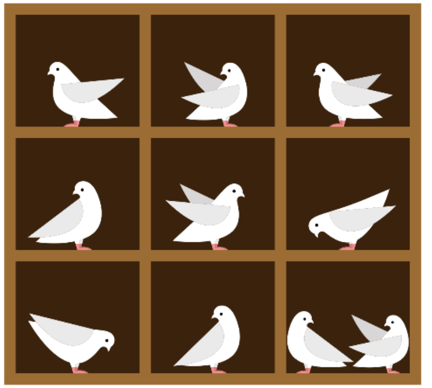
Lecture 4 (2R)

Logic 3



Today:

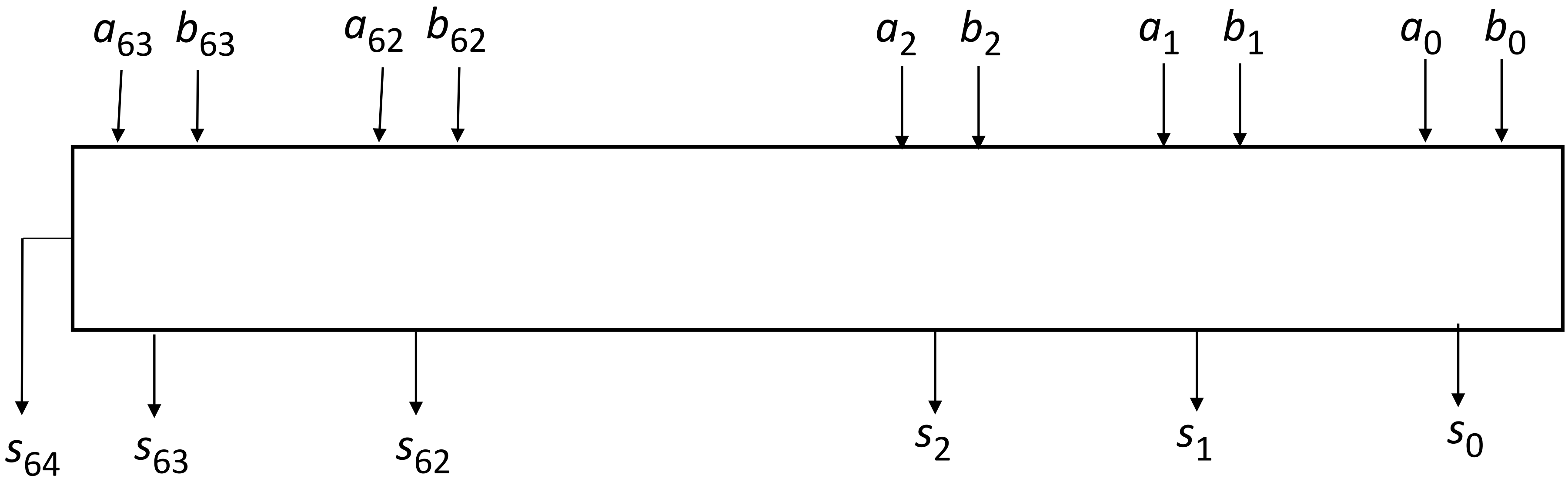
- ▢ Doing things with circuits
- ▢ Tautologies
- ▢ Formalizing proofs – an important theorem
- ▢ Adding in quantifiers – first-order logic
- ▢ The math/English gap
- ▢ Negating quantified formulas

Announcements:

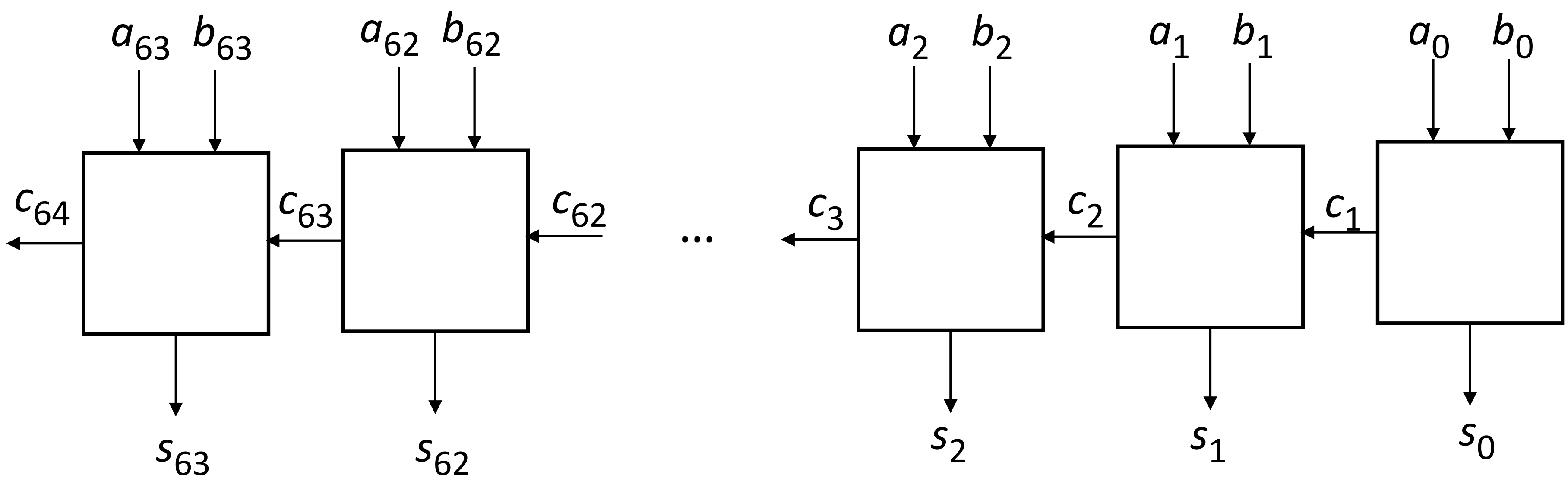
- Quiz 1. Easy. Tomorrow 7am to 7pm
- PS2 is up. More challenging than PS1

Phillip Rogaway

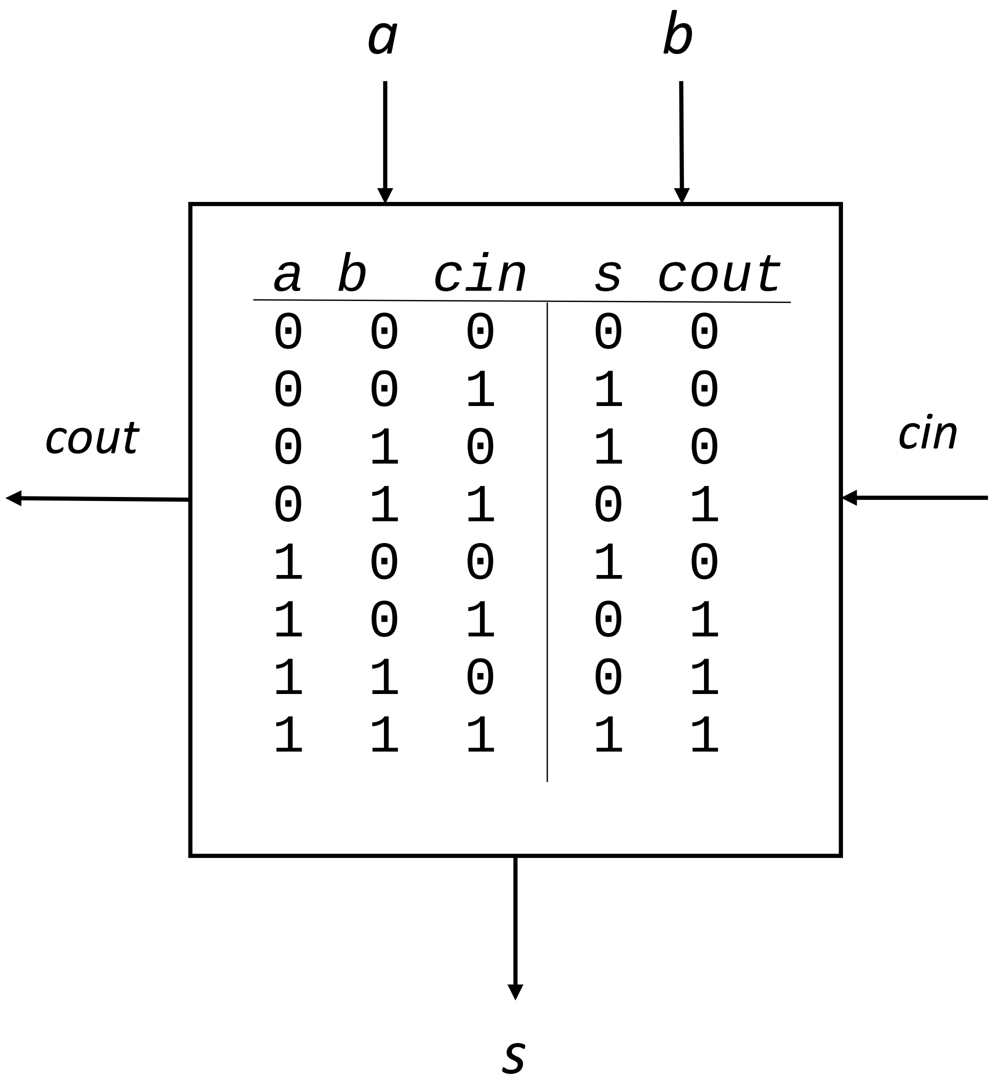
```
    1 0 0 1 1 0 1 0
+   1 0 0 1 1 0 1 1
-------------------
  1 0 0 1 1 0 1 0 1
```



