Finite Automata Theory and Formal Languages

TMV027/DIT321– LP4 2017

Lecture 9
Ana Bove

April 20th 2017

Overview of today's lecture:

- Equivalence between FA and RE: from RE to FA;
- Pumping Lemma for RL;
- Closure properties of RL.

Recap: Regular Expressions

- Algebraic representation of (regular) languages;
- $R, S ::= \emptyset \mid \epsilon \mid a \mid R + S \mid RS \mid R^*$ ...
- ... representing the languages $\emptyset, \{\epsilon\}, \{a\}, \mathcal{L}(R) \cup \mathcal{L}(S), \mathcal{L}(R)\mathcal{L}(S)$ and $\mathcal{L}(R)^*$ respectively;
- Algebraic laws for RE and how to prove them;
- How to transform a FA into a RE:
  - By eliminating states;
  - With a system of linear equations and Arden's lemma.

From Regular Expressions to Finite Automata

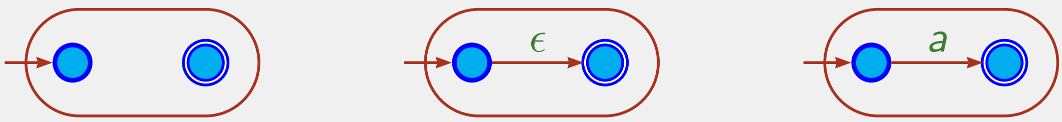
Proposition: *Every language defined by a RE is accepted by a FA.*

Proof: Let $\mathcal{L} = \mathcal{L}(R)$ for some RE R.

By induction on R we construct a ϵ-NFA E with only one final state and no arcs into the initial state or out of the final state.

Moreover, E is such that $\mathcal{L} = \mathcal{L}(E)$.

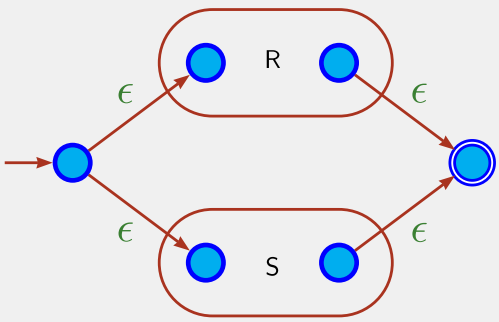
Base cases are the RE $\emptyset$, ϵ and $a \in \Sigma$.
The corresponding ϵ-NFA recognising the languages $\emptyset$, $\{\epsilon\}$ and $\{a\}$ are:

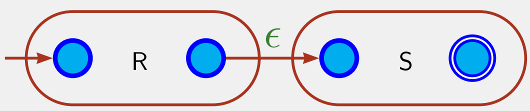


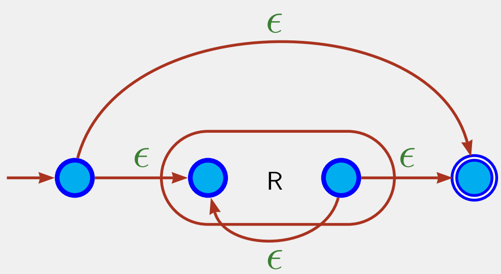
From RE to FA: Inductive Step (Cont.)

IH: Consider the RE R and S and ϵ-NFA for them.

We construct the ϵ-NFA for $R + S$, RS and R^* recognising $\mathcal{L}(R) \cup \mathcal{L}(S)$, $\mathcal{L}(R)\mathcal{L}(S)$ and $\mathcal{L}(R)^*$ respectively:

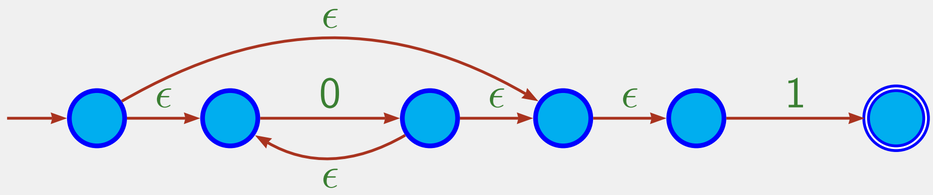




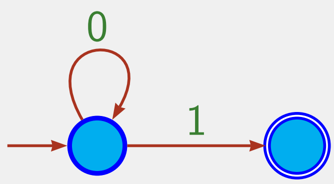


Example: From RE to FA

If we follow this method for the RE 0^*1 we obtain the ϵ-NFA



Compare it with the following DFA for the same language:



How to Identify Regular Languages?

