Writing Formal Language Theory Proofs: Finite-State Automata Example

11-711: Algorithms for NLP

September 11, 2015

Writing Formal Proofs

Proof Writing

Formal proof:

Starting from definitions, axioms, and stated assumptions, demonstrate step by step that a statement is necessarily true.

Proof Writing

Formal proof:

Starting from definitions, axioms, and stated assumptions, demonstrate step by step that a statement is necessarily true.

What are the starting axioms?

What kind of steps should I take?

How do I justify each step?

When am I done?

Proof Writing

When writing a proof, you are the instructor!

Make sure the reader fully understands the material.

This is often the case for research papers.

Proof Writing

What are the starting axioms?

- The reader is a student in your class
- Definitions from course material and problem statements
- Don't rely on knowledge beyond basic mathematical operations

Proof Writing

What steps should I take?

- Not just trying to convince the reader that you know what you're talking about
- Reader should be able to understand well enough to explain to someone else
- Don't try to impress the reader with great logical jumps
- If you think it, write it out!

Proof Writing

How do I justify each step?

- Definitions from course notes or problem description
- References to previously proven statements in the same problem/paper/homework
- Basic logical and mathematical operations and identities
- If you think a step may need more justification or even a few sentences explaining why it is correct, write it out!

Proof Writing

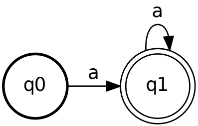
When am I done?

- Your last statement should be the statement you are trying to prove, with full justification
- When in doubt, go one step further!
- Ask yourself if a skeptical reader would be convinced

FSA Proof Examples

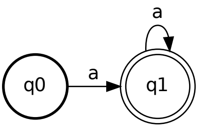
Dual Control Finite State Automaton

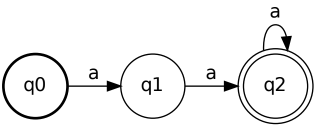
FSA



Dual Control Finite State Automaton

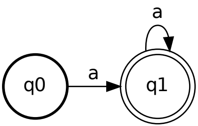
DCFSA





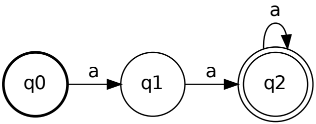
Dual Control Finite State Automaton

DCFSA



One input

Two machines



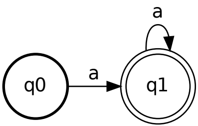
Dual Control Finite State Automaton

DCFSA $A = (Q_1, Q_2, \Sigma, \delta_1, \delta_2, q_0^1, q_0^2, F_1, F_2)$

- One alphabet
- Two independent sets of states (including initial and final)
- Two independent transition functions
- For each input symbol, both controls change state
- Accepts if both controls are in final state at end of input

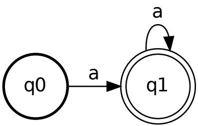
Dual Control Finite State Automaton

A_1



Dual Control Finite State Automaton

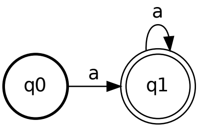
A_1



aa^*

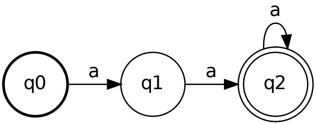
Dual Control Finite State Automaton

A_1



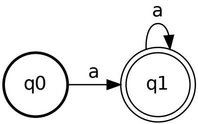
aa^*

A_2



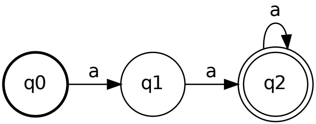
Dual Control Finite State Automaton

A_1



aa^*

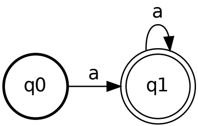
A_2



aaa^*

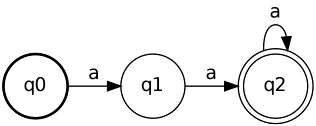
Dual Control Finite State Automaton

A_1



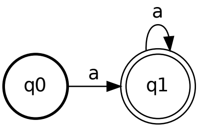
$$aa^* \cap aaa^*$$

A_2



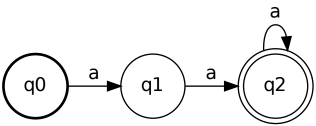
Dual Control Finite State Automaton

A_1



$aa^* \cap aaa^*$ Regular?

