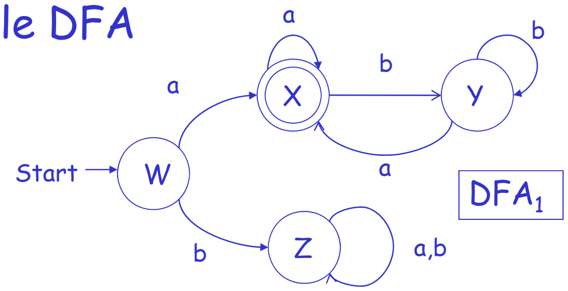
Deterministic Finite Automata (DFAs)

Friday, October 1, 2010
Reading: Sipser Sec. 1.1 (required), Kozen Ch. 3 (optional)

CS235 Languages and Automata

Department of Computer Science
Wellesley College

An Example DFA

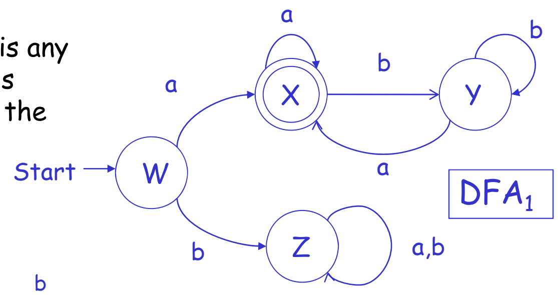


A **deterministic finite automaton (DFA)** is a kind of directed graph with labeled edges. It has:

- a set of nodes (**states**): {W,X, Y,Z}
- an **alphabet** to label the edges (transitions): {a, b}
- a unique **start state**: W
- a set of **final states**: {X} (here only one, but can be any number)
- every node must have exactly one labeled out-edge for every symbol in alphabet (**transition function**). Multiple edges between the same two states are sometimes drawn as one edge with multiple labels.

Labeled Paths

A **labeled path** in a DFA is any path in which each edge is labeled by the symbol on the transition taken.



Here are some labeled paths in DFA$_1$:

$W \overset{a}{\Rightarrow} X \overset{b}{\Rightarrow} Y \overset{b}{\Rightarrow} Y \overset{a}{\Rightarrow} X \overset{b}{\Rightarrow} Y$; label of this path is abbab

$Z \overset{b}{\Rightarrow} Z \overset{a}{\Rightarrow} Z \overset{a}{\Rightarrow} Z$; label of this path is baa

Y ; label of this path is ε

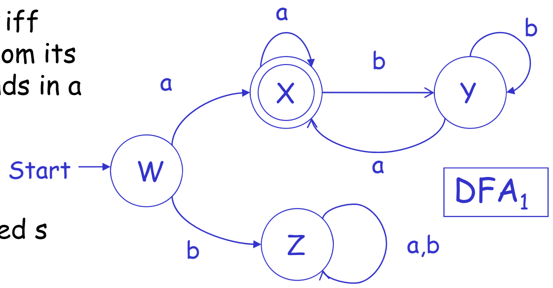
The **label of a path** is the concatenation of all the edge symbols.

In a DFA, for any state Q and string s there is a unique labeled path labeled s starting at Q.

Accepting/Rejecting Strings

A DFA **accepts** a string *s* iff the unique labeled path from its start state with label s ends in a final state.

Otherwise the DFA **rejects** the string (i.e., the unique path labeled s ends in a nonfinal state).



DFA$_1$ accepts abbabaa via $W \overset{a}{\Rightarrow} X \overset{b}{\Rightarrow} Y \overset{b}{\Rightarrow} Y \overset{a}{\Rightarrow} X \overset{b}{\Rightarrow} Y \overset{a}{\Rightarrow} X \overset{a}{\Rightarrow} X$

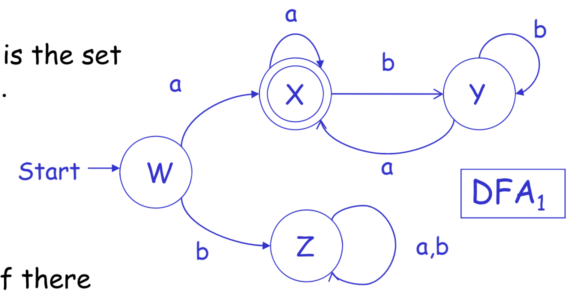
DFA$_1$ rejects abb via $W \overset{a}{\Rightarrow} X \overset{b}{\Rightarrow} Y \overset{b}{\Rightarrow} Y$

DFA$_1$ rejects babba via $W \overset{b}{\Rightarrow} Z \overset{a}{\Rightarrow} Z \overset{b}{\Rightarrow} Z \overset{b}{\Rightarrow} Z \overset{a}{\Rightarrow} Z$

A **dead state** is a state (like Z) that guarantees rejection if reached.

