CS 5000: Lecture 5

Vladimir Kulyukin
Department of Computer Science
Utah State University

Outline

- Regular Expressions

Two Language Operations

If L_1 and L_2 are the languages,
$L_1L_2 = \{x_1x_2 \mid x \in L_1 \text{ and } x_2 \in L_2\}$.

If L is a language, the Kleene closure of L
is $L^* = \{x_1x_2...x_n \mid n \geq 0, x_{0\leq i\leq n} \in L\}$.

Atomic Regular Expressions

A regular expression is a string r that denotes the language $L(r)$ over some alphabet Σ.

Three types of atomic regular expressions:

1. If $a \in \Sigma$, then $L(a) = \{a\}$.
2. $L(\varepsilon) = \{\varepsilon\}$.
3. $L(\varnothing) = \{\ \}$.

Compound Regular Expressions

Let r_1 and r_2 be regular expressions.

1. $(r_1 + r_2)$ is a regular expression. Then $L(r_1 + r_2) = L(r_1) \cup L(r_2)$.
2. $(r_1 r_2)$ is a regular expression. Then $L(r_1 r_2) = L(r_1) L(r_2)$.
3. $(r)^*$ is a regular expression. Then $L\left((r)^*\right) = \left(L(r)\right)^*$.

Examples

$$ab; L(ab) = \{ab\}.$$

$$ab + c; L(ab + c) = \{ab, c\}.$$

$$ba^*; L(ba^*) = \{b, ba, baa, baaa, \ldots\} = \{ba^n \mid n \geq 0\}.$$

Examples

$$(a+b)^*;\ L\left((a+b)^*\right)=\{a,b\}^*.$$

$$(ab+\varepsilon);\ L\left((ab+\varepsilon)\right)=\{ab,\varepsilon\}.$$

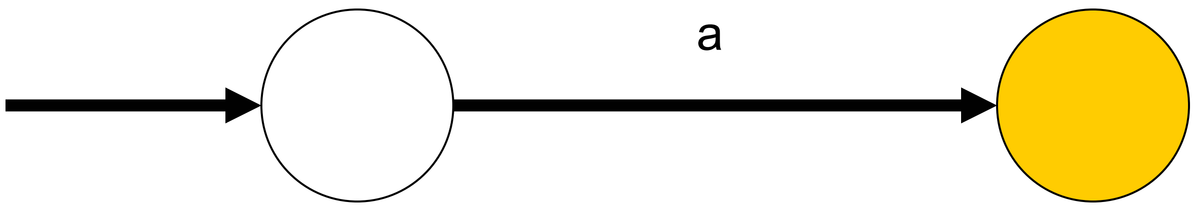
$$(a+b)(c+d)=\{ac,ad,bc,bd\}.$$

$$(abc)^* = L\left((abc)^*\right)=\{\varepsilon, abc, abcabc, abcabcabc, \ldots\}.$$

$$a^*b^*;\ L\left(a^*b^*\right)=\{\varepsilon, a, b, aa, bb, abb, aaab, \ldots\}.$$