A_2



FSA-DCFSA Equivalence

Task 1:

Prove that the set of languages accepted by a Dual Control Finite State Automaton is regular.

FSA-DCFSA Equivalence

Task 1:

Prove that the set of languages accepted by a Dual Control Finite State Automaton is regular.

For DCFSA A, construct FSA A' that accepts the same language

Proof 1 Part 1

Proof Writing: First define your building blocks

Proof 1 Part 1

DCFSA $A = (Q_1, Q_2, \Sigma, \delta_1, \delta_2, q_0^1, q_0^2, F_1, F_2)$

Proof 1 Part 1

DCFSA $A = (Q_1, Q_2, \Sigma, \delta_1, \delta_2, q_0^1, q_0^2, F_1, F_2)$

FSA $A' = (Q', \Sigma', \delta', q_0', F')$

$$
\begin{aligned}
Q' &= ? \\
\Sigma' &= ? \\
\delta' &= ? \\
q_0' &= ? \\
F' &= ?
\end{aligned}
$$

$L(A') = L(A)$

Proof 1 Part 1

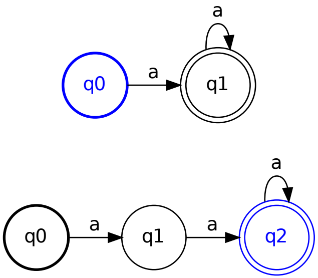
$$\Sigma' = \Sigma$$

Proof 1 Part 1

$Q' =$

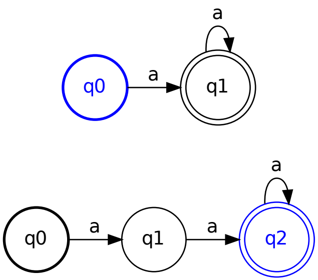
Proof 1 Part 1

$$Q' = \{[q_1, q_2] \mid q_1 \in Q_1, q_2 \in Q_2\}$$



Proof 1 Part 1

$$Q' = \{[q_1, q_2] \mid q_1 \in Q_1, q_2 \in Q_2\}$$



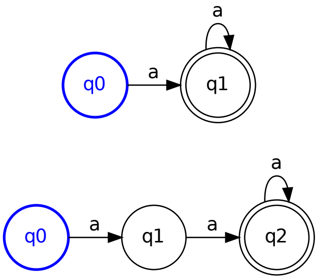
You don't have to draw a picture, but the reader should be able to!

Proof 1 Part 1

q'_0

Proof 1 Part 1

$q'_0 = [q^1_0, q^2_0]$

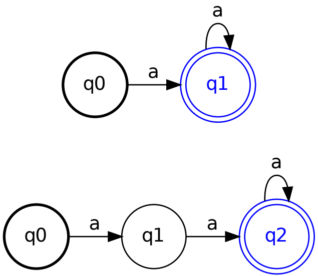


Proof 1 Part 1

$F' =$

Proof 1 Part 1

$$F' = \{[f_1, f_2] \mid f_1 \in F_1, f_2 \in F_2\}$$

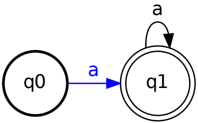


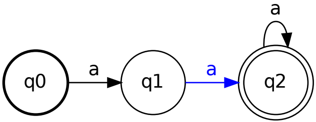
Proof 1 Part 1

$\delta'([q_1, q_2], a) =$

Proof 1 Part 1

$$\delta'([q_1, q_2], a) = [p_1, p_2] \iff \delta_1(q_1, a) = p_1 \land \delta_2(q_2, a) = p_2$$
$$q_1 \in Q_1,\ q_2 \in Q_2,\ a \in \Sigma$$





