COSC76/276 Artificial Intelligence
Fall 2022
Logical agents

Soroush Vosoughi
Computer Science
Dartmouth College
Soroush@Dartmouth.edu


HAPPY
Halloween

Reminders

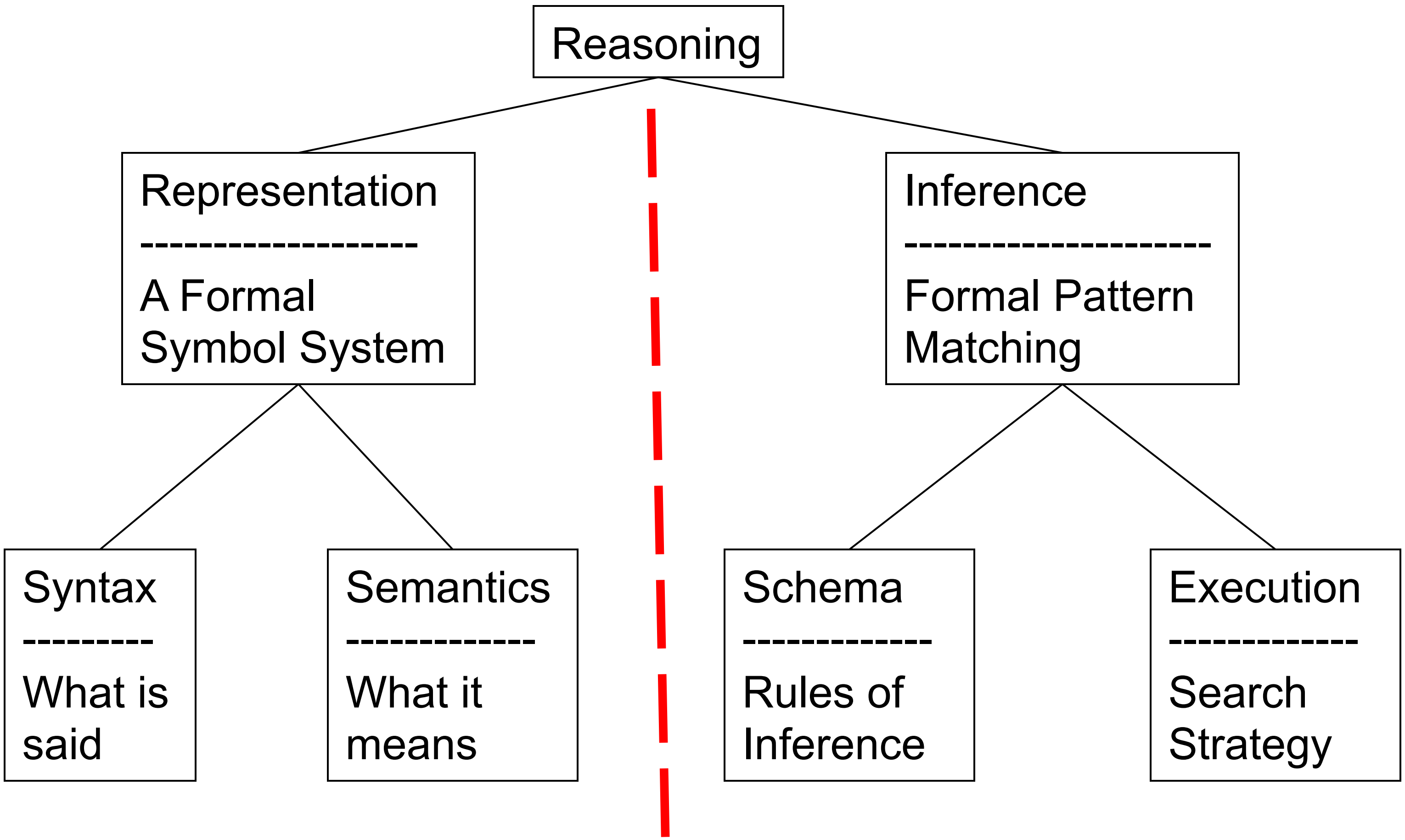
- PA4 (due Nov $4^{th)}$
- SA6 (due Nov 2nd)

Recap

- Logical agents apply inference to a knowledge base to derive new information and make decisions
  - Tell, knowledge base, ask
- Basic concepts of logic:
  - syntax: formal structure of sentences
  - Model assignment
  - semantics: truth of sentences wrt models
  - entailment: necessary truth of one sentence given another
  - inference: deriving sentences from other sentences
- Propositional logic to state specific facts about the world

Ontology:
What kind of things exist in the world?
What do we need to describe and reason about?

