cs302 Theory of Computation	UVa Spring 2008

Notes: Deterministic Finite Automata

Thursday, 24 January

Upcoming Schedule:

- Monday, 2-3pm Office Hours (Olsson 236A)
- Monday, 5:30-6:30 Problem-Solving Session (Olsson 226D)
- **Tuesday, 29 January**: Problem Set 1 is due at the beginning of class
- Reading for next week: Finish Chapter 1

Problems and Machines

One of the main goals in this class is to understand what set of problems can be solved by different abstract machines. A *complexity class* is a set of problems of similar complexity. One way to define a complexity class is as the set of problems that can be solved by a specified abstract machine. Later, we will see complexity classes that further constrain the abstract machine by limiting the time or space available as a function of the size of the problem input.

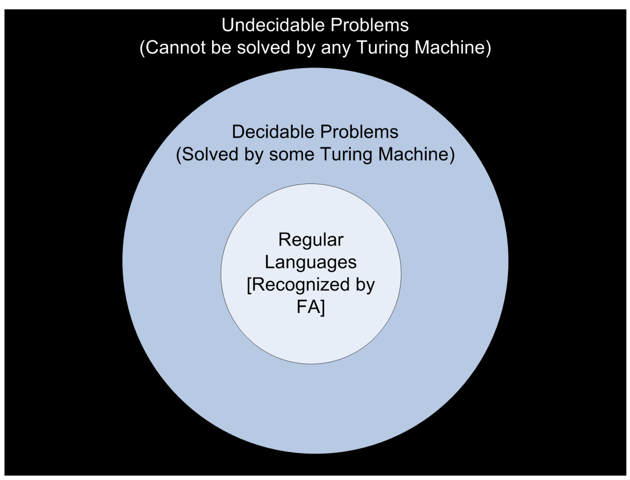

Figure 1: Complexity Classes

Where are the problems with finite input domains in the figure?

Are they any problems that can be solved by Finite Automata that cannot be solved by Turing Machine?

Definition: A *finite automaton* (DFA) is a 5-tuple (Q, Σ, δ, q_0, F):

1. Q — a *finite* set (the states)
2. Σ — a *finite* set (the alphabet)
3. δ: $Q \times \Sigma \to Q$ — a function from a state and alphabet symbol to a state (the transition function)
4. $q_0 \in Q$ — the start state
5. $F \subseteq Q$ — the set of accept states

How many start states can there be?
How many accept states can there be?
If we describe δ using a table, how big is the table?

$Q = \{A, B\}$
$\Sigma = \{0, 1\}$
δ is described by:

	0	1
A	A	B
B	B	A

$q_0 = A$
$F = \{B\}$

Definition: A *regular language* is a language that can be recognized by some finite automaton.

Language Recognition: $\mathcal{L}(A)$ represents the language recognized by DFA A:

$w \in \mathcal{L}(A) \Leftrightarrow$ running A on input w ends in an accept state

Computation Model for DFA
Define δ^* as the *extended transition function*:

$\delta^*(\qquad, w) \in \qquad \Leftrightarrow w \in \mathcal{L}(A)$