CSC242: Introduction to Artificial Intelligence

Lecture 2.3