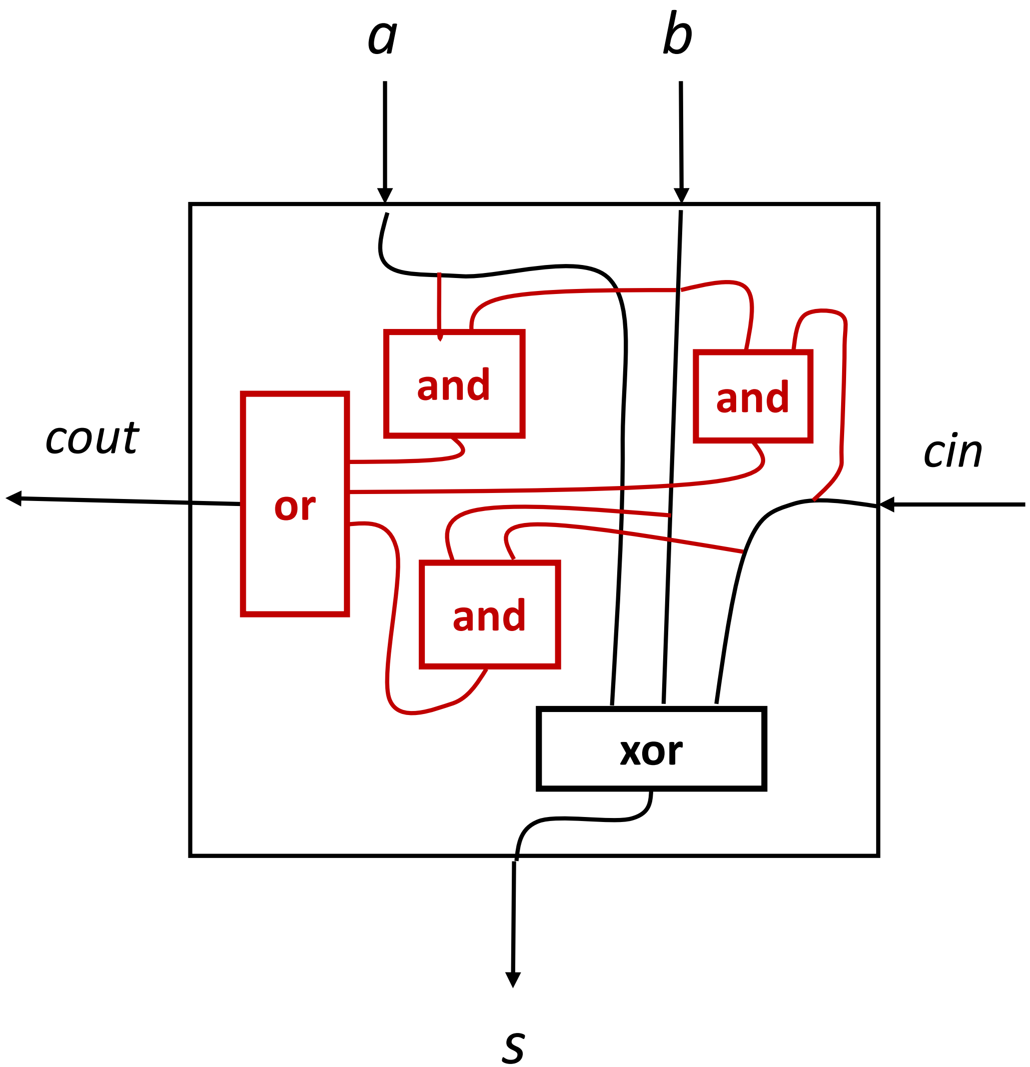
a63 b63 a62 b62 a2 b2 a1 b1 a0 b0
s64 s63 s62 s2 s1 s0



