CS 181U
Applied Logic & Automated Theorem Proving

Lecture 2

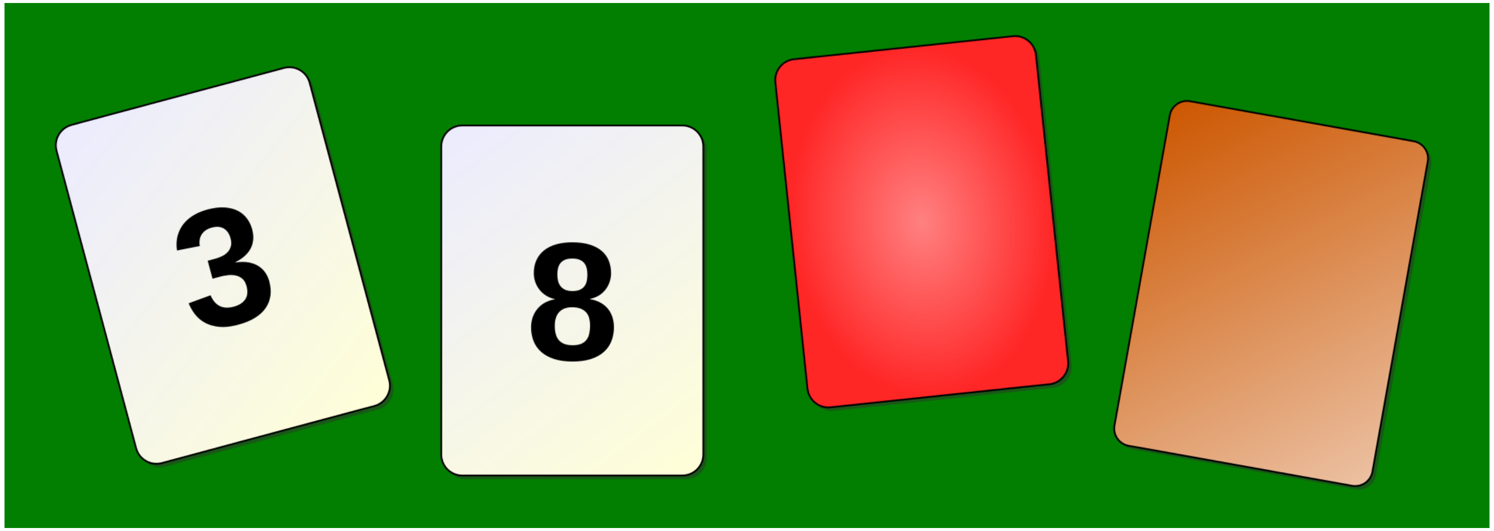
Human Reasoning,
Logic and Language,
Boolean Logic,
Object-Oriented Structural Recursion

Announcement: office hours
Still converging. Informal Poll.

Announcement: grutoring
no grutors .

Announcement: HW01
90% ready.
Will release today and email, or just check website.
Due Tues Jan 31

Human Reasoning,
Logic & Language



Each card has a number on one side, and a patch of color on the other.

Rule: if a card shows an even number on one face, then its opposite face is red.

Which card or cards must be turned over to test the rule?

Each card has an age on one side, and a beverage on the other.

Rule: if a person is under 21, then they must be drinking a non-alcoholic beverage.

Which card or cards must be turned over to test the rule?

The Wason Selection Task

Wason, P. C. (1968). "Reasoning about a rule". Quarterly Journal of Experimental Psychology. Volume 20 (3): pages 273-281.

Main takeaways:

90% of people get the abstract version (with numbers and colors) wrong.

Only 25% get it wrong with a real-world (age vs. drinking) example.

Ongoing debates about logical reasoning vs. enforcement of social contracts.

Conditionals

Remember $A \rightarrow B$?

"A implies B"

"If A, then B"

A	B	$A \rightarrow B$
F	F	
F	T	
T	F	
T	T	

Refresh your memory: fill this in

Biscuit Conditionals

Your friend says the following:

"If you are hungry, then there are biscuits on the table."

Let $H =$ you are hungry
Let $B =$ there are biscuits on table

$$H \to B$$

What does this mean if you interpret this statement in formal logic compared to everyday language?

Biscuit Conditionals

Austin, J.L. 1961. Ifs and cans. In Philosophical Papers, ed. J.O. Urmson and G.J. Warnock, pages 153-180. Oxford: Oxford University Press.

Elder, Chi-Hé. "Biscuit Conditionals, Conditional Speech Acts and Speech-Act Conditionals." (2019).

Main takeaways:

Natural language often does not match up with formal logic interpretations.

People have been arguing about this for over 50 years.

Meta Takeaways

Turing a formal logic statement into a natural language statement can help understanding.

Turing a formal logic statement into a natural language statement can hurt understanding.

Be careful about mixing natural langauge and logic!

