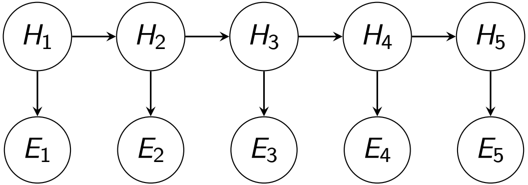
CS-541: Artificial Intelligence
Lecture 10

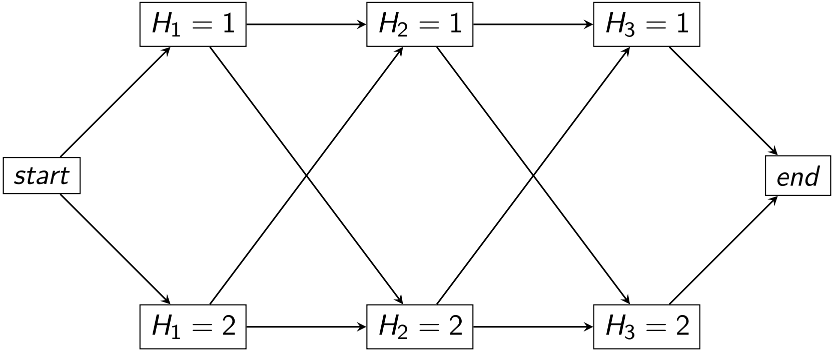
Abdul Rafae Khan
Department of Computer Science
Stevens Institute of Technology
akhan4@stevens.edu

April 18, 2022

Recap: Hidden Markov Model



Recap: Lattice Representation



Edge *start* $\Rightarrow H_1 = 1$ has a probability $p(h_1)p(e_1|h_1)$

Edge $H_{i-1} = h_{i-1} \Rightarrow H_i = h_i$ has weight $p(h_i|h_{i-1})p(e_i|h_i)$

Each path from *start* to *end* is an assignment with weights equal to the product of edge weights

Forward-Backward

Smoothing queries:

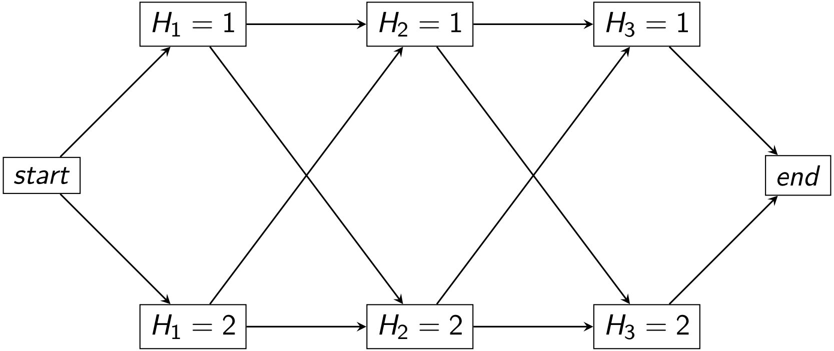
$$\mathbb{P}(H_i = h_i | E_i = e_i) \propto S_i(h_i)$$

Forward-Backward Algorithm:

```
Compute F_1, F_2, ..., F_n
Compute B_n, B_{n-1}, ..., B_1
Compute S_i for each i and normalize
```

Running time: $O(nK^2)$

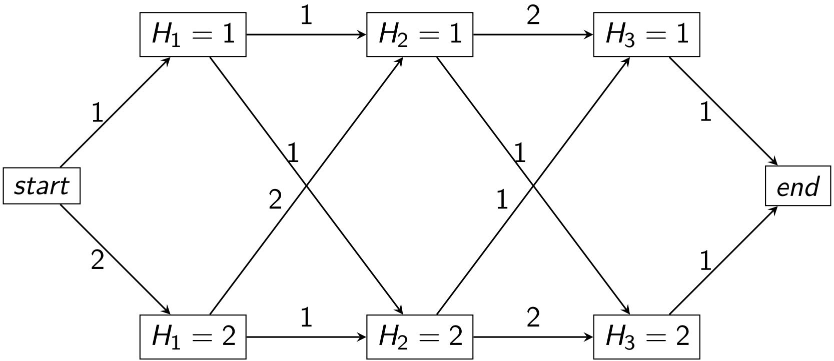
Recap: Forward-Backward Algorithm



Question:

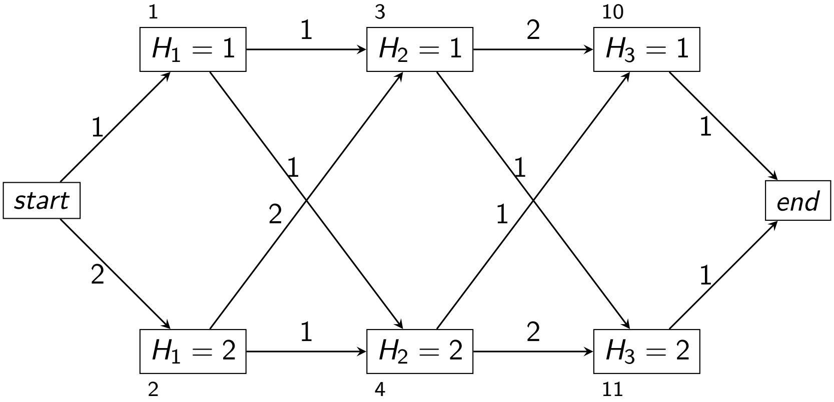
$\mathbb{P}(H_2 = 2 | E_1 = 1, E_2 = 2, E_3 = 1) = ?$

