ECS120 Introduction to the Theory of Computation
Fall Quarter 2007

Discussion Notes
Wednesday, October 10, 2007

Notes on Notation

Σ^* : Notation for any string of any length. (p64)
Σ^n : Notation for any string of length n.
x^R : Notation for the reverse of string x. (p14)

If you are using LaTeX to type your solutions, here are some useful commands:

Output:	**Source:**
Σ^n	`$\Sigma^{n}$`
Γ^*	`$\Gamma^{*}$`
$\overline{L}$	`$\overline{L}$`
$\widehat{\delta}(q_0, x_i)$	`$\widehat{\delta}( q_{0}, x_{i} )$`
$w \in L$	`$w \in L$`
$w \notin L$	`$w \not\in L$`
$\{0, 1\}$	`$\{ 0, 1 \}$`
$M = (Q, \Sigma, \delta, q_0, F)$	`$M = ( Q, \Sigma, \delta, q_{0}, F )$`

Using JFLAP

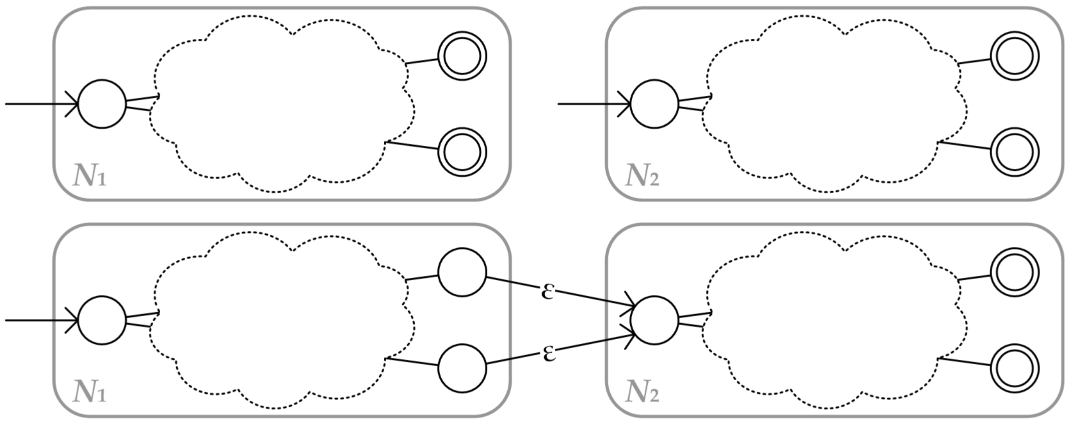
JFLAP is a great tool for playing around with state machines. However, it should only be used *as a tool.* You must be able to perform by hand many of the operations done automatically in JFLAP.

For example, it is appropriate to use JFLAP to test if your DFA does what you designed it for. It is not appropriate to attach output *generated by* JFLAP as an answer. (For example, for the first part of problem 4.) You need to exhibit enough work to prove that you performed the operation yourself.

See Professor Filkov's post on the discussion newsgroup for more.

Other Operations

Concatenation



Let $L_1(N_1)$ and $L_2(N_2)$ where $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$.

We construct the NFA N_3 such that $L_3(N_3) = L_1 \circ L_2$ as follows:

- $Q_3 = Q_1 \cup Q_2$ is set of states
- Σ is the alphabet
- δ_3 is the transition function such that for any $q \in Q_3$ and $a \in \Sigma_\epsilon$:

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & \text{where } q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q, a) & \text{where } q \in F_1 \text{ and } a \neq \epsilon \\ \delta_1(q, a) \cup \{ q_2 \} & \text{where } q \in F_1 \text{ and } a = \epsilon \\ \delta_2(q, a) & \text{where } q \in Q_2 \end{cases}$$

- q_1 is the start state, and
- F_2 is the set of accept states

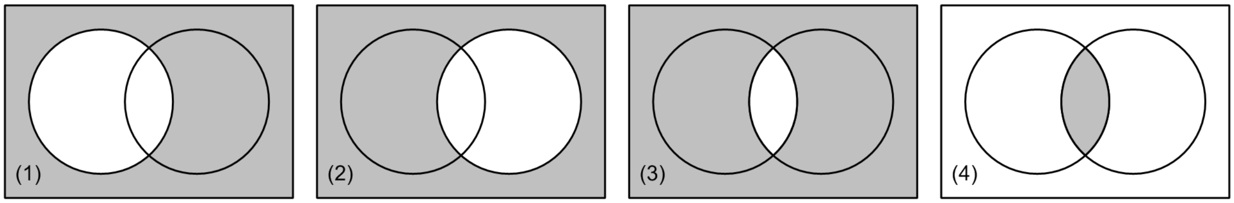
Thus we have $N_3 = (Q_3, \Sigma, \delta_3, q_1, F_2)$. This is also given in the book on page 61.

