CSC242: Introduction to Artificial Intelligence

Lecture 2.3

Please put away all electronic devices

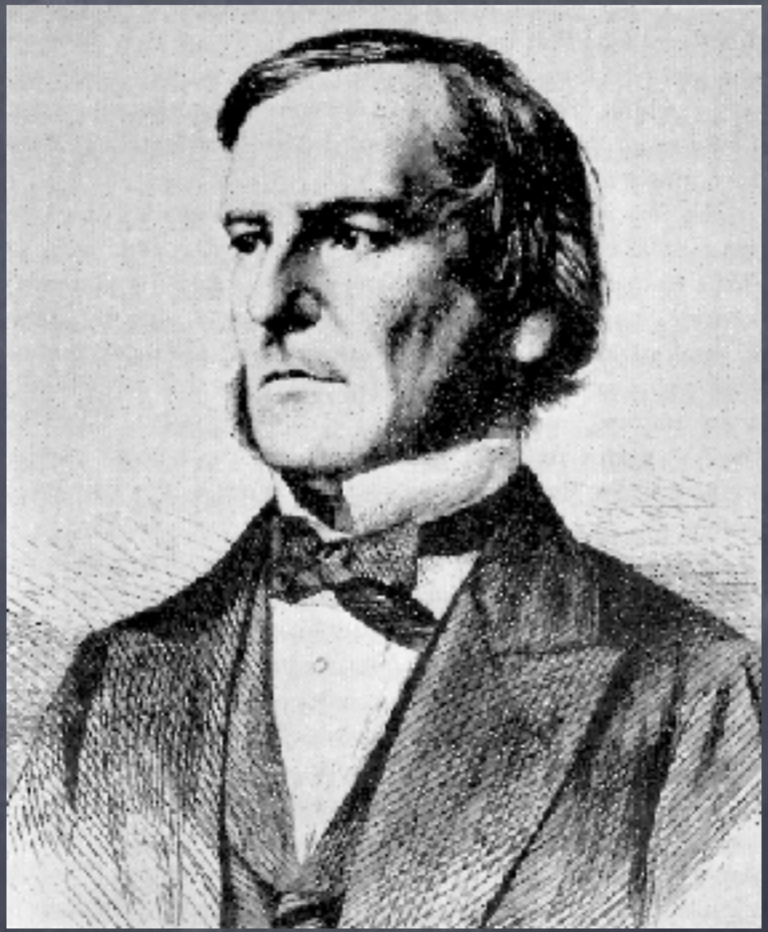
Boolean CSP

- Variables describing attributes or features of the world
  - Factored representation
- Domains: Boolean → { true, false}
- Constraints: Identify possible combinations of the boolean variables

Propositional Logic

Aristole

(384BC - 332BC)



George Boole

(1815-1864)

Propositions

$Hungry$	$Cranky$

Assignment

$Hungry$	$Cranky$
true	false
false	true
true	true
false	false

Possible Worlds

$Hungry$=true, $Cranky$=false

$Hungry$=false, $Cranky$=true

$Hungry$=true, $Cranky$=true

$Hungry$=false, $Cranky$=false

Knowledge

$Hungry \vee \ Cranky$

$Hungry$=true,

$Cranky$=false

$Hungry$=false,

$Cranky$=true

$Hungry$=true,

$Cranky$=true

$Hungry$=false,

$Cranky$=false

Knowledge

$$Hungry \Rightarrow Cranky$$

$Hungry$=true,

$Cranky$=false

$Hungry$=false,

$Cranky$=true

$Hungry$=true,

$Cranky$=true

$Hungry$=false,

$Cranky$=false

Propositional Logic

- Syntax:
  - What counts as a well-formed statement, formula, sentence, or program
- Semantics:
  - What these statements, formulas, sentences, or programs mean

Truth Table

$B_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{1,2} \vee P_{2,1}$	$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
F	F	F	F	T
F	F	T	T	F
F	T	F	T	F
F	T	T	T	F
T	F	F	F	F
T	T	F	T	T
T	F	T	T	T
T	T	T	T	T

Model (Possible World)

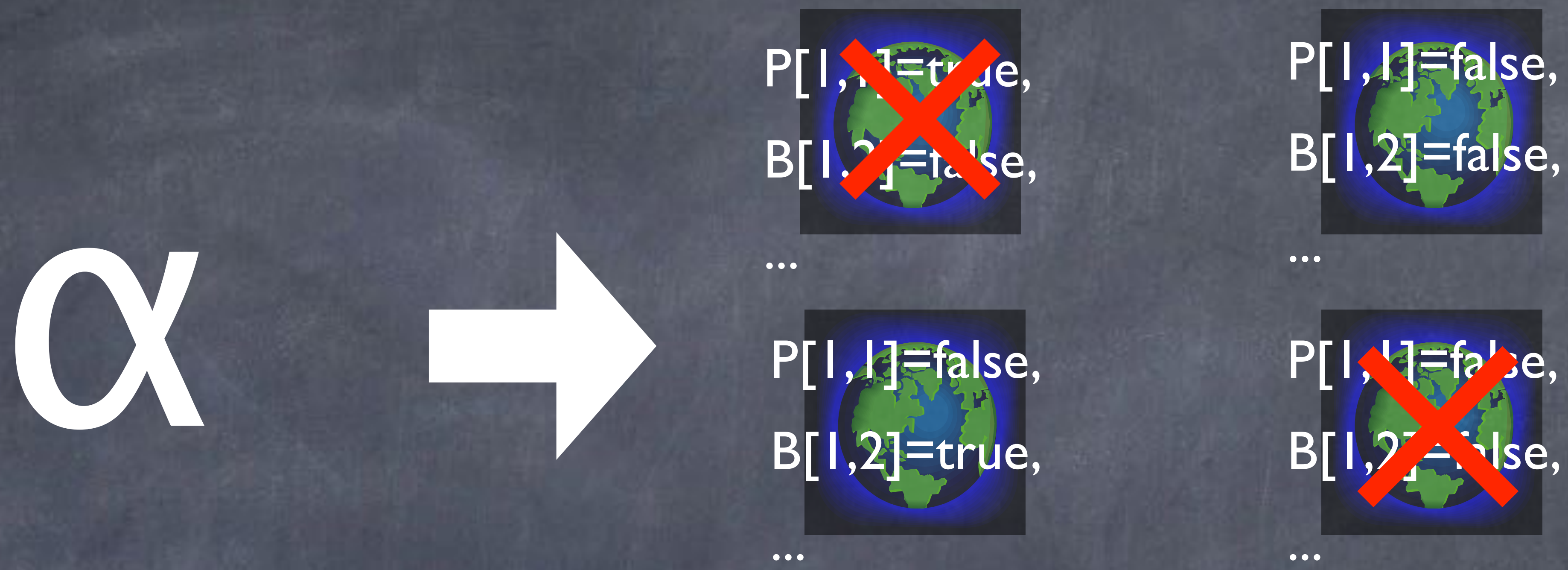
- Assignment of true or false to <u>all</u> the propositional variables
- A model <u>satisfies</u> a sentence if it makes the sentence true
  - "A model of the sentence"