We have seen that a language is regular iff there is a DFA that accepts the language.

Then we saw that DFA, NFA and ϵ-NFA are equivalent in the sense that we can convert between them.

Hence FA accept all and only the regular languages (RL).

Now we have seen how to convert between FA and RE.

Thus RE also define all and only the RL.

How to Prove that a Language is NOT Regular?

In a FA with n states, any path

$$q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \xrightarrow{a_3} \dots \xrightarrow{a_{m-1}} q_m \xrightarrow{a_m} q_{m+1}$$

has a loop if $m \geqslant n$.

That is, we have $i < j$ such that $q_i = q_j$ in the path above.

This is an application of the *Pigeonhole Principle*.

How to Prove that a Language is NOT Regular?

Example: Let us prove that $\mathcal{L} = \{0^m 1^m | m \geqslant 0\}$ is not a RL.

Let us assume it is: then $\mathcal{L} = \mathcal{L}(A)$ for some FA A with n states, $n > 0$.

Let $k \geqslant n > 0$ and let $w = 0^k 1^k \in \mathcal{L}$.

Then there must be an accepting path $q_0 \xrightarrow{w} q_f \in F$.

Since $k \geqslant n$, there is a loop (pigeonhole principle) when reading the 0's.

Then $w = xyz$ with $|xy| = j \leqslant n$, $y \neq \epsilon$ and $z = 0^{k-j}1^k$ such that

$$q_0 \xrightarrow{x} q_l \xrightarrow{y} q_l \xrightarrow{z} q_f \in F$$

Observe that the following path is also an accepting path

$$q_0 \xrightarrow{x} q_l \xrightarrow{z} q_f \in F$$

However y must be of the form 0^i with $i > 0$ hence $xz = 0^{k-i}1^k \notin \mathcal{L}$.

This contradicts the fact that A accepts $\mathcal{L}$.

The Pumping Lemma for Regular Languages

Theorem: *Let $\mathcal{L}$ be a RL.*
Then, there exists a constant n—which depends on $\mathcal{L}$—such that for every string $w \in \mathcal{L}$ with $|w| \geqslant n$, it is possible to break w into 3 strings x, y and z such that $w = xyz$ and

1. $y \neq \epsilon$;
2. $|xy| \leqslant n$;
3. $\forall k \geqslant 0.\ xy^k z \in \mathcal{L}$.

Proof of the Pumping Lemma

Assume we have a FA A that accepts the language, then $\mathcal{L} = \mathcal{L}(A)$.

Let n be the number of states in A.

Then any path of length $m \geqslant n$ has a loop.

Let us consider $w = a_1 a_2 \dots a_m \in \mathcal{L}$.

We have an accepting path and a loop such that

$$q_0 \xrightarrow{x} q_l \xrightarrow{y} q_l \xrightarrow{z} q_f \in F$$

with $w = xyz \in \mathcal{L}$, $y \neq \epsilon$, $|xy| \leqslant n$.

Then we also have

$$q_0 \xrightarrow{x} q_l \xrightarrow{y^k} q_l \xrightarrow{z} q_f \in F$$

for any k, that is, $\forall k \geqslant 0.\ xy^k z \in \mathcal{L}$.

Example: Application of the Pumping Lemma

We use the Pumping lemma to prove that $\mathcal{L} = \{0^m1^m | m \geqslant 0\}$ is not a RL.

We assume it is. Then the Pumping lemma applies.

Let n be the constant given by the lemma and let $w = 0^n1^n \in \mathcal{L}$, then $|w| \geqslant n$.

By the lemma we know that $w = xyz$ with $y \neq \epsilon$, $|xy| \leqslant n$ and $\forall k \geqslant 0.\ xy^kz \in \mathcal{L}$.