a63 b63 a62 b62 a2 b2 a1 b1 a0 b0
c64 c63 c62 ... c3 c2 c1
s63 s62 s2 s1 s0



a	b	cin	s	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1


a
b
cout
cin
and
and
or
and
xor
s

Tautologies

$$\neg (A \lor B) \equiv \neg A \land \neg B$$

Another way to say this:

$$\neg (A \lor B) \leftrightarrow \neg A \land \neg B$$

is always true

Or: the truth table for

$$\neg (A \lor B) \leftrightarrow \neg A \land \neg B$$

is always 1

Def: A boolean formula

$$\varphi(x_1, \ldots, x_n)$$

is a **tautology** if it is **always** true (no matter how you set its variables)

We write $\models \varphi$ to mean that φ is a tautology – it is always true.

Is there an algorithm to figure out if φ is a tautology?

How efficient is your algorithm?

Is there a more efficient algorithm?

What about proving that φ is true without building a truth table?

$$(PQ)(P \vee R \to S)(SQ \to U) \to U$$

1. PQ assumption
2. $P \vee R \to S$ assumption
3. $SQ \to U$ assumption
4. $PQ \to P$ AND-1 (eliminate conjunction)
5. P *modus ponens* on (1), (4)
6. $PQ \to Q$ AND-2 (eliminate conjunction)
7. Q *modus ponens* on (1), (6)
8. $P \to P \vee R$ OR-1 (introduce disjunction)
9. $P \vee R$ *modus ponens* on (5), (8)
10. S *modus ponens* on (2) and (9)
11. $S \to (Q \to SQ)$ AND-3 (introduce conjunction)
12. $Q \to SQ$ *modus ponens* on (10), (11)
13. SQ *modus ponens* on (7), (12)
14. U *modus ponens* on (3) and (13)

Therefore

$\{PQ,\ P \vee R \to S,\ SQ \to U\} \vdash U$ or

$\vdash (PQ)(P \vee R \to S)(SQ \to U) \to U$ //The given statement is provable

Axioms

Name	Axiom Schema	Description
THEN-1	$\phi \to (\chi \to \phi)$	Add hypothesis χ, implication introduction
THEN-2	$(\phi \to (\chi \to \psi)) \to ((\phi \to \chi) \to (\phi \to \psi))$	Distribute hypothesis ϕ over implication
AND-1	$\phi \land \chi \to \phi$	Eliminate conjunction
AND-2	$\phi \land \chi \to \chi$	
AND-3	$\phi \to (\chi \to (\phi \land \chi))$	Introduce conjunction
OR-1	$\phi \to \phi \lor \chi$	Introduce disjunction
OR-2	$\chi \to \phi \lor \chi$	
OR-3	$(\phi \to \psi) \to ((\chi \to \psi) \to (\phi \lor \chi \to \psi))$	Eliminate disjunction
NOT-1	$(\phi \to \chi) \to ((\phi \to \neg\chi) \to \neg\phi)$	Introduce negation
NOT-2	$\phi \to (\neg\phi \to \chi)$	Eliminate negation
NOT-3	$\phi \lor \neg\phi$	Excluded middle, classical logic
IFF-1	$(\phi \leftrightarrow \chi) \to (\phi \to \chi)$	Eliminate equivalence
IFF-2	$(\phi \leftrightarrow \chi) \to (\chi \to \phi)$	
IFF-3	$(\phi \to \chi) \to ((\chi \to \phi) \to (\phi \leftrightarrow \chi))$	Introduce equivalence

