Harvard CS 121 and CSCI E-207

Lecture 6: Regular Expressions

Harry Lewis

September 21, 2010

- **Reading:** Sipser, §1.3.

Reprise on the optimality of the subset construction

Could it be that for any NFA N, there is an equivalent DFA with fewer than 2^n states, where n is the number of states of N?

To disprove this, show that there exist "bad" NFAs of every size

Theorem: For every $n \geq 1$, there is a language L_n such that

1. There is an $(n+1)$-state NFA recognizing L_n.
2. There is no DFA recognizing L_n with fewer than 2^n states.

$$\begin{aligned} L_n \quad = \quad & \{w \in \{a,b\}^* : \text{the } n\text{th symbol} \\ & \quad \text{from the right end of } w \text{ is an } a\} \end{aligned}$$

Conclusion: For finite automata, nondeterminism provides an *exponential savings* (in the worst case).

Regular Expressions

- Let $\Sigma = \{a, b\}$. The **regular expressions** over Σ are certain expressions formed using the symbols $\{a, b, (,), \varepsilon, \emptyset, \cup, \circ, {}^*\}$

- We use **red** for the strings under discussion (the **object language**) and **black** for the ordinary notation we are using for doing mathematics (the **metalanguage**).

- Construction Rules (= inductive/recursive definition):

1. a, b, ε, $\emptyset$ are regular expressions
2. If R_1 and R_2 are RE's, then so are $(R_1 \circ R_2)$, $(R_1 \cup R_2)$, and (R_1^*).

- Examples:

$(a \circ b)$ $\quad ((((a \circ (b^*)) \circ c) \cup ((b^*) \circ a))^*)$ $\quad (\emptyset^*)$

What REs Do

- Regular expressions (which are strings) represent languages (which are sets of strings), via the function L:

$$
\begin{array}{lrcl}
(1) & L(a) & = & \{a\} \\
(2) & L(b) & = & \{b\} \\
(3) & L(\varepsilon) & = & \{\varepsilon\} \\
(3) & L(\emptyset) & = & \emptyset \\
(4) & L((R_1 \circ R_2)) & = & L(R_1) \circ L(R_2) \\
(5) & L((R_1 \cup R_2)) & = & L(R_1) \cup L(R_2) \\
(6) & L((R_1^*)) & = & L(R_1)^*
\end{array}
$$

- Example:

$$L(((a^*) \circ (b^*))) = \{a\}^* \circ \{b\}^*$$

- $L(\cdot)$ is called the **semantics** of the expression.

Syntactic Shorthand

- Omit many parentheses, because union and concatenation of languages are associative. For example,

  for any languages L_1, L_2, L_3:

  $$(L_1L_2)L_3 = L_1(L_2L_3)$$

  and therefore for any regular expressions R_1, R_2, R_3,

  $$L\big((R_1 \circ (R_2 \circ R_3))\big) = L\big(((R_1 \circ (R_2 \circ R_3)))\big)$$

- Omit $\circ$ symbol

- Drop the distinction between red and black, between object language and metalanguage.

Semantic equivalence

The following are equivalent:

$$((ab)c) \qquad (a(bc)) \qquad abc$$

or strictly speaking

$$((a \circ b) \circ c) \qquad (a \circ (b \circ c))$$

- **Equivalent** means:

  "same semantics—same $L(\cdot)$-value—maybe different syntax"

More syntactic sugar

- By convention, $*$ takes precedence over $\circ$, which takes precedence over $\cup$.

  So $a \cup bc^*$ is equivalent to $(a \cup (b \circ (c^*)))$.

- Σ is shorthand for $a \cup b$ (or the analogous RE for whatever alphabet is in use).

Examples of Regular Languages

Strings ending in $a = \Sigma^* a$

Strings containing the substring $abaab =$?

Strings of even length $= (aa \cup ab \cup ba \cup bb)^*$

Strings with even # of a's $= (b \cup ab^*a)^*$
$= b^*(ab^*ab^*)^*$

Strings with $\leq$ two a's $=$?