Unsatisfiable

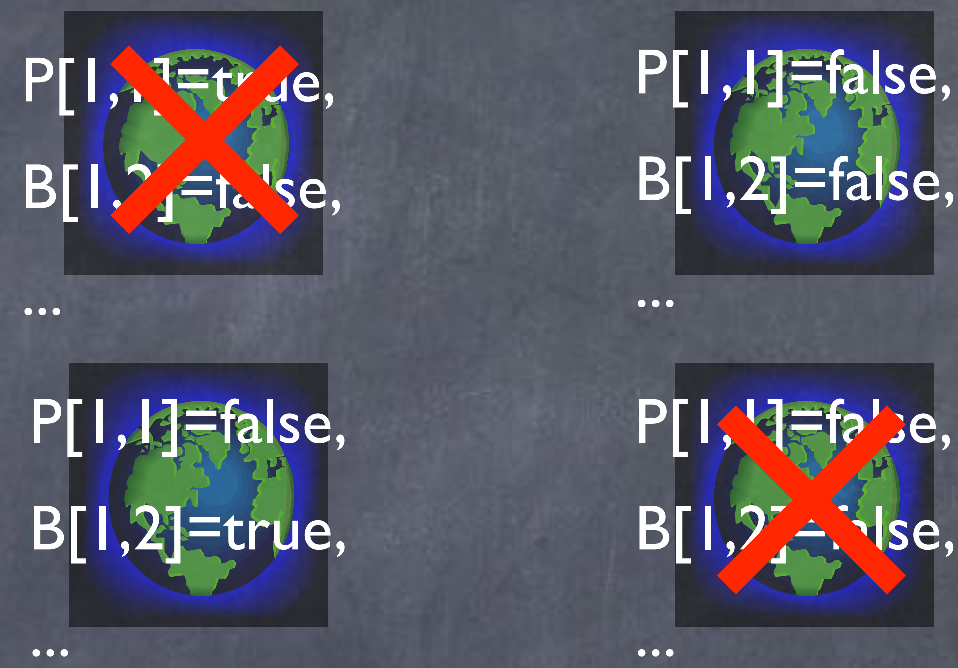
- No complete, consistent assignment of truth values to the propositions that makes the sentence or set of sentences true
- Rules out all possible worlds
- Cannot describe the actual world

Inference

- Given a set of statements in propositional logic (facts and background knowledge)
- Test whether some other statement is true
- "If I know α is true, is β also true?"
- "Does β follow from my knowledge α?"


α
P[1,1]=true,
B[1,2]=false,
...
P[1,1]=false,
B[1,2]=false,
...
P[1,1]=false,
B[1,2]=true,
...
P[1,1]=false,
B[1,2]=false,
...

α →


P[1,1]=true,
B[1,2]=false,
...
P[1,1]=false,
B[1,2]=false,
...
P[1,1]=false,
B[1,2]=true,
...
P[1,1]=false,
B[1,2]=false,
...

β →

α

β


P[1,1]=true,
B[1,2]=false,
...
P[1,1]=false,
B[1,2]=false,
...
P[1,1]=false,
B[1,2]=true,
...
P[1,1]=false,
B[1,2]=false,
...

α →


P[1,1]=true,
B[1,2]=false,
...


P[1,1]=false,
B[1,2]=false,
...


P[1,1]=false,
B[1,2]=true,
...


P[1,1]=false,
B[1,2]=false,
...

β →

α

P[1,1]=true,
B[1,2]=false,
...
P[1,1]=false,
B[1,2]=false,
...
P[1,1]=false,
B[1,2]=true,
...
P[1,1]=false,
B[1,2]=false,
...

β

Entailment

- If α entails β: $\alpha \models \beta$
  - Whenever α is true, so is β
  - Every model of α is a model of β
  - Models(α) $\subseteq$ Models(β)
  - α is at least as strong an assertion as β
    - Rules out no fewer possible worlds

Entailment is conservative

- Only accept a conclusion that is <u>guaranteed</u> to be true whenever the premises are true

Propositional Logic

- Language for expressing knowledge
- Definition of entailment (“follows from”)
- Need to be able to compute whether β logically follows from α
  - That is, whether α entails β

Entailment

- If α entails β: $\alpha \models \beta$
  - Whenever α is true, so is β
  - Every model of α is a model of β
  - Models(α) $\subseteq$ Models(β)
  - α is at least as strong an assertion as β
    - Rules out no fewer possible worlds

Entailment

P	Q	$\ldots$	α	β
F	F	...	F	T
F	T	...	F	F
F	F	...	F	F
F	T	...	T	T
T	F	...	T	T
T	T	...	T	T
T	F	...	F	F
T	T	...	F	T

Entailment

P	Q	$\ldots$	α	β
F	F	...	F	T
F	T	...	F	F
F	F	...	F	F
F	T	...	T	T
T	F	...	T	T
T	T	...	T	T
T	F	...	F	F
T	T	...	F	T

Model Checking

P	Q	$\ldots$	α	β
F	F	...	F	T
F	T	...	F	F
F	F	...	F	F
F	T	...	T	T
T	F	...	T	T
T	T	...	T	T
T	F	...	F	F
T	T	...	F	T

$At_{1,1}$

α $\neg S_{1,1}$

$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$

β $\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$

α $\neg S_{1,1}$

$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$

β $\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$

α $\neg S_{1,1}$

$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$

β $\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$	$S_{1,1}$	$W_{2,1}$	$W_{1,2}$	

$$At_{1,1}$$

α $\neg S_{1,1}$

$$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$$

β $\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$	$S_{1,1}$	$W_{2,1}$	$W_{1,2}$	
F	F	F	F	
F	F	F	T	
F	F	T	F	
F	F	T	T	
F	T	F	F	
F	T	F	T	
F	T	T	F	
F	T	T	T	
T	F	F	F	
T	F	F	T	
T	F	T	F	
T	F	T	T	
T	T	F	F	
T	T	F	T	
T	T	T	F	
T	T	T	T	

