Properties of FA Languages

I. Closure Properties

Union: The union of two DFA languages is a DFA language.

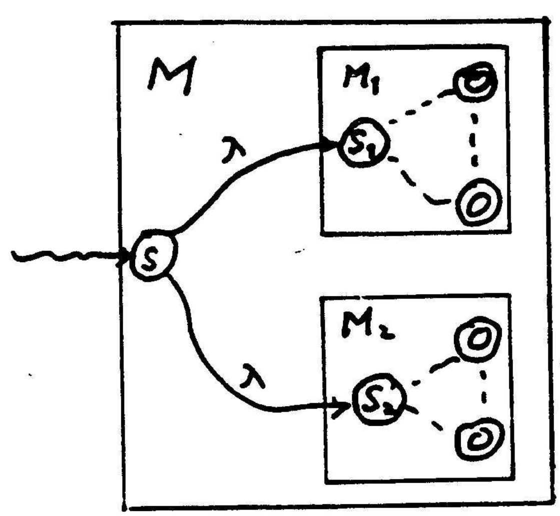
Proof: Given $L_1 = L(M_1)$ **and** $L_2 = L(M_2)$, $L_1,\ L_2$ **are generic DFA languages.**

$$M_1 = (Q_1,\ \Sigma_1,\ \delta_1,\ s_1,\ F_1) \text{ and}$$

$$M_2 = (Q_2,\ \Sigma_2,\ \delta_2,\ s_2,\ F_2).$$

Construct a λ-NFA $M = (Q,\ \Sigma,\ \delta,\ s,\ F)$ **where**

$$\begin{aligned} Q &= Q_1 \cup Q_2 \cup \{s\} \\ \Sigma &= \Sigma_1 \cup \Sigma_2 \\ \delta &= \delta_1 \cup \delta_2 \cup \{(s, \lambda, s_1), (s, \lambda, s_2)\} \\ F &= F_1 \cup F_2 \end{aligned}$$



Now we claim that $L(M) = L_1 \cup L_2$

(1) Let x be a generic word in $\underline{L(M)}$.
$x \in L(M)$ **implies**
$sx \vdash_M^+ f$ **for some** $f \in F_1$ **or** $f \in F_2$**.**
If $f \in F_1$, then $sx \vdash_M s_1x \vdash_M^* f$ **and then** $s_1x \vdash_{M_1}^* f$**. We know that** $x \in L(M_1)$**.**
If $f \in F_2$, then $sx \vdash_M s_2x \vdash_M^* f$**. Since** $s_2x \vdash_{M_2}^* f$**, we know that** $x \in L(M_2)$**,**
therefore $x \in L(M_1) \cup L(M_2)$**, i.e.,** $x \in L_1 \cup L_2$**.**
So $L(M) \subseteq L_1 \cup L_2$**.**

(2) Let x be a generic word in $L_1 \cup L_2$.
Without losing of generality we can assume that $x \in L_1$**, i.e.,** $x \in L(M_1)$**.**
Then $s_1x \vdash_{M_1}^* f$**, for some** $f \in F_1$**.**
Then $sx \vdash_M s_1x \vdash_M^* f$**. Therefore** $x \in L(M)$**.**
So, $L_1 \cup L_2 \subseteq L(M)$**.**

(3) By the results of (1) and (2), we conclude $L(M) = L_1 \cup L_2$**.**

Since every λ-NFA language is a DFA language, $L(M)$ is a DFA language.

2. Complementation:

The complement of a DFA language $L \subseteq \Sigma^*$ is a DFA language.

Proof: There is a complete DFA $M = (Q,\ \Sigma,\ \delta,\ s,\ F)$ **with** $L = L(M)$**. Define** $\overline{M} = (Q,\ \Sigma,\ \delta,\ s,\ Q - F)$

3. Intersection:

L_1 **and** L_2

$$L \;=\; L_1 \cap L_2 \;=\; \overline{\overline{L_1} \cup \overline{L_2}}$$

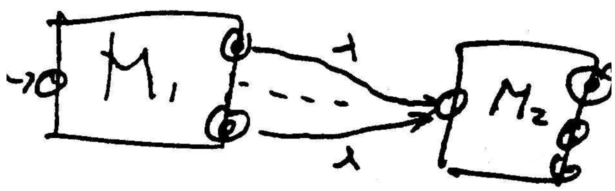
4. Catenation:

The catenation of two DFA languages is a DFA language:

$$M_1 = (Q_1,\ \Sigma_1,\ \delta_1,\ s_1,\ F_1);$$
$$M_2 = (Q_2,\ \Sigma_2,\ \delta_2,\ s_2,\ F_2).$$

$$L(M) \ = \ L(M_1)L(M_2)$$

$$M = (Q_1 \cup Q_2,\ \Sigma_1 \cup \Sigma_2,\ \delta,\ s_1,\ F_2),$$
$$\delta = \delta_1 \cup \delta_2 \cup \{(f, \lambda, s_2) \mid f \in F_1\}$$



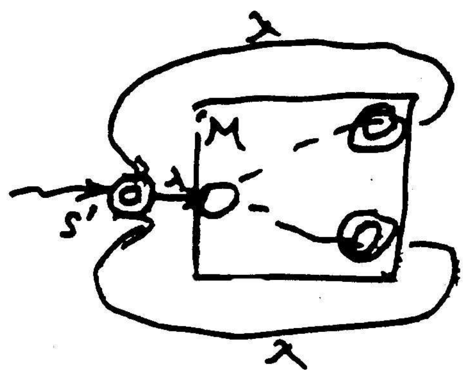
$M = (Q,\ \Sigma,\ \delta,\ s,\ F)$ **where**

$$\begin{array}{rcl} Q & = & Q_1 \cup Q_2 \\ \Sigma & = & \Sigma_1 \cup \Sigma_2 \\ \delta & = & \delta_1 \cup \delta_2 \cup \{(f, \lambda, s_2) \mid f \in F_1\} \\ s & = & s_1 \\ F & = & F_2 \end{array}$$

5. Star:

The star of a DFA language is a DFA language

$L = L(M)$ **where** $M = (Q,\ \Sigma,\ \delta,\ s,\ F)$

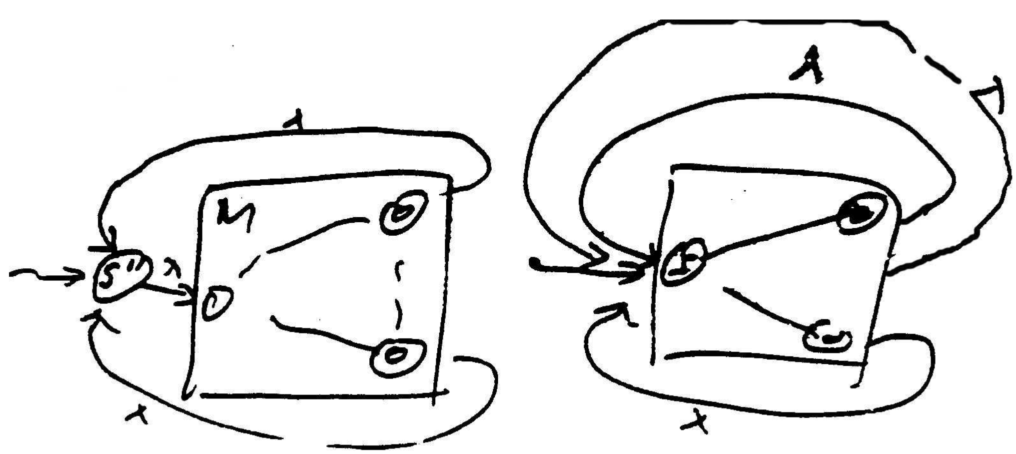


$L^* = L(M')$

$M' = (Q',\ \Sigma',\ \delta',\ s',\ F')$ **where**

$$\begin{aligned} Q' &= Q \cup \{s'\} \\ \Sigma' &= \Sigma \\ \delta' &= \delta \cup \{(s', \lambda, s)\} \cup \{(f, \lambda, s') \mid f \in F\} \\ F' &= F \cup \{s'\} \quad F' = \{s'\} \end{aligned}$$

6. Plus:



II. Decidability Properties

A decision problem is a problem each instance of which is either false or true.

A decision algorithm is an algorithm whose result for each possible input is either false or true.

A decision problem is decidable if there exists a decision algorithm for it. Otherwise it is undecidable.

1. DFA membership

INSTANCE: A DFA $M = (Q,\ \Sigma,\ \delta,\ s,\ F)$ **and a word** $x \in \Sigma^*$**.**

QUESTION: Is $x \in L(M)$ **?**

Theorem: DFA membership is decidable.

