CMSC 330: Organization of Programming Languages

Theory of Regular Expressions
DFAs and NFAs

Reminders

- Project 1 due Sep. 24
- Homework 1 posted
- Exam 1 on Sep. 25

- Exam topics list posted
- Practice homework (and solutions) posted
Previous Course Review

- \{s \mid s \text{ defined}\} \text{ means the set of string } s \text{ such that } s \text{ is chosen or defined as given}
- s \in A \text{ means } s \text{ is an element of the set } A
- De Morgan’s Laws:
  \[(A \cap B)^c = A^c \cup B^c\]
  \[(A \cup B)^c = A^c \cap B^c\]

- There exists and for all symbols
- Etc…

Review

- Basic parts of a regular expression?
  composition, |, *, ε, ∅, \{a\}

- What does a DFA do?

- Basic parts of a DFA?
  alphabet, set of states, start state, final states, transition function \((Σ, Q, q_0, F, δ)\)
Example DFA

- $S_0$ = "Haven't seen anything yet" OR "seen zero or more b's" OR "Last symbol seen was a b"
- $S_1$ = "Last symbol seen was an a"
- $S_2$ = "Last two symbols seen were ab"
- $S_3$ = "Last three symbols seen were abb"

• Language?
  • $(a|b)^*abb$

Notes about the DFA definition

• Can not have more than one transition leaving a state on the same symbol
  – the transition function must be a valid function)

• Can not have transitions with no or empty labels
  – the transitions must be labeled by alphabet symbols
Nondeterministic Finite Automata (NFA)

• An NFA is a 5-tuple \((\Sigma, Q, q_0, F, \delta)\) where
  – \(\Sigma\) is an alphabet
  – \(Q\) is a nonempty set of states
  – \(q_0 \in Q\) is the start state
  – \(F \subseteq Q\) is the set of final states
  • There may be 0, 1, or many
  – \(\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q\) specifies the NFA's transitions
    • Transitions on \(\varepsilon\) are allowed – can optionally take these transitions without consuming any input
    • Can have more than one transition for a given state and symbol

• An NFA accepts \(s\) if there is at least one path from its start to final state on \(s\)

NFA for \((a|b)^*abb\)

• \(ba\)
  – Has paths to either \(S_0\) or \(S_1\)
  – Neither is final, so rejected

• \(babaabb\)
  – Has paths to different states
  – One leads to \(S_3\), so accepted
Another example DFA

Language?
• (ab|aba)*

NFA for (ab|aba)*

• aba
  – Has paths to states S0, S1
• ababa
  – Has paths to S0, S1
  – Need to use $\varepsilon$-transition
Relating R.E.'s to DFAs and NFAs

- Regular expressions, NFAs, and DFAs accept the same languages!

DFA  can transform  NFA
    can transform
    can transform
    r.e.
    (we'll discuss this next)

Reducing Regular Expressions to NFAs

- Goal: Given regular expression $e$, construct NFA: $<e> = (\Sigma, Q, q_0, F, \delta)$
  - Remember r.e. defined recursively from primitive r.e. languages
  - Invariant: $|F| = 1$ in our NFAs
    - Recall $F = \text{set of final states}$

- Base case: $a$

  $<a> = ((a), \{S0, S1\}, S0, \{S1\}, \{(S0, a, S1)\})$
Reduction (cont’d)

- Base case: $\varepsilon$

\[
\langle \varepsilon \rangle = (\varepsilon, \{S0\}, S0, \{S0\}, \emptyset)
\]

- Base case: $\emptyset$

\[
\langle \emptyset \rangle = (\emptyset, \{S0, S1\}, S0, \{S1\}, \emptyset)
\]

Reduction (cont’d)

- Induction: $AB$

\[
\langle A \rangle \quad \langle B \rangle
\]
Reduction (cont’d)

- Induction: $AB$

\[
\begin{align*}
\langle A \rangle & = (\Sigma_A, Q_A, q_A, \{f_A\}, \delta_A) \\
\langle B \rangle & = (\Sigma_B, Q_B, q_B, \{f_B\}, \delta_B) \\
\langle AB \rangle & = (\Sigma_A \Sigma_B, Q_A \cup Q_B, q_A, \{f_B\}, \delta_A \cup \delta_B \cup \{(f_A, \epsilon, q_B)\})
\end{align*}
\]

