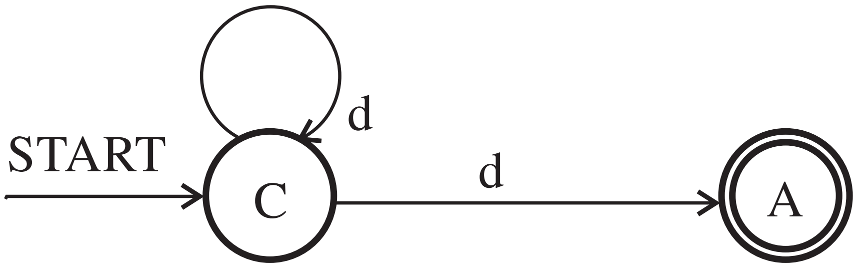
Relationship between RG and FA

Example

$G = (\{C\}, \{d\}, \{C \to d \mid dC\}, C)$.

$M = (\{C, A\}, \{d\}, \delta, C, \{A\})$, where δ:

δ	d
C	$\{C, A\}$
A	



Relationship between RG and FA

Example

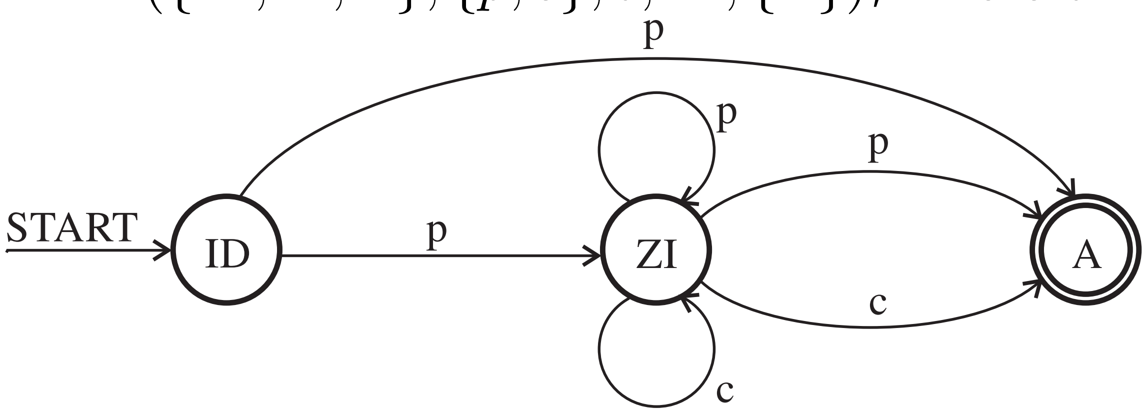
$G = (\{\mathit{ID}, \mathit{ZI}\}, \{p, c\}, P, \mathit{ID})$, where P:

$\mathit{ID} \rightarrow p\,\mathit{ZI} \mid p$

$\mathit{ZI} \rightarrow p\,\mathit{ZI} \mid c\,\mathit{ZI} \mid p \mid c.$

(Grammar generates identifiers according to the usual definition (p – alphabet letter, c – digit).)

$M = (\{\mathit{ID}, \mathit{ZI}, A\}, \{p, c\}, \delta, \mathit{ID}, \{A\})$, where δ



δ	p	c
ID	$\{\mathit{ZI}, A\}$	
ZI	$\{\mathit{ZI}, A\}$	$\{\mathit{ZI}, A\}$
A		

Relationship between RG and FA

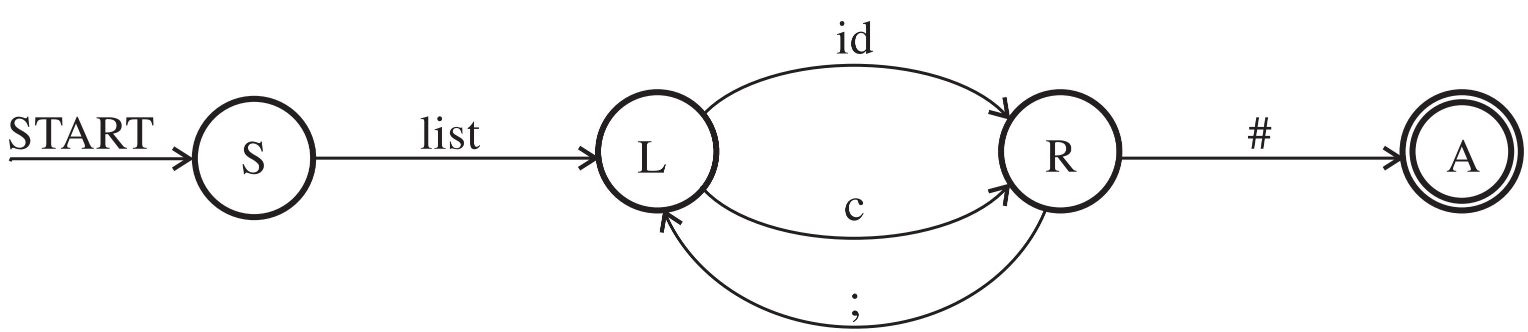
Example

Language describing strings of the form: $list\ \ id; c; id; id; \ldots; c; c; id\#$

$G = (\{S, L, R\}, \{list, id, c, ;, \#\}, P, S)$, where P:
$S \rightarrow list\ L,\ L \rightarrow id\ R \mid c\,R,\ R \rightarrow ;\ L \mid \#$

$M = (\{S, L, R, A\}, \{list, id, c, \#, ;\}, \delta, S, \{A\})$, where δ:

δ	$list$	id	c	$;$	$\#$
S	L				
L		R	R		
R				L	A
A					



Relationship between FA and RG

Algorithm Regular grammar for a given NFA

Input: NFA $M = (Q, \Sigma, \delta, q_0, F)$, $Q \cap \Sigma = \emptyset$.