Strings of form $x_1 x_2 \cdots x_k$, $k \geq 0$, each $x_i \in \{aab, aaba, aaa\}$ = ?

Decimal numerals, no leading zeroes

$$= 0 \cup ((1 \cup \ldots \cup 9)(0 \cup \ldots \cup 9)^*)$$

All strings with an even # of a's <u>and</u> an even # of b's

$= (b \cup ab^*a)^* \cap (a \cup ba^*b)^*$ <u>but this isn't a regular expression</u>

Equivalence of REs and FAs

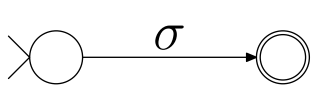
Recall: we call a language **regular** if there is a finite automaton that recognizes it.

Theorem: For every regular expression R, $L(R)$ is regular.

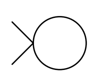
Proof (going back to hyper-formality for a moment):

Induct on the construction of regular expressions ("structural induction").

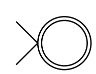
Base Case: R is a, b, ε, or $\emptyset$



accepts $\{\sigma\}$



accepts $\emptyset$



accepts $\{\varepsilon\}$

Equivalence of REs and FAs, continued

Inductive Step: If R_1 and R_2 are REs and $L(R_1)$ and $L(R_2)$ are regular (inductive hyp.), then so are:

$$\begin{aligned} L((R_1 \circ R_2)) &= L(R_1) \circ L(R_2) \\ L((R_1 \cup R_2)) &= L(R_1) \cup L(R_2) \\ L((R_1^*)) &= L(R_1)^* \end{aligned}$$

(By the closure properties of the regular languages).

Proof is constructive (actually produces the equivalent finite automaton, not just proves its existence).

Example Conversion of a RE to a FA

$$(a \cup \varepsilon)(aa \cup bb)^*$$

The Other Direction

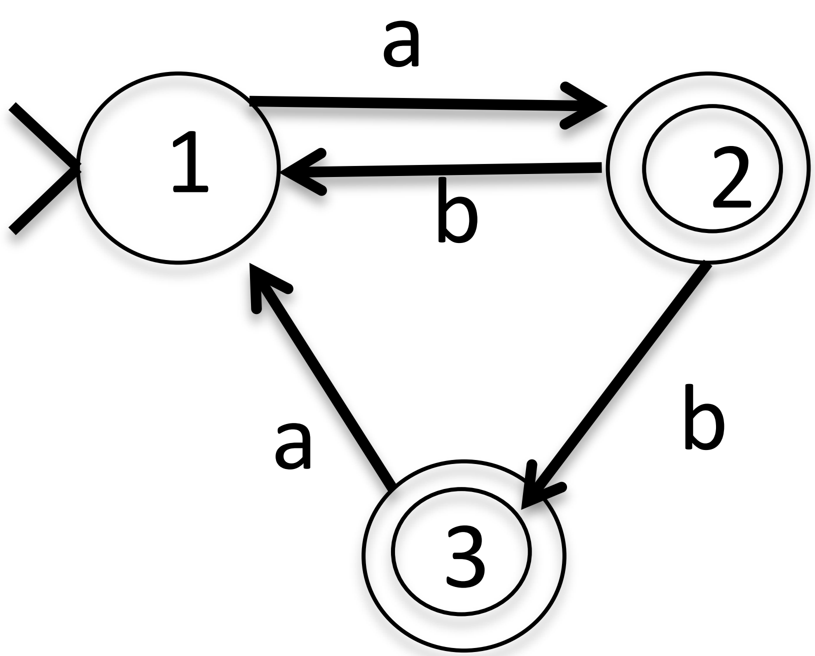
Theorem: For every regular language L, there is a regular expression R such that $L(R) = L$.

Proof:

Define generalized NFAs (GNFAs) (of interest only for this proof)

- Transitions labelled by regular expressions (rather than symbols).
- One start state q_{start} and only one accept state q_{accept}.
- Exactly one transition from q_i to q_j for every two states $q_i \neq q_{\text{accept}}$ and $q_j \neq q_{\text{start}}$ (including self-loops).