Review



Logic in general

- Logics are formal languages for representing information such that conclusions can be drawn from formal inference patterns

- Syntax defines the well-formed sentences in the language

- Semantics define the "meaning" or interpretation of sentences:
  - connect symbols to real events in the world
  - i.e., define truth of a sentence in a world

Entailment – formalism

- Let α and β be sentences.

- We say that $\alpha \vDash \beta$ iff for every model in which α is true, β is true.

- We let $M(\alpha)$ be the set of models for which a sentence α is true. Then $\alpha \vDash \beta$ means $M(\alpha) \subset M(\beta)$

Entailment for the logic agent

- Entailment means that one thing follows from another set of things:

$$KB \models \alpha$$

- Knowledge base *KB* entails sentence α if and only if α is true in all worlds wherein *KB* is true

- The entailed α MUST BE TRUE in ANY world in which KB IS TRUE.

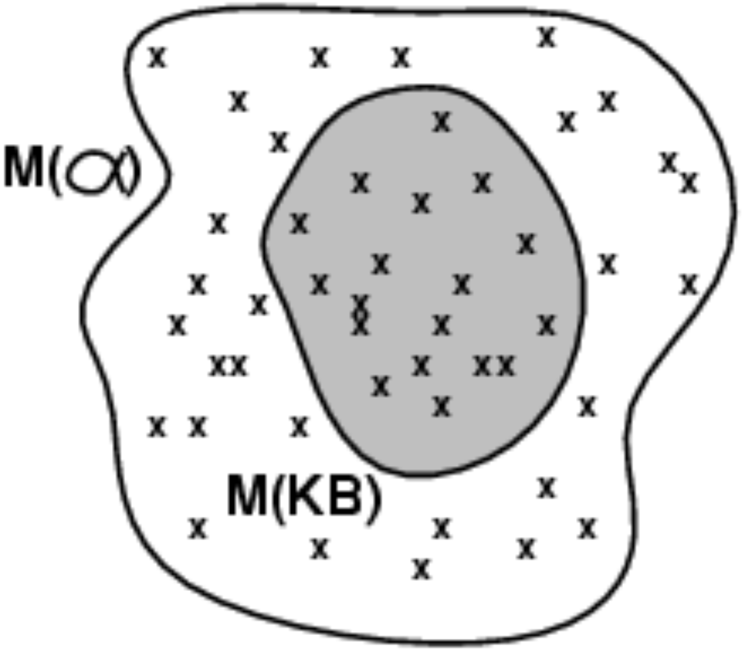
Models

- Logicians typically think in terms of models, which are formally structured worlds with respect to which truth can be evaluated

- We say *m* is a model of a sentence α if α is true in *m*

- *M(α)* is the set of all models of α

- Then KB $\models$ α iff *M(KB)* $\subseteq$ *M*(α)



Entailment examples

- E.g., the KB = “the Giants won and the Reds won” entails α = “The Giants won”.
- Example. In arithmetic, we say that $x=0 \vDash xy=0$. If you choose a model (say $x=0$, $y=6$) such that $x=0$ is true, then the sentence $xy=0$ is also true.
- E.g., KB = “$x+y = 4$” entails α = “$4 = x+y$”
- E.g., KB = “Mary is Sue’s sister and Amy is Sue’s daughter” entails α = “Mary is Amy’s aunt.”

Monotonicity

- Monotonicity: Each new sentence added to the knowledge base further constrains the set of models that holds.

- ⇒ if we can prove that some sentence is entailed by a set of sentences in the knowledge base, then adding new sentences to the knowledge base will never invalidate that proof.

$$\text{if } KB \models \alpha \text{ then } KB \land \beta \models \alpha$$

Propositional logic

- Winter ∧ NiceWeatherSunday ⇒ Procrastinated

- Atomic sentence: a symbol that can take on the value true or false.
- Literal: atomic sentence, or negated atomic sentence
- Logical connectives: ¬∨∧⇒⇔

Backus-Naur form

- Backus-Naur Form gives a recursive definition of syntax, the set of all legal sentences

