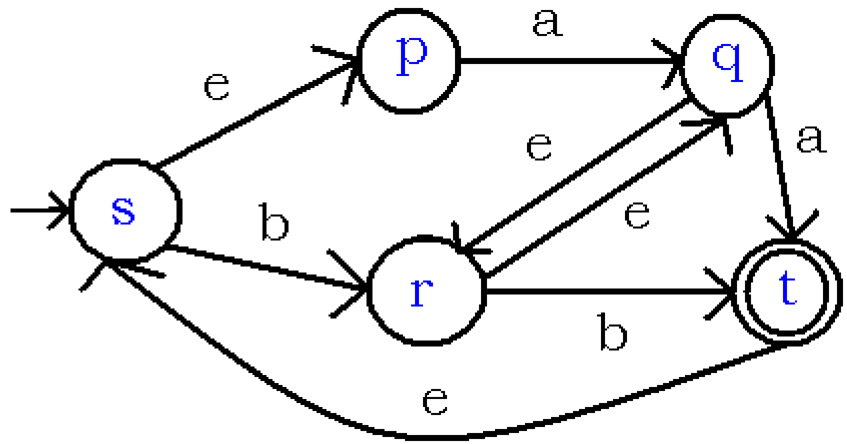
Convert to a DFA:

Start state:

Final States:



State	Symbol	Read- Q	E(Q)
	a		
	b		
	a		
	b		
	a		
	b		

Announcements

Assignment #2 is posted:

Due Friday June 2.

There is a tutorial today.

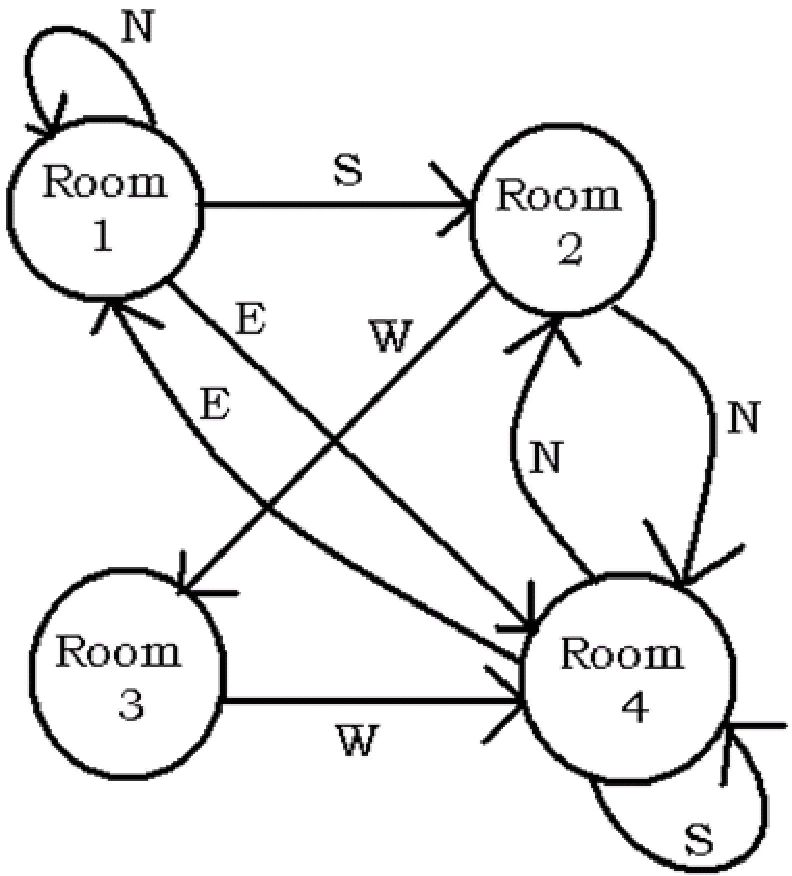
Colossal Cave Adventure (from Wikipedia)

In the mid 1970s, programmer, caver, and role-player William Crowther developed a program called *Colossal Cave Adventure*. The game used a text interface to create an interactive adventure through a spectacular underground cave system. Crowther's work was later modified and expanded by programmer Don Woods, and Colossal Cave Adventure became wildly popular among early computer enthusiasts, spreading across the nascent ARPANET throughout the 1970s.