Example conversion of an NFA to a RE



Steps toward the proof

Lemma: For every NFA N, there is an equivalent GNFA G.

- Add new start state, new accept state. Transitions?
- If multiple transitions between two states, combine. How?
- If no transition between two states, add one. With what transition?

Lemma: For every GNFA G, there is an equivalent RE R.

- By induction on the number of states k of G.
- Base case: $k = 2$. Set R to be the label of the transition from q_{start} to q_{accept}.

Ripping and repairing GNFAs to reduce the number of states

- Inductive Hypothesis: Suppose every GNFA G of k or fewer states has an equivalent RE (where $k \geq 2$).

- Induction Step: Given a $(k+1)$-state GNFA G, we will construct an equivalent k-state GNFA G'.

  Rip: Remove a state q_r (other than q_{start}, q_{accept}).

  Repair: For every two states $q_i \notin \{q_{\text{accept}}, q_r\}$, $q_j \notin \{q_{\text{start}}, q_r\}$, let $R_{i,j}$, $R_{i,r}$, $R_{r,r}$, $R_{r,j}$ be REs on transitions $q_i \to q_j$, $q_i \to q_r$, $q_r \to q_r$ and $q_r \to q_j$ in G, respectively,

  In G', put RE $R_{ij} \cup R_{i,r}R_{r,r}^{*}R_{r,j}$ on transition $q_i \to q_j$.

  Argue that $L(G') = L(G)$, which is regular by IH.

Also constructive.

Examples of Regular Languages

- $\{w \in \{a,b\}^* : |w|$ even & every 3rd symbol is an $a\}$

- $\{w \in \{a,b\}^* :$ There are not 7 a's or 7 b's in a row$\}$

- $\{w \in \{a,b\}^* : w$ has both an even number of a's and an even number of b's$\}$

- Are there non-regular languages???

Goal: Existence of Non-Regular Languages

Intuition:

- Every regular language can be described by a finite string (namely a regular expression).
- To specify an arbitrary language requires an infinite amount of information.
  - For example, an infinite sequence of bits would suffice:
  - Σ^* has a lexicographic ordering, and the i'th bit of an infinite sequence specifying a language would say whether or not the i'th string is in the language.

$\Rightarrow$ Some language must not be regular.

How to formalize?

Countability

- A set S is finite if there is a bijection $\{1, \ldots, n\} \leftrightarrow S$ for some $n \geq 0$.

- Countably infinite if there is a bijection $f : \mathcal{N} \leftrightarrow S$

  This means that S can be "enumerated," i.e. listed as $\{s_0, s_1, s_2, \ldots\}$ where $s_i = f(i)$ for $i = 0, 1, 2, 3, \ldots$

  So $\mathcal{N}$ itself is countably infinite

  So is $\mathcal{Z}$ (integers) since $\mathcal{Z} = \{0, -1, 1, -2, 2, \ldots\}$

  Q: What is f?

- Countable if S is finite or countably infinite

- Uncountable if it is not countable

Facts about Infinite Sets

- **Proposition:** The union of 2 countably infinite sets is countably infinite.

  If $A = \{a_0, a_1, \ldots\}$, $B = \{b_0, b_1, \ldots\}$

  Then $A \cup B = C = \{c_0, c_1, \ldots\}$

  where $c_i = \begin{cases} a_{i/2} & \text{if } i \text{ is even} \\ b_{(i-1)/2} & \text{if } i \text{ is odd} \end{cases}$

  Q: If we are being fussy, there is a small problem with this argument. What is it?

