Streets of Philadelphia

Can you avoid a ticket?

1. Parking your car to the right of the pole at 2 PM on Friday
2. Waiting for your friend at the right of the pole at 5 PM on Saturday

System Description

The system is in multiuser state if and only if it is operating normally. If the system is operating normally, the kernel is functioning. The kernel is not functioning or the system is in interrupt mode. If the system is not in multiuser state, then it is in interrupt mode. The system is not in interrupt mode.

Would the above text *logically* make sense? Why?

Two Main Types of Logic

- Propositional logic [today]
  - Can express true/false statements and their combinations E.g., The system is in multiuser state.
  - Basic component of first-order logic
- First-order logic (FOL) [next lecture]
  - Can express quantifiers: "*every*", "*some*"
  - Can specify structures

many more types of logic ...

B5: Propositional Logic

Today

- Understand how to use propositional logic
  - Symbols/Syntax
  - Semantics
  - Proofs
- Take-home exercises
  - System description, Street sign, Analysis of "if"

Can be easy or challenging, depending your previous experience

Section 1

Building Blocks

- Proposition: A statement that is either true or false (i.e., truth value)
  - E.g., "it is raining", "there is no class"
- Symbols non-propositions?
  - Propositional variables: ***p***, ***q***, ***r***, ...
  - Connectives: Unary: Ø, Binary: Ù, Ú, ®, «
- Syntax: how to combine these symbols
  - E.g., **Ø(*p* Ù *q*)**, **Ø*p* Ú Ø*q***, **(*p* ® *q*) ® (*r* ® (*q* Ú *r*))**

Well-Formed Formula (wff)

- Legal statements in propositional logic
- Primitive wff's syntax: good/bad
  - Propositional variables
- Complex wff's
  - Adding Ø in front of a wff j, i.e., (Øj)
  - Combining two wff's j and ψ, using binary connectives: (j Ù y), (j Ú y), (j ® y), (j « y)
  - Repeated application of these processes

( ) may be omitted when no confusion arises.

Exercise

Identify wff's and non-wff's

A. ***p***
B. Ø(Ø***p***)
C. ØØ ***p***
D. Ø***p*** Ø***q***
E. ***p*** ØÚ ***q***
F. Ø***p*** Ú ***q***
G. (***p*** ® ***q***) ® ((***r*** Ú ***p***) ® (***q*** Ú ***r***))
H. ( )

- Primitive wff's
  - Propositional variables
- Complex wff's
  - Adding $\neg$ in front of a wff φ, i.e., $(\neg\varphi)$
  - Combining two wff's φ and ψ, using binary connectives: $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$, $(\varphi \leftrightarrow \psi)$
  - Repeated application of these processes

Meaning?
What do we need?

Section Summary

- Propositional variables represent true/false statements.
- The syntax of propositional logic is a specific way of combining propositional variables using connectives.

no meaning yet

Section 2

Semantics

- Assignment (of a propositional variable): Function that maps the set of variables to {**T**, **F**}, **T** for 'true' and **F** for 'false'
- Interpretation: The semantics of the logic proper, i.e., meaning of logical connectives

Truth Table

- Tabular representation of truth value computation

p	***q***	Ø***p***	***p*** Ù ***q***	***p*** Ú ***q***	***p*** ® ***q***	***p*** « ***q***
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

Connective Names

- Ø***p***: Negation (not)
- ***p*** Ù ***q***: Conjunction (and)
- ***p*** Ú ***q***: Disjunction (or)
- ***p*** ® ***q***: Conditional (if)
- ***p*** « ***q***: Biconditional (if and only if or iff)

Truth table vs. Meaning in English

Special Cases

- Tautology: A wff that is true regardless of the truth assignment to variables.
  - E.g., ***p*** Ú Ø***p***
- Contradiction: A wff that is false regardless of the truth assignment to variables.
  - E.g., ***p*** Ù Ø***p***

Group Exercise

A. Evaluate the truth value of the following wff when ***p***, ***q***, ***r*** are all **F**:

(Ø*p* ® *q*) Ù (Ø*p* Ú *r*)

B. Determine whether the following wff's are tautology (always true), contradiction (always false), or neither:

B1: **(Ø(*p* Ú *q*)) « (Ø*p* Ù Ø*q*)**

B2: **(Ø*p* Ú *q*) Ú (*p* Ú Ø*q*)**

Use as reference

Analysis Techniques (1)