Intersection

How do we take the intersection of languages? The operations introduced so far include union, concatenation, complement, and star. In fact, we can use these operations to form the intersection:

1. Find the complement of L_1.
2. Find the complement of L_2.

3. Find the union of the complement of L_1 with the complement of L_2.
4. Take the complement of the union.



Examples

Example 1.13 and 1.15

Sometimes, it may be easier to define the transition function directly versus trying to draw a DFA or NFA.

Lets look at examples 1.13 and 1.15 from the book:

Let $\Sigma = \{ \langle\text{RESET}\rangle, 0, 1, 2 \}$. For each $i \geq 1$ let A_i be the language of all strings where the sum of the numbers is a multiple of i, except that the sum is reset to 0 whenever the symbol $\langle\text{RESET}\rangle$ appears.

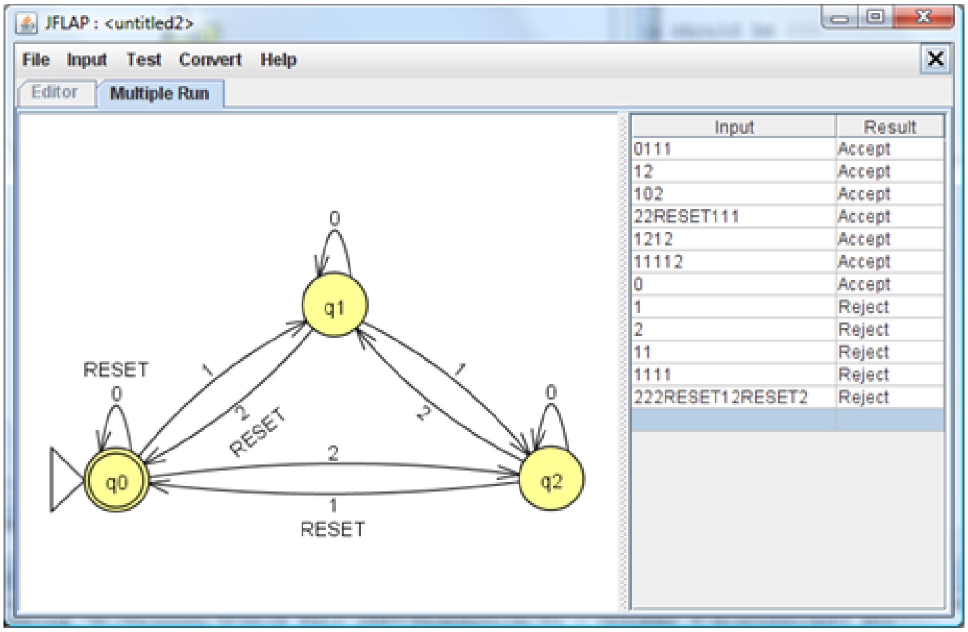
For example, let $i = 3$. Consider the string $11\langle\text{RESET}\rangle12$. After reading the 11, the sum should be 2. After the $\langle\text{RESET}\rangle$, the sum should be 0. Finally, after the input 12, the sum should be 3. Therefore, since 3 is a multiple of 3, we accept the string.

Let each state represent the sum modulo 3. Therefore the start state q_0 represents $0 \mod 3$. If we see a 0 our sum is still $0 \mod 3$ so we stay in q_0. If we see a $\langle\text{RESET}\rangle$, the sum is reset to 0 so we stay in q_0. If we see a 1 our sum is now $1 \mod 3$ so we must move to q_1. If we see a 2 our sum is now $2 \mod 3$ so we must move to q_2.

Eventually, we end up with the following table:

δ	0	1	2	$\langle\text{RESET}\rangle$
q_0	q_0	q_1	q_2	q_0
q_1	q_1	q_2	q_0	q_0
q_2	q_2	q_0	q_1	q_0

When we draw this DFA we get:



Notice that on a 0 we always stay in the same state, and on a $\langle\text{RESET}\rangle$ we always go back to q_0.