α

$At_{1,1}$

$\neg S_{1,1}$

$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$

β

$\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$	$S_{1,1}$	$W_{2,1}$	$W_{1,2}$	$\neg S_{1,1}$	
F	F	F	F	T	
F	F	F	T	T	
F	F	T	F	T	
F	F	T	T	T	
F	T	F	F	F	
F	T	F	T	F	
F	T	T	F	F	
F	T	T	T	F	
T	F	F	F	T	
T	F	F	T	T	
T	F	T	F	T	
T	F	T	T	T	
T	T	F	F	F	
T	T	F	T	F	
T	T	T	F	F	
T	T	T	T	F	

$At_{1,1}$

α $\neg S_{1,1}$

$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$

β $\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$	$S_{1,1}$	$W_{2,1}$	$W_{1,2}$	$\neg S_{1,1}$	$W_{2,1} \vee W_{1,2}$	
F	F	F	F	T	F	
F	F	F	T	T	T	
F	F	T	F	T	T	
F	F	T	T	T	T	
F	T	F	F	F	F	
F	T	F	T	F	T	
F	T	T	F	F	T	
F	T	T	T	F	T	
T	F	F	F	T	F	
T	F	F	T	T	T	
T	F	T	F	T	T	
T	F	T	T	T	T	
T	T	F	F	F	F	
T	T	F	T	F	T	
T	T	T	F	F	T	
T	T	T	T	F	T	

$$At_{1,1}$$

α $\neg S_{1,1}$

$$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$$

β $\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$	$S_{1,1}$	$W_{2,1}$	$W_{1,2}$	$\neg S_{1,1}$	$W_{2,1} \vee W_{1,2}$	$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$	
F	F	F	F	T	F	T	
F	F	F	T	T	T	F	
F	F	T	F	T	T	F	
F	F	T	T	T	T	F	
F	T	F	F	F	F	F	
F	T	F	T	F	T	T	
F	T	T	F	F	T	T	
F	T	T	T	F	T	T	
T	F	F	F	T	F	T	
T	F	F	T	T	T	F	
T	F	T	F	T	T	F	
T	F	T	T	T	T	F	
T	T	F	F	F	F	F	
T	T	F	T	F	T	T	
T	T	T	F	F	T	T	
T	T	T	T	F	T	T	

$$At_{1,1}$$

α $\neg S_{1,1}$

$$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$$

β $\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$	$S_{1,1}$	$W_{2,1}$	$W_{1,2}$	$\neg S_{1,1}$	$W_{2,1} \vee W_{1,2}$	$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$	$\neg W_{1,2}$	$\neg W_{2,1}$
F	F	F	F	T	F	T	T	T
F	F	F	T	T	T	F	F	T
F	F	T	F	T	T	F	T	F
F	F	T	T	T	T	F	F	F
F	T	F	F	F	F	F	T	T
F	T	F	T	F	T	T	F	T
F	T	T	F	F	T	T	T	F
F	T	T	T	F	T	T	F	F
T	F	F	F	T	F	T	T	T
T	F	F	T	T	T	F	F	T
T	F	T	F	T	T	F	T	F
T	F	T	T	T	T	F	F	F
T	T	F	F	F	F	F	T	T
T	T	F	T	F	T	T	F	T
T	T	T	F	F	T	T	T	F
T	T	T	T	F	T	T	F	F

$At_{1,1}$

α $\neg S_{1,1}$

$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$

β $\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$	$S_{1,1}$	$W_{2,1}$	$W_{1,2}$	$\neg S_{1,1}$	$W_{2,1} \vee W_{1,2}$	$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$	$\neg W_{1,2}$	$\neg W_{2,1}$
F	F	F	F	T	F	T	T	T
F	F	F	T	T	T	F	F	T
F	F	T	F	T	T	F	T	F
F	F	T	T	T	T	F	F	F
F	T	F	F	F	F	F	T	T
F	T	F	T	F	T	T	F	T
F	T	T	F	F	T	T	T	F
F	T	T	T	F	T	T	F	F
T	F	F	F	T	F	T	T	T
T	F	F	T	T	T	F	F	T
T	F	T	F	T	T	F	T	F
T	F	T	T	T	T	F	F	F
T	T	F	F	F	F	F	T	T
T	T	F	T	F	T	T	F	T
T	T	T	F	F	T	T	T	F
T	T	T	T	F	T	T	F	F

α

$At_{1,1}$

$\neg S_{1,1}$

$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$

β

$\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$	$S_{1,1}$	$W_{2,1}$	$W_{1,2}$	$\neg S_{1,1}$	$W_{2,1} \vee W_{1,2}$	$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$	$\neg W_{1,2}$	$\neg W_{2,1}$
F	F	F	F	T	F	T	T	T
F	F	F	T	T	T	F	F	T
F	F	T	F	T	T	F	T	F
F	F	T	T	T	T	F	F	F
F	T	F	F	F	F	F	T	T
F	T	F	T	F	T	T	F	T
F	T	T	F	F	T	T	T	F
F	T	T	T	F	T	T	F	F
T	F	F	F	T	F	T	T	T
T	F	F	T	T	T	F	F	T
T	F	T	F	T	T	F	T	F
T	F	T	T	T	T	F	F	F
T	T	F	F	F	F	F	T	T
T	T	F	T	F	T	T	F	T
T	T	T	F	F	T	T	T	F
T	T	T	T	F	T	T	F	F

$At_{1,1}$

α $\neg S_{1,1}$

$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$

β $\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$	$S_{1,1}$	$W_{2,1}$	$W_{1,2}$	$\neg S_{1,1}$	$W_{2,1} \vee W_{1,2}$	$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$	$\neg W_{1,2}$	$\neg W_{2,1}$
F	F	F	F	T	F	T	T	T
F	F	F	T	T	T	F	F	T
F	F	T	F	T	T	F	T	F
F	F	T	T	T	T	F	F	F
F	T	F	F	F	F	F	T	T
F	T	F	T	F	T	T	F	T
F	T	T	F	F	T	T	T	F
F	T	T	T	F	T	T	F	F
T	F	F	F	T	F	T	T	T
T	F	F	T	T	T	F	F	T
T	F	T	F	T	T	F	T	F
T	F	T	T	T	T	F	F	F
T	T	F	F	F	F	F	T	T
T	T	F	T	F	T	T	F	T
T	T	T	F	F	T	T	T	F
T	T	T	T	F	T	T	F	F

