Propositional Logic

Jinghao Zhao

11/05/2020

Logic

- Logic: knowledge representation language
  - Represent human knowledge as "**sentences**" (a.k.a *axiom*)
    - **Knowledge base (KB)**: a set of sentences

- Example
  - **Propositional logic**
    - Boolean logic
  - **First-order logic**
    - Quantifiers $\forall, \exists$, objects and relations

- Key components in Logic
  - Syntax
  - Semantics
  - Reasoning/Inference

Key Components of Logic

- **Syntax:** how to write sentences
  - What kind of sentences are well-formed?
  - Example - Arithmetic system:
    - $x + y = 4$ ok, $x4y = +$ wrong

- **Semantics**: how to interpret sentences
  - Is this sentence True given this **possible world(model)**?
  - Example - Arithmetic system:
    - Sentence: $x + y = 4$
    - Possible world 1: $\{x = 2, y = 2\}$
      - sentence is True for possible world 1
    - Possible world 2: $\{x = 1, y = 0\}$
      - sentence is False for possible world 2]

- **Reasoning/Inference**
  - We have some known facts. What new knowledge can we derive from those known facts?
  - x mod 4 = 0 => x mod 2 = 0
  - Will get to details later

Propositional Logic

A.k.a. **Boolean logic**

- Syntax
- Semantics
- Inference - Entailment
  - How to prove it?
    - Proof by enumeration – **Model Checking**
    - Theorem proving – **Proof by refutation** (use resolution)

Syntax

- Atomic sentence
  - A single propositional symbol, like A (A can be True or False)
- Logical connectives
  - $\neg$ not
  - $\wedge$ and (conjunction)
  - $\vee$ or (disjunction)
  - $\Rightarrow$ $(or \rightarrow)$ implication
  - $\Leftrightarrow$ if and only if
- Complex sentence
  - $A \vee B,\ A \vee \neg C \Rightarrow B, \ldots$

A special type of sentence: Horn clause

Syntactic Forms – CNF, DNF

- **CNF (Conjunction Normal Form)**: $(A \ \vee \neg B) \wedge (A \vee \neg C \vee D)$
  - CNF consists of clauses that are connective by conjunction. Within each clause, literals are connected by **disjunction**
  - $(A \ \vee \neg B) \wedge (A \vee \neg C \vee D)$
    - 2 **clauses**: $(A \ \vee \neg B), (A \vee \neg C \vee D)$
    - 4 **variables**: A, B, C, D
    - **Literals**: $A, \neg B, \neg C, D$

- DNF (Disjunction Normal Form): $(A \ \wedge \neg B) \vee (A \wedge \neg C \wedge D)$

Completeness

All propositional sentences can be converted to CNF/DNF. (complete)

We will mainly use CNF.

For most algorithms, you will need to standardize the sentence by converting it to CNF first.

Syntactic Forms – Horn Clause

- Horn clause
  - A <u>subset</u> of CNF
  - **Each clause has at most one positive literal**
  - Not all sentences can be converted to Horn clause! (Not complete)

$A \vee B \vee \neg C$ X

$\neg A \vee B \vee \neg C$ ✓ $\equiv A \wedge C \Rightarrow B$

$\neg A \vee \neg B \vee \neg C$ ✓

A typical form (before converting to CNF)

Why do we care about Horn clause?

It's a special type! If the sentences are Horn clauses, inference can be done in linear time (exponential for general sentences)

Semantics

When is this sentence True?

$P \Rightarrow Q$ is equivalent to $\neg P \lor Q$

P	Q	$\neg P$	$P \land Q$	$P \lor Q$	$\boxed{P \Rightarrow Q}$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

Figure 7.8 Truth tables for the five logical connectives. To use the table to compute, for example, the value of $P \lor Q$ when P is true and Q is false, first look on the left for the row where P is *true* and Q is *false* (the third row). Then look in that row under the $P \lor Q$ column to see the result: *true*.

Semantics – Logical Equivalence

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) && \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) && \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) && \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) && \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) && \text{De Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) && \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) && \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) && \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

Figure 7.11 Standard logical equivalences. The symbols α, β, and γ stand for arbitrary sentences of propositional logic.