Recap: Forward-Backward Algorithm



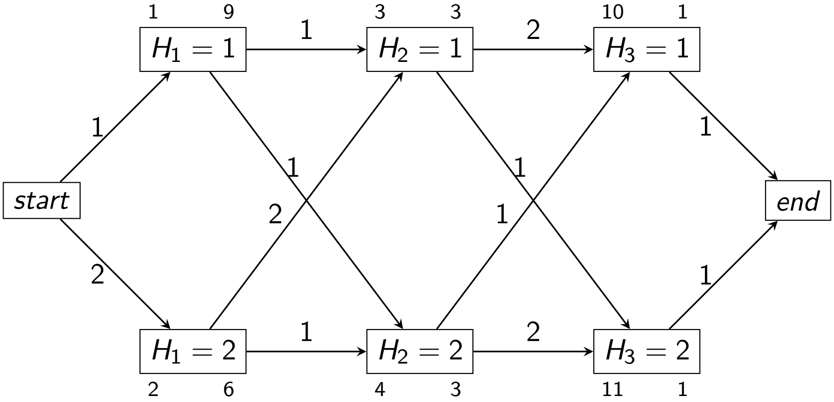
Recap: Forward-Backward Algorithm

Computing forward values F_i



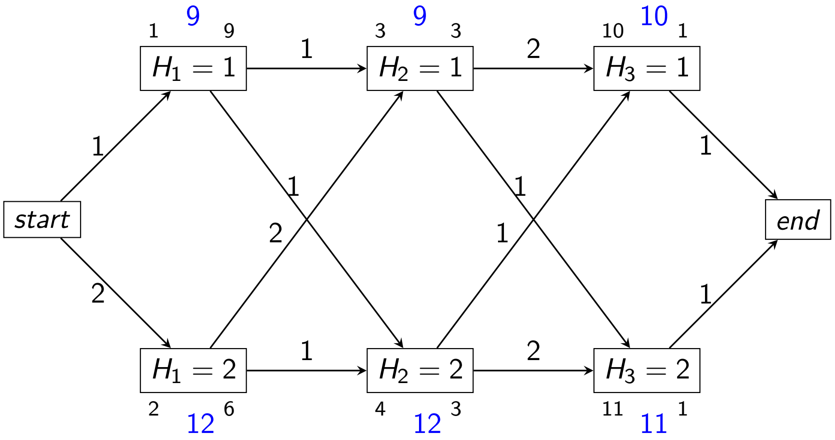
Recap: Forward-Backward Algorithm

Computing backward values B_i

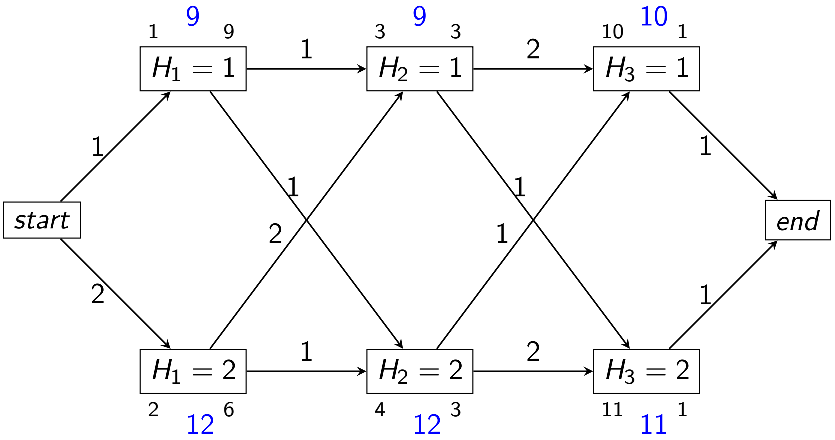


Recap: Forward-Backward Algorithm

Computing S_i



Recap: Forward-Backward Algorithm

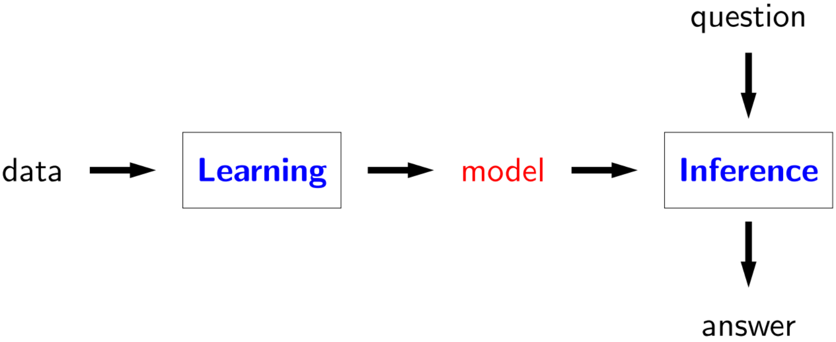


Question:
$\mathbb{P}(H_2 = 2 | E_1 = 1, E_2 = 2, E_3 = 1) = \frac{12}{12+9}$

Question

If $X_1 + X_2 = 10$ and $X_1 X_2 = 4$, what is X_1?

