Propositional logic

Artificial intelligence
Kristóf Karacs
PPKE-ITK

Outline

- Logics of different order: 0, 1, 2, higher
- Basic concepts and nomenclature
  - Syntax vs. semantics
  - Entailment
- Propositional logic
- Entailment and proof methods
  - Truth table, equivalence, resolution

Logics of different order

- Propositional logic (a. k. a. Boolean logic)
  - Only constant Boolean statements
- First order predicate logic (FOPL)
  - Introduces variables, predicates, functions, and quantifiers
- Higher order logics
  - Quantifiers can also be applied to predicates and functions
  - Meta level reasoning

Logic

- A formal language in which knowledge can be expressed
- In problem solving we enumerate states
- Logic provides a means of describing set of states and carrying out reasoning
  - “Peter is hungry”: refers to all world states in which Peter is hungry regardless of other things influencing the state

Basic concepts

- Syntax: specifies what expressions are legal
  - Well-formed sentences
- Semantics: meaning of sentences
  - Interpretation: assigns meaning to logic symbols
  - Semantics define the truth of sentences w. r. t. all possible interpretations
  - An interpretation *i* is a model of a set of sentences iff each of the sentences is true in interpretation *i*
- Logical inference: entailment
  - A set of sentences KB entails φ ($KB \vDash \varphi$) iff every model of KB is also a model of φ
  - Sentence φ logically follows from KB

Syntax of propositional logic

- Atomic sentences: Propositions
  - Symbols: P, Q, R, … (uppercase letters)
  - Special cases: T (true) and F (false)
- Complex sentences
  - Brackets
  - Connectives in order of precedence (high to low)
    - not ($\neg$), and ($\wedge$), or ($\vee$), implies ($\rightarrow$), equivalent ($\leftrightarrow$)
  - If φ and ψ are sentences, then

    $$(\varphi),\ \neg\varphi,\ \varphi \wedge \psi,\ \varphi \vee \psi,\ \varphi \rightarrow \psi \text{ and } \varphi \leftrightarrow \psi$$

    are also sentences

Semantics

- Meaning of a sentence is a truth value
  - {***T*, *F***}
- An interpretation is an assignment of truth values to the propositional variables
  - $\models_i \varphi$ Sentence φ is ***T*** in interpretation i
  - $\not\models_i \varphi$ Sentence φ is ***F*** in interpretation i

Semantic rules

- $\models_i$ ***T*** for all i
- $\not\models_i$ ***F*** for all i
- $\models_i \neg\varphi$ iff $\not\models_i \varphi$
- $\models_i \varphi \wedge \psi$ iff $\models_i \varphi$ and $\models_i \psi$ (conjunction)
- $\models_i \varphi \vee \psi$ iff $\models_i \varphi$ or $\models_i \psi$ (disjunction)
- $\models_i$ P iff i(P) = ***T***

Properties of sentences

- Equivalence $\varphi \equiv \psi$
  - φ and ψ are true for the same models
- Validity $\vDash \varphi$
  - A sentence is valid iff its truth value is ***T*** in all interpretations
  - Valid sentences are called tautologies
  - Examples: ***T***, $P \lor \neg P$, $A \rightarrow A$
- Satisfiability
  - A sentence is satisfiable iff it has at least one model

Entailment theorem

$$KB \vDash \varphi \quad \text{iff} \quad \vDash (KB \rightarrow \varphi)$$

- Enables proving **entailment** if we have means to prove the **validity** of a sentence
- This theorem is valid for all logics

Proving validity

- Truth table
- Equivalence rules
- Resolution

- $(X \rightarrow (Y \wedge Z)) \leftrightarrow ((X \rightarrow Y) \wedge (X \rightarrow Z))$

Proving by truth table

Proving by truth table

X	Y	Z	Y∧Z	X→Y	X → Z	X →(Y∧Z)	((X→Y)∧(X→Z))	S

Proving by truth table

X	Y	Z	Y∧Z	X→Y	X → Z	X →(Y∧Z)	((X→Y)∧(X→Z))	S
T	T	T	T	T	T	T	T	T
T	T	F	F	T	F	F	F	T
T	F	T	F	F	T	F	F	T
T	F	F	F	F	F	F	F	T
F	T	T	T	T	T	T	T	T
F	T	F	F	T	T	T	T	T
F	F	T	F	T	T	T	T	T
F	F	F	F	T	T	T	T	T