Since $y \neq \epsilon$ and $|xy| \leqslant n$, we know that $y = 0^i$ with $i \geqslant 1$.

However, we have a contradiction since $xy^kz \notin \mathcal{L}$ for $k \neq 1$ since it either has too few 0's ($k = 0$) or too many ($k > 1$).

Note: This is connected to the fact that a FA has *finite memory*!
If we could build a machine with infinitely many states it would be able to recognise the language.

Example: Application of the Pumping Lemma

Example: Let us prove that $\mathcal{L} = \{0^i1^j | i \leqslant j\}$ is not a RL.

Let us assume it is, hence the Pumping lemma applies.

Let n be given by the Pumping lemma and let $w = 0^n1^{n+1} \in \mathcal{L}$, hence $|w| \geqslant n$.

Then we know that $w = xyz$ with $y \neq \epsilon$, $|xy| \leqslant n$ and $\forall k \geqslant 0.\ xy^kz \in \mathcal{L}$.

Since $y \neq \epsilon$ and $|xy| \leqslant n$, we know that $y = 0^r$ with $r \geqslant 1$.

However, we have a contradiction since $xy^kz \notin \mathcal{L}$ for $k > 2$ since it will have more 0's than 1's.

(Even for $k = 2$ if $r > 1$.)

Exercise: What about the languages $\{0^i1^j \mid i \geqslant j\}$, $\{0^i1^j \mid i > j\}$ and $\{0^i1^j \mid i \neq j\}$?

Pumping Lemma is not a Sufficient Condition

By showing that the Pumping lemma does not apply to a certain language $\mathcal{L}$ we prove that $\mathcal{L}$ is not regular.

However, if the Pumping lemma *does* apply to $\mathcal{L}$, we *cannot* conclude whether $\mathcal{L}$ is regular or not!

Example: We know $\mathcal{L} = \{b^m c^m \mid m \geqslant 0\}$ is not regular.

Let us consider $\mathcal{L}' = a^+\mathcal{L} \cup (b + c)^*$.

Using clousure properties (to come later) we can prove that $\mathcal{L}'$ is not regular.

However, the Pumping lemma does apply for $\mathcal{L}'$ with $n = 1$.

This shows the Pumping lemma is not a sufficient condition for a language to be regular.

Closure Properties for Regular Languages

Let $\mathcal{M}$ and $\mathcal{N}$ be RL. Then $\mathcal{M} = \mathcal{L}(R) = \mathcal{L}(D)$ and $\mathcal{N} = \mathcal{L}(S) = \mathcal{L}(F)$ for RE R and S, and DFA D and F.

We have seen that RL are closed under the following operations:

Union: $\mathcal{M} \cup \mathcal{N} = \mathcal{L}(R + S)$ or $\mathcal{M} \cup \mathcal{N} = \mathcal{L}(D \uplus F)$ (s.21, l.5);

Complement: $\overline{\mathcal{M}} = \mathcal{L}(\overline{D})$ (slide 23, lec. 5)

Intersection: $\mathcal{M} \cap \mathcal{N} = \overline{\overline{\mathcal{M}} \cup \overline{\mathcal{N}}}$ or $\mathcal{M} \cap \mathcal{N} = \mathcal{L}(D \times F)$ (s.20, l.5);

Difference: $\mathcal{M} - \mathcal{N} = \mathcal{M} \cap \overline{\mathcal{N}}$;

Concatenation: $\mathcal{M}\mathcal{N} = \mathcal{L}(RS)$;

Closure: $\mathcal{M}^* = \mathcal{L}(R^*)$.

More Closure Properties for Regular Languages

RL are also closed under the following operations:

Prefix: See additional exercise 3 on DFA.
Hint: in D, make final all states in a path from the start state to final state.

Reversal: Recall that $\text{rev}(a_1 \dots a_n) = a_n \dots a_1$ and $\forall x.\text{rev}(\text{rev}(x)) = x$ (slides 14 & 16, lec. 4).

