Harvard CS121 and CSCI E-207

Lecture 3: Proofs and DFAs

Harry Lewis

September 9, 2010

Reading: Sipser, Chapter 0

What is a proof?

A proof is a formal argument of the truth of some mathematical statement.

- "Formal" means that the successive statements are unambiguous, and the steps interlock in a logical vise-grip.
- "Formal" also means that the argument could, in principle, be put in syntax that a machine could check.
- But proofs are meant to be read by human beings, and ordinary conventions and courtesies of human communication should be observed!

Why do we do proofs?

- To be absolutely positive the statement is true
  - For example, it is commonly believed that there are no fast, completely correct algorithms for the Traveling Salesman Problem. But until someone proves ”TSP is hard” we won’t know to stop looking
- To understand why it is true, so we can tell whether it can be extended or restricted and still remain true
- (Sometimes) so we can solve a problem associated with the proposition
  - if TSP is not hard, we’d like to find the algorithm

An example (Sipser, Theorem 0.20)

Prove that $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

An example (Sipser, Theorem 0.20)

Prove that $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

1. Do we know what the statement to be proved means?
   - What kinds of things does it talk about? (What are A and B?)
   - What does the notation mean? (What are $\cup$ and $\overline{\ }$?)

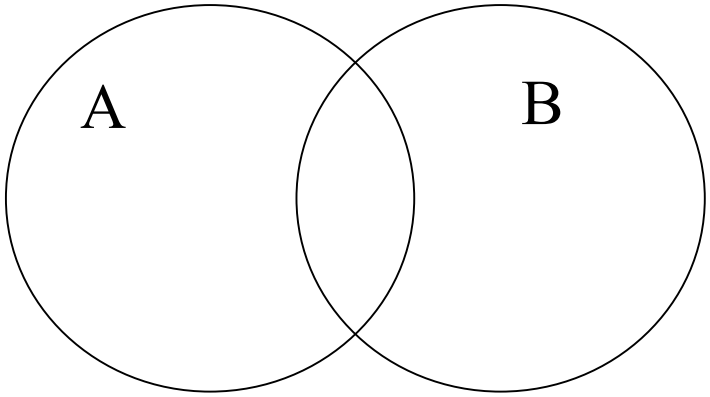
An example (Sipser, Theorem 0.20)

Prove that $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

1. Do we know what the statement to be proved means?
   - What kinds of things does it talk about? (What are A and B?)
   - What does the notation mean? (What are $\cup$ and $\overline{\ }$?)
2. Try a simple example first.
   - Say, $A = \{1, 2\}$ and $B = \{2, 3\}$.
   - Back up to Step 1 if necessary! Ask `cs121@seas.harvard.edu` or `cscie207@seas.harvard.edu` if necessary but most of the time the information is in the notes and problem sets.

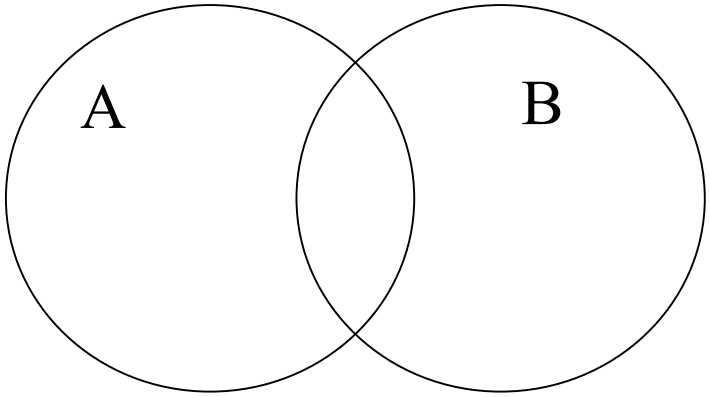
Proof example, continued

3. Try drawing a picture and play with it.



Proof example, continued

3. Try drawing a picture and play with it.



4. Decide on a proof strategy

   - If we are trying to prove two sets equal, a good strategy is *mutual inclusion*: Prove everything in the first is in the second, and then that everything in the second is in the first.
   - This illustrates a more general strategy: Try breaking the problem into simpler chunks and solving them separately.