Proof: Compute for the given M and x the terminating state q such that $sx \ \vdash^* \ q$. If $q \in F$ then answer "True" else answer "false".

2. **DFA Emptiness**

INSTANCE: A DFA $M = (Q,\ \Sigma,\ \delta,\ s,\ F)$.
QUESTION: Is $L(M) = \emptyset$ **?**

Theorem DFA emptiness is decidable.

Proof: $L(M) = \emptyset$ **iff there is no path in the state diagram of** M **from** s **to a final state. If** $F = \emptyset$**, then this holds immediately.**
Otherwise, enumerate the states that can be reached from s**. (Mark** s**. Now mark all states reachable by one transition from one of the marked states. Repeat this until no newly marked state is introduced.**
The marked states are the reachable states).

3. **DFA Universality**

INSTANCE: A DFA $M = (Q,\ \Sigma,\ \delta,\ s,\ F)$**.**
QUESTION: Is $L(M) = \Sigma^*$ **?**

Theorem DFA universality is decidable.
Proof:

4. **DFA Containment**

INSTANCE: Two DFA
$M_1 = (Q_1,\ \Sigma_1,\ delta_1,\ s_1,\ F_1)$ **and**
$M_2 = (Q_2,\ \Sigma_2,\ delta_2,\ s_2,\ F_2)$**.**
QUESTION: Is $L(M_1) \subseteq L(M_2)$ **?**

Theorem DFA containment is decidable.
Proof:

$L(M_1) \cap \overline{L(M_2)} = \emptyset$ **means that** $L(M_1) \subseteq L(M_2)$

5. <u>DFA Equivalence</u>

INSTANCE: Two DFA
$M_1 = (Q_1,\ \Sigma_1,\ delta_1,\ s_1,\ F_1)$ **and**
$M_2 = (Q_2,\ \Sigma_2,\ delta_2,\ s_2,\ F_2)$**.**
QUESTION: Is $L(M_1) = L(M_2)$ **?**

<u>**Theorem**</u> **DFA equivalence is decidable.**

$L(M_1) \subseteq L(M_2)$ **? yes**

$L(M_2) \subseteq L(M_1)$ **? yes**

Pumping Lemma and Non-DFA Language.

The DFA Pumping Lemma

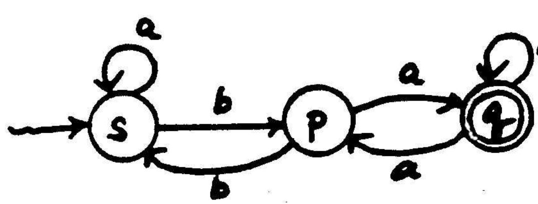
Let $M = (Q,\ \Sigma,\ \delta,\ s,\ F)$ **be a DFA and let** $p = \#Q$**. For all words** x **in** $L(M)$ **such that** $|x| \geq p$, x **can be decomposed into** uvw**, for some** u, v, **and** w **in** Σ^* **such that**

(i) $|uv| \leq p$;

(ii) $|v| \geq 1$; **and**

(iii) **for all** $i \geq 0$, uv^iw **is in** $L(M)$.

Example $M = (Q, \Sigma, \delta, s, F)$ **where**
$Q = \{s,\ p,\ q\}$, $\Sigma = \{a,\ b\}$,
$F = \{q\}$. $\#Q = 3$



Let x **be an arbitrary word of length** $\geq \#Q$ **in** $L(M)$, **e.g.** $x = baaa$**. The accepting conf. sequence for** x **is:**
$\underline{s}baaa \vdash paaa \vdash \underline{q}aa \vdash pa \vdash \underline{q}$.
So, $u = b,\ v = aa,\ w = a$ **and** $b(aa)^ia \in L(M)$ **for all** $i \geq 0$.