$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathit{True} \mid \mathit{False} \mid P \mid Q \mid R \mid \ldots \\
\mathit{ComplexSentence} & \rightarrow & (\ \mathit{Sentence}\ ) \mid [\ \mathit{Sentence}\ ] \\
& \mid & \neg\ \mathit{Sentence} \\
& \mid & \mathit{Sentence} \land \mathit{Sentence} \\
& \mid & \mathit{Sentence} \lor \mathit{Sentence} \\
& \mid & \mathit{Sentence} \ \Rightarrow\ \mathit{Sentence} \\
& \mid & \mathit{Sentence} \ \Leftrightarrow\ \mathit{Sentence} \\
\\
\textsc{Operator Precedence} & : & \neg, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

Model and propositional logic

- model: true false values for every atomic sentence

- A world with the atomic sentences isSnowing and isSunny would have the four models (true, true), (true, false), (false, true), and (false, false).

Propositional logic: semantics

- Take a model and sentence and evaluate to T/F. Easy for atomic sentences. For complex sentences, write some rules using **truth tables** and apply recursively.

P	Q	P∧ Q
F	F	F
F	T	F
T	F	F
T	T	T

Propositional logic: semantics

- Definition of the **implies** connective:
- $P \Rightarrow Q$ is true in models for which either P is false, or both P and Q are true.