Now, we must generalize this concept. To generalize this, we build a machine B_i as follows:

- $Q_i = \{ q_0, q_1, \ldots, q_{i-1} \}$ is the set of states
- $\Sigma = \{ \langle\text{RESET}\rangle, 0, 1, 2 \}$ is the alphabet
- For each $q_j \in Q_i$ we define:

$$\begin{aligned}
\delta_i(q_j, 0) &= q_j \\
\delta_i(q_j, 1) &= q_k \text{ where } k = j + 1 \mod i \\
\delta_i(q_j, 2) &= q_k \text{ where } k = j + 2 \mod i \\
\delta_i(q_j, \langle\text{RESET}\rangle) &= q_0
\end{aligned}$$

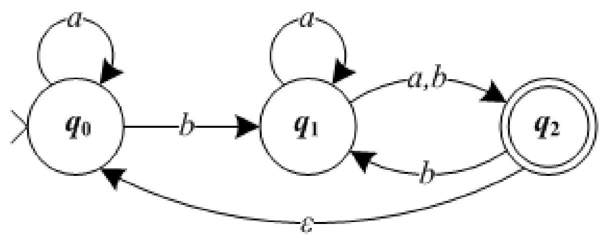
- q_0 is the start state
- $F_i = \{ q_0 \}$ is the set of accept states

Thus giving us $B_i = (Q_i, \Sigma, \delta_i, q_0, F_i)$.

Converting NFA $\rightarrow$ DFA

Let $N = (Q, \Sigma, \delta, q_0, F)$ where:

- The set of states Q is defined as $Q = \{q_0, q_1, q_2\}$.
- The alphabet Σ is defined as $\Sigma = \{a, b\}$.
- The transition function δ be defined by the following diagram:



- The set of accepting states F is defined as $F = \{q_2\}$.

Lets try to convert N into a DFA $M = (Q', \Sigma, \delta', q_0', F')$ using the table method. (This avoids unnecessary states ending up in the DFA.)

Note that I'll be following ϵ-transitions *after* a symbol is read! We only have one ϵ-transition to worry about here, from $q_2 \to q_0$. Therefore, whenever we put q_2 in our table, we also need to put down q_0 as a reachable state.

First, create a table that looks like this:

state	a	b
$\{q_0\}$		

Now, we get to fill in this table. Start at q_0 and put down all the states reachable from q_0:

state	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_1\}$

Next, add any new states introduced in the right of the table to the left column:

state	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_1\}$
$\{q_1\}$		

The next step illustrates what must happen when encountering an ϵ-transition. Notice that q_2 is reachable from q_1 on an a. Since q_2 has an ϵ-transition to q_0, this means q_0 is also reachable from q_1 as follows:

$$q_1 \xrightarrow{a} q_2 \xrightarrow{\epsilon} q_0$$

After filling everything in for q_1, we get:

state	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_1\}$
$\{q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$

Again, we add any new states (or sets of states) to the left column in the table:

state	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_1\}$
$\{q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$		
$\{q_0, q_1, q_2\}$		

Time to figure out the transitions for $\{q_0, q_2\}$. First, fill in the states reachable from q_0 (which is already in our table):

state	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_1\}$
$\{q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0 \ldots$	$\{q_1 \ldots$
$\{q_0, q_1, q_2\}$		

Then, fill in the states reachable from q_2 (after a symbol is read). In this particular step, there are no new states reachable from q_2 that we didn't have before:

state	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_1\}$
$\{q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0\}$	$\{q_1\}$
$\{q_0, q_1, q_2\}$		

This process keeps going until our table is full and no new states are added:

state	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_1\}$
$\{q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0\}$	$\{q_1\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

Tada! Notice we only had to consider four states here: $\{q_0\}, \{q_1\}, \{q_0, q_2\}$, and $\{q_0, q_1, q_2\}$. What happened to the rest of the states in $\mathcal{P}(Q)$? Well, you could still include them if you want. However, those states are not reachable from the start state, and can be eliminated from the DFA.

We aren't done yet! We need to define all parts of our new DFA. So here it goes. Let $M = (Q', \Sigma, \delta', q_0', F')$ such that:

- The set of states Q' is defined as $Q' = \{\{q_0\}, \{q_1\}, \{q_0, q_2\}, \{q_0, q_1, q_2\}\}$.
- The transition function δ' is defined by the table:

state	a	b
$\{q_0\}$	$\{q_0\}$	$\{q_1\}$
$\{q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0\}$	$\{q_1\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

- The start state q_0' is defined as $q_0' = \{q_0\}$.
- The set of final states F' is defined as: $F' = \{\{q_0, q_2\}, \{q_0, q_1, q_2\}$. (This should be the set of all states in Q' which contain any state from F. In our case, that includes any state in Q' which contains q_2.)

Granted, sometimes this method does not save you any work. However, when it does it is worth it!

Finally, here is a picture of the final DFA. (Note that I am using the shorthand q_{ij} to denote state $\{q_i, q_j\}$, etc.)