$At_{1,1}$

α $\neg S_{1,1}$

$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$

β $\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$	$S_{1,1}$	$W_{2,1}$	$W_{1,2}$	$\neg S_{1,1}$	$W_{2,1} \vee W_{1,2}$	$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$	$\neg W_{1,2}$	$\neg W_{2,1}$
F	F	F	F	T	F	T	T	T
F	F	F	T	T	T	F	F	T
F	F	T	F	T	T	F	T	F
F	F	T	T	T	T	F	F	F
F	T	F	F	F	F	F	T	T
F	T	F	T	F	T	T	F	T
F	T	T	F	F	T	T	T	F
F	T	T	T	F	T	T	F	F
T	F	F	F	T	F	T	T	T
T	F	F	T	T	T	F	F	T
T	F	T	F	T	T	F	T	F
T	F	T	T	T	T	F	F	F
T	T	F	F	F	F	F	T	T
T	T	F	T	F	T	T	F	T
T	T	T	F	F	T	T	T	F
T	T	T	T	F	T	T	F	F

$$At_{1,1}$$

α $\neg S_{1,1}$

$$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$$

β $\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$	$S_{1,1}$	$W_{2,1}$	$W_{1,2}$	$\neg S_{1,1}$	$W_{2,1} \vee W_{1,2}$	$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$	$\neg W_{1,2}$	$\neg W_{2,1}$
F	F	F	F	T	F	T	T	T
F	F	F	T	T	T	F	F	T
F	F	T	F	T	T	F	T	F
F	F	T	T	T	T	F	F	F
F	T	F	F	F	F	F	T	T
F	T	F	T	F	T	T	F	T
F	T	T	F	F	T	T	T	F
F	T	T	T	F	T	T	F	F
T	F	F	F	T	F	T	T	T
T	F	F	T	T	T	F	F	T
T	F	T	F	T	T	F	T	F
T	F	T	T	T	T	F	F	F
T	T	F	F	F	F	F	T	T
T	T	F	T	F	T	T	F	T
T	T	T	F	F	T	T	T	F
T	T	T	T	F	T	T	F	F

$At_{1,1}$

α $\neg S_{1,1}$

$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$

β $\neg W_{1,2}$?

$\neg W_{2,1}$?

$At_{1,1}$	$S_{1,1}$	$W_{2,1}$	$W_{1,2}$	$\neg S_{1,1}$	$W_{2,1} \vee W_{1,2}$	$S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})$	$\neg W_{1,2}$	$\neg W_{2,1}$
F	F	F	F	T	F	T	T	T
F	F	F	T	T	T	F	F	T
F	F	T	F	T	T	F	T	F
F	F	T	T	T	T	F	F	F
F	T	F	F	F	F	F	T	T
F	T	F	T	F	T	T	F	T
F	T	T	F	F	T	T	T	F
F	T	T	T	F	T	T	F	F
T	F	F	F	T	F	T	T	T
T	F	F	T	T	T	F	F	T
T	F	T	F	T	T	F	T	F
T	F	T	T	T	T	F	F	F
T	T	F	F	F	F	F	T	T
T	T	F	T	F	T	T	F	T
T	T	T	F	F	T	T	T	F
T	T	T	T	F	T	T	F	F

$$\{At_{1,1},\ \neg S_{1,1},\ S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})\} \models \neg W_{1,2} \quad ✔$$

$$\{At_{1,1},\ \neg S_{1,1},\ S_{1,1} \Leftrightarrow (W_{2,1} \vee W_{1,2})\} \models \neg W_{2,1} \quad ✔$$

Propositional Logic

- Programming language for knowledge
- Factored representation of state
  - Propositions, connectives, sentences
- Model (possible world)
- Entailment ("follows from"): $\alpha \models \beta$
  - Every model of α is a model of β

Computing Entailment

```
boolean tt_entails(Set<Sentence> kb, Sentence alpha) {
    List<Symbol> symbols = new List(kb.getSymbols());
    symbols.append(alpha.getSymbols());
    return tt_check_all(kb, alpha, symbols, new Model());
}

boolean tt_check_all(Set<Sentence> kb, Sentence alpha,
                     List<Symbol> symbols, Model model) {
    if (symbols.isEmpty()) {
        if (model.satisfies(kb)) {
            return model.satisfies(alpha);
        } else {
            return true;
        }
    } else {
        Symbol p = symbols.removeFirst();
        return (tt_check_all(kb, alpha, symbols,
                             model.clone().assign(p, Boolean.TRUE)) &&
                tt_check_all(kb, alpha, symbols,
                             model.clone().assign(p, Boolean.FALSE)));
    }
}
```

$O(2^n)$

P	Q	$\ldots$	α	$\ldots$	β
F	F	...	F	...	T
F	T	...	F	...	F
F	F	...	F	...	F
F	T	...	T	...	T
T	F	...	T	...	T
T	T	...	T	...	T
T	F	...	F	...	F
T	T	...	F	...	T

Model Checking

2^n

P	Q	...	α	...	β
F	F	...	F	...	T
F	T	...	F	...	F
F	F	...	F	...	F
F	T	...	T	...	T
T	F	...	T	...	T
T	T	...	T	...	T
T	F	...	F	...	F
T	T	...	F	...	T

Intuition: Math

Intuition: Math

$$\begin{array}{r} 123 \\ +456 \\ \hline 579 \end{array}$$

Intuition: Math

$x + 3 = 7$

$x + 3 - 3 = 7 - 3$

$x = 7 - 3$

$x = 4$

Mathematical Identities

- Allow us to rewrite equations

Mathematical Identities

- Allow us to rewrite equations
- Truth-preserving:
  - If the original equation holds, then so does the rewritten one

Inference Rules

- Look for rules that allow us to rewrite sentences in a truth-preserving way
- We'll call these <u>inference rules</u>, since they will allow us to do inference (draw conclusions, make implicit knowledge explicit)

$$Hungry \Rightarrow Cranky$$

$$Hungry$$

$$\Rightarrow$$

$$Cranky$$

$\alpha \Rightarrow \beta$

α

β

Modus Ponens

$$\frac{\alpha \Rightarrow \beta,\ \alpha}{\beta}$$

Derivation

- β can be derived from α using inference rules: $\alpha \vdash \beta$

Derivation

- β can be derived from α using inference rules: $\alpha \vdash \beta$

$$\{\ Hungry \Rightarrow Cranky,\ Hungry\ \} \vdash Cranky$$

Properties of Inference Rules

Soundness

- Derives only logically entailed sentences
- Truth-preserving

$$\text{if } \alpha \vdash \beta \quad \text{then } \alpha \models \beta$$

α	β	$\alpha \Rightarrow \beta$
F	F	T
F	T	T
T	F	F
T	T	T

Entailment

- α entails β when:
  - β is true in **every** world considered possible by α
  - Every model of α is also a model of β
  - Models(α) $\subseteq$ Models(β)

$$\{\ \alpha,\ \alpha \Rightarrow \beta\ \} \models \beta$$

$$\{ \alpha, \alpha \Rightarrow \beta \} \models \beta$$

If $\alpha \vdash \beta$ using modus ponens,

then $\alpha \models \beta$.