Some Very Important Definitions - Model

- Model (a.k.a. possible world)
  - In propositional logic, a model is an assignment for this sentence

$$f = (\neg A \wedge B) \leftrightarrow C$$
$$w = \{A : 1, B : 1, C : 0\}$$

- If a sentence α is true in model m, we say that model m **satisfies** α
- $M(\alpha) \coloneqq$ the set of all the models that satisfy α

Model

We say two sentences are α and β equivalent iff $M(\alpha) = M(\beta)$

α and β are inconsistent $M(\alpha \wedge \beta) = \emptyset$

α and β are consistent $M(\alpha \wedge \beta) \neq \emptyset$

α and β are mutually exclusive

- $M(\alpha) \wedge M(\beta) = \emptyset$
- $M(\alpha \wedge \beta) = \emptyset$

The same conditions to inconsistent

Some Very Important Definitions

- Knowledge Base Δ
  - A set of sentences $\{\alpha_1, \alpha_2, \ldots\}$
  - We can consider the whole knowledge base as **a single long sentence** $\alpha_1 \land \alpha_2 \land \cdots$
    - All sentences are connected by conjunction

Example – Knowledge Base

Determine models for the following (variables R, S, C (rainy, sunny, cloudy)

$$
\begin{aligned}
KB = & R \vee S \vee C; \\
& R \rightarrow (C \wedge \neg S); \\
& C \leftrightarrow \neg S
\end{aligned}
$$

$KB = \{(R = 1, S = 0, C = 1), (R = 0, C = 1, S = 0), (R = 0, C = 0, S = 1)\}$

Some Very Important Definitions - Entailment

- **Entailment**
  - $\Delta \models \beta$ iff for every model in which Δ is True β is also True
    - Essentially, wheneverΔ is True, β must be True
    - Formal definition: $\mathrm{M}(\Delta) \subseteq M(\beta)$, or $\mathrm{M}(\Delta \wedge \beta) = M(\Delta)$

- Why is entailment so important?
  - We have some known facts represented as a knowledge base Δ
  - Now we make a new claim β
  - Does our known facts support this new claim β?

- Why do we only consider the case where Δ is True?

Some Very Important Definitions - Satisfiability

- **Satisfiability**
  - α is **satisfiable** if $M(\alpha) \neq \emptyset$
    - There is some assignment (model) that makes α true.
    - For example, $\alpha \land \neg\alpha$ is unsatisfiable.

- **Validity**
  - α is **valid** if α is *always true* in all models
    - For example, $\alpha \lor \neg\alpha$ is valid.

Some More Definitions

KB entails a sentence α denoted as $\Delta \models \alpha$ if $M(\Delta \wedge \alpha) = M(\Delta)$

KB is consistent with sentence α if $M(\Delta \wedge \alpha)$ is non-empty.

KB contradicts sentence α if $\Delta \wedge \alpha$ is not satisfiable.

Inference

- Determine entailment
  - Given two sentence Δ, β, does $\Delta \vDash \beta$ hold?

Inference method

- Proof by enumeration – **Model Checking**
  - List all the models where Δ is True, check whether β is also True
- Theorem proving – **Proof by refutation (resolution)**
  - Use resolution rule

- **Soundness**: is this inference rule/algorithm correct in all cases
- **Completeness**: can it determine entailment for <u>any</u> $\Delta \vDash \beta$

Equivalence Review

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) \quad \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) \quad \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) \quad \text{De Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) \quad \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

Figure 7.11 Standard logical equivalences. The symbols α, β, and γ stand for arbitrary sentences of propositional logic.

Inference Rules

We assume that we work on CNF.
We can omit the conjunction connectives among clauses and use commas.

- Modus Ponen: $$\frac{\alpha, \alpha \rightarrow \beta}{\therefore \beta}$$
  - Example: $\Delta = \{A, B, B \vee C, B \rightarrow D\}$
- And-Elimination $$\frac{\alpha \wedge \beta}{\therefore \alpha}$$
- Resolution $$\frac{\alpha \vee \beta, \neg\beta \vee \delta}{\therefore \alpha \vee \delta}$$

$(\alpha \vee \beta) \wedge (\neg\beta \vee \delta)$

(1) Model Checking - Example

$\Delta : \{A, A \vee B \rightarrow C\}$

$\alpha : c$

Determine if $\Delta \models \alpha$

(not A and not B) or C

- Draw a truth table
- For every model (assignment):
  - If Δ is True:
    - if α is True:
      - Continue to next model
    - else (α is False):
      - Return False (no entailment)
  - else (Δ is False):
    - skip
- Return True (after scanning the whole table without returning False)