Take a step back



Examples: search problems, MDPs, Bayesian networks

Modeling paradigm

State-based models: search problems, MDPs
Applications: route finding, game playing, etc.
Think in terms of states, actions, and costs

Variable-based models: Bayesian networks
Applications: tracking, medical diagnosis, etc. Think in terms of variables and factors

Logic-based models: propositional logic, first-order logic
Applications: theorem proving, verification, reasoning
Think in terms of logical formulas and inference rules

A historical note

Logic was dominant paradigm in AI before 1990s

```
(ING/BY
   (PUSH NP/ T
        (SETR SUBJ *)
        (TO VP/VP

           (* IF THE SUBJECT WAS NOT PROPERLY DETERMINED IN A
           POSS-ING COMPLEMENT, LOOK FOR IT HERE.)

)))

(NP/
   (CAT DET T

          ((GETF POSSPRO                           (* START OF THE NP
                                                   NETWORK.))
            (ADDL ADJS (BUILDQ (POSS (NP (PRO *)))))
            (SETRQ DET THE

           (* IF THE DETERMINER IS A POSSESSIVE PRONOUN
           (MY, YOUR), CONSTRUCT THE POSSESIVE MODIFIER AND USE
           'THE' FOR THE DETERMINER)

))
          (T (SETR DET *)))
        (TO NP/ART))
   (CAT PRO T
        (SETR N (BUILDQ (PRO *))                   (* A PRONOUN MAY PICK UP
                                                   PP MODIFIERS IN NP/HEAD)
              )
        (SETR NU (GETF NUMBER))
        (TO NP/NP))
   (MEM (WHETHER IF)
        T
        (SETR NTYPE *)
        (TO COMPL/NTYPE

           (* CONSTRUCT THE COMPLEMENT STRUCTURE FOR SENTENCES
           SUCH AS 'I DON'T KNOW WHETHER HE LEFT.')
```

- Problem 1: *deterministic*, didn't handle uncertainty (probability addresses this)
- Problem 2: rule-based, didn't allow fine tuning from *data* (machine learning addresses this)
- Strength: provides *expressiveness* in a compact way

Natural Language

Example:

- A dime is better than a nickel.
- A nickel is better than a penny.
- Therefore, a dime is better than a penny.

Natural Language

Example:

- A dime is better than a nickel.
- A nickel is better than a penny.
- Therefore, a dime is better than a penny.

Example:

- A penny is better than nothing.
- Nothing is better than world peace.
- Therefore, a penny is better than world peace???

Natural Language

Example:

- A dime is better than a nickel.
- A nickel is better than a penny.
- Therefore, a dime is better than a penny.

Example:

- A penny is better than nothing.
- Nothing is better than world peace.
- Therefore, a penny is better than world peace???

Natural language is slippery!!!

Language

Language is a mechanism for expression.

Natural languages (informal):
English: Two divides even numbers.
German: Zwei dividieren geraden zahlen.

Programming languages (formal):

Python: `def even(x): return x % 2 == 0`
C++: `bool even(int x) { return x % 2 == 0; }`

Logical languages (formal):

First-order-logic: $\forall x.\, \mathrm{Even}(x) \rightarrow \mathrm{Divides}(x, 2)$

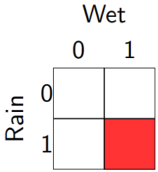
Two goals of a logic language

- Represent knowledge about the world
- Reason with that knowledge

Ingredients of a logic

Syntax: defines a set of valid formulas (Formulas)
Example: Rain $\wedge$ Wet

Semantics: for each formula, specify a set of models (assignments/configurations of the world)
Example:



Inference rules: given f, what new formulas g can be added that are guaranteed to follow $(\frac{f}{g})$?
Example: from Rain $\wedge$ Wet, derive Rain

Syntax versus semantics

Syntax: what are valid expressions in the language?
Semantics: what do these expressions mean?
Different syntax, same semantics (5):

$$2 + 3 \Leftrightarrow 3 + 2$$

Same syntax, different semantics (1 versus 1.5):

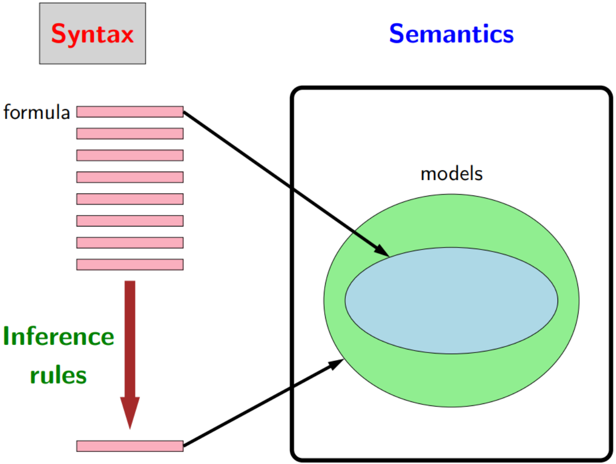
$$3/2 \text{ (Python 2.7)} \not\Leftrightarrow 3/2 \text{ (Python 3)}$$

Logics

- Propositional logic
- Propositional logic with only Horn clauses
- First-order logic
- First-order logic with only Horn clauses

Tradeoff: Balance expressivity and computational efficiency.

Propositional logic



Syntax of propositional logic

Propositional symbols (atomic formulas): A, B, C

Logical connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

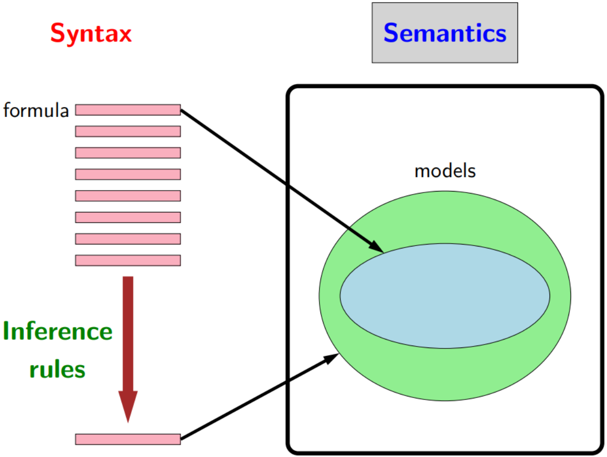
Build up formulas recursively — if f and g are formulas, so are the following:

- Negation: $\neg f$
- Conjunction: $f \wedge g$
- Disjunction: $f \vee g$
- Implication: $f \rightarrow g$
- Biconditional: $f \leftrightarrow g$

Syntax of propositional logic

- Formula: A
- Formula: $\neg A$
- Formula: $\neg B \rightarrow C$
- Formula: $\neg A \wedge (\neg B \rightarrow C) \vee (\neg B \vee D)$
- Formula: $\neg\neg A$
- Non-formula: $A \neg B$
- Non-formula: $A + B$

Propositional logic



Model

A **model** w in propositional logic is an assignment of truth values to propositional symbols

Example:

- 3 propositional symbols: A, B, C
- $2^3 = 8$ possible models w:

$$\{A{:}0, B{:}0, C{:}0\}$$

$$\{A{:}0, B{:}0, C{:}1\}$$

$$\{A{:}0, B{:}1, C{:}0\}$$

$$\{A{:}0, B{:}1, C{:}1\}$$

$$\{A{:}1, B{:}0, C{:}0\}$$

$$\{A{:}1, B{:}0, C{:}1\}$$

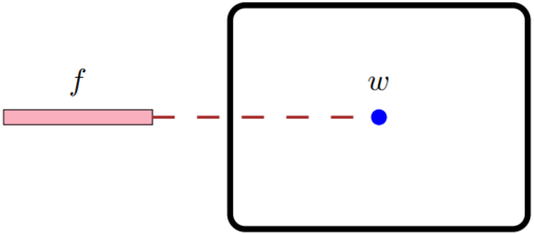
$$\{A{:}1, B{:}1, C{:}0\}$$

$$\{A{:}1, B{:}1, C{:}1\}$$

Model

Let f be a formula.
Let w be a model.
An interpretation function $\mathcal{I}(f, w)$ returns:

  true (1) (say that w satisfies f)
  false (0) (say that w does not satisfy f)



Interpretation function: definition

Base case:

- For a propositional symbol p (e.g., A, B, C): $\mathcal{I}(p, w) = w(p)$

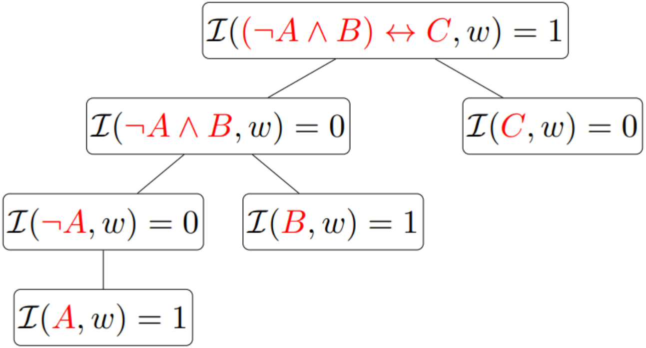
Recursive case:

- For any two formulas f and g, define:

$\mathcal{I}(f, w)$	$\mathcal{I}(g, w)$	$\mathcal{I}(\neg f, w)$	$\mathcal{I}(f \wedge g, w)$	$\mathcal{I}(f \vee g, w)$	$\mathcal{I}(f \rightarrow g, w)$	$\mathcal{I}(f \leftrightarrow g, w)$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Example: inverted v-structure

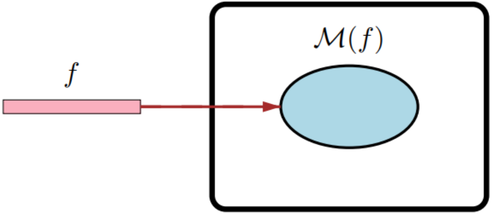
Formula: $f = (\neg A \wedge B) \leftrightarrow C$
Model: $w = \{A : 1, B : 1, C : 0\}$
Interpretation:



Formula represents a set of models

Each formula f and model w has an interpretation $\mathcal{I}(f, w) \in \{0, 1\}$

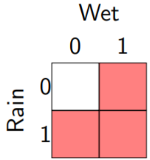
Model:
Let $\mathcal{M}(f)$ be the set of models w for which $\mathcal{I}(f, w) = 1$



Models: example

Formula:
$f = \text{Rain} \lor \text{Wet}$

Models:
$\mathcal{M}(f) =$



Compact representation:
A formula compactly represents a set of models.

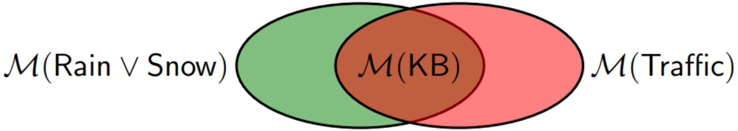
Knowledge base

A **knowledge base** KB is a set of formulas representing their conjunction/intersection:

$$\mathcal{M}(KB) = \cap_{f \in KB} \mathcal{M}(f)$$

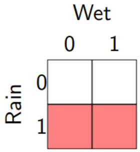
Intuition: KB specifies constraints on the world. $\mathcal{M}(KB)$ is the set of all worlds satisfying those constraints.

Let $KB = \{Rain \vee Snow, Traffic\}$.

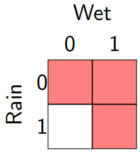


Knowledge base: example

$\mathcal{M}(\text{Rain})$

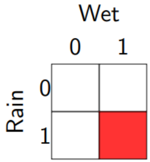


$\mathcal{M}(\text{Rain} \rightarrow \text{Wet})$



Intersection:

$\mathcal{M}(\{\text{Rain}, \text{Rain} \rightarrow \text{Wet}\})$



Adding to the knowledge base

Adding more formulas to the knowledge base:

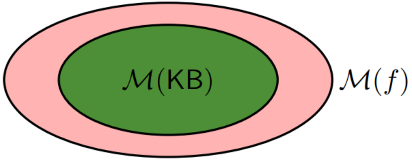
$$KB \longrightarrow KB \cup \{f\}$$

Shrinks the set of models:

$$\mathcal{M}(KB) \longrightarrow \mathcal{M}(KB) \cup \mathcal{M}(f)$$

How much does M(KB) shrink?