Modus ponens is sound

Soundness

- Derives only logically entailed sentences
- Truth-preserving

if $\alpha \vdash \beta$ then $\alpha \models \beta$

Completeness

- Derives all logically entailed sentences

if $\alpha \models \beta$ then $\alpha \vdash \beta$

Modus ponens is NOT complete

Properties of Inference Rules

Soundness: if $\alpha \vdash \beta$ then $\alpha \models \beta$

Completeness: if $\alpha \models \beta$ then $\alpha \vdash \beta$

Inference Rules

$$\frac{\alpha \wedge \beta}{\alpha}$$

And-elimination

$$\frac{\neg\neg\alpha}{\alpha}$$

Double negation

$$\frac{\neg(\alpha \wedge \beta)}{\neg\alpha \vee \neg\beta} \qquad \frac{\neg(\alpha \vee \beta)}{\neg\alpha \wedge \neg\beta}$$

DeMorgan's Laws

$$\frac{\alpha \Rightarrow \beta, \alpha}{\beta}$$

Modus Ponens

$$\frac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)} \qquad \frac{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta}$$

Definition of biconditional

Want to compute whether $\alpha \models \beta$

For a sound inference rule:

if $\alpha \vdash \beta$ then $\alpha \models \beta$

if $\alpha \vdash \gamma$ and $\gamma \vdash \beta$ then $\alpha \models \beta$

Proof (Derivation)

- Sequence of inference rule applications that lead from the premises to the conclusion

Background knowledge:

$R_1 : \neg P_{1,1}$

$R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

$R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

Perceptions:

$R_4 : \neg B_{1,1}$

$R_5 : B_{2,1}$

Biconditional elimination on R_2:

$R_6 : ((B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1}) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

And-elimination on R_6:

$R_7 : (P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}$

Logical equivalence for contrapositives on R_7:

$R_8 : \neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1})$

Modus Ponens on R_8 and R_4:

$R_9 : \neg(P_{1,2} \vee P_{2,1})$

DeMorgan's Rule R_9:

$R_{10} : \neg P_{1,2} \wedge \neg P_{2,1}$

And-elimination R_{10}:

$R_{11} : \neg P_{1,2}$

Propositional Inference As Search

- Initial state: Background knowledge and current observations
- Actions: Apply an inference rule to the sentences that match their premises
- Result: Add conclusions of inference rule to knowledge base
- Goal: The knowledge base contains the sentence we want to prove

Theorem Proving

- Searching for proofs is an alternative to enumerating models
- "In many practical cases, finding a proof can be more efficient because the proof can ignore irrelevant propositions, no matter how many of them there are."

Propositional Inference

- Entailment: "follows from our knowledge"
- Model checking
- Inference rules: soundness, completeness
- Proof: search for sequence of applications of inference rules from premises to conclusion

Propositional Inference As Search

- Initial state: Background knowledge and current observations
- Actions: Apply an inference rule to the sentences that match their premises
- Result: Add conclusions of inference rule to knowledge base
- Goal: The knowledge base contains the sentence we want to prove

Inference Rules

$$\frac{\alpha \wedge \beta}{\alpha}$$

And-elimination

$$\frac{\neg\neg\alpha}{\alpha}$$

Double negation

$$\frac{\neg(\alpha \wedge \beta)}{\neg\alpha \vee \neg\beta} \qquad \frac{\neg(\alpha \vee \beta)}{\neg\alpha \wedge \neg\beta}$$

DeMorgan's Laws

$$\frac{\alpha \Rightarrow \beta, \alpha}{\beta}$$

Modus Ponens

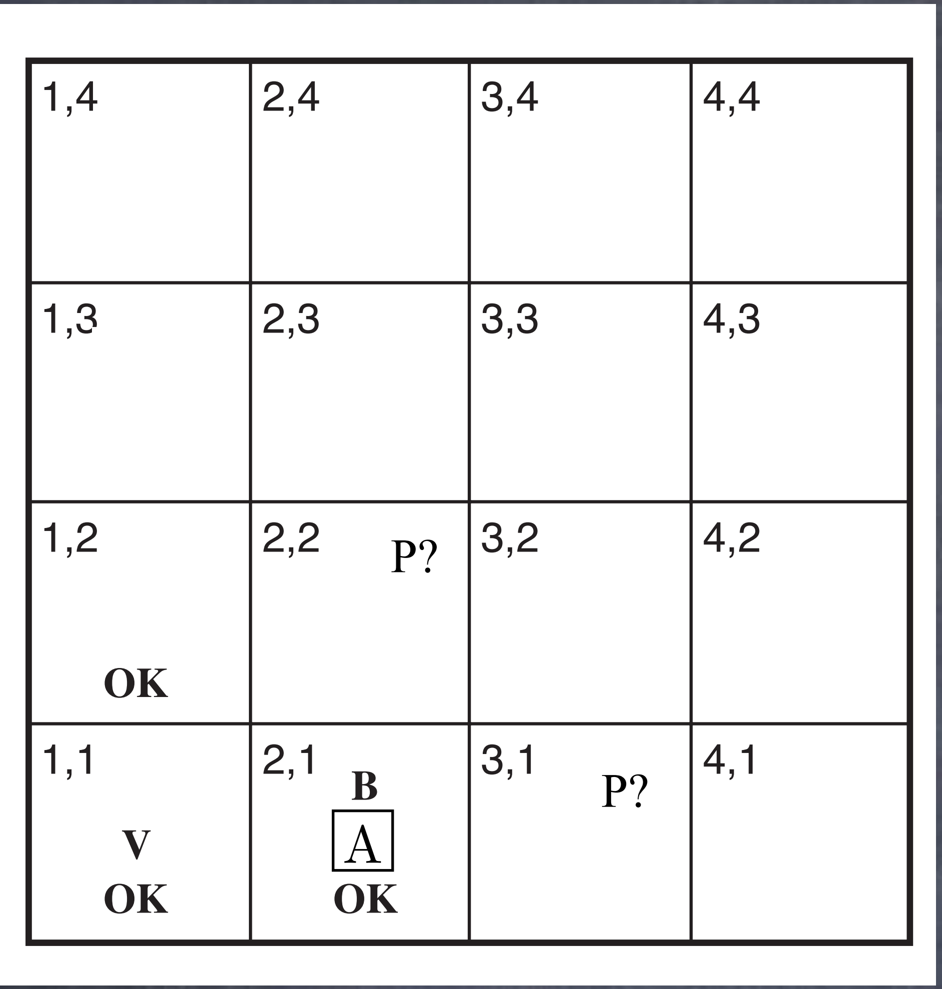
$$\frac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)} \qquad \frac{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta}$$