Now really do the proof

1. Prove that for any sets A and B, $\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$.

2. Prove that for any sets A and B, $\overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$.

Hints for writing up good proofs (thanks largely to Tom Leighton)

1. State the game plan, including the general proof technique you are using.

Hints for writing up good proofs (thanks largely to Tom Leighton)

1. State the game plan, including the general proof technique you are using.

2. Keep the flow linear and use English to explain when you are moving from step to step.

Hints for writing up good proofs (thanks largely to Tom Leighton)

1. State the game plan, including the general proof technique you are using.

2. Keep the flow linear and use English to explain when you are moving from step to step.

3. Proofs are read by human beings, not machines.

Hints for writing up good proofs (thanks largely to Tom Leighton)

1. State the game plan, including the general proof technique you are using.

2. Keep the flow linear and use English to explain when you are moving from step to step.

3. Proofs are read by human beings, not machines.

4. Use as little new symbolism as possible, and use old symbolism correctly.

Hints for writing up good proofs (thanks largely to Tom Leighton)

1. State the game plan, including the general proof technique you are using.

2. Keep the flow linear and use English to explain when you are moving from step to step.

3. Proofs are read by human beings, not machines.

4. Use as little new symbolism as possible, and use old symbolism correctly.

5. Avoid “clearly,” which bullies the reader and often hides errors.

Hints for writing up good proofs (thanks largely to Tom Leighton)

1. State the game plan, including the general proof technique you are using.

2. Keep the flow linear and use English to explain when you are moving from step to step.

3. Proofs are read by human beings, not machines.

4. Use as little new symbolism as possible, and use old symbolism correctly.

5. Avoid “clearly,” which bullies the reader and often hides errors.

6. When you are done, explain why you are done.

Pigeonhole Principle

- If there are more pigeons than pigeonholes and every pigeon is in a pigeonhole, then some pigeonhole must contain at least two pigeons

- or more formally

- For any finite sets S and T and any function $f : S \to T$, if $|S| > |T|$ then there exist $s_1, s_2 \in S$ such that $s_1 \neq s_2$ but $f(s_1) = f(s_2)$

- A proof by pigeonhole: In any group of people, two have the same number of friends

A Little LaTeX

For any finite sets S and T and any function $f : S \rightarrow T$, if $|S| > |T|$ then there exist $s_1, s_2 \in S$ such that $s_1 \neq s_2$ but $f(s_1) = f(s_2)$

```
For any finite sets $S$ and $T$

and any function $f: S\rightarrow T$,

if $|S|>|T|$ then there exist $s_1,s_2\in S$

such that $s_1\neq s_2$ but $f(s_1)=f(s_2)$
```

LaTeX section

Friday (tomorrow) morning, 11Am, 1 Story St., Room 304 – will be taped for all to see

Nonconstructive proofs

- We have already seen a constructive proof–a proof that actually delivers the goods
  - Not every symmetric, transitive relation is reflexive
- The proof about people and their friends is nonconstructive

Nonconstructive Proof Example #2: Numbers with a certain property

- Proof that there exist irrational a, b such that a^b is rational
  - Is $\sqrt{2}^{\sqrt{2}}$ rational? Don't know. But consider both possibilities:
  1. $\sqrt{2}^{\sqrt{2}}$ is rational. In that case we are done ($a = b = \sqrt{2}$)
  2. $\sqrt{2}^{\sqrt{2}}$ is irrational. But then

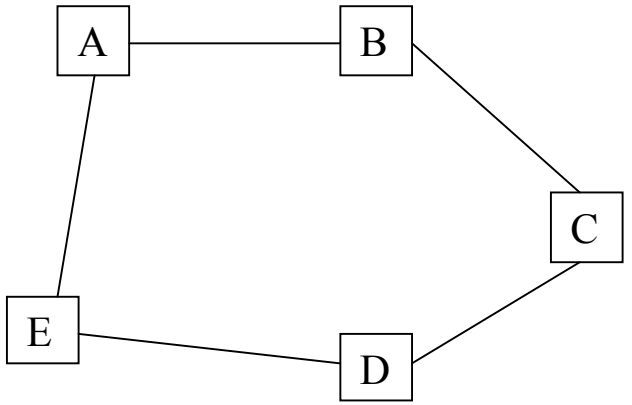
$$\left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}} = \left(\sqrt{2}\right)^{\sqrt{2}\cdot\sqrt{2}} = \left(\sqrt{2}\right)^2 = 2$$

     which is rational and we are done ($a = \sqrt{2}^{\sqrt{2}}, b = \sqrt{2}$)
- Proof does not tell us whether case 1 or case 2 holds, but one or the other must be true
- Law of the Excluded Middle or *tertium non datur*

A combined constructive and nonconstructive proof

(A) In any group of six people there are either three that know each other or three that don't, but (B) in some groups of five people there are no three who mutually know each other and also no three who mutually don't know each other.