Output: Regular grammar G, $L(M) = L(G)$.

$N \leftarrow Q$; $S \leftarrow q_0$

$P \leftarrow \{B \to aC : C \in \delta(B, a), B, C \in Q, a \in \Sigma\}$

$P \leftarrow P \cup \{B \to a : C \in \delta(B, a), B \in Q, C \in F, a \in \Sigma\}$

if $q_0 \in F$ **then**

    if S is at righthand side of no rule **then**

        $P \leftarrow P \cup \{S \to \varepsilon\}$

    else

        $N \leftarrow N \cup \{S'\}, S' \notin Q$

        $P \leftarrow P \cup \{S' \to \varepsilon\} \cup \{S' \to \alpha : (S \to \alpha) \in P\}$

        $S \leftarrow S'$ $\triangleright$ S' is set as the start symbol of grammar G

    end if

end if

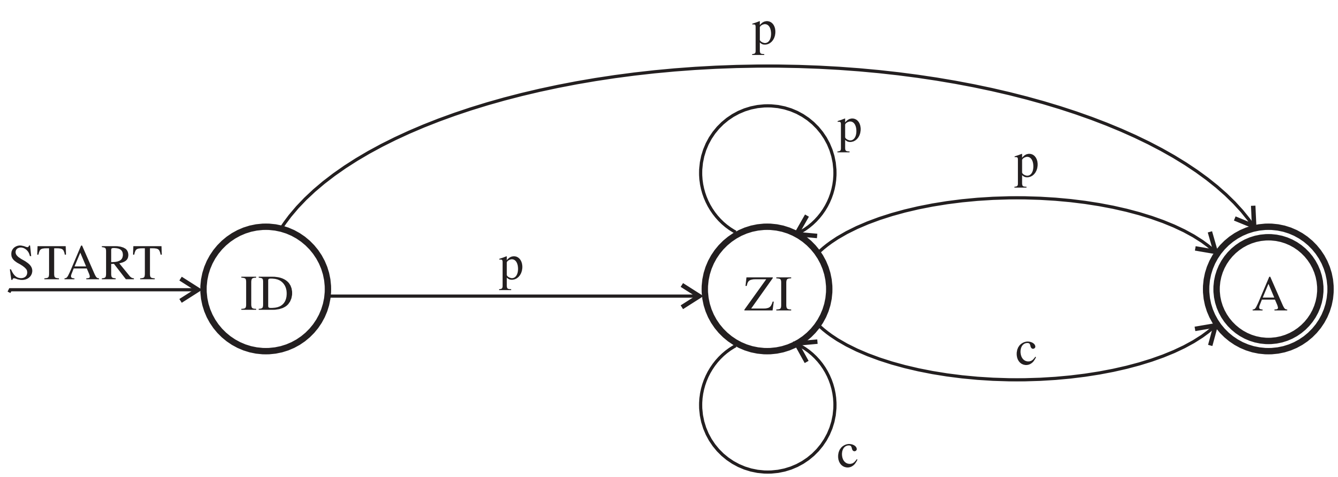
$G \leftarrow (N, \Sigma, P, S)$

return G

Relationship between FA and RG

Example

We construct a regular grammar for NFA:



The resulting regular grammar $G = (\{ID, ZI, A\}, \{p, c\}, P, ID)$, where P:

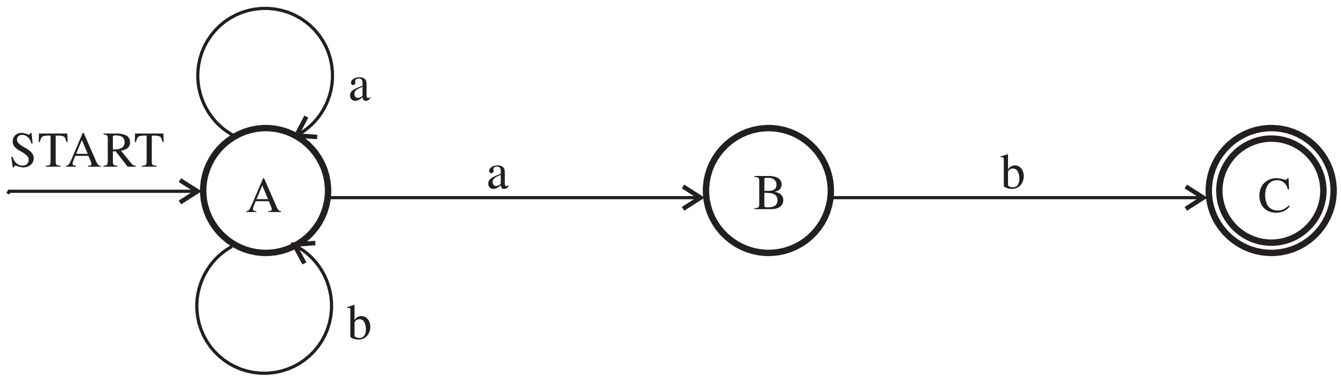
$ID \rightarrow pZI \mid p \mid pA$

$ZI \rightarrow pZI \mid pA \mid cZI \mid cA \mid p \mid c$.

Relationship between FA and RG

Example

We construct a regular grammar for NFA:



The resulting regular grammar $G = (\{A, B, C\}, \{a, b\}, P, A)$, where P:

$A \to aA \mid aB \mid bA$

$B \to bC \mid b$.