Definition of biconditional

$$Hungry \lor \ Cranky$$

$$\neg Hungry$$

$$Cranky$$



$$P_{1,1} \lor P_{2,2} \lor P_{3,1}$$

$$\neg P_{1,1}$$

$$P_{2,2} \lor P_{3,1}$$

$$\neg P_{2,2}$$

$$P_{3,1}$$

If A or B is true and you know it's not A,

then it must be B

$$\frac{l_1 \vee \ \dots \ \vee l_i \vee \ \dots \ \vee l_k \qquad\qquad \neg l_i}{l_1 \vee \ \dots \ \vee l_{i\text{-}1} \vee l_{i+1} \vee \ \dots \ \vee l_k}$$

Literal

$$\frac{l_1 \vee \ldots \vee l_i \vee \ldots \vee l_k \qquad \neg l_i}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k}$$

Literal

Complementary

$$\frac{l_1 \vee \ldots \vee l_i \vee \ldots \vee l_k \qquad\qquad \neg l_i}{l_1 \vee \ldots \vee l_{i\text{-}1} \vee l_{i+1} \vee \ldots \vee l_k}$$

Literal

Complementary

$$\frac{l_1 \vee \ldots \vee l_i \vee \ldots \vee l_k \qquad \neg l_i}{\underbrace{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k}_{\text{Clause}}}$$

Unit Resolution

Clause

Unit Clause

$$\frac{l_1 \vee \cdots \vee l_i \vee \cdots \vee l_k, \quad m}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \cdots \vee l_k}$$

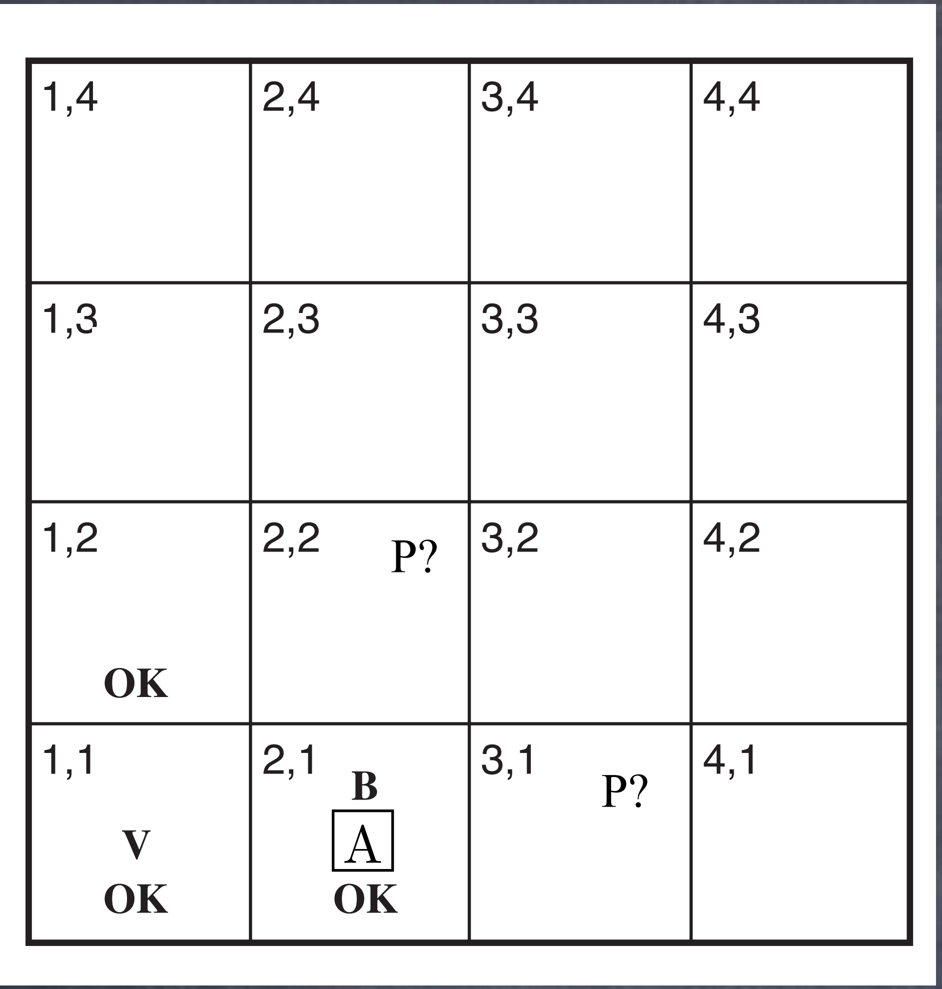
$l_1, \ldots, l_k$ and m are literals

l_i and m are complementary

$$Hungry \vee Cranky$$

$$\neg Hungry$$

$$Cranky$$



$$P_{1,1} \vee P_{2,2} \vee P_{3,1}$$

$$\neg P_{1,1}$$

$$P_{2,2} \vee P_{3,1}$$

$$\neg P_{2,2}$$

$$P_{3,1}$$

Unit Resolution

$$\frac{l_1 \vee \cdots \vee l_i \vee \cdots \vee l_k, \quad m}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \cdots \vee l_k}$$

$l_1, \ldots, l_k$ and m are literals

l_i and m are complementary

Unit Resolution

- Sound: if $\alpha \vdash \beta$ then $\alpha \models \beta$

- Not complete:

  if $\alpha \models \beta$ then $\alpha \vdash \beta$

Resolution

$$\frac{l_1 \vee \cdots \vee l_i \vee \cdots \vee l_k, \quad m_1 \vee \cdots \vee m_j \vee \cdots \vee m_n}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \cdots \vee l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \cdots \vee m_n}$$

$l_1, \ldots, l_k$, $m_1, \ldots, m_n$ are literals

l_i and m_j are complementary

Technical note: Resulting clause must be <u>factored</u> to contain only one copy of each literal.