Proof 1 Part 2

We want to show that $w \in L(A')$ if and only if $w \in L(A)$:

$$\hat{\delta}'(q_0', w) = [p_1, p_2] \iff \left(\hat{\delta}_1(q_0^1, w) = p_1\right) \wedge \left(\hat{\delta}_2(q_0^2, w) = p_2\right)$$

This also includes states $\{[q_1, q_2] \mid q_1 \in F_1, q_2 \in F_2\}$

Proof 1 Part 2

Need to prove:

$$\hat{\delta}'(q_0', w) = [p_1, p_2] \iff \left(\hat{\delta}_1(q_0^1, w) = p_1\right) \wedge \left(\hat{\delta}_2(q_0^2, w) = p_2\right)$$

Proof by induction on length of w

Split $\boldsymbol{w}$: $\boldsymbol{w} = \boldsymbol{x}\sigma$, $\boldsymbol{x} \in \Sigma^*$ is a string and $\sigma \in \Sigma$ is a vocabulary item:

$$|w| = 0 \quad \underbrace{\qquad}_{\boldsymbol{x}}\underbrace{\qquad}_{\sigma} \quad \text{(easy)}$$

Proof 1 Part 2

Need to prove:

$$\hat{\delta}'(q_0', w) = [p_1, p_2] \iff \left(\hat{\delta}_1(q_0^1, w) = p_1\right) \wedge \left(\hat{\delta}_2(q_0^2, w) = p_2\right)$$

Proof by induction on length of w

Split $\boldsymbol{w}$: $\boldsymbol{w} = \boldsymbol{x}\sigma$, $\boldsymbol{x} \in \Sigma^*$ is a string and $\sigma \in \Sigma$ is a vocabulary item:

$$|w| = 0 \quad \underbrace{\hphantom{xx}}_{x}\underbrace{\hphantom{xx}}_{\sigma} \quad \text{(easy)}$$

$$|w| = 1 \quad \underbrace{\hphantom{xx}}_{x}\underbrace{a}_{\sigma} \quad n + 1 \text{ (still not bad)}$$

Proof 1 Part 2

Need to prove:

$$\hat{\delta}'(q_0', w) = [p_1, p_2] \iff \left(\hat{\delta}_1(q_0^1, w) = p_1\right) \wedge \left(\hat{\delta}_2(q_0^2, w) = p_2\right)$$

Proof by induction on length of w

Split $\boldsymbol{w}$: $\boldsymbol{w} = \boldsymbol{x}\sigma$, $\boldsymbol{x} \in \Sigma^*$ is a string and $\sigma \in \Sigma$ is a vocabulary item:

$$|w| = 0 \quad \underbrace{}_{x}\underbrace{}_{\sigma} \quad \text{(easy)}$$

$$|w| = 1 \quad \underbrace{}_{x}\underbrace{a}_{\sigma} \quad n+1 \text{ (still not bad)}$$

$$|w| = 2 \quad \underbrace{\sigma}_{x}\underbrace{a}_{\sigma} \quad n+1$$

Proof 1 Part 2

Need to prove:

$$\hat{\delta}'(q_0', w) = [p_1, p_2] \iff \left(\hat{\delta}_1(q_0^1, w) = p_1\right) \wedge \left(\hat{\delta}_2(q_0^2, w) = p_2\right)$$

Proof by induction on length of w

Split $\boldsymbol{w}$: $\boldsymbol{w} = \boldsymbol{x}\sigma$, $\boldsymbol{x} \in \Sigma^*$ is a string and $\sigma \in \Sigma$ is a vocabulary item:

$\lvert w \rvert = 0$	$\underbrace{\quad}_{x}\underbrace{\quad}_{\sigma}$	(easy)
$\lvert w \rvert = 1$	$\underbrace{\quad}_{x}\underbrace{a}_{\sigma}$	$n+1$ (still not bad)
$\lvert w \rvert = 2$	$\underbrace{\sigma}_{x}\underbrace{a}_{\sigma}$	$n+1$
$\lvert w \rvert = 3$	$\underbrace{aa}_{x}\underbrace{a}_{\sigma}$	$n+1$