Entailment

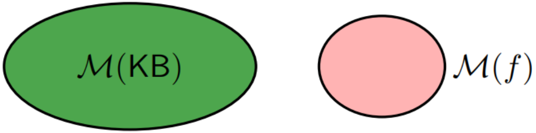


Intuition: f added no information/constraints (it was already known)

Entailment: KB entails f (written $KB \models f$) iff $\mathcal{M}(KB) \subseteq \mathcal{M}(f)$

Example: Rain $\wedge$ Snow $\models$ Snow

Contradiction

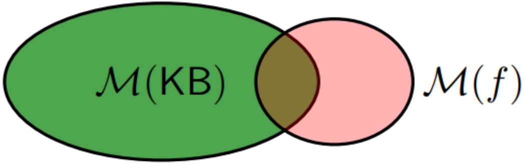


Intuition: f contradicts what we know (captured in KB)

Contradiction: KB contradicts f iff $\mathcal{M}(KB) \cap \mathcal{M}(f) = \emptyset$

Example: Rain $\wedge$ Snow contradicts $\neg$Snow

Contingency



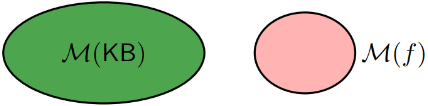
Intuition: f adds non-trivial information to KB

$$\emptyset \subsetneq \mathcal{M}(KB) \cap \mathcal{M}(f) \subsetneq \mathcal{M}(KB)$$

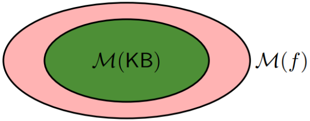
Example: Rain and Snow

Contradiction and Entailment

Contradiction:

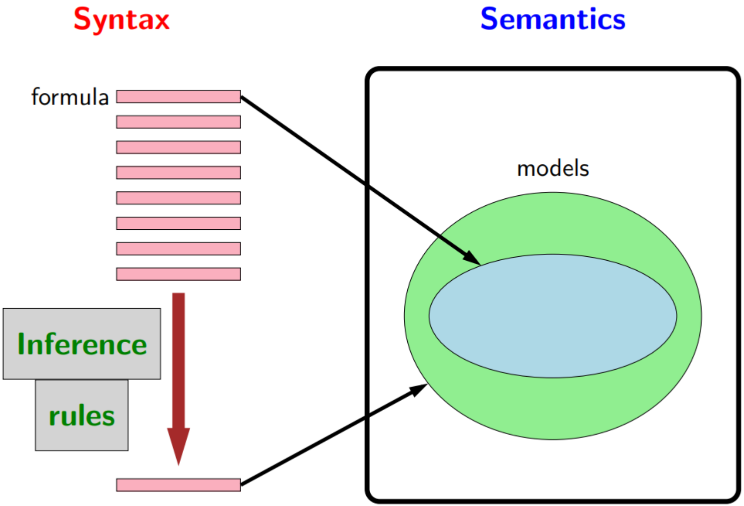


Entailment:



KB contradicts f iff KB entails $\neg f$

Propositional logic



Inference rules

Example of making an inference:
It is raining. (Rain)
If it is raining, then it is wet. (Rain $\rightarrow$ Wet)
Therefore, it is wet. (Wet)

$$\frac{Rain, Rain \rightarrow Wet}{Wet} \quad \frac{(premises)}{(conclusion)}$$

Modus ponens inference rule
For any propositional symbols p and q:

$$\frac{p, p \rightarrow q}{q}$$

Inference framework

If $f_1, \cdots, f_k, g$ are formulas, then the following is an inference rule:

$$\frac{f_1, \cdots, f_k}{g}$$

Rules operate directly on **syntax**, not on **semantics**

Inference algorithm

```
Input: set of inference rules Rules.
Repeat until no changes to KB:
    Choose set of formulas f_1, ···., f_k KB.
    If matching rule (f_1, ···., f_k)/g exists:
        Add g to KB
```

KB derives/proves f ($KB \vdash f$) iff f eventually gets added to KB

Inference example

Modus ponens inference

Starting point:

$$KB = \{Rain, Rain \rightarrow Wet, Wet \rightarrow Slippery\}$$

Apply modus ponens to Rain and Rain $\rightarrow$ Wet:

$$KB = \{Rain, Rain \rightarrow Wet, Wet \rightarrow Slippery, Wet\}$$

Apply modus ponens to Wet and Wet $\rightarrow$ Slippery:

$$KB = \{Rain, Rain \rightarrow Wet, Wet \rightarrow Slippery, Wet, Slippery\}$$

Converged

Inference example

Modus ponens inference

Starting point:

$$KB = \{Rain, Rain \rightarrow Wet, Wet \rightarrow Slippery\}$$

Apply modus ponens to Rain and Rain $\rightarrow$ Wet:

$$KB = \{Rain, Rain \rightarrow Wet, Wet \rightarrow Slippery, Wet\}$$

Apply modus ponens to Wet and Wet $\rightarrow$ Slippery:

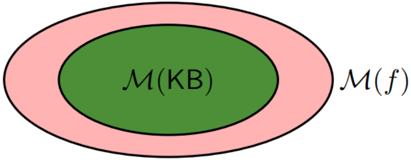
$$KB = \{Rain, Rain \rightarrow Wet, Wet \rightarrow Slippery, Wet, Slippery\}$$

Converged **Can't derive some formulas: $\neg$Wet, Rain $\rightarrow$ Slippery**

Desiderata for inference rules

Semantics:

Interpretation defines entailed/true formulas: $KB \models f$:



Syntax:

Inference rules derive formulas: $KB \vdash f$

How does $f : KB \models f$ relate to $f : KB \vdash f$?