$$\frac{P_{1,1} \vee P_{3,1}, \quad \neg P_{1,1} \vee \neg P_{2,2}}{P_{3,1} \vee \neg P_{2,2}}$$

1. $Hungry \lor Cranky$

2. $\neg Sleepy \lor \neg Hungry$

3. $Cranky \lor Sleepy$

4. $\neg Sleepy \lor Cranky$ (1,2)

5. $Cranky \lor Cranky$ (3,4)

6. $Cranky$ (factoring)

Resolution

- Sound: if $\alpha \vdash \beta$ then $\alpha \models \beta$
  - Easy to show
- Complete: if $\alpha \models \beta$ then $\alpha \vdash \beta$
  - Proof by contradiction (see book)

Resolution

$$\frac{l_1 \vee \cdots \vee l_i \vee \cdots \vee l_k, \quad m_1 \vee \cdots \vee m_j \vee \cdots \vee m_n}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \cdots \vee l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \cdots \vee m_n}$$

$l_1, \ldots, l_k$, $m_1, \ldots, m_n$ are literals

l_i and m_j are complementary

Technical note: Resulting clause must be <u>factored</u> to contain only one copy of each literal.

$A \wedge B$ $\qquad$ $P \Rightarrow Q$ $\qquad$ $(A \vee B) \Rightarrow \neg(B \vee C)$

Conjunctive Normal Form (CNF)

- Eliminate $\Leftrightarrow$: $\alpha \Leftrightarrow \beta \;\rightarrow\; \alpha \Rightarrow \beta \wedge \beta \Rightarrow \alpha$
- Eliminate $\Rightarrow$: $\alpha \Rightarrow \beta \;\rightarrow\; \neg\alpha \vee \beta$
- Move negation in:
  - $\neg\neg\alpha \;\rightarrow\; \alpha$
  - $\neg(\alpha \vee \beta) \;\rightarrow\; (\neg\alpha \wedge \neg\beta)$
  - $\neg(\alpha \wedge \beta) \;\rightarrow\; (\neg\alpha \vee \neg\beta)$
- Distribute $\vee$ over $\wedge$:
  - $(\alpha \vee (\beta \wedge \gamma)) \rightarrow ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$

$$A \wedge B$$

$$A \wedge B$$

$$A,\ B$$

$$P \Rightarrow Q$$

$$\neg P \vee Q$$

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

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

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

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

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

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

Inference Using Resolution

- Convert sentences (KB) to CNF (set of clauses)
- Apply resolution inference rule to pairs of clauses with complementary literals
- Add resulting clause to set of clauses
- Until...

Proof by Contradiction

- $\alpha \models \beta$ if and only if $(\alpha \wedge \neg\beta)$ is unsatisfiable
- If negation of goal is inconsistent with our knowledge
- Then the goal itself is entailed by our knowledge

Resolution Refutation

- Convert $(KB \wedge \neg\beta)$ to CNF
- Apply resolution rule until:
  - No new clauses can be added
    - $KB \not\models \beta$
  - Two clauses resolve to yield the <u>empty</u> clause (contradiction)

    $KB \models \beta$

1. $Flu \Rightarrow Sneezing$
2. $Cold \Rightarrow Congested$
3. $Congested \Rightarrow Coughing$

1. $Flu \Rightarrow Sneezing$
2. $Cold \Rightarrow Congested$
3. $Congested \Rightarrow Coughing$
4. $Flu \vee Cold$
5. $\neg Sneezing$

1. $Flu \Rightarrow Sneezing$

2. $Cold \Rightarrow Congested$

3. $Congested \Rightarrow Coughing$

4. $Flu \vee Cold$

5. $\neg Sneezing$

To prove: $Coughing$

1. $Flu \Rightarrow Sneezing$

2. $Cold \Rightarrow Congested$

3. $Congested \Rightarrow Coughing$

4. $Flu \vee Cold$

5. $\neg Sneezing$

To prove: $Coughing$

$\neg Flu \vee Sneezing$

$\neg Cold \vee Congested$

$\neg Congested \vee Coughing$

$Flu \vee Cold$

$\neg Sneezing$

1. $\neg Flu \vee Sneezing$
2. $\neg Cold \vee Congested$
3. $\neg Congested \vee Coughing$
4. $Flu \vee Cold$
5. $\neg Sneezing$

To prove: $Coughing$

1. $\neg Flu \vee Sneezing$
2. $\neg Cold \vee Congested$
3. $\neg Congested \vee Coughing$
4. $Flu \vee Cold$
5. $\neg Sneezing$
6. $\neg Coughing$

1. $\neg Flu \vee Sneezing$
2. $\neg Cold \vee Congested$
3. $\neg Congested \vee Coughing$
4. $Flu \vee Cold$
5. $\neg Sneezing$
6. $\neg Coughing$

1 & 4: 7. $Cold \vee Sneezing$

1. $\neg Flu \lor Sneezing$
2. $\neg Cold \lor Congested$
3. $\neg Congested \lor Coughing$
4. $Flu \lor Cold$
5. $\neg Sneezing$
6. $\neg Coughing$

1 & 4: 7. $Cold \lor Sneezing$

7 & 5: 8. $Cold$

1. $\neg Flu \lor Sneezing$
2. $\neg Cold \lor Congested$
3. $\neg Congested \lor Coughing$
4. $Flu \lor Cold$
5. $\neg Sneezing$
6. $\neg Coughing$

1 & 4: 7. $Cold \lor Sneezing$

7 & 5: 8. $Cold$

2 & 3: 9. $\neg Cold \lor Coughing$

9 & 6: 10. $\neg Cold$

1. $\neg Flu \vee Sneezing$
2. $\neg Cold \vee Congested$
3. $\neg Congested \vee Coughing$
4. $Flu \vee Cold$
5. $\neg Sneezing$
6. $\neg Coughing$

1 & 4: 7. $Cold \vee Sneezing$

7 & 5: 8. $Cold$

2 & 3: 9. $\neg Cold \vee Coughing$

9 & 6: 10. $\neg Cold$

8 & 10: $\varnothing$

1. $\neg Flu \vee Sneezing$
2. $\neg Cold \vee Congested$
3. $\neg Congested \vee Coughing$
4. $Flu \vee Cold$
5. $\neg Sneezing$
6. $\neg Coughing$