Proof 1, Part 2

Defined: $\boldsymbol{w}$, $\hat{\delta}$, $(Q', \Sigma', \delta', q_0', F')$, $(Q_1, Q_2, \Sigma, \delta_1, \delta_2, q_0^1, q_0^2, F_1, F_2)$

Base:
$|\boldsymbol{w}| = 0 \iff \boldsymbol{w} = \varepsilon$

$$\hat{\delta}'(q_0', \varepsilon) = [q_0^1, q_0^2] \iff \left(\hat{\delta}_1(q_0^1, \varepsilon) = q_0^1\right) \wedge \left(\hat{\delta}_2(q_0^2, \epsilon) = q_0^2\right) \quad (1)$$

by sub. ε and def. of δ', δ_1, δ_2

Proved for $|\boldsymbol{w}| = 0$

Proof 1 Part 2

Defined: $\boldsymbol{w}$, $\hat{\delta}$, $(Q', \Sigma', \delta', q_0', F')$, $(Q_1, Q_2, \Sigma, \delta_1, \delta_2, q_0^1, q_0^2, F_1, F_2)$

Induction: $|\boldsymbol{w}| = n + 1$

Proof 1 Part 2

Defined: $\boldsymbol{w}$, $\hat{\delta}$, $(Q', \Sigma', \delta', q_0', F')$, $(Q_1, Q_2, \Sigma, \delta_1, \delta_2, q_0^1, q_0^2, F_1, F_2)$

Induction: $|\boldsymbol{w}| = n + 1$

$$\hat{\delta}'(q_0', \boldsymbol{w}) = \hat{\delta}'(q_0', \boldsymbol{x}\sigma) \tag{2}$$

by definition of $\boldsymbol{w}$

Proof 1 Part 2

Defined: $\boldsymbol{w}$, $\hat{\delta}$, $(Q', \Sigma', \delta', q_0', F')$, $(Q_1, Q_2, \Sigma, \delta_1, \delta_2, q_0^1, q_0^2, F_1, F_2)$

Induction: $|\boldsymbol{w}| = n + 1$

$$\hat{\delta}'(q_0', \boldsymbol{w}) = \hat{\delta}'(q_0', \boldsymbol{x}\sigma) \qquad \text{For } n = 0,\ |\boldsymbol{w}| = 1 \text{ so } \boldsymbol{x} = \varepsilon \tag{2}$$

by definition of $\boldsymbol{w}$

Proof 1 Part 2

Defined: $\boldsymbol{w}$, $\hat{\delta}$, $(Q', \Sigma', \delta', q_0', F')$, $(Q_1, Q_2, \Sigma, \delta_1, \delta_2, q_0^1, q_0^2, F_1, F_2)$

Induction: $|\boldsymbol{w}| = n + 1$

$$\hat{\delta}'(q_0', \boldsymbol{w}) = \hat{\delta}'(q_0', \boldsymbol{x}\sigma) \quad \text{For } n = 0,\ |\boldsymbol{w}| = 1 \text{ so } \boldsymbol{x} = \varepsilon \tag{2}$$

by definition of $\boldsymbol{w}$

$$\hat{\delta}'(q_0', \boldsymbol{x}) = [p_1, p_2] \iff \left(\hat{\delta}_1(q_0^1, \boldsymbol{x}) = p_1\right) \wedge \left(\hat{\delta}_2(q_0^2, \boldsymbol{x}) = p_2\right) \tag{3}$$

by inductive hyp.