Properties

Truth:

$$f : KB \models f$$

Soundness: A set of inference rules Rules is sound if:

$$f : KB \vdash f \subseteq f : KB \models f$$

Completeness: A set of inference rules Rules is complete if:

$$f : KB \vdash f \supseteq f : KB \models f$$

Soundness and completeness

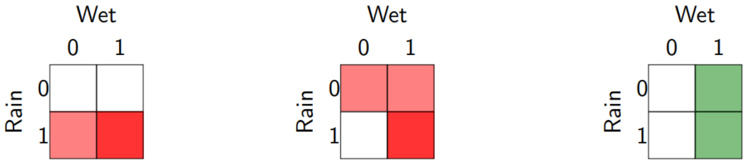
The truth, the whole truth, and nothing but the truth.

- **Soundness:** nothing but the truth
- **Completeness:** whole truth

Soundness: example

Is $\frac{Rain, Rain \rightarrow Wet}{Wet}$ (Modus ponens) sound?

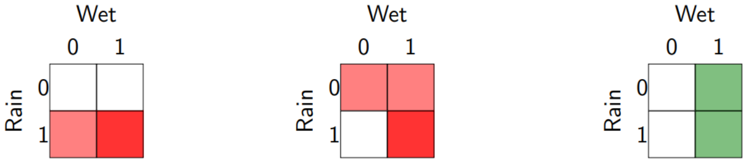
$$\mathcal{M}(Rain) \; \cap \; \mathcal{M}(Rain \rightarrow Wet) \; \subseteq ? \; \mathcal{M}(Wet)$$



Soundness: example

Is $\frac{Rain, Rain \to Wet}{Wet}$ (Modus ponens) sound?

$$\mathcal{M}(Rain) \ \cap \ \mathcal{M}(Rain \to Wet) \ \subseteq ? \ \mathcal{M}(Wet)$$

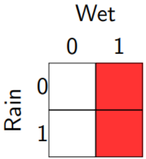


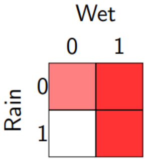
Sound!

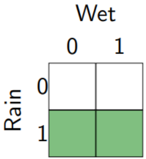
Soundness: example

Is $\frac{Rain, Rain \to Wet}{Wet}$ (Modus ponens) sound?

$$\mathcal{M}(Wet) \;\cap\; \mathcal{M}(Rain \to Wet) \;\subseteq ?\; \mathcal{M}(Rain)$$



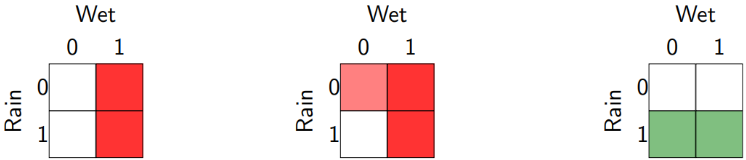




Soundness: example

Is $\frac{Rain, Rain \rightarrow Wet}{Wet}$ (Modus ponens) sound?

$$\mathcal{M}(Wet) \;\cap\; \mathcal{M}(Rain \rightarrow Wet) \;\subseteq ?\; \mathcal{M}(Rain)$$



Unsound!

Completeness: example

Recall completeness: inference rules derive all entailed formulas (f such that $KB \models f$)

Setup:

$$KB = \{Rain, Rain \vee Snow \to Wet\}$$
$$f = Wet$$

Completeness: example

Recall completeness: inference rules derive all entailed formulas (f such that $KB \models f$)

Setup:

$KB = \{Rain, Rain \vee Snow \to Wet\}$

$f = Wet$

Rules $= \{\frac{f, f \to g}{g}\}$ (Modus ponens)

Semantically: $KB \models f$ (f is entailed)

Syntactically: $KB \not\vdash f$ (can't derive f)

Incomplete!

Fixing completeness

Option 1: Restrict the allowed set of formulas

propositional logic

$\Downarrow$

propositional logic with only Horn clauses

Fixing completeness

Option 1: Restrict the allowed set of formulas

propositional logic

$\downarrow$

propositional logic with only Horn clauses

Option 2: Use more powerful inference rules

Modus ponens

$\downarrow$

resolution

Definite clauses

A **definite clause** has the following form:

$$(p_1 \wedge \cdots \wedge p_k) \rightarrow q$$

where $p_1 \wedge \cdots \wedge p_k, q$ are propositional symbols

Intuition: if $p_1, \cdots, p_k$ hold, then q holds
Example: (Rain $\wedge$ Snow) $\rightarrow$ Traffic
Example: Traffic
Non-example: $\neg$Traffic
Non-example: (Rain $\wedge$ Snow) $\rightarrow$ (Traffic $\vee$ Peaceful)

Horn clauses

A **Horn clause** is either:

- a definite clause $(p_1 \wedge \cdots \wedge p_k \rightarrow q)$
- a goal clause $(p_1 \wedge \cdots \wedge p_k \rightarrow \mathit{false})$

Example (definite): (Rain $\wedge$ Snow) $\rightarrow$ Traffic
Example (goal): Traffic $\wedge$ Accident $\rightarrow$ false

equivalent: $\neg$(Traffic $\wedge$ Accident)