Unsatisfiable

1 & 4: 7. $Cold \vee Sneezing$

7 & 5: 8. $Cold$

2 & 3: 9. $\neg Cold \vee Coughing$

9 & 6: 10. $\neg Cold$

8 & 10: $\varnothing$

1. $\neg Flu \vee Sneezing$
2. $\neg Cold \vee Congested$
3. $\neg Congested \vee Coughing$
4. $Flu \vee Cold$
5. $\neg Sneezing$
6. $\neg Coughing$

Satisfiable

Unsatisfiable

1 & 4: 7. $Cold \vee Sneezing$

7 & 5: 8. $Cold$

2 & 3: 9. $\neg Cold \vee Coughing$

9 & 6: 10. $\neg Cold$

8 & 10: $\varnothing$

1. $\neg Flu \vee Sneezing$
2. $\neg Cold \vee Congested$
3. $\neg Congested \vee Coughing$
4. $Flu \vee Cold$
5. $\neg Sneezing$
6. $\neg Coughing$

Satisfiable

Unsatisfiable

1 & 4: 7. $Cold \vee Sneezing$

7 & 5: 8. $Cold$

2 & 3: 9. $\neg Cold \vee Coughing$

9 & 6: 10. $\neg Cold$

8 & 10: $\varnothing$

1. $\neg Flu \lor Sneezing$
2. $\neg Cold \lor Congested$
3. $\neg Congested \lor Coughing$
4. $Flu \lor Cold$
5. $\neg Sneezing$
6. $\neg Coughing$

Satisfiable

Unsatisfiable

Proven: $Coughing$

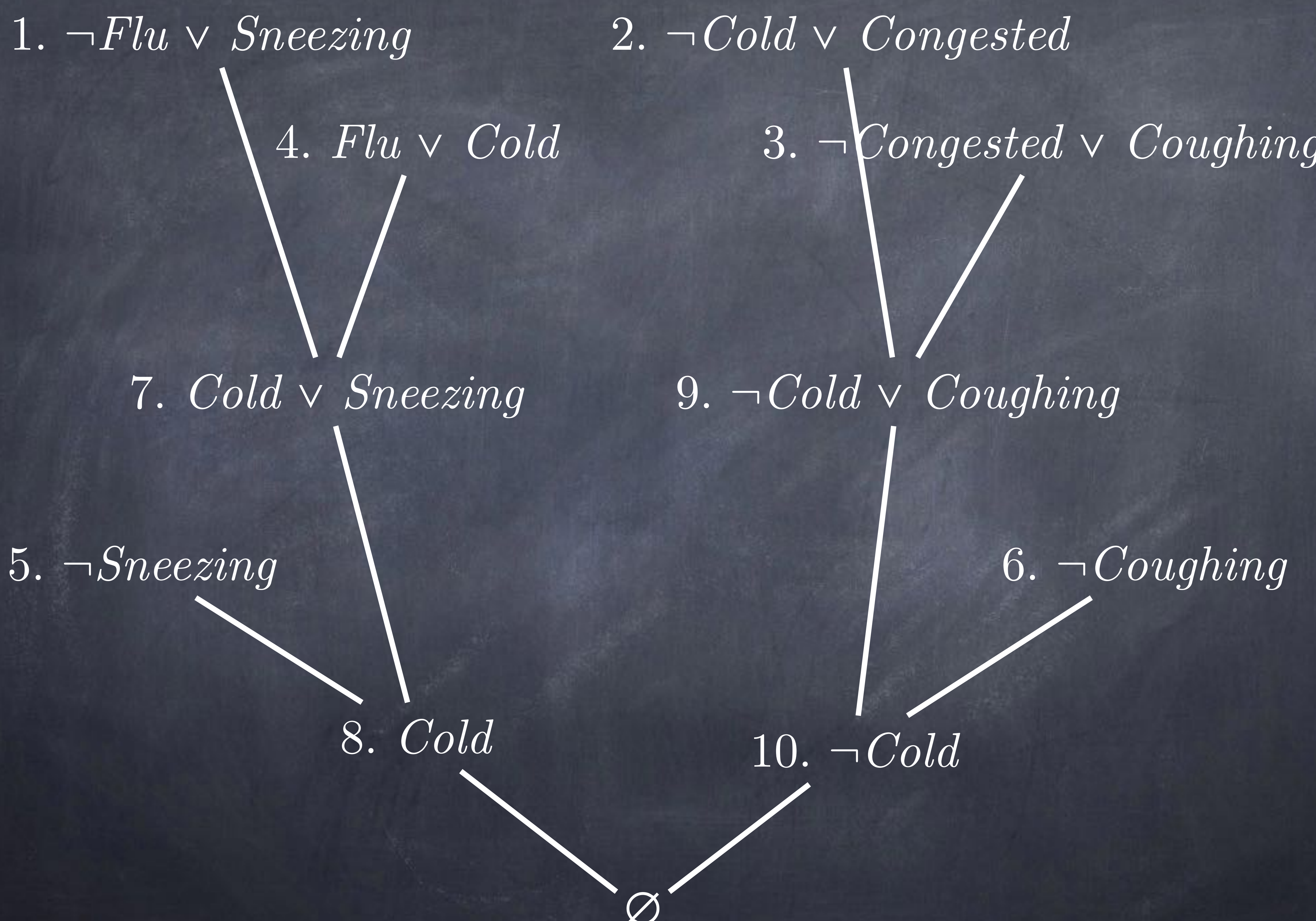
1 & 4: 7. $Cold \lor Sneezing$

7 & 5: 8. $Cold$

2 & 3: 9. $\neg Cold \lor Coughing$

9 & 6: 10. $\neg Cold$

8 & 10: $\varnothing$


1. ¬Flu ∨ Sneezing
2. ¬Cold ∨ Congested
4. Flu ∨ Cold
3. ¬Congested ∨ Coughing
7. Cold ∨ Sneezing
9. ¬Cold ∨ Coughing
5. ¬Sneezing
6. ¬Coughing
8. Cold
10. ¬Cold
∅

Resolution Refutation

- Convert $(KB \wedge \neg\beta)$ to CNF
- Apply resolution rule until:
  - No new clauses can be added
    - $KB \not\models \beta$
  - Two clauses resolve to yield the <u>empty</u> clause (contradiction)

    $KB \models \beta$

Propositional Inference

- Entailment: $\alpha \models \beta$
- Model Checking
  - Intractable (but see AIMA 7.6)
- Inference rules: Soundness, Completeness
- Proof: $\alpha \vdash \beta$
  - Searching for proofs is an alternative to enumerating models
    - May be faster in practice
- Resolution is a sound and complete inference rule
  - Works on clauses (CNF); requires refutation

For next time:

AIMA 8.0–8.3;
8.1.1–8.1.2 fyi