Atomic Reg Exps ➔ Reg Languages

$$a \in \Sigma$$



Atomic Reg Exps ➔ Reg Languages

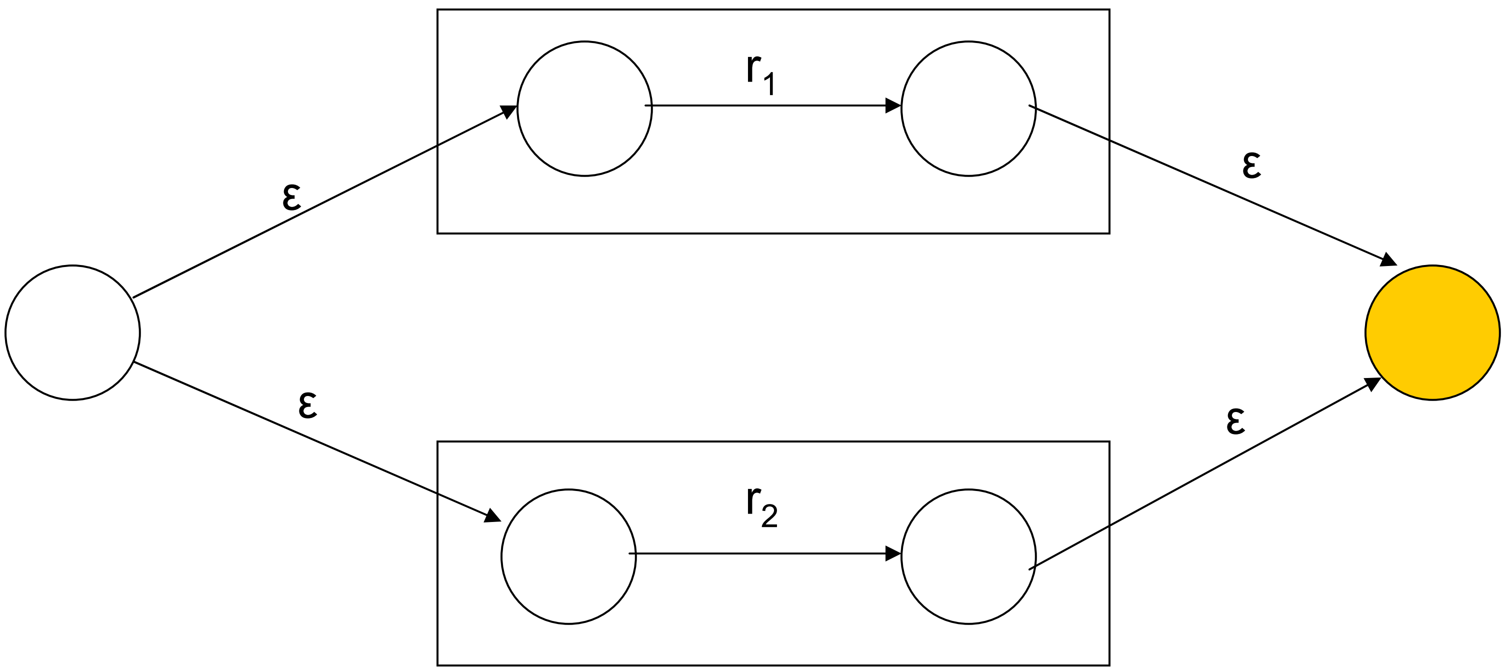
$$\mathcal{E}$$

Atomic Reg Exps ➔ Reg Languages

$\emptyset$

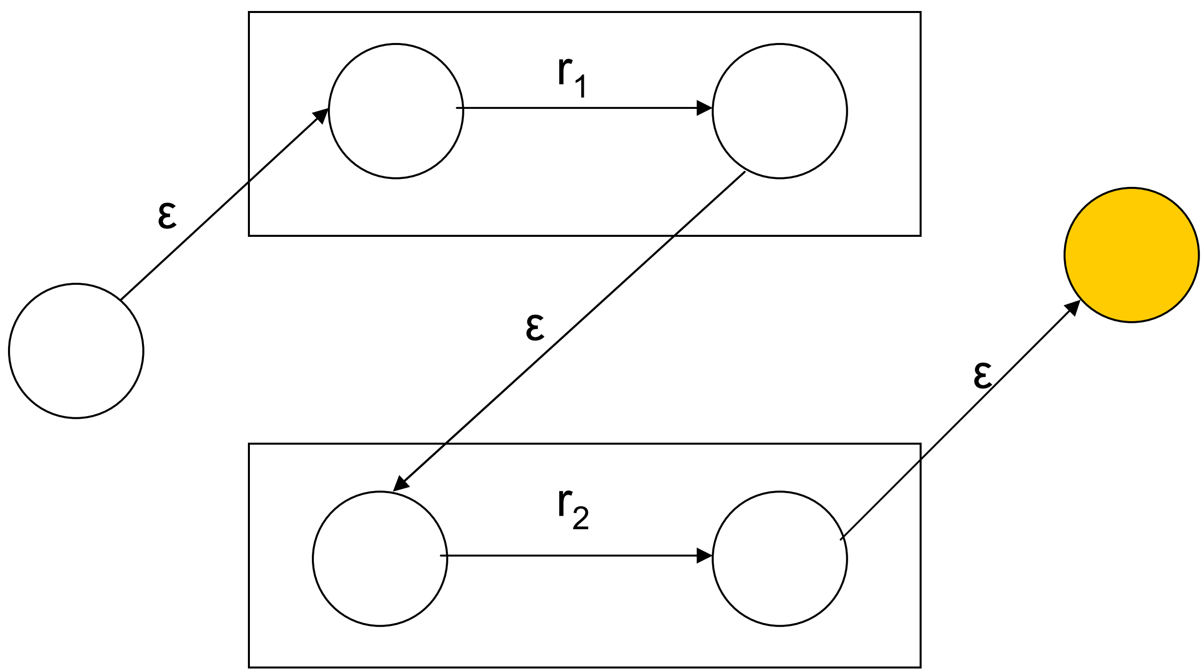
Compound Reg Exps ➔ Reg Languages

$$\left(r_1 + r_2\right)$$



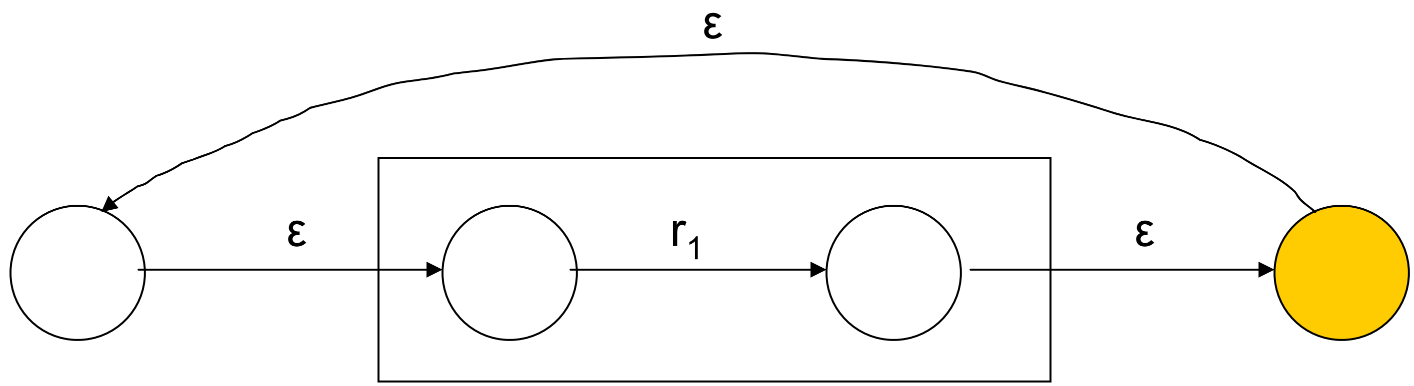
Compound Reg Exps ➔ Reg Languages

$$\left(r_1 \bullet r_2\right)$$



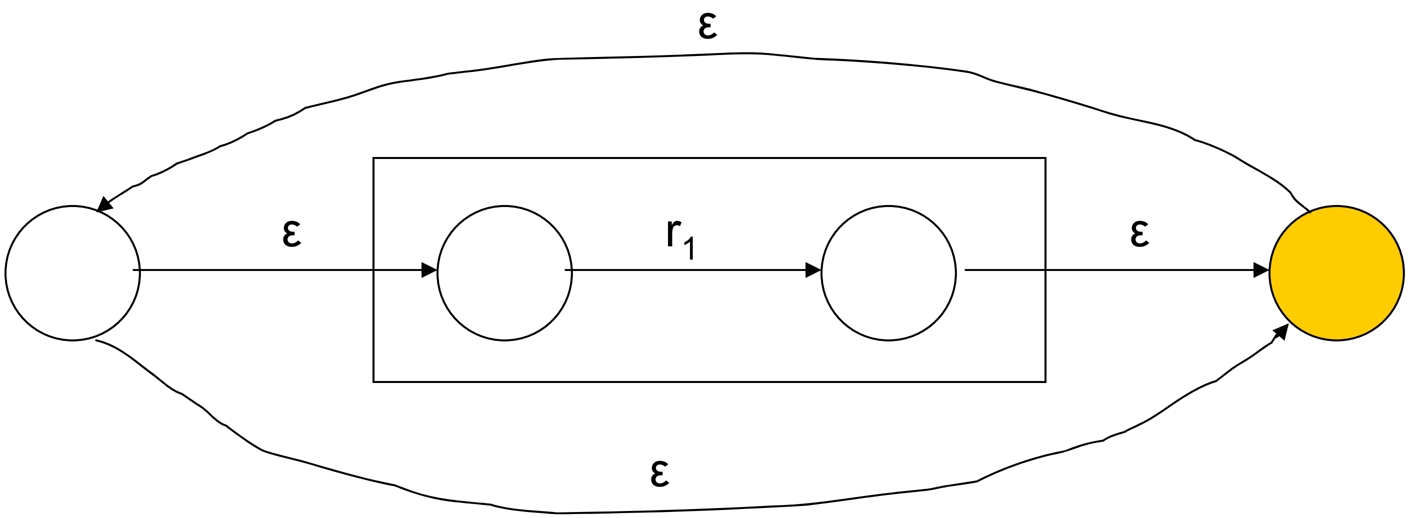
Compound Reg Exps ➔ Reg Languages

$$\left(r_1\right)^+$$



Compound Reg Exps → Reg Languages

$$(r_1)^*$$



Reg Exps ➔ NFA

Theorem: If r is a regular expression, then there is an NFA N such that $L(N) = L(r)$.

Proof: We use induction on the number of operators in r. The base case is true for $n = 0$. Assume that the statement is true for k. Consider a regular expression q that has $k+1$ operators. Then q can be of the form $(r)^*, (r)^+, r_1 + r_2, r_1 r_2$. We know that r, r_1, and r_2. have fewer than $k+1$ operators. Thus, by the inductive hypothesis, there are NFAs with the same languages. By construction techniques explained on the previous slides, these regular expressions can be combined into larger NFAs with the same languages.