$\vdash \phi$ can prove ϕ

$\vDash \phi$ ϕ is always true — it's a tautology

Theorem. Fix a proof system like the one sketched. Then:

Soundness: If $\vdash \phi$ then $\vDash \phi$

Completeness: If $\vDash \phi$ then $\vdash \phi$

$\vdash \phi$ can prove ϕ

$\models \phi$ ϕ is always true — it's a tautology

Theorem. Fix a proof system like the one sketched. Then:

Soundness: If $\Gamma \vdash \phi$ then $\Gamma \models \phi$

Completeness: If $\Gamma \models \phi$ then $\Gamma \vdash \phi$

Leon Henkin (1921-2006)

Invented the standard proof this result, and taught me it as an undergraduate.

Coming clean – the result isn't just about sentential logic, but first-order logic. You get to have ∀ and ∃ quantifiers, and a proof system with rules for manipulating them.

Adding in quantifiers

"All apples are bad"

$$(\forall x)\ (A(x) \rightarrow B(x))$$

"Some apples are bad"

$$(\exists x)\ (A(x) \wedge B(x))$$

"BILLY has beat up every boy at Caesar-Chavez elementary school"

$$(\forall x)\ ((\text{CCstudent}(x) \wedge \text{Boy}(x) \wedge (x \neq \text{BILLY}) \rightarrow \text{HasBeatenUp}\ (\text{BILLY}, x))$$

All lions are fierce $(\forall x)\ (L(x) \rightarrow F(x))$

Some lions do not drink coffee $(\exists\ x)\ (L(x) \wedge \neg\ C(x))$

Some fierce creatures do not drink coffee $(\exists x)\ (F(x) \wedge \neg\ C(x))$

"Nobody likes a sore loser"

Universe of discourse = human beings

L(x, y) - predicate - true iff person x likes y

S(x) - person x is a sore loser

$$(\forall x)\ (S(x) \rightarrow (\forall y)\ (\neg\ L(y, x)))$$

(apparently, a sore loser doesn't like even himself)

Vocabulary of first-order logic consists of:

- Parenthesis and logical connectives: ( ) $\neg \wedge \vee \rightarrow \leftrightarrow$
- Equality symbol: = *usually included*
- $\forall$, $\exists$ *universal and existential quantifiers*
- Variables: $x_1, x_2, \ldots$ *name points in the universe U*
- Constant symbols 0, 1, BILLY *name a point in the universe U*
- Function symbols f, +, … *map a tuple of points in U to a point in U*
- Predicate symbols P, <, PRIME, … *functions from universe U to $\mathbb{B}$*

The English/Logic Gap

CSE majors take ECS120 **or** ECS122A

Biology majors take MAT17A **or** MAT21A

Hosts ask if you want BEEF **or** FISH **or** VEG

Either Facebook lives **or** democracy dies.

Either Facebook lives **or** the square root of 10 exceeds 3.

If the forecast calls for rain **then** I will grab my umbrella.

$$R \rightarrow U$$

If the forecast calls for rain **then** one plus one is two

If one plus one is three **then** I will grab my umbrella.

English language propositions (= simple declarative statements about the state of the world) routinely **don't** have clear T/F values.

"It is raining outside"   Either true/false?
Not really!

English: Words have constructed, contextualized meaning.

Math: Same!

A particular fragment math: A few words and symbols are given a precise meaning.

Negating quantified boolean formulas

$$\neg (A \vee B) \equiv \neg A \wedge \neg B$$

$$\neg (A \wedge B) \equiv \neg A \vee \neg B$$

$$\neg (\forall x)\, \phi \equiv (\exists x)\, (\neg \phi)$$

$$\neg (\exists x)\, \phi \equiv (\forall x)\, (\neg \phi)$$