- Identify special conditions on subformulas
  - j Ù y: To be **T**, both j and y must be **T**
  - j Ú y: To be **F**, both j and y must be **F**
  - j ® y: To be **F**, j must be **T** and y must be **F**
  - j « y: To be **T**, j and y must agree
- Using a new formula without affecting value
  - If j is **T**, j Ú y and y Ú j must be **T** for *any* y

Use as reference

Analysis Techniques (2)

- Use of equivalence
  - ØØj is equivalent to j — double negation
  - j Ú y ⇔ y Ú j — commutativity
  - j Ù y ⇔ y Ù j
  - Ø(j Ú y) ⇔ Øj Ù Øy — 'De Morgan'
  - Ø(j Ù y) ⇔ Øj Ú Øy
  - Øj Ú y ⇔ j ® y — '→' shorthand
  - j « y ⇔ (j ® y) Ù (y ® j) — '↔' shorthand
  - j ® y ⇔ Øy ® Øj — 'contrapositive'

System Description, Analyzed

Logically make sense ⇔ Consistent

System description: The system is in multiuser state (*m*) if and only if it is operating normally (*o*). If the system is operating normally (*o*), the kernel is functioning (*f*). The kernel is not functioning (¬*f*) or the system is in interrupt mode (*i*). If the system is not in multiuser state (¬*m*), then it is in interrupt mode (*i*). The system is not in interrupt mode (¬*i*).

Set of Statements

- Consistent: *Some* truth value assignment makes *all* statements true.
- *Not* consistent

⇔ Inconsistent

⇔ contradictory

⇔ *No* truth value assignment makes *all* statements true.

System Description, Formally

1. ***m*** « ***o***
2. ***o*** ® ***f***
3. Ø***f*** Ú ***i***
4. Ø***m*** ® ***i***
5. Ø***i***

System description:

1. The system is in multiuser state (*m*) if and only if it is operating normally (*o*).
2. If the system is operating normally (*o*), the kernel is functioning (*f*).
3. The kernel is not functioning (¬*f*) or the system is in interrupt mode (*i*).
4. If the system is not in multiuser state (¬*m*), then it is in interrupt mode (*i*).
5. The system is not in interrupt mode (¬*i*).

Easiest way of analyzing consistency?

Section Summary

- The semantics of propositional logic is based on interpretation and assignment.
- In general, the brute-force approach of trying all possible truth value combinations is too tedious.
  - Try to find easier (still complete) analysis by using cases and equivalences

Section 3

Proofs

- In some cases, semantic analysis can be tedious.
- Proof is a mechanical procedure that is used to derive a conclusion (theorem) from a hypothesis.
- Notes
  - Hypothesis: a set of logical statements
  - Proof steps must consist of individually-justified steps.

Overview: Proof Structure

Hypothesis: ***o*** ® ***f***, ***o***

Proof of ***f*** from the above hypothesis

1. ***o*** ® ***f*** [hyp]
2. ***o*** [hyp]
3. ***f*** [MP: 1, 2]

proof

conclusion/theorem

justification for each step

line numbering

- If the system is operating normally (*o*), the kernel is functioning (*f*).
- The system is operating normally (*o*).

Justification for Each Step

- Use of the hypothesis
- Modus Ponens (MP) as a Rule of inference:
  - If we have ***X*** ® ***Y*** and ***X***,
  - Conclude ***Y***

  - If the system is operating normally (*o*), the kernel is functioning (*f*).
  - The system is operating normally (*o*).
- Use of justified equivalence
  - Use of '∧'/'∨', commutativity, double negation, De Morgan, '→' shorthand, contrapositive, etc.

  See reference slide
- Use of already proven statements

Example: Direct Proof

Hypothesis: ***m*** « ***o***, Ø***m*** ® ***i***, Ø***i***

Proof of ***o*** from the above hypothesis

1. Ø***m*** ® ***i*** [hyp]
2. Ø***i*** ® ***m*** [contrapositive/double negation: 1]
3. Ø***i*** [hyp]
4. ***m*** [MP: 2, 3]
5. ***m*** « ***o*** [hyp]
6. **(*m* ® *o*) Ù (*o* ® *m*)** ['↔' shorthand: 5]
7. ***m*** ® ***o*** [use of '∧': 6]
8. ***o*** [MP: 7, 4]

Example: Proof by Cases

Hypothesis: none

Proof of **(Ø*p* Ú *q*) Ú (*p* Ú Ø*q*)** [from empty hypothesis]