Proof 1 Part 2

Defined: $\boldsymbol{w}$, $\hat{\delta}$, $(Q', \Sigma', \delta', q_0', F')$, $(Q_1, Q_2, \Sigma, \delta_1, \delta_2, q_0^1, q_0^2, F_1, F_2)$

Induction: $|\boldsymbol{w}| = n + 1$

$$\hat{\delta}'(q_0', \boldsymbol{w}) = \hat{\delta}'(q_0', \boldsymbol{x}\sigma) \tag{2}$$

For $n = 0$, $|\boldsymbol{w}| = 1$ so $\boldsymbol{x} = \varepsilon$

by definition of $\boldsymbol{w}$

$$\hat{\delta}'(q_0', \boldsymbol{x}) = [p_1, p_2] \iff \left(\hat{\delta}_1(q_0^1, \boldsymbol{x}) = p_1\right) \wedge \left(\hat{\delta}_2(q_0^2, \boldsymbol{x}) = p_2\right) \tag{3}$$

by inductive hyp.

$$\delta'([p_1, p_2], \sigma) = [r_1, r_2] \iff (\delta_1(p_1, \sigma) = r_1) \wedge (\delta_2(p_2, \sigma) = r_2) \tag{4}$$

by definition of δ'

Proof 1 Part 2

Defined: $\boldsymbol{w}$, $\hat{\delta}$, $(Q', \Sigma', \delta', q_0', F')$, $(Q_1, Q_2, \Sigma, \delta_1, \delta_2, q_0^1, q_0^2, F_1, F_2)$

Induction: $|\boldsymbol{w}| = n + 1$

$$\hat{\delta}'(q_0', \boldsymbol{w}) = \hat{\delta}'(q_0', \boldsymbol{x}\sigma) \qquad \text{For } n = 0,\ |\boldsymbol{w}| = 1 \text{ so } \boldsymbol{x} = \varepsilon \tag{2}$$

by definition of $\boldsymbol{w}$

$$\hat{\delta}'(q_0', \boldsymbol{x}) = [p_1, p_2] \iff \left(\hat{\delta}_1(q_0^1, \boldsymbol{x}) = p_1\right) \wedge \left(\hat{\delta}_2(q_0^2, \boldsymbol{x}) = p_2\right) \tag{3}$$

by inductive hyp.

$$\delta'([p_1, p_2], \sigma) = [r_1, r_2] \iff (\delta_1(p_1, \sigma) = r_1) \wedge (\delta_2(p_2, \sigma) = r_2) \tag{4}$$

by definition of δ'

$$\delta'\left(\hat{\delta}'(q_0', \boldsymbol{x}), \sigma\right) = [r_1, r_2] \iff$$
$$\left(\delta_1\left(\hat{\delta}_1(q_0^1, \boldsymbol{x}), \sigma\right) = r_1\right) \wedge \left(\delta_2\left(\hat{\delta}_2(q_0^2, \boldsymbol{x}), \sigma\right) = r_2\right) \tag{5}$$

sub. (3) into (4)

Proof 1 Part 2

$$\delta'\left(\hat{\delta}'(q_0', \boldsymbol{x}), \sigma\right) = [r_1, r_2] \iff$$
$$\left(\delta_1\left(\hat{\delta}_1(q_0^1, \boldsymbol{x}), \sigma\right) = r_1\right) \wedge \left(\delta_2\left(\hat{\delta}_2(q_0^2, \boldsymbol{x}), \sigma\right) = r_2\right) \quad (5)$$

sub. (3) into (4)

Proof 1 Part 2

$$\delta'\left(\hat{\delta}'(q_0', \boldsymbol{x}), \sigma\right) = [r_1, r_2] \iff$$
$$\left(\delta_1\left(\hat{\delta}_1(q_0^1, \boldsymbol{x}), \sigma\right) = r_1\right) \wedge \left(\delta_2\left(\hat{\delta}_2(q_0^2, \boldsymbol{x}), \sigma\right) = r_2\right) \quad (5)$$

sub. (3) into (4)

$$\hat{\delta}'(q_0', \boldsymbol{x}\sigma) = [r_1, r_2] \iff \left(\hat{\delta}_1(q_0^1, \boldsymbol{x}\sigma) = r_1\right) \wedge \left(\hat{\delta}_2(q_0^2, \boldsymbol{x}\sigma) = r_2\right) \quad (6)$$