Proof of DFA Pumping Lemma:

Let x **be in** $L(M)$ **with** $|x| \geq p$**.**
Then M **has an accepting configuration sequence**

$$q_0 a_1 a_2 \dots a_r \ \vdash \ q_1 a_2 \dots a_r \ \vdash \ \dots \ \vdash q_{r-1} a_r \ \vdash \ q_r$$

where $q_0 = s, \ q_r \in F$**.**
Consider the first p **transitions.** $s, \ q_1, \ \dots, \ q_p$ **cannot all be distinct since there are only** p **distinct states (by pigeonhole principle).**
This means that $q_i = q_j$ **for some** $0 \leq i < j \leq p$**.**
Let

$$u = a_1 \dots a_i,$$

$$v = a_{i+1} \dots a_j,$$

$$w = a_{j+1} \dots a_r.$$

So, we have $q_0 u \ \vdash^* \ q_i$
$q_i v \ \vdash^* \ q_j \qquad (q_i = q_j)$
$q_j w \ \vdash^* \ q_r$**.**
Since $q_i = q_j$**,** $q_i v^k \ \vdash^* \ q_j$ **for any** $k \geq 0$

(i) $|uv| \leq p,$ **since** $j \leq p$
(ii) $|v| \geq 1,$ **since** $i < j$
(iii) $uv^k w \in L(M)$ **for all** $k \geq 0,$ **since**

$$q_0 uv^k w \ \vdash^* \ q_i v^k w \ \vdash^* \ q_j w \ \vdash \ q_r \ \in F$$

qed.

<u>Re-state Pumping Lemma</u>
<u>If</u> L **is a <u>DFA Language</u>**
<u>then</u>
($L = L(M)$ **for some** M **of** p **states;)**
for <u>every</u> word x **of length** $\geq p$ **in** L**,**
there <u>exists one</u> decomposition $x = uvw$
which satisfies

(i) $|uv| \leq p,$
(ii) $|v| \geq 1,$
(iii) $uv^k w \in L(M)$ **for all** $k \geq 0.$

Comments: The DFA pumping lemma shows a property of DFA languages. It can be used possitively; but it is mainly used negatively to show that some languages are not in $\mathcal{L}_{DFA}$. The DFA pumping lemma gives a necessary condition for DFA languages. The condition is not sufficient.

Review of logic:

if A **then** $B \Leftrightarrow$ **if** $\neg B$ **then** $\neg A$
if A **then** $B \not\Leftrightarrow$ **if** B **then** A
if A **then** $B \not\Leftrightarrow$ **if** $\neg A$ **then** $\neg B$

Example

if it is sugar then it is sweet.
if it is not sweet then it is not sugar.
if it is sweet then it is sugar.

So, pumping lemma can be used to show that a language is not a DFA language, but cannot be used to show that a language is a DFA language.

Non-DFA Language

$L = \{a^i b^i \mid i \geq 0\}$ **is not a DFA Language**

Proof (I):

Assume L **is a DFA language. Then** $L = L(M)$ **for some DFA** $M = (Q,\ \Sigma,\ \delta,\ s,\ F)$ **with** $|Q| = p$**.**
Consider $x = a^t b^t,\ \ |x| \geq p.$
By DFA pumping lemma, there is a decomposition

$$x = uvw$$

which satisfies (i), (ii), and (iii).
Consider all the possible decompositions

$$\begin{array}{ll}
\underline{\textbf{Case 1}}: & v = a^k,\ \ k \geq 1.\ k \leq t \\
 & \textbf{But } uv^0w = a^{t-k}b^t \notin L. \\
\underline{\textbf{Case 2}}: & v = b^k,\ \ k \geq 1. \\
 & uv^0w = a^t b^{t-k} \notin L. \\
\underline{\textbf{Case 3}}: & v = a^k b^l,\ \ k, l \geq 1. \\
 & \textbf{Then } uv^2w = a^t b^l a^k b^t \notin L.
\end{array}$$

All the possible decompositions fail.
Therefore, L **is not a DFA language.**

Proof (II):

....

$|Q| = p$.

Consider $x = a^p b^p$ in L. Obviously, $|x| \geq p$.

By DFA pumping lemma, there is a decomposition

$$x = uvw$$

that satisfies (i), (ii), and (iii).

The only decompositions that satisfy (i) and (ii) are the following:

$$v = a^k \textbf{ for } p \geq k \geq 1.$$

But $uv^0w = a^{p-k}b^p \notin L$.
So, there is no such decomposition.
L is not a DFA language.

(i) $|uv| \leq p$;
(ii) $|v| \geq 1$;

Notes on proving that a language is not a DFA language by pumping lemma

1) **Assume L is a DFA language.**
   Then we have a constant p.
2) **Choose a word in** L **of length** $\geq p$.
3) **Consider all the possible decompositions of** x. **If none of them satisfy (i), (ii), and (iii) at the same time, then conclude that such decomposition does not exist.**

So, L **is not a DFA language.**