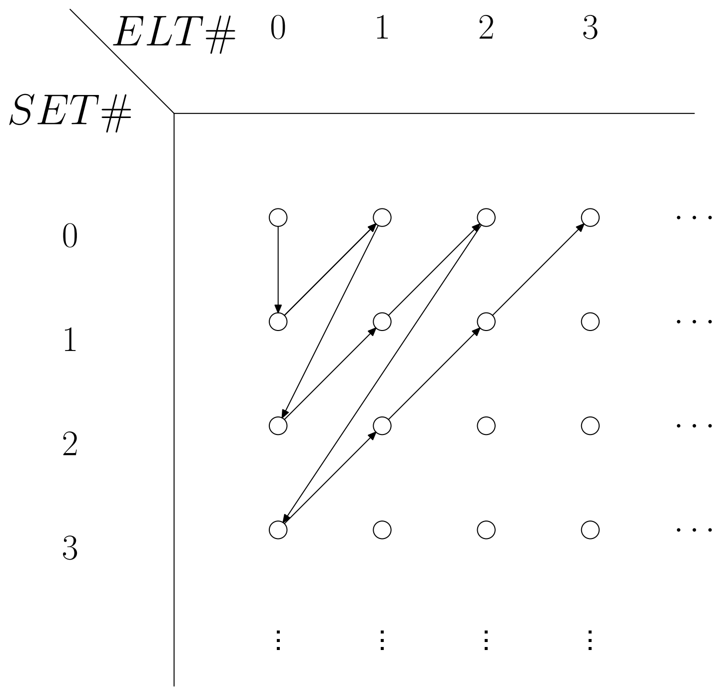
- **Proposition:** If there is a function $f : \mathcal{N} \to S$ that is onto S then S is countable.

Countable Unions of Countable Sets

- **Proposition:** The union of countably many countably infinite sets is countably infinite

Countable Unions of Countable Sets

- **Proposition:** The union of countably many countably infinite sets is countably infinite



$$(0,0),(0,1),(1,0),(0,2),(1,1),(2,0),\ldots$$

Each element is “reached” eventually in this ordering

Q: What is the bijection $\mathcal{N} \leftrightarrow \mathcal{N} \times \mathcal{N}$?

Are there uncountable sets?
(Infinite but not countably infinite)

Theorem: $P(\mathcal{N})$ is uncountable

(The set of all sets of natural numbers)

Proof by contradiction:

(i.e. assume that $P(\mathcal{N})$ is countable and show that this results in a contradiction)

- Suppose that $P(\mathcal{N})$ were countable.
- Then there is an enumeration of all subsets of $\mathcal{N}$ say $P(\mathcal{N}) = \{S_0, S_1, \ldots\}$

Diagonalization

$$\begin{array}{c|cccccc}
S_i \diagdown j = & 0 & 1 & 2 & 3 & 4 & \\
\hline
S_0 & Y & N & N & Y & N & \cdots \\
S_1 & N & N & N & N & N & \cdots \\
S_2 & Y & Y & N & Y & Y & \cdots \\
S_3 & N & N & N & Y & N & \cdots \\
\vdots & & & & D & &
\end{array}$$

"Y" in row i, column j means $j \in S_i$

- Let D = $\{i \in \mathcal{N} : i \in S_i\}$ be the diagonal.
- $D = YNNY \ldots = \{0, 3, \ldots\}$
- Let $\overline{D} = \mathcal{N} - D$ be its complement.
- $\overline{D} = NYYN \ldots = \{1, 2, \ldots\}$
- **Claim:** $\overline{D}$ is omitted from the enumeration, contradicting the assumption that every set of natural numbers is one of the S_is.

  Pf: $\overline{D}$ is different from each row because they differ at the diagonal.

Cardinality of Languages

- An alphabet Σ is finite by definition

- **Proposition:** Σ^* is countably infinite

- So every language is either finite or countably infinite

- $P(\Sigma^*)$ is uncountable, being the set of subsets of a countable infinite set.

  i.e. There are uncountably many languages over any alphabet

  Q: Even if $|\Sigma| = 1$?

Existence of Non-regular Languages

Theorem: For every alphabet Σ, there exists a non-regular language over Σ.

Proof:

- There are only countably many regular expressions over Σ.

  $\Rightarrow$ There are only countably many regular languages over Σ.

- There are uncountably many languages over Σ.
- Thus at least one language must be non-regular.

$\Rightarrow$ In fact, "almost all" languages must be non-regular.

Q: Could we do this proof using DFAs instead?

Q: Can we get our hands on an *explicit* non-regular language?