CS 161

Propositional Logic

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

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

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)

Inference algorithm for Horn clause:

- Forward Chaining
- Backward Chaining

(Will discuss later)

Syntactic Forms – Horn Clause

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

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)

Inference algorithm for Horn clause:

- Forward Chaining
- Backward Chaining

(Will discuss later)

$A \vee B \vee \neg C$ 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)

Semantics

When is this sentence True?

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

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
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 $P \vee Q$ 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 $P \vee Q$ column to see the result: *true*.

Semantics – Logical Equivalence

$(\alpha \wedge \beta)$	$\equiv$	$(\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv$	$(\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv$	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv$	$(\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

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

Exercise – Convert a Sentence to CNF

$$B \Leftrightarrow (P \lor Q)$$

Exercise – Convert a Sentence to CNF

$$B \Leftrightarrow (P \vee Q)$$

1. $(B \Rightarrow P \vee Q) \wedge ((P \vee Q) \Rightarrow B)$
2. $(\neg B \vee P \vee Q) \wedge (\neg(P \vee Q) \vee B)$
3. $(\neg B \vee P \vee Q) \wedge ((\neg P \wedge \neg Q) \vee B)$
4. $(\neg B \vee P \vee Q) \wedge (\neg P \vee B) \wedge (\neg Q \vee B)$

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

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

α	$A \wedge (\neg A \vee B)$	$A \wedge (\neg A \vee \neg B)$
β	B	B

Some Very Important Definitions

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

Example – Knowledge Base

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

$$KB = \quad R \vee S \vee C,$$
$$R \Rightarrow (C \wedge \neg S),$$
$$C \Leftrightarrow \neg S$$

Example – Knowledge Base

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

$$KB = \quad R \vee S \vee C,$$
$$R \Rightarrow (C \wedge \neg S),$$
$$C \Leftrightarrow \neg S$$

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

Some Very Important Definitions - Entailment

- **Entailment**
  - $\Delta \vDash \beta$ 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

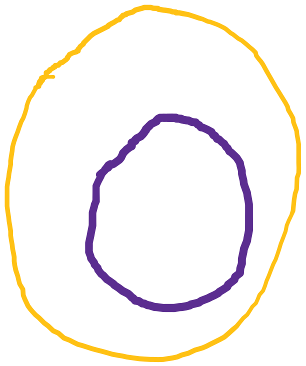
- **Satisfiability**
  - α is **satisfiable** if $M(\alpha) \neq \emptyset$
    - There is some assignment (model) that makes α true.
    - For example, $\alpha \wedge \neg\alpha$ is unsatisfiable.
- **Validity**
  - α is **valid** if α is *always true* in all models
    - For example, $\alpha \vee \neg\alpha$ is valid.

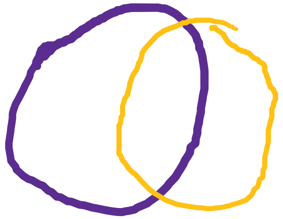
Some More Definitions

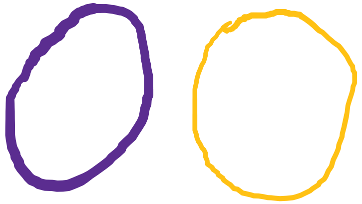
KB entails a sentence α denoted as $\Delta \models \alpha$ if $M(\Delta \wedge \alpha) = M(\Delta)$

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> $\Delta \vDash \beta$
  - (For example, forward chaining and backward chaining are not complete because it only works for Horn clause)

Equivalence Review

$(\alpha \wedge \beta)$	$\equiv$	$(\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv$	$(\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv$	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv$	$(\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

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 $$\frac{\alpha \vee \beta, \neg\beta \vee \delta}{\therefore \alpha \vee \delta}$$ $\longleftarrow (\alpha \vee \beta) \wedge (\neg\beta \vee \delta)$

(1) Model Checking - Example

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

$\alpha : c$

Determine if $\Delta \models \alpha$

- 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!**

Example

$$\Delta : A \vee \neg B \to C$$

$$(C \to D) \vee \neg E$$

$$E \vee D$$

$$\alpha : A \to D$$

Determine if $\Delta \models \alpha$

- Convert sentences to CNF first!
- Use resolution rule

Example

$$\Delta : A \vee \neg B \to C$$
$$(C \to D) \vee \neg E$$
$$E \vee D$$

$$\alpha : A \to D$$

Determine if $\Delta \models \alpha$

- Convert sentences to CNF first!
- Use resolution rule

Δ: $\neg A \vee C$,
$B \vee C$,
$\neg C \vee D \vee \neg E$,
$E \vee D$,

α: $\neg A \vee D$

Example

$$\Delta : A \vee \neg B \rightarrow C$$
$$(C \rightarrow D) \vee \neg E$$
$$E \vee D$$