1. Case 1: **Ø*p* Ú *q*** [hyp]
2. | **(Ø*p* Ú *q*) Ú (*p* Ú Ø*q*)** [use of '∨': 1]
3. Case 2: **Ø(Ø*p* Ú *q*)** [hyp]
4. | ***p* Ù Ø *q*** [De Morgan/double neg: 2]
5. | ***p*** [use of '∧': 4]
6. | ***p* Ú Ø *q*** [use of '∨': 5]
7. | **(Ø*p* Ú *q*) Ú (*p* Ú Ø *q*)** [use of '∨': 6]
8. **(Ø*p* Ú *q*) Ú (*p* Ú Ø*q*)** [exhaustive cases: 1-2, 3-7]

System Description, Revisited

Hypothesis: ***m*** « ***o***, ***o*** ® ***f***, Ø***f*** Ú ***i***, Ø***m*** ® ***i***, Ø***i***

Proof of **F** from the above hypothesis

1. ***o*** [Theorem (a few slides ago)]
2. ___________ [hyp]
3. ***f*** [MP: 1, 2]
4. Ø***f*** Ú ***i*** [hyp]
5. ___________ [____________: 4]
6. ***i*** [_________: 3, 5]
7. __________ [____________]
8. **F** [contradiction: 6,7]

Example: Proof by Contradiction

Hypothesis: ***m*** « ***o***, Ø***m*** ® ***i***, Ø***i***

Proof of ***m*** from the above hypothesis

1. Ø***m*** ® ***i*** [hyp]
2. | Ø***m*** [hyp to be rejected]
3. | ***i*** [MP: 1, 2]
4. | Ø***i*** [hyp]
5. | **F** [contradiction: 3, 5]
6. ***m*** [proof by contradiction: 2-5]

Proof Methods Summary

- Direct proof: Straightforward proof of a conclusion from hypotheses (if any)
  - Proof by cases: Prove exclusive and exhaustive cases separately
- Indirect proof: Uses a proof of a proposition different from the conclusion
  - Proof by contradiction: Assume the negation of the conclusion and derive a contradiction
  - Proof by contrapositive: To prove $p \rightarrow q$, prove $\neg q \rightarrow \neg p$ [application of a tautology: $(p \rightarrow q) \Leftrightarrow (\neg q \rightarrow \neg p)$]

Special Theorems

- Corollary: A theorem that can be derived from another theorem 'easily'
- Lemma: A preliminary theorem that is used to prove a 'main' theorem

Proofs in Practice

- Guidelines
  - Find an appropriate proof method, e.g., proof by cases, proof by contradiction.
  - Structure proofs modularly. I.e., prove lemmas to shorten the main proof.
  - Label every step.
  - Write justification for each step. I.e., refer to hypothesis, rules of inference, lemmas, theorems, and derived steps used in the justification.

Example: Informal Proof

Following the formal pattern
Not limited to propositional logic

Proof of: $A \subseteq B$ iff $A \cap B = B$

1. '$\Rightarrow$': if $A \subseteq B$, then $A \cap B = B$
2. | For every $x \in A$, $x \in B$ [def: '$\subseteq$']
3. | $A \cap B = \{x \mid x \in A$ and $x \in B\}$ [def: '$\cap$']
4. | $\{x \mid x \in A$ and $x \in B\} = \{x \mid x \in B$ and $x \in B\}$ [2, 3]
5. | $B = \{x \mid x \in B\}$ [def: set]
6. | $A \cap B = B$ [set identity: 4, 5]
7. '$\Leftarrow$': if $A \cap B = B$, then $A \subseteq B$
8. | analogous
9. $A \subseteq B$ iff $A \cap B = B$ [both directions of 'iff': 1-6, 7-8]

Example: Non-Proof

Statement: “God exists”

Proof attempt

1. The word ‘God’ exists. [any dictionary]
2. | Suppose that God does not exists. [hyp]
3. | Then there must not be the word ‘God’. [2]
4. | A contradiction. [1, 3]
5. Therefore, God exists.[proof by contradiction: 2-4]

Section Summary

- In propositional logic, we can prove any true theorem.
  - This is based on the truth-preserving nature of the proof procedure.
- Depending on the problem, choose either semantic analysis (Section 2) or proof (Section 3)
  - Occasionally, one is either than the other.

Summary Exercise

- Depending on your previous experience with logic, this unit may have been difficult. Describe your situation.
- Questions/Comments/Suggestions