by definition of $\hat{\delta}$

Proof 1 Part 2

$$\delta'\left(\hat{\delta}'(q_0', \boldsymbol{x}), \sigma\right) = [r_1, r_2] \iff$$
$$\left(\delta_1\left(\hat{\delta}_1(q_0^1, \boldsymbol{x}), \sigma\right) = r_1\right) \wedge \left(\delta_2\left(\hat{\delta}_2(q_0^2, \boldsymbol{x}), \sigma\right) = r_2\right) \quad (5)$$

sub. (3) into (4)

$$\hat{\delta}'(q_0', \boldsymbol{x}\sigma) = [r_1, r_2] \iff \left(\hat{\delta}_1(q_0^1, \boldsymbol{x}\sigma) = r_1\right) \wedge \left(\hat{\delta}_2(q_0^2, \boldsymbol{x}\sigma) = r_2\right) \quad (6)$$

by definition of $\hat{\delta}$

$$\hat{\delta}'(q_0', \boldsymbol{w}) = [r_1, r_2] \iff \left(\hat{\delta}_1(q_0^1, \boldsymbol{w}) = r_1\right) \wedge \left(\hat{\delta}_2(q_0^2, \boldsymbol{w}) = r_2\right) \quad (6)$$

by definition of $\boldsymbol{w}$

Proved for $|\boldsymbol{w}| = n + 1$

FSA-DCFSA Equivalence

Task 2:

Given:

FSA $A_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1)$
FSA $A_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2)$
DCFSA $A = (Q_1, Q_2, \Sigma, \delta_1, \delta_2, q_0^1, q_0^2, F_1, F_2)$

Prove:

$L(A) = L(A_1) \ \cap \ L(A_2)$

FSA-DCFSA Equivalence

Task 2:

Given:

FSA $A_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1)$
FSA $A_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2)$
DCFSA $A = (Q_1, Q_2, \Sigma, \delta_1, \delta_2, q_0^1, q_0^2, F_1, F_2)$

Prove:

$L(A) = L(A_1) \ \cap \ L(A_2)$

Prove by dual containment

Proof 2 Direction 1

Direction 1: $L(A) \subseteq L(A_1) \cap L(A_2)$

Let $w \in L(A)$	define	(1)
$\hat{\delta}_1(q_0^1, w) \in F_1$ and $\hat{\delta}_2(q_0^2, w) \in F_2$	by definition of A	(2)
$w \in L(A_1)$	by definition of A_1	(3)
$w \in L(A_2)$	by definition of A_2	(4)
$w \in L(A_1) \cap L(A_2)$	by definition of intersection	(5)

Proof 2 Direction 2

Direction 2: $L(A_1) \cap L(A_2) \subseteq L(A)$

Let $w \in L(A_1) \cap L(A_2)$	define	(6)
$w \in L(A_1)$ and $w \in L(A_2)$	by definition of intersection	(7)
$\hat{\delta}_1(q_0^1, w) \in F_1$	by definition of A_1	(8)
$\hat{\delta}_2(q_0^2, w) \in F_2$	by definition of A_2	(9)
$w \in L(A)$	by definition of A	(10)

Proof 2 Conclusion

We have proven $L(A) \subseteq L(A_1) \cap L(A_2)$ and $L(A_1) \cap L(A_2) \subseteq L(A)$.

Thus $L(A) = L(A_1) \ \cap \ L(A_2)$ by dual containment.

Proof 2 Conclusion

We have proven $L(A) \subseteq L(A_1) \cap L(A_2)$ and $L(A_1) \cap L(A_2) \subseteq L(A)$.

Thus $L(A) = L(A_1) \ \cap \ L(A_2)$ by dual containment.

Important: explicitly write out what you just proved and why.
Don't leave out the last step!

Writing Formal Language Theory Proofs: Finite-State Automata Example

11-711: Algorithms for NLP

September 11, 2015