P	Q	$P \Rightarrow Q$
F	F	T
F	T	T
T	F	F
T	T	T

Truth tables for all logical connectives

P	Q	¬P	P∧Q	PVQ	P⇒Q	P⇔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

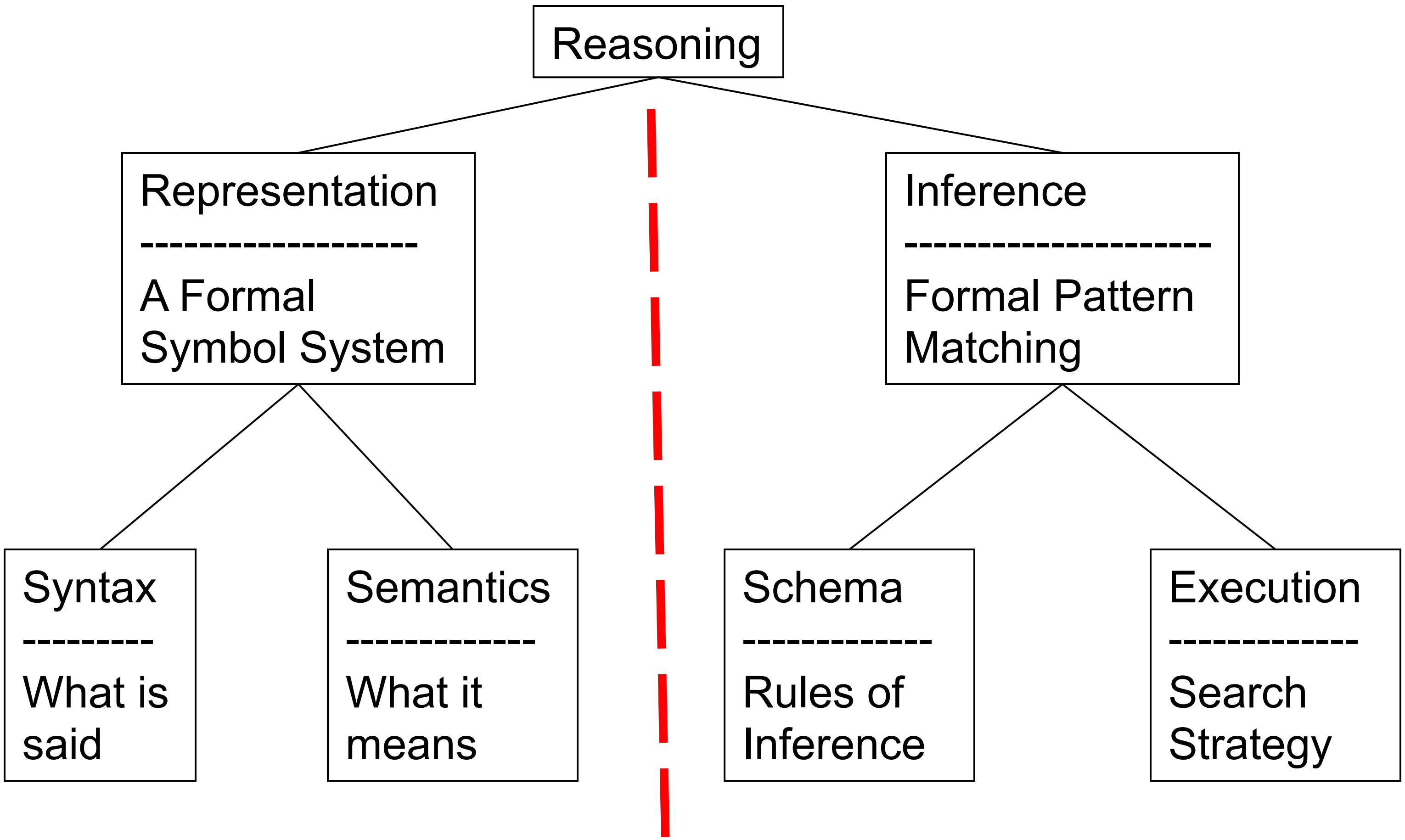
Summary

- Model: assignment of values to variables
- Sentences: used to select a set of models (winter)
- Syntax: description of legal sentences
- Semantics: maps (sentence + model) to T/F
- Entailment: $\alpha \models \beta$. ("it is greater than 100 degrees" entails "it is greater than 32 degrees")
- Propositional logic with symbols and connectives

Next

- How to make inference?

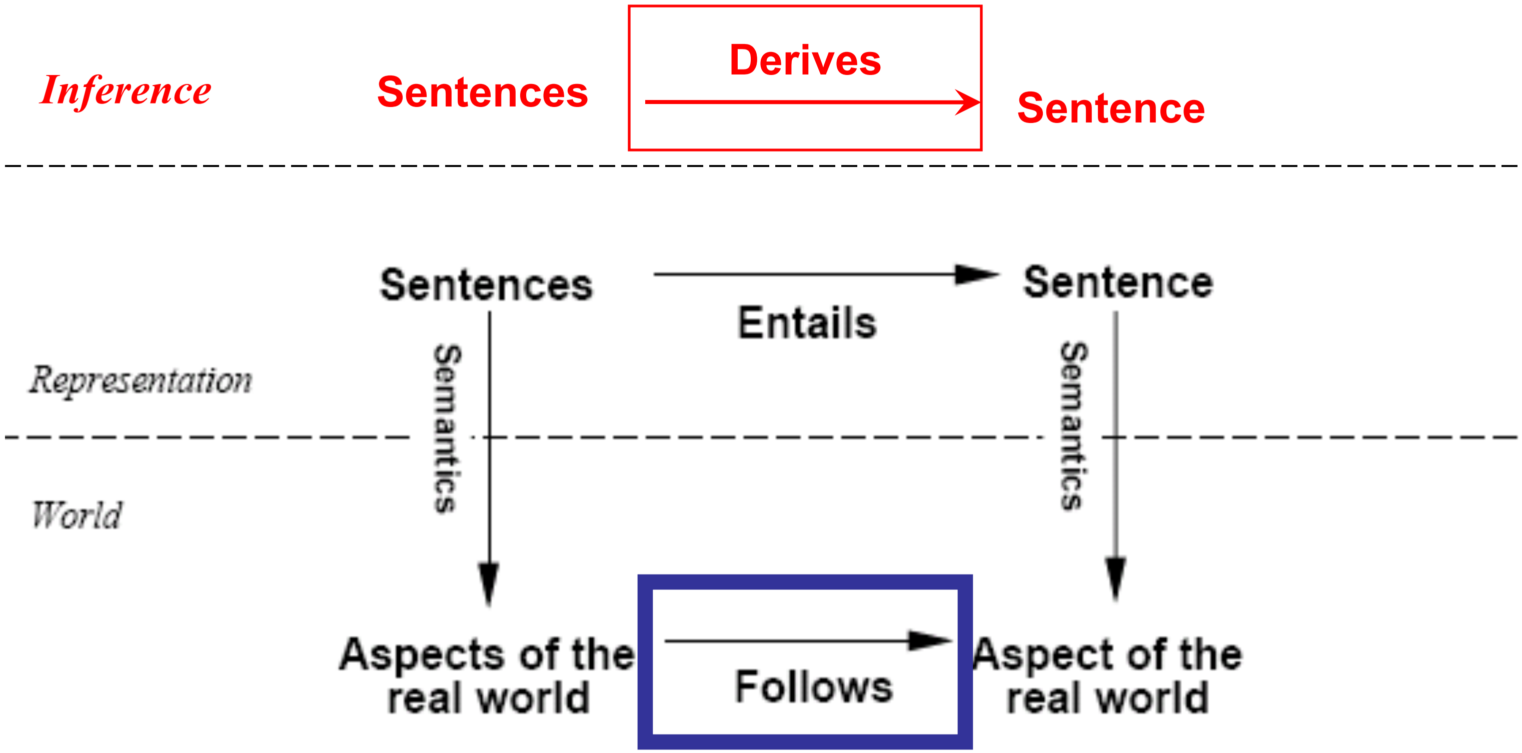
Ontology:
What kind of things exist in the world?
What do we need to describe and reason about?