$$\alpha : A \rightarrow D$$

Determine if $\Delta \models \alpha$

- Convert sentences to CNF first!
- Use resolution rule

Δ: $\neg A \vee C$,
$B \vee C$,
$\neg C \vee D \vee \neg E$,
$E \vee D$,

α: $\neg A \vee D$

$\Delta \wedge \neg\alpha$: $\Delta \wedge A \wedge \neg D$

Example

$\Delta : A \vee \neg B \rightarrow C$

$(C \rightarrow D) \vee \neg E$

$E \vee D$

$\alpha : A \rightarrow D$

Determine if $\Delta \models \alpha$

- Convert sentences to CNF first!
- Use resolution rule

Δ: $\neg A \vee C$,
$B \vee C$,
$\neg C \vee D \vee \neg E$,
$E \vee D$,

α: $\neg A \vee D$

$\Delta \wedge \neg\alpha$: $\Delta \wedge A \wedge \neg D$

$$\frac{A \quad \neg A \vee C}{C} \qquad \frac{C \quad \neg D \quad \neg C \vee D \vee \neg E}{\neg E} \qquad \frac{E \vee D \quad \neg E}{D}$$

$$\frac{\neg D \quad D}{\text{False}}$$

Exercise

$$\Delta : A \wedge B \rightarrow C, A, C \rightarrow D$$

$$\alpha : C$$

Determine if $\Delta \models \alpha$

Exercise

$$\Delta : A \wedge B \to C, A, C \to D$$

$$\alpha : C$$

Determine if $\Delta \models \alpha$

$\Delta: \neg A \vee \neg B \vee C,$
$A,$
$\neg C \vee D.$

$$\frac{\begin{array}{c} C \\ \neg C \vee D \end{array}}{D} \qquad \frac{\begin{array}{c} A \\ \neg A \vee \neg B \vee C \end{array}}{\neg B \vee C}$$

$A=1,\ B=0,\ C=1,\ D=1.$

Exercise

$$\Delta : P \vee Q, P \rightarrow R, Q \rightarrow R$$

$$\alpha : \neg R$$

Determine if $\Delta \models \alpha$

Exercise

$$\Delta : P \vee Q, P \rightarrow R, Q \rightarrow R$$

$$\alpha : \neg R$$

Determine if $\Delta \models \alpha$

$\Delta : P \vee Q,$
$\neg P \vee R,$
$\neg Q \vee R.$

$$\frac{\neg R \quad \neg P \vee R}{\neg P} \qquad \frac{\neg R \quad \neg Q \vee R}{\neg Q} \qquad \frac{\neg P \quad \neg Q \quad P \vee Q}{\text{False}}$$

Horn Clause – Forward, Backward Chaining

- α:
  $\forall x \; \mathrm{King}(x) \Rightarrow \mathrm{Person}(x)$
  $\forall x, y \; \mathrm{Person}(x) \land \mathrm{Brother}(x, y) \Rightarrow \mathrm{Person}(y)$
  $\mathrm{King}(\mathrm{Richard})$
  $\mathrm{Brother}(\mathrm{Richard}, \mathrm{John})$
- β: $\mathrm{Person}(\mathrm{John})$

Horn Clause – Forward, Backward Chaining

- α:
  $\forall x \; \mathrm{King}(x) \Rightarrow \mathrm{Person}(x)$
  $\forall x, y \; \mathrm{Person}(x) \wedge \mathrm{Brother}(x, y) \Rightarrow \mathrm{Person}(y)$
  $\mathrm{King}(\mathrm{Richard})$
  $\mathrm{Brother}(\mathrm{Richard}, \mathrm{John})$
- β: $\mathrm{Person}(\mathrm{John})$

Forward.

King(Richard)
King(x) ⇒ Person(x)
x = Richard

Person(Richard)

Brother(Richard, John)
Person(x) ∧ Brother(x,y) ⇒ Person(y)
x = Richard, y = John

Person(John)

Backward.

y = John
Person(x) ∧ Brother(x,y) ⇒ Person(y)

∀x, Person(x) ∧ Brother(x, John) ⇒ Person(John)

x = Richard
Brother(x, John) ✓
Person(Richard) ?

King(x) ⇒ Person(x)
x = Richard

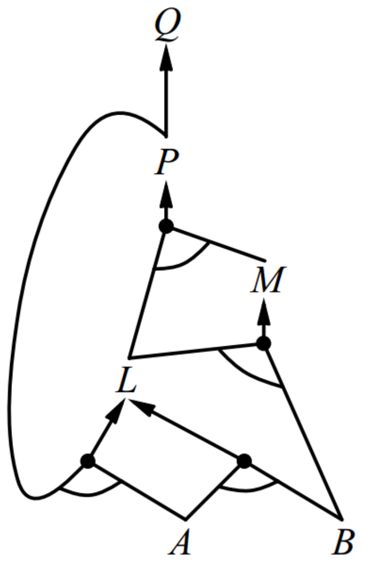
Person(Richard) ✓

Horn Clause – Forward, Backward Chaining

- α:
  $\forall x\ \mathrm{King}(x) \Rightarrow \mathrm{Person}(x)$
  $\forall x, y\ \mathrm{Person}(x) \land \mathrm{Brother}(x, y) \Rightarrow \mathrm{Person}(y)$
  $\mathrm{King}(\mathrm{Richard})$
  $\mathrm{Brother}(\mathrm{Richard}, \mathrm{John})$
- β: $\mathrm{Person}(\mathrm{John})$

$$
\begin{aligned}
P &\Rightarrow Q \\
L \land M &\Rightarrow P \\
B \land L &\Rightarrow M \\
A \land P &\Rightarrow L \\
A \land B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$

(a)



(b)

Figure 7.16 (a) A set of Horn clauses. (b) The corresponding AND–OR graph.