Equivalence (re-write) rules

- Logical equivalence
  - Different syntax
  - Same semantics
- Usage
  - Proving via showing equivalence
  - Modifying to a particular syntax to allow the use of other techniques (e.g. resolution)

Commutativity and associativity of connectives

- Commutativity:
  - $P \wedge Q$ can be replaced by $Q \wedge P$ (& vice-versa)
  - $P \vee Q$ can be replaced by $Q \vee P$ (& vice-versa)
  - $P \leftrightarrow Q$ can be replaced by $Q \leftrightarrow P$ (& vice-versa)
- Associativity
  - $((P \wedge Q) \wedge R)$ can be replaced by $(P \wedge (Q \wedge R))$ (& vice-versa)
  - $((P \vee Q) \vee R)$ can be replaced by $(P \vee (Q \vee R))$ (& vice-versa)

Distributivity of connectives

- And over or, or over and:
  - $(P \wedge (Q \vee R))$ can be replaced by $((P \wedge Q) \vee (P \wedge R))$
  - $(P \vee (Q \wedge R))$ can be replaced by $((P \vee Q) \wedge (P \vee R))$

- Over the implies sign
  - $(P \rightarrow (Q \vee R))$ can be replaced by $((P \rightarrow Q) \vee (P \rightarrow R))$
  - $(P \rightarrow (Q \wedge R))$ can be replaced by $((P \rightarrow Q) \wedge (P \rightarrow R))$

Double negation

- Double negations can be removed
  - $\neg\neg P$ is equivalent to P

- Caution when translating from natural language

de Morgan's laws and contraposition

- de Morgan's laws
  - $\neg(P \wedge Q)$ is equivalent to $(\neg P \vee \neg Q)$
  - $\neg(P \vee Q)$ is equivalent to $(\neg P \wedge \neg Q)$

- Contraposition
  - $(P \rightarrow Q)$ is equivalent to $(\neg Q \rightarrow \neg P)$

Other equivalences

- $(P \rightarrow Q)$ is equivalent to $(\neg P \vee Q)$
- $(P \leftrightarrow Q)$ is equivalent to $((P \rightarrow Q) \wedge (Q \rightarrow P))$
- $(P \leftrightarrow Q)$ is equivalent to $((P \wedge Q) \vee (\neg P \wedge \neg Q))$
- $(P \wedge \neg P)$ is equivalent to ***F***
- $(P \vee \neg P)$ is equivalent to ***T***

Propositional implication rules

- Re-write rules are good for bidirectional search
  - What if equivalence does not hold
- Modus Ponens

$$\frac{A \rightarrow B,\ A}{B}$$

  - Comma used for conjunction
  - Above the line: what we know
  - Below the line: what we can deduce

Proving Modus Ponens

A	B	A→B	α: A→B, A	β: B
True	True	True	True	True
True	False	False	False	False
False	True	True	False	True
False	False	True	False	True

Elimination and introduction of “and”

- “and” elimination

$$\frac{A_1, A_2, \ldots, A_n}{A_i}$$

[1 ≤ i ≤ n]

- “and” introduction

$$\frac{A_1, A_2, \ldots, A_n}{A_1 \wedge A_2 \wedge \ldots \wedge A_n}$$

Introduction of “or”; Unit resolution

- “or” introduction

$$\frac{A_i}{A_1 \vee A_2 \vee \ldots \vee A_n}$$

[1 ≤ i ≤ n]

- Unit resolution
  - Basis for theorem proving

$$\frac{(A \vee B) \wedge \neg B}{A}$$

Problems

- Too many predicates
  - Sample r.: “If you see a stop sign, then stop!”
  - A new predicate for every stop sign
- Slow inference
- No variables (many constants needed)
  - Even more predicates