- Constructive proof of (B):



Proof, continued

- Nonconstructive proof of (A) by contradiction
  - Suppose not. Then in some particular group of 6 people, there are no 3 who mutually know each other and no 3 who mutually don't.

Proof, continued

- Pick some individual X. Either X knows 3 of the other 5, or there are some 3 of the other 5 whom X does not know. (Pigeonhole)
  - If X knows 3, say A, B, C, then no two of them can know each other. For example if A knew B, then X, A, B would all know each other. But then no two of A, B, C know each other, contradiction.
  - If there are 3 whom X does not know, say A, B, C, then each two of those must know each other. For example, if A and B did not know each other, then no two of X, A, B would know each other. But then A, B, C all know each other, contradiction.

Proof, finis

- NB: The opposite of a “for all” statement is a “for some . . . not” or “there exists . . . not” statement.
- In theory at least, (A) could also be solved by **exhaustive search**.
- It would be enough to say that the second case is “symmetrical”
- This is also an example of case analysis

A nonconstructive proof that tells us almost nothing

- There exists an unbiased 7-sided die
- ???

Deterministic Finite Automata (DFAs)

Example: Home Stereo

- P = power button (ON/OFF)
- S = source button (CD/Radio/TV), only works when stereo is ON, but source remembered when stereo is OFF.
- Starts OFF, in CD mode.
- A computational problem: does a given a sequence of button presses $w \in \{P, S\}^*$ leave the system with the radio on?

Formal Definition of a DFA

- A DFA M is a 5-Tuple $(Q, \Sigma, \delta, q_0, F)$

  Q : Finite set of states
  Σ : Alphabet
  δ : "Transition function", Q x $\Sigma \rightarrow Q$
  q_0 : Start state, $q_0 \in Q$
  F : Accept (or final) states, $F \subseteq Q$

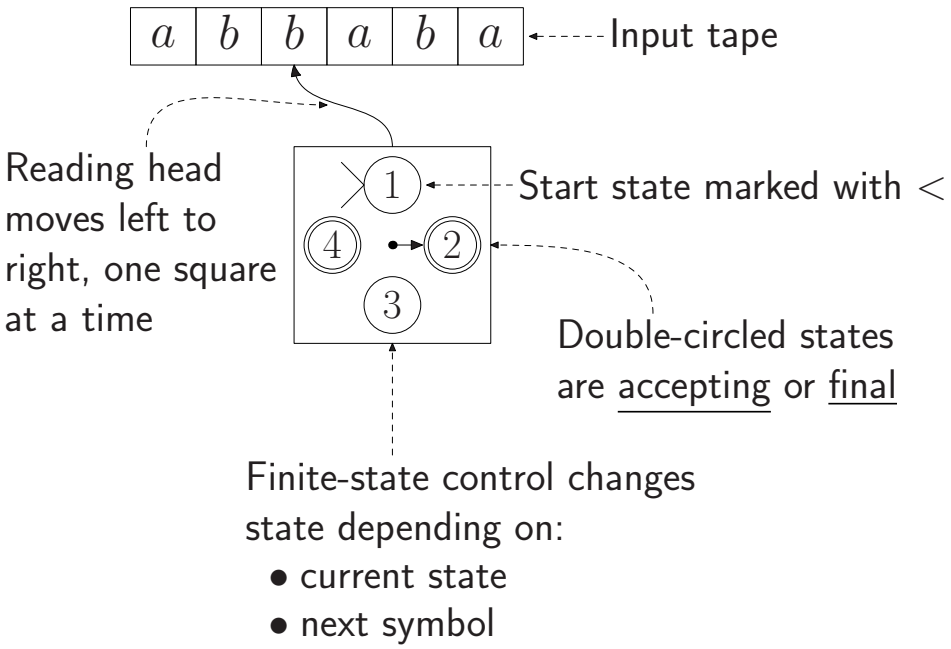
- If $\delta(p, \sigma) = q$,

  then if M is in state p and reads symbol $\sigma \in \Sigma$

  then M enters state q (while moving to next input symbol)