Boolean Logic

Syntax, Well-Formed Formulas

Boolean logic formulas defined recursively:

Base Cases:

T is a formula

F is a formula

v is a formula, where $v \in V$, and V is a variable set

Recursive Cases:

If α and β are well-formed formulas, then

$\neg\alpha$ is a formula

$(\alpha \wedge \beta)$ is a formula

$(\alpha \vee \beta)$ is a formula

$(\alpha \to \beta)$ is a formula

$(\alpha \leftrightarrow \beta)$ is a formula

WFF 'N Proof (1970)

Atoms and Literals

Atoms: the smalled building blocks of our logical universe

T is an atom

F is an atom

v is an atom, for any $v \in V$

nothing else is an atom

Literals: if a is any atom then

a is a literal

$\neg a$ is a literal

Is $\neg\neg a$ is a literal?

This is the only way to make literals. Said another way, a literal is either an atom or the negation of an atom.

Interpretations

Given a set of variables, V, an interpretaion for V is a total mapping, $I : V \to \{T, F\}$.

I.e. an interpretation I assigns each variable to be either true or false.

In python, we will use dictionaries for interpretations:

Example: `interp = { A : T, B : F, C : F }`

Semantics for $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

For all possible interpretations of the variables in a formulas that uses a single connective, we need to define the resulting interpretation.

A	$\neg A$
F	T
T	F

A	B	$A \wedge B$
F	F	F
F	T	F
T	F	F
T	T	T

A	B	$A \vee B$
F	F	F
F	T	T
T	F	T
T	T	T

A	B	$A \rightarrow B$
F	F	T
F	T	T
T	F	F
T	T	T

A	B	$A \leftrightarrow B$
F	F	T
F	T	F
T	F	F
T	T	T

Satisfiability

We say that an interpretation I satisfies, or models, a logical formula ϕ if ϕ evalautes to T when raplacing all variables in ϕ with their corresponding interpretation from I. We write:

$$I \models \phi$$

A formula ϕ is satisfiable if there exists an interpretation I such that $I \models \phi$.

Is $(A \vee \neg B) \wedge C$ satisfiable?

Is $(A \wedge B) \wedge (\neg A \vee \neg B)$ satisfiable?

Tautology, Validity

We say that a formula ϕ is a tautology, if it evalautes to T under all interpretations. We can also say that ϕ is a valid formula.

$$((A \wedge B) \rightarrow C) \leftrightarrow (A \rightarrow (B \rightarrow C)).$$

A	B	C	$A \wedge B$	$(A \wedge B) \rightarrow C$	$B \rightarrow C$	$A \rightarrow (B \rightarrow C)$	$((A \wedge B) \rightarrow C) \Leftrightarrow (A \rightarrow (B \rightarrow C))$
T	T	T	T	T	T	T	T
T	T	F	T	F	F	F	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	F	T	T	T	T
F	T	F	F	T	F	T	T
F	F	T	F	T	T	T	T
F	F	F	F	T	T	T	T

Unsatisfiable Formulas

A formula ϕ is said to be unsatisfiable if there does not exist any interpretation I such that $I \models \phi$.

Example: $(A \wedge B) \wedge (\neg A \vee \neg B)$

Adequate Sets of Connectives

A set of connectives is adequate if every truth function can be expressed using only those connectives.

$\{\wedge, \vee, \neg\}$ is an adequate set of connectives.

$\{\wedge, \neg\}$ is an adequate set of connectives.

$\{\vee, \neg\}$ is an adequate set of connectives.

Less Common (but useful!) operators

Boolean logic formulas defined recursively:

Base Cases:

T is a formula

F is a formula

v is a formula, where $v \in V$, and V is a variable set

Recursive Cases:

If α and β are well-formed formulas, then

$\neg\alpha$ is a formula

$(\alpha \land \beta)$ is a formula

$(\alpha \lor \beta)$ is a formula

$(\alpha \to \beta)$ is a formula

$(\alpha \leftrightarrow \beta)$ is a formula

same slide as before, with new stuff

$(\alpha \bar{\land} \beta)$ is a formula (NAND)

$(\alpha \bar{\lor} \beta)$ is a formula (NOR)

$(\alpha \oplus \beta)$ is a formula (XOR)

Semantics of $\bar{\wedge}, \bar{\vee}, \oplus$

A	B	$A \bar{\wedge} B$
F	F	T
F	T	T
T	F	T
T	T	F

NAND

A	B	$A \bar{\vee} B$
F	F	T
F	T	F
T	F	F
T	T	F

NOR

A	B	$A \oplus B$
F	F	F
F	T	T
T	F	T
T	T	F

XOR

Observe: $\bar{\wedge}$ is adequate to rewrite any Boolean formula:

$$\neg A \equiv (A \bar{\wedge} A)$$
$$A \wedge B \equiv (A \bar{\wedge} B) \bar{\wedge} (A \bar{\wedge} B)$$

This is enough

$A \vee B \equiv ?????$ But, try this out for fun

Negation Normal Form

A formula is in negation normal form if all occurrences of $\neg$ occur on atoms and the only other logical connectives are $\wedge$ and $\vee$.

$A \wedge B$ is in NNF

$\neg C$ is in NNF

$(A \wedge B) \vee (X \rightarrow \neg Y)$ is in not NNF

$\neg(A \wedge B)$ is not in NNF

$\neg\neg A$ is not in NNF

Object-Oriented Structural Recursion in Python for HW