Practice

- Draw the NFA for these regular expressions using exactly the reduction method:
  - ab
  - hello

- Write the formal (5-tuple) NFA for the same regular expressions
Reduction (cont’d)

• Induction: \((A|B)\)

\[
\begin{align*}
\implies A & = (G^2d0, A, QA, qA, \{fA\}, G^2e1, A) \\
\implies B & = (G^2d0, B, QB, qB, \{fB\}, G^2e1, B) \\
\implies (A|B) & = (G^2d0, A, QA, qA, \{fA\}, G^2e1, A) \\
\end{align*}
\]

Reduction (cont’d)

• Induction: \((A|B)\)

\[
\begin{align*}
\implies A & = (\Sigma_A, Q_A, q_A, \{f_A\}, \delta_A) \\
\implies B & = (\Sigma_B, Q_B, q_B, \{f_B\}, \delta_B) \\
\implies (A|B) & = (\Sigma_A\cup\Sigma_B, Q_A\cup Q_B\cup\{S0,S1\}, S0, \{S1\}, \\
\delta_A\cup\delta_B\cup\{(S0,\epsilon,q_A), (S0,\epsilon,q_B), (f_A,\epsilon,S1), (f_B,\epsilon,S1)\})
\end{align*}
\]

CMSC 330

17

CMSC 330

18
Practice

- Draw the NFA for these regular expressions using exactly the reduction method:
  - $ab | bc$
  - $hello | hi$

- Write the formal NFA for the same regular expressions

Reduction (cont’d)

- Induction: $A^*$
Reduction (cont’d)

• Induction: $A^*$

- $<A> = (\Sigma_A, Q_A, q_A, \{f_A\}, \delta_A)$
- $<A^*> = (\Sigma_A, Q_A \cup \{S0, S1\}, S0, \{S1\},$
  $\delta_A \cup \{(f_A, \epsilon, S1), (S0, \epsilon, q_A), (S0, \epsilon, S1), (S1, \epsilon, S0)\})$

Practice

• Draw the NFA for these regular expressions using exactly the reduction method:
  – $(ab \mid bc^*)^*$
  – hello $\mid (hi)^*$

• Write the formal NFA for the same regular expressions
Reduction Complexity

- Given a regular expression $A$ of size $n$...
  
  Size = # of symbols + # of operations

- How many states does $<A>$ have?
  - 2 added for each $\mid$, 2 added for each $^*$
  - $O(n)$
  - That’s pretty good!

Practice

Draw NFAs for the following regular expressions and languages:

- $(0|1)^*110^*$
- $101^*111$
- all binary strings ending in 1 (odd numbers)
- all alphabetic strings which come after “hello” in alphabetic order
- $(ab^*c|d^*a|ab)d$
Handling $\varepsilon$-transitions

What if we want to remove all those unneeded $\varepsilon$-transitions?

First, some definitions:
- We say: $p \xrightarrow{\varepsilon} q$
  - if it is possible to transition from state $p$ to state $q$ taking only $\varepsilon$-transitions
  - if $\exists p, p_1, p_2, \ldots, p_n, q \in Q (p \neq q)$ such that
    $\{p, \varepsilon, p_1\} \in \delta, \{p_1, \varepsilon, p_2\} \in \delta, \ldots, \{p_n, \varepsilon, q\} \in \delta$

$\varepsilon$-closure

- For any state $p$, the $\varepsilon$-closure of $p$ is defined as the set of states $q$ such that $p \xrightarrow{\varepsilon} q$
- $\{q \mid p \xrightarrow{\varepsilon} q\}$
Example

- What’s the $\varepsilon$-closure of S2 in this NFA?

$\{S_2, S_0\}$

Example

- Find the $\varepsilon$-closure for each of the states in this NFA:
Example

- Make the NFA for the regular expression
  \(- (0|1^*)111(0^*1)\)

- Find the epsilon closure for each of the states of your NFA