Regular Language = Language of a DFA

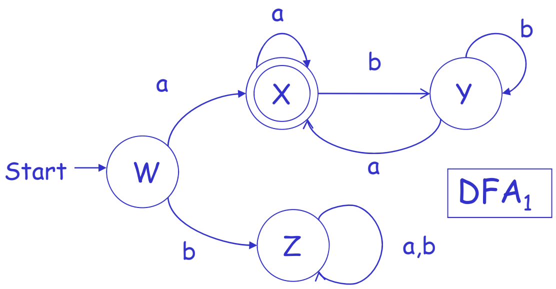
The **language** of a DFA is the set of all strings it accepts.



A language is **regular** iff there is some DFA that accepts it.

How would you describe the language of DFA1 in English?

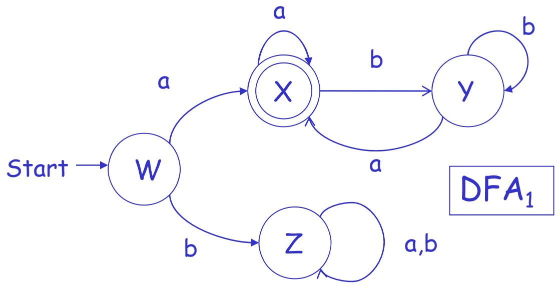
Another Formalization of DFAs



A DFA is a quintuple (Q, Σ, δ, s, F)

1. Q is a finite set of **states**	({W,X,Y,Z} ,
2. Σ is an **alphabet**	{a, b} ,
3. δ is a **transition function** in Q x Σ → Q	*see next slide,*
4. s in Q is a **start state**	W,
5. F ⊆ *Q* is a set of **final states**	{X})

Specifying The Transition Function δ



δ as a table:

	a	b
W	X	Z
X	X	Y
Y	X	Y
Z	Z	Z

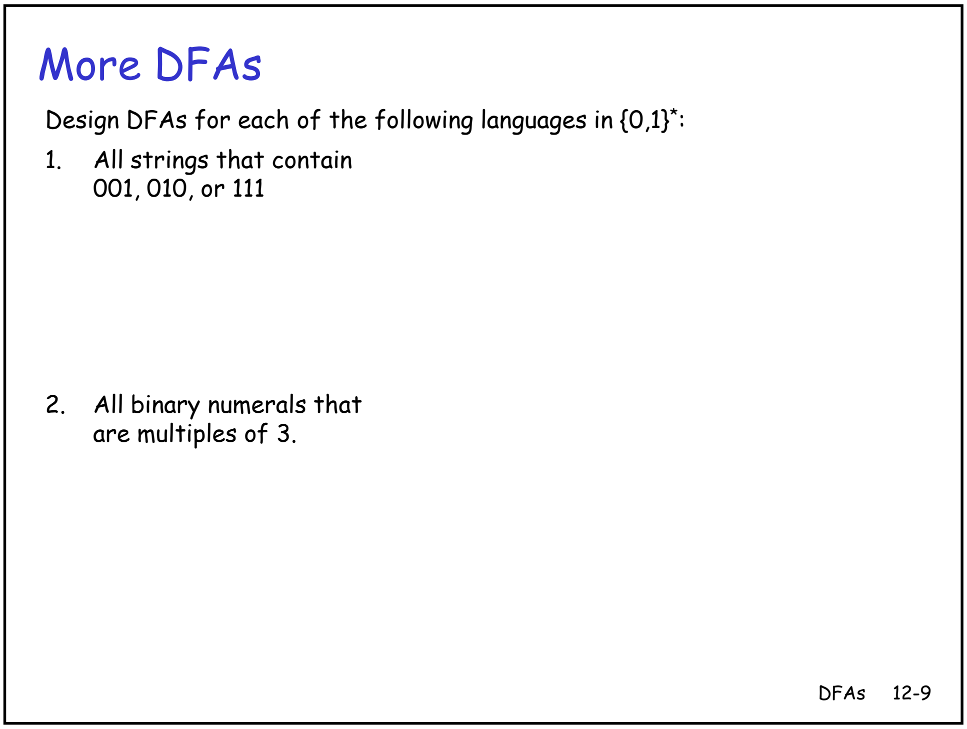
δ as a triple:

({W,X,Y,Z} x {a,b},
{W,X,Y,Z},
{ ((W,a), X), ((W,b), Z),
((X,a), X), ((X,b), Y),
((Y,a), X), ((Y,b), Y),
((Z,a), Z), ((Z,b), Z) }

Designing Some DFAs

Design DFAs for each of the following languages in {a,b}*:

1. All strings beginning with a
2. All strings ending in a
3. All strings that begin with a or end in a
4. All strings that begin and end with the same symbol



More DFAs

Design DFAs for each of the following languages in $\{0,1\}^*$:

1. All strings that contain
   001, 010, or 111

2. All binary numerals that
   are multiples of 3.