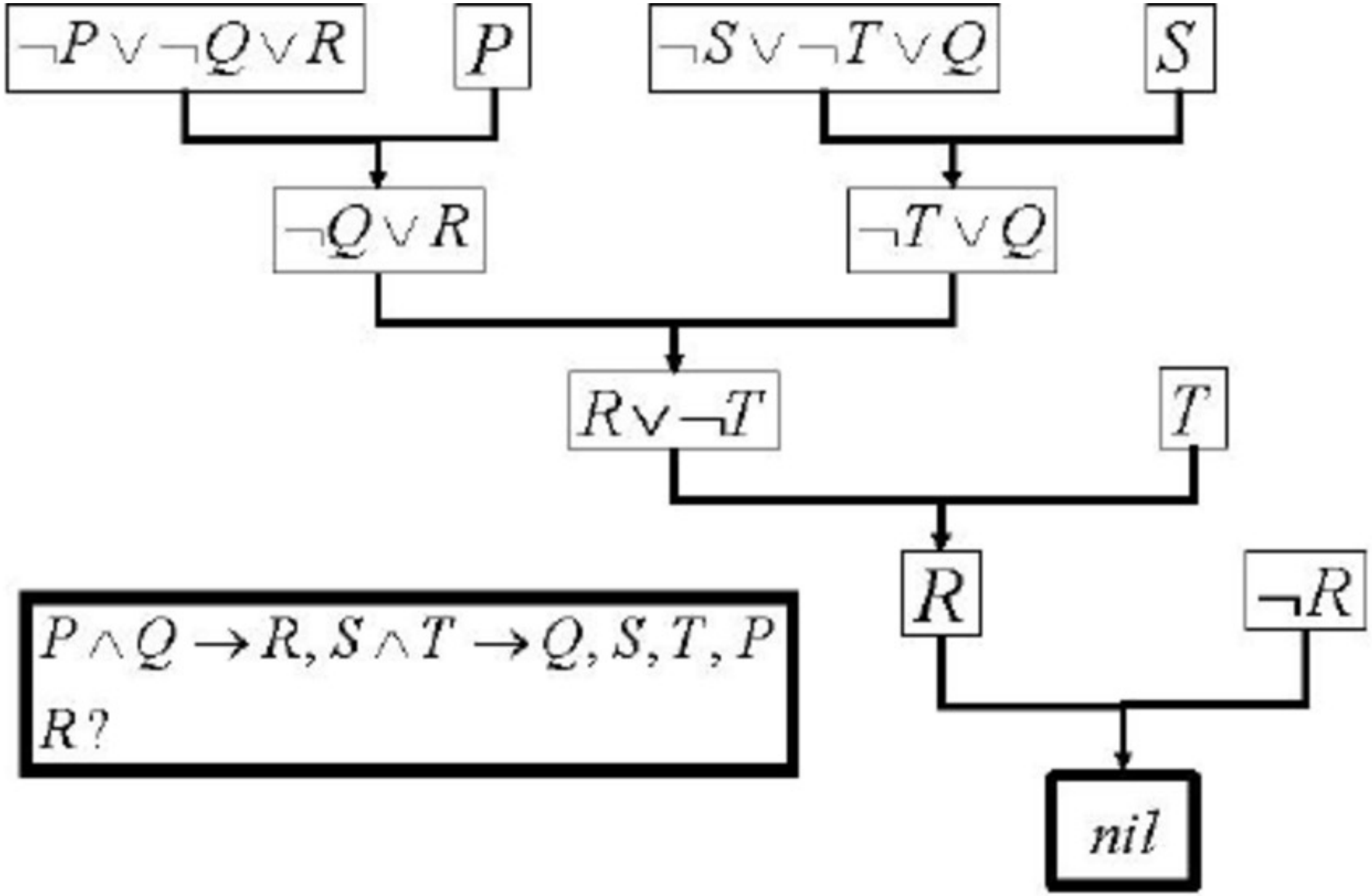
(2)Theorem Proving – Proof by Refutation (Resolution)

How do we determine whether $\Delta \models \alpha$?
Proof by refutation: $\Delta \models \alpha$ if and only if the sentence $(\Delta \wedge \neg\alpha)$ is unsatisfiable.

How do we determine whether $(\Delta \wedge \neg\alpha)$ is unsatisfiable?
Proof by Resolution (a.k.a. a resolution-based algorithm): Use the resolution inference rule. This algorithm is sound and complete. It applies to any kind of Δ and α.

This algorithm is **sound and complete!**

Exercise


$\neg P \vee \neg Q \vee R$
P
$\neg S \vee \neg T \vee Q$
S
$\neg Q \vee R$
$\neg T \vee Q$
$R \vee \neg T$
T
R
$\neg R$
nil
$P \wedge Q \rightarrow R, S \wedge T \rightarrow Q, S, T, P$
R?

Exercise

$\Delta = [(A \lor \neg B) \Rightarrow C] \land [C \Rightarrow D \lor \neg E] \land [E \lor D]$
$\alpha = A \Rightarrow D$

1. $\neg A \lor C$
2. $B \lor C$
3. $\neg C \lor D \lor \neg E$
4. $E \lor D$
5. A
6. $\neg D$
7. C (1,5)
8. $D \lor \neg E$ (3,7)
9. E (4,6)
10. D (8,9)
11. $\neg E$ (6,8)