Modus ponens

Inference rule:

$$\frac{p_1, \cdots, p_k, (p_1 \wedge \cdots \wedge p_k) \rightarrow q}{q}$$

Example:

$$\frac{\mathit{Wet}, \mathit{Weekday}, \mathit{Wet} \wedge \mathit{Weekday} \rightarrow \mathit{Traffic}}{\mathit{Traffic}}$$

Completeness of modus ponens

Claim: Modus ponens on Horn clauses

Modus ponens is complete with respect to Horn clauses:

- Suppose KB contains only Horn clauses and p is an entailed propositional symbol
- Then applying modus ponens will derive p

$KB \models p$ (entailment) is the same as $KB \vdash p$ (derivation)

Example: Modus ponens

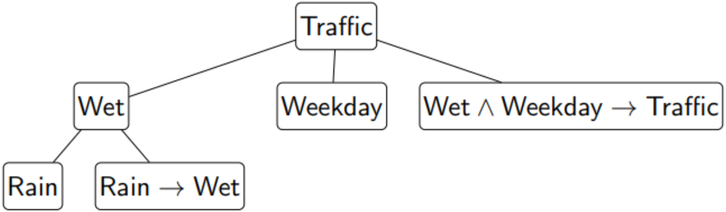
KB

Rain
Weekday
Rain $\rightarrow$ Wet
Wet $\wedge$ Weekday $\rightarrow$ Traffic
Traffic $\wedge$ Careless $\rightarrow$ Accident

Modus ponens

$$\frac{p_1, \cdots, p_k, (p_1 \wedge \cdots \wedge p_k) \rightarrow q}{q}$$

Question: $KB \models Traffic \leftrightarrow KB \vdash Traffic$



Horn clauses and disjunction

Written with implication

$A \rightarrow C$

$A \land B \rightarrow C$

Written with disjunction

$\neg A \lor C$

$\neg A \lor \neg B \lor C$

Horn clauses and disjunction

Written with implication
$A \rightarrow C$
$A \wedge B \rightarrow C$

Written with disjunction
$\neg A \vee C$
$\neg A \vee \neg B \vee C$

- **Literal:** either p or $\neg p$, where p is a propositional symbol
- **Clause:** disjunction of literals
- **Horn clauses:** at most one positive literal

Modus ponens (rewritten):

$$\frac{A, \neg A \vee C}{C}$$

Intuition: cancel out A and $\neg$A

Resolution

General clauses have any number of literals:

$$\neg A \vee B \vee \neg C \vee D \vee \neg E \vee F$$

Resolution inference

$$\frac{f_1 \vee \cdots \vee f_n \vee p, p \vee g_1 \vee \cdots \vee g_m}{f_1 \vee \cdots \vee f_n \vee g_1 \vee \cdots \vee g_m}$$

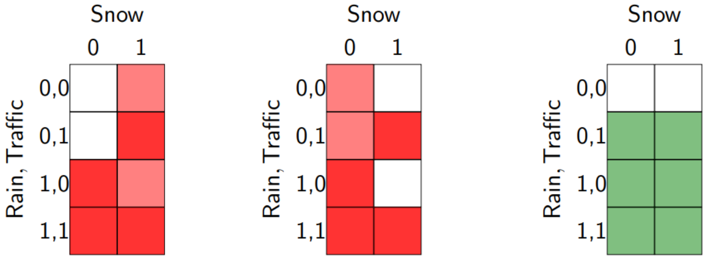
Example:

$$\begin{aligned} \mathit{Rain} \vee \mathit{Snow}, \neg \mathit{Snow} \vee \mathit{Traffic} \\ \mathit{Rain} \vee \mathit{Traffic} \end{aligned}$$

Soundness of resolution

$$\frac{Rain \vee Snow, \vee Traffic}{Rain \vee Traffic}\text{(resolution rule)}$$

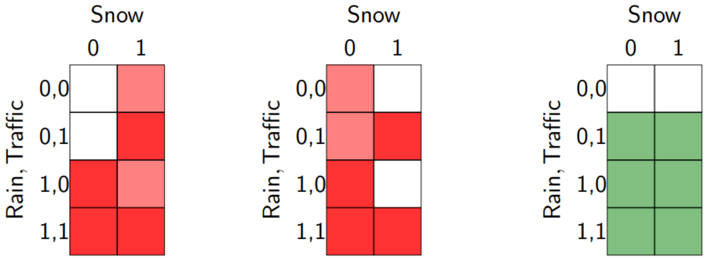
$$\mathcal{M}(Rain \vee Snow) \;\cap\; \mathcal{M}(\neg Snow \vee Traffic) \;\subseteq?\; \mathcal{M}(Rain \vee Traffic)$$



Soundness of resolution

$$\frac{\mathit{Rain} \lor \mathit{Snow}, \lor \mathit{Traffic}}{\mathit{Rain} \lor \mathit{Traffic}} \text{(resolution rule)}$$

$$\mathcal{M}(\mathit{Rain} \lor \mathit{Snow}) \ \cap \ \mathcal{M}(\neg \mathit{Snow} \lor \mathit{Traffic}) \ \subseteq ? \ \mathcal{M}(\mathit{Rain} \lor \mathit{Traffic})$$



Sound!

Conjunctive normal form

So far Resolution only works on clauses

Conjunctive normal form (CNF)
A **CNF formula** is a conjunction of clauses
Example: $(A \lor B \lor \neg C) \land (\neg B \lor D)$

Equivalent: knowledge base where each formula is a clause

Conjunctive normal form

So far Resolution only works on clauses

Conjunctive normal form (CNF)
A **CNF formula** is a conjunction of clauses
Example: $(A \vee B \vee \neg C) \wedge (\neg B \vee D)$

Equivalent: knowledge base where each formula is a clause

Conversion to CNF formula
Every formula f in propositional logic can be converted into an equivalent CNF formula f':

$$\mathcal{M}(f) = \mathcal{M}(f')$$

Conversion to CNF: example

Initial formula:

$(\mathsf{Summer} \rightarrow \mathsf{Snow}) \rightarrow \mathsf{Bizzare}$

Conversion to CNF: example

Initial formula:
(Summer $\rightarrow$ Snow) $\rightarrow$ Bizzare

Remove implication ($\rightarrow$):
$\neg$(Summer $\rightarrow$ Snow) $\vee$ Bizzare

Conversion to CNF: example

Initial formula:
$(\text{Summer} \rightarrow \text{Snow}) \rightarrow \text{Bizzare}$

Remove implication ($\rightarrow$):
$\neg(\neg\ \text{Summer} \vee \text{Snow}) \vee \text{Bizzare}$

Conversion to CNF: example

Initial formula:
(Summer $\rightarrow$ Snow) $\rightarrow$ Bizzare

Remove implication ($\rightarrow$):
$\neg$($\neg$ Summer $\vee$ Snow) $\vee$ Bizzare

Push negation ($\neg$) inwards (de Morgan):
($\neg\neg$Summer $\wedge$ $\neg$Snow) $\vee$ Bizzare

Conversion to CNF: example

Initial formula:
(Summer $\rightarrow$ Snow) $\rightarrow$ Bizzare

Remove implication ($\rightarrow$):
$\neg$($\neg$ Summer $\vee$ Snow) $\vee$ Bizzare

Push negation ($\neg$) inwards (de Morgan):
($\neg\neg$Summer $\wedge$ $\neg$Snow) $\vee$ Bizzare

Remove double negation:
(Summer $\wedge$ $\neg$ Snow) $\vee$ Bizzare

Conversion to CNF: example

Initial formula:
(Summer $\rightarrow$ Snow) $\rightarrow$ Bizzare

Remove implication ($\rightarrow$):
$\neg$($\neg$ Summer $\vee$ Snow) $\vee$ Bizzare

Push negation ($\neg$) inwards (de Morgan):
($\neg\neg$Summer $\wedge$ $\neg$Snow) $\vee$ Bizzare

Remove double negation:
(Summer $\wedge$ $\neg$ Snow) $\vee$ Bizzare

Distribute $\vee$ over $\wedge$:
(Summer $\vee$ Bizzare) $\wedge$ ($\neg Snow \vee$ Bizzare)

Conversion to CNF: general

Conversion rules:

- Eliminate $\leftrightarrow$: $\frac{f \leftrightarrow g}{(f \rightarrow g)(g \rightarrow f)}$
- Eliminate $\rightarrow$: $\frac{f \rightarrow g}{fg}$
- Move $\neg$ inwards: $\frac{\neg(f \wedge g)}{\neg f \vee \neg g}$
- Move $\neg$ inwards: $\frac{\neg(f \vee g)}{\neg f \wedge \neg g}$
- Eliminate double negation: $\frac{\neg\neg f}{f}$
- Distribute $\vee$ over $\wedge$: $\frac{f \vee (g \wedge h)}{(f \vee g) \wedge (f \vee h)}$

Resolution algorithm

Relationship between entailment and contradiction (basically "proof by contradiction")

$$KB \models f \iff KB \cup \{\neg f\} \text{ is unsatisfiable}$$

```
Add ¬f into KB
Convert all formulas into CNF.
Repeatedly apply resolution rule.
Return entailment iff derive false.
```

Resolution: example

$$KB = \{A \rightarrow (B \vee C), A, \neg B\}, \;\; f = \{C\}$$

Question:

$KB \models f$

- negate f
- convert to CNF
- Check if contradiction occurs

Resolution: example

$KB = \{A \rightarrow (B \vee C), A, \neg B\}$, $f = \{C\}$

Resolution: example

$KB' \models \{A \to (B \vee C), A, \neg B, \neg C\}$

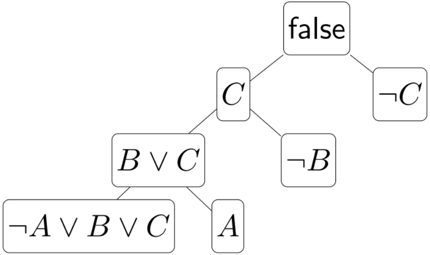
Resolution: example

$KB' \models A \rightarrow (B \vee C), A, \neg B, \neg C$

Convert to CNF:

$KB' = \neg A \vee B \vee C, A, \neg B, \neg C$

Repeatedly apply **resolution** rule:



Conclusion: KB **entails** f

Time complexity

Modus ponens inference rule

$$\frac{p_1, \cdots, p_k, (p_1 \wedge \cdots \wedge p_k) \rightarrow q}{q}$$

Each rule application adds clause with **one** propositional symbol $\Rightarrow$ linear time

$$\frac{f_1 \vee \cdots \vee f_n \vee p, \neg p \vee g_1 \vee \cdots \vee g_m}{f_1 \vee \cdots \vee f_n \vee g_1 \vee \cdots \vee g_m}$$

Each rule application adds clause with **many** propositional symbols $\Rightarrow$ exponential time

Modes ponens vs Resolution

Horn clauses	Any clause
Modus ponens	Resolution
Linear time	Exponential time
Less expressive	More expressive

Recap

Propositional logic
Syntax vs Semantics
Inference Rules
Modus Ponens
Horn Clause
Resolution

References

Stuart Russell and Xiaodong Song (2021)
CS 188 — Introduction to Artificial Intelligence
University of California, Berkeley

Chelsea Finn and Nima Anari (2021)
CS221 — Artificial Intelligence: Principles and Techniques
Stanford University

The End