Closure under Prefix

Another way to prove that the language of prefixes of a RL is regular:

Define the function:

$$
\begin{array}{l}
pre : RE \to RE \\
pre(\emptyset) = \emptyset \\
pre(\epsilon) = \epsilon \\
pre(a) = \epsilon + a \\
pre(R_1 + R_2) = pre(R_1) + pre(R_2) \\
pre(R_1 R_2) = pre(R_1) + R_1 pre(R_2) \\
pre(R^*) = R^* pre(R)
\end{array}
$$

and prove that $\mathcal{L}(pre(R)) = \text{Prefix}(\mathcal{L}(R))$.

Then, if $\mathcal{L} = \mathcal{L}(R)$ for some RE R then $\text{Prefix}(\mathcal{L}) = \text{Prefix}(\mathcal{L}(R)) = \mathcal{L}(pre(R))$.

Closure under Reversal

We define the function:

$$
\begin{array}{ll}
\_^{\mathsf{r}} : RE \to RE & \\
\emptyset^{\mathsf{r}} = \emptyset & (R_1 + R_2)^{\mathsf{r}} = R_1^{\mathsf{r}} + R_2^{\mathsf{r}} \\
\epsilon^{\mathsf{r}} = \epsilon & (R_1 R_2)^{\mathsf{r}} = R_2^{\mathsf{r}} R_1^{\mathsf{r}} \\
a^{\mathsf{r}} = a & (R^*)^{\mathsf{r}} = (R^{\mathsf{r}})^*
\end{array}
$$

Theorem: *If $\mathcal{L}$ is regular so is $\mathcal{L}^{\mathsf{r}}$.*

Proof: (See theo. 4.11, pages 139–140).

Let R be a RE such that $\mathcal{L} = \mathcal{L}(R)$.
We need to prove by induction on R that $\mathcal{L}(R^{\mathsf{r}}) = (\mathcal{L}(R))^{\mathsf{r}}$.
Hence $\mathcal{L}^{\mathsf{r}} = (\mathcal{L}(R))^{\mathsf{r}} = \mathcal{L}(R^{\mathsf{r}})$ and $\mathcal{L}^{\mathsf{r}}$ is regular.

Example: The reverse of the language defined by $(0+1)^*0$ can be defined by $0(0+1)^*$.

Closure under Reversal

Another way to prove this result is by constructing a ϵ-NFA for $\mathcal{L}^{\mathsf{r}}$.

Proof: Let $N = (Q, \Sigma, \delta_N, q_0, F)$ be a NFA such that $\mathcal{L} = \mathcal{L}(N)$.

Define a ϵ-NFA $E = (Q \cup \{q\}, \Sigma, \delta_E, q, \{q_0\})$ with $q \notin Q$ and δ_E such that

$$
\begin{array}{l}
r \in \delta_E(s, a) \text{ iff } s \in \delta_N(r, a) \text{ for } r, s \in Q \\
r \in \delta_E(q, \epsilon) \text{ iff } r \in F
\end{array}
$$

Using Closure Properties

Example: Consider $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $\mathcal{L}_1$ is regular, $\mathcal{L}_2$ is not regular but $\mathcal{L}_1 \cap \mathcal{L}_2$ is regular.

Is $\mathcal{L}_1 \cup \mathcal{L}_2$ is regular?

Let us assume that $\mathcal{L}_1 \cup \mathcal{L}_2$ is regular.

Then $(\mathcal{L}_1 \cup \mathcal{L}_2 - \mathcal{L}_1) \cup (\mathcal{L}_1 \cap \mathcal{L}_2)$ should also be regular.

But this is actually $\mathcal{L}_2$ which is not regular!

We arrive to a contradiction.
Hence $\mathcal{L}_1 \cup \mathcal{L}_2$ cannot be regular.

Overview of next Week

Mon 24	Tue 25	Wed 26	Thu 27	Fri 28
	Ex 10-12 EA RL.		**Ex 10-12 HA2** RL.	
Lec 13-15 HB3 RL.			**Lec 13-15 HB3** CFG.	
15-17 6213 6215 **Individual help**		**15-17 EL41** **Consultation**		

Overview of Next Lecture

Sections 4.3–4.4:

- Decision properties for RL;
- Equivalence of RL;
- Minimisation of automata.