- Home Stereo example:

Another Visualization



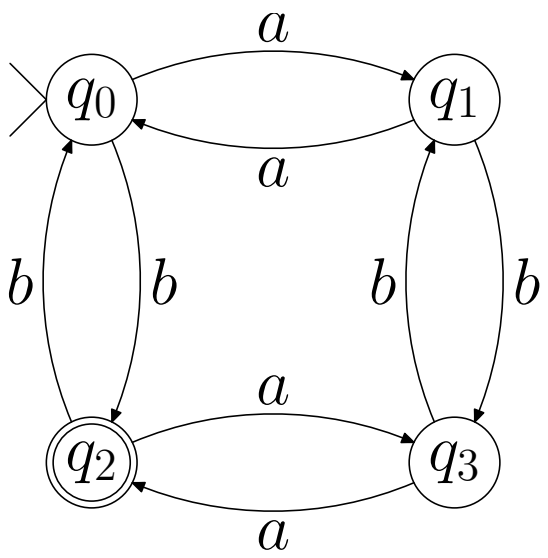
M accepts string X if

- After starting M in the start[initial] state with head on first square,
- when all of X has been read,
- M winds up in a final state.

Examples

- Bounded Counting: A DFA for

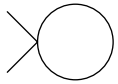
  $\{x : x$ has an even # of a's and an odd # of b's$\}$



Transition function δ:

	a	b
q_0	q_1	q_2
q_1	q_0	q_3
q_2	q_3	q_0
q_3	q_2	q_1.

i.e. $\delta(q_0, a) = q_1$, etc.

 = start state  = final state

$Q = \{q_0, q_1, q_2, q_3\}$ $\Sigma = \{a, b\}$ $F = \{q_2\}$

Another Example, to work out together

- Pattern Recognition: A DFA that accepts $\{ x : x$ has aab as a substring$\}$.

Another Example

- A DFA that accepts $\{ x : x$ has $ababa$ as a substring$\}$.

Another Example

- A DFA that accepts $\{ x : x$ has $ababa$ as a substring$\}$.

You are going through a constructive process

string $\rightarrow$ DFA

that is automated in every text editor!

Really a compiler that generates DFA code from an input string pattern

An interesting problem

How big is the smallest automaton that can tell two given strings apart?

Formal Definition of Computation

$M = (Q, \Sigma, \delta, q_0, F)$ accepts $w = w_1 w_2 \cdots w_n \in \Sigma^*$ (where each $w_i \in \Sigma$) if there exist $r_0, \ldots, r_n \in Q$ such that

1. $r_0 = q_0$,
2. $\delta(r_i, w_{i+1}) = r_{i+1}$ for each $i = 0, \ldots, n-1$, and
3. $r_n \in F$.

The language recognized (or accepted) by M, denoted $L(M)$, is the set of all strings accepted by M.

Transition function on an entire string

More formal (not necessary for us, but notation sometimes useful):

- Inductively define $\delta^* : Q \times \Sigma^* \to Q$ by $\delta^*(q, \varepsilon) = q$, $\delta^*(q, w\sigma) = \delta(\delta^*(q, w), \sigma)$.
- Intuitively, $\delta^*(q, w) =$ "state reached after starting in q and reading the string w.
- M accepts w if $\delta^*(q_0, w) \in F$.

Transition function on an entire string

More formal (not necessary for us, but notation sometimes useful):

- Inductively define $\delta^* : Q \times \Sigma^* \to Q$ by $\delta^*(q, \varepsilon) = q$, $\delta^*(q, w\sigma) = \delta(\delta^*(q, w), \sigma)$.
- Intuitively, $\delta^*(q, w) =$ "state reached after starting in q and reading the string w.
- M accepts w if $\delta^*(q_0, w) \in F$.

Determinism: Given M and w, the states $r_0, \ldots, r_n$ are uniquely determined. Or in other words, $\delta^*(q, w)$ is well defined for any q and w: There is precisely one state to which w "drives" M if it is started in a given state.