A big fan of Tolkien, Woods introduced additional fantasy elements, such as elves and a troll. Adventure was the first game to feature objects that could be picked up, used, and dropped (and that could be carried by a non-player character).

You are in a maze of twisty little passages, all alike.

If the maze has n rooms and each one has trails exiting to the N, S, W, E. How many trails must be traversed before some room is visited more than once?



CSC 320 Lecture 10: DFA's and Regular Expressions

(Comic by Randall Munroe.)

Lecture 11: Regular Languages

The goal of the lecture is to prove that:

R= { L : L is generated by a regular expression}, and

S= { L : L is L(M) for a DFA(M)}

are the same sets of languages (R=S).

We already proved that S is the same set as

T= {L : L = L(M) for some NDFA M} (last class).

Before proving this, it is helpful to first show that the set S is closed under union, concatenation and Kleene star.

A set S is closed with respect to a binary operation · if for all s and t in S, s · t is in S.

Examples:

N= {0, 1, 2, 3, …}

Closed for addition and multiplication.

Not closed under subtraction or division.

The set S = { L : L is L(M) for some DFA M} is closed under union.

Let $M_1 = (K_1, \Sigma, \delta_1, s_1, F_1)$ accept L_1

and $M_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$ accept L_2 .

Proof 1: A construction for a new DFA $M = (K, \Sigma, \delta, s, F)$ which accepts $L_1 \cup L_2$.

Proof 2: A construction for a new NDFA $M = (K, \Sigma, \Delta, s, F)$ which accepts $L_1 \cup L_2$.

The set $S = \{ L : L \text{ is } L(M) \text{ for some DFA } M\}$ is closed under concatenation.

Let $M_1 = (K_1, \Sigma, \delta_1, s_1, F_1)$ accept L_1

and $M_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$ accept L_2.

Proof: A construction for a new NDFA $M = (K, \Sigma, \Delta, s, F)$ which accepts $L_1 \cdot L_2$.

The set S= { L : L is L(M) for some DFA M} is closed under Kleene star.

Let M_1= (K_1, Σ, δ_1, s_1, F_1) accept L_1.

Proof: A construction for a new NDFA M= (K, Σ, Δ, s, F) which accepts L_1*.

Theorem:

If L is a regular language, then L is L(M) for some DFA M.

Proof:

By showing how to construct a NDFA for L.

Last lecture, we proved by construction that for every NDFA, there is an equivalent DFA.

So this indirectly gives a construction for a DFA.

Regular expressions over Σ:

[Basis] 1. Φ and σ for each $\sigma \in \Sigma$ are regular expressions.

[Inductive step] If α and β are regular expressions, then so are:

2. $(\alpha\beta)$

3. $(\alpha \cup \beta)$ and

4. α^*

Note: Regular expressions are strings over

Σ U { (,) , Φ , U , * }

for some alphabet Σ.

Theorem: If L is accepted by a DFA M, then there is a regular expression which generates L.

There is a proof which constructs a regular expression from the DFA in the text (in the proof of Theorem 2.3.2). I expect you to know that this theorem is true but you are not responsible for the proof.

Conclusion: A language is regular if and only if it is accepted by a finite automaton.

The set of S= { L : L is L(M) for some DFA M} is closed under complement.

Let $M_1 = (K_1, \Sigma, \delta_1, s_1, F_1)$ accept L_1.

Proof: A construction for a new DFA M= (K, Σ, δ, s, F) which accepts the complement of L_1.

Regular languages are also closed for intersection (assignment #2).