- How to make inference given the knowledge base?

“Einstein Simplified: Cartoons on Science” by Sydney Harris, 1992, Rutgers University Press

Schematic perspective



If KB is true in the real world,

then any sentence α ***derived*** *from KB*

by a sound inference procedure

is also true in the real world.

Examples of Sound Inference Patterns

Classical Syllogism (due to Aristotle)

All Ps are Qs	All Men are Mortal
X is a P	Socrates is a Man
Therefore, X is a Q	Therefore, Socrates is Mortal

Implication (Modus Ponens)

P implies Q	Smoke implies Fire
P	Smoke
Therefore, Q	Therefore, Fire

Contrapositive (Modus Tollens)

P implies Q	Smoke implies Fire
Not Q	Not Fire
Therefore, Not P	Therefore, not Smoke

Law of the Excluded Middle (due to Aristotle)

A Or B	Alice is a Democrat or a Republican
Not A	Alice is not a Democrat
Therefore, B	Therefore, Alice is a Republican

Logical inference

- The notion of entailment can be used for logic inference.

- *KB* $|-_i \alpha$ means *KB* derives a sentence α using inference procedure *i*

- *Sound* (or *truth preserving*):

  The algorithm **only** derives entailed sentences.

  i is sound iff whenever KB $|-_i \alpha$ *it is also true that KB* $|= \alpha$

- *Complete*:

  The algorithm can derive **every** entailed sentence.

  i is complete iff whenever KB $|= \alpha$ *it is also true that KB* $|-_i \alpha$

Example: College life

- Let's tell the knowledge base some sentences. Then ask if some other sentence is entailed.

1. If I did not eat dinner, that implies that either the fridge was empty, or an assignment was due and I procrastinated. If the fridge was empty, or an assignment was due, and I procrastinated, then I did not eat dinner.
2. If it is winter, and there was nice weather Sunday, I procrastinated.
3. If the fridge is empty, my housemate will be mad.

Write these sentences using propositional logic

Write sentences

Example: College life

- $\neg$ Dinner $\Leftrightarrow$ FridgeEmpty $\vee$ (AssignmentDue $\wedge$ Procrastinated)
- Winter $\wedge$ NiceWeatherSunday $\Rightarrow$ Procrastinated
- FridgeEmpty $\Rightarrow$ HousemateMad

- ask: if it is not winter, and I did not eat dinner, does that imply that my housemate is mad?
- KnowledgeBase $\wedge \neg$W $\wedge$ $\neg$D$\vDash$H?

Model checking

Method #1 for inference

- Seven symbols: D, F, P, A, W, H, N.
- Each symbol can take the value true or false.
- Consider all assignments of true/false values.

- If H is true for all models in which all sentences in (KB and ¬W¬D) are true, then H is entailed.

Propositional logic example

Stated in English

- "Laws of Physics" in the Wumpus World:
  - "A breeze in B11 is equivalent to a pit in P12 or a pit in P21."
- Particular facts about a specific instance:
  - "There is no breeze in B11."
- Goal or query sentence:
  - "Is it true that P12 does not have a pit?"

Write sentences

Propositional logic example

Stated in Propositional Logic

- "Laws of Physics" in the Wumpus World:
  - "A breeze in B11 is equivalent to a pit in P12 or a pit in P21."

    $(B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1}))$
- Particular facts about a specific instance:
  - "There is no breeze in B11."

    $(\neg B_{1,1})$
- Goal or query sentence:
  - "Is it true that P12 does not have a pit?"

    $(\neg P_{1,2})$

Truth table for inference

Proposition symbols

Sentences

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

Enumerate rows (different assignments to symbols),
if KB is true in row, check that α is too

Inference by enumeration

```
function TT-ENTAILS?(KB, α) returns true or false
   inputs: KB, the knowledge base, a sentence in propositional logic
           α, the query, a sentence in propositional logic

   symbols ← a list of the proposition symbols in KB and α
   return TT-CHECK-ALL(KB, α, symbols, { })
```

```
function TT-CHECK-ALL(KB, α, symbols, model) returns true or false
   if EMPTY?(symbols) then
      if PL-TRUE?(KB, model) then return PL-TRUE?(α, model)
      else return true // when KB is false, always return true
   else do
      P ← FIRST(symbols)
      rest ← REST(symbols)
      return (TT-CHECK-ALL(KB, α, rest, model ∪ {P = true})
              and
              TT-CHECK-ALL(KB, α, rest, model ∪ {P = false }))
```

O(2^n) for n symbols

Time complexity?

Validity and satisfiability

A sentence is valid if it is true in all models,

e.g., *True*, $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$

A sentence is satisfiable if it is true in some model

e.g., $A \vee B$, C

A sentence is unsatisfiable if it is false in all models

e.g., $A \wedge \neg A$

Inference rules

Method #2 for inference

1. Apply some inference rules to the knowledge base (together with ¬W¬D) to construct new sentences.
2. Show that the sentence H is entailed

We haven't seen inference rules yet. Can be effective, but it may be difficult to find inference rules that allow the construction of the particular sentence.

Contradiction

Method #3 for inference

1. Add ¬W¬D and ¬H to the knowledge base
2. Show that this induces a contradiction: there exists a sentence that is entailed, and the negation of that sentence is also entailed.

Example: Show that are no models for which ¬W¬D holds, but H does not.

Conjunctive Normal Form (CNF)

- Boolean formulae are central to CS
  - Boolean logic is the way our discipline works
- Two canonical Boolean formulae representations:
  - CNF = Conjunctive Normal Form
    - A conjunct of disjuncts = (AND (OR …) (OR …))
    - "…" = a list of literals (= a variable or its negation)
    - CNF is used by Resolution Theorem Proving
- Can convert any Boolean formula to CNF

Conjunctive Normal Form (CNF)

We'd like to prove: KB $\models$ α
(This is equivalent to KB $\land \neg \alpha$ is unsatisfiable.)

We first rewrite $KB \land \neg\alpha$ into **conjunctive normal form (CNF).**

A "conjunction of disjunctions"

literals

$$\underbrace{(A \lor \neg B)}_{\text{Clause}} \land \underbrace{(B \lor \neg C \lor \neg D)}_{\text{Clause}}$$

- Any KB can be converted into CNF

Review: Equivalence & Implication

- Equivalence is a conjoined double implication
  - $(X \Leftrightarrow Y) \;=\; [(X \Rightarrow Y) \wedge (Y \Rightarrow X)]$

Review: de Morgan's rules

- How to bring $\neg$ inside parentheses
  - (1) Negate everything inside the parentheses
  - (2) Change operators to “the other operator”

- $\neg(X \land Y \land \ldots \land Z) = (\neg X \lor \neg Y \lor \ldots \lor \neg Z)$

- $\neg(X \lor Y \lor \ldots \lor Z) = (\neg X \land \neg Y \land \ldots \land \neg Z)$

Review: Boolean Distributive Laws

- **Both** of these laws are valid:

- AND distributes over OR
  - $X \wedge (Y \vee Z) = (X \wedge Y) \vee (X \wedge Z)$
  - $(W \vee X) \wedge (Y \vee Z) = (W \wedge Y) \vee (X \wedge Y) \vee (W \wedge Z) \vee (X \wedge Z)$

- OR distributes over AND
  - $X \vee (Y \wedge Z) = (X \vee Y) \wedge (X \vee Z)$
  - $(W \wedge X) \vee (Y \wedge Z) = (W \vee Y) \wedge (X \vee Y) \wedge (W \vee Z) \wedge (X \vee Z)$

Logical equivalence

- To manipulate logical sentences we need some rewrite rules.
- Two sentences are logically equivalent iff they are true in same models: $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

These are important to know

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

CNF rules

1. Eliminate $\Leftrightarrow$, replace with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
2. Eliminate $\Rightarrow$, replace with $\neg\alpha \vee \beta$
3. Move $\neg$ inwards:
   - $\neg(\neg\alpha)$:α
   - $\neg(\alpha \wedge \beta)$:$\neg\alpha \vee \neg\beta$ (DeMorgan)
   - $\neg(\alpha \vee \beta)$:$\neg\alpha \wedge \neg\beta$ (DeMorgan)
4. Distribute $\vee$ over $\wedge$:
   - $\alpha \vee (\beta \wedge \gamma) \rightarrow (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

Now we have a conjunction of disjunctions of literals:

- $(A \vee \neg B \vee C) \wedge$
- $(C \vee \neg D) \wedge \ldots$

CNF example

- College life
  - $\neg D \Leftrightarrow F \vee (A \wedge P)$
  - $W \wedge N \Rightarrow P$
  - $F \Rightarrow H$

1. Eliminate $\Leftrightarrow$, replace with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
2. Eliminate $\Rightarrow$, replace with $\neg \alpha \vee \beta$
3. Move $\neg$ inwards:
   - $\neg(\neg \alpha)$:α
   - $\neg(\alpha \wedge \beta)$:$\neg \alpha \vee \neg \beta$ (DeMorgan)
   - $\neg(\alpha \vee \beta)$:$\neg \alpha \wedge \neg \beta$ (DeMorgan)
4. Distribute $\vee$ over $\wedge$:
   - $\alpha \vee (\beta \wedge \gamma) \rightarrow (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

Write together

CNF example

1. Eliminate double implications
   - $\neg D \Leftrightarrow F \vee (A \wedge P)$
     - $\neg D \Rightarrow F \vee (A \wedge P)$
     - $F \vee (A \wedge P) \Rightarrow \neg D$
   - $W \wedge N \Rightarrow P$
   - $F \Rightarrow H$

CNF example

2. Replace implications with $\neg\alpha \vee \beta$:

- $\neg D \Rightarrow F \vee (A \wedge P)$
  - $\neg(\neg D) \vee (F \vee (A \wedge P))$
- $F \vee (A \wedge P) \Rightarrow \neg D$
  - $\neg(F \vee (A \wedge P)) \vee \neg D$
- $W \wedge N \Rightarrow P$
  - $\neg(W \wedge N) \vee P$
- $F \Rightarrow H$
  - $\neg F \vee H$

CNF example

3. Move ¬ inwards:
   - $\neg(\neg D) \vee (F \vee (A \wedge P))$
     - $D \vee (F \vee (A \wedge P))$
   - $\neg(F \vee (A \wedge P)) \vee \neg D$
     - $(\neg F \wedge \neg(A \wedge P)) \vee \neg D$
   - $(\neg F \wedge (\neg A \vee \neg P)) \vee \neg D$
     - $\neg(W \wedge N) \vee P$
   - $\neg W \vee \neg N \vee P$
     - $\neg F \vee H$

CNF example

4. Distribute $\vee$ over $\wedge$:

  – $D \vee (F \vee (A \wedge P))$
  – $(\neg F \wedge (\neg A \vee \neg P)) \vee \neg D$
  – $\neg W \vee \neg N \vee P$
  – $\neg F \vee H$

- After:

  – $D \vee ((F \vee A) \wedge (F \vee P))$
    - $(D \vee F \vee A) \wedge (D \vee F \vee P))$
  – …

CNF example

- It's in CNF! A conjunction of disjunctions!
  - $(D \vee F \vee A) \wedge (D \vee F \vee P))$
  - $(\neg F \vee \neg D) \wedge (\neg A \vee \neg P \vee \neg D))$
  - $\neg W \vee \neg N \vee P$
  - $\neg F \vee H$

CNF example

- Sanity check. Is
- $\neg D \Leftrightarrow F \vee (A \wedge P) \equiv (D \vee F \vee A) \wedge (D \vee F \vee P))$?
- Let's assume $\neg D$. Then $F \vee (A \wedge P)$ is true, from the double implication. Now looking at the CNF form:
- $(False \vee F \vee A) \wedge (False \vee F \vee P))$
- $(F \vee A) \wedge (F \vee P))$
- So either F is true, or both A and P are true, which is the same result we got from the original form. Sane.

Summary

- Methods for inference:
  - Model checking
  - Proofs, which need the sentences in CNF
- CNF
  - Conjunction of disjunctions
  - Any propositional logic sentence can be written in CNF using the logical equivalences

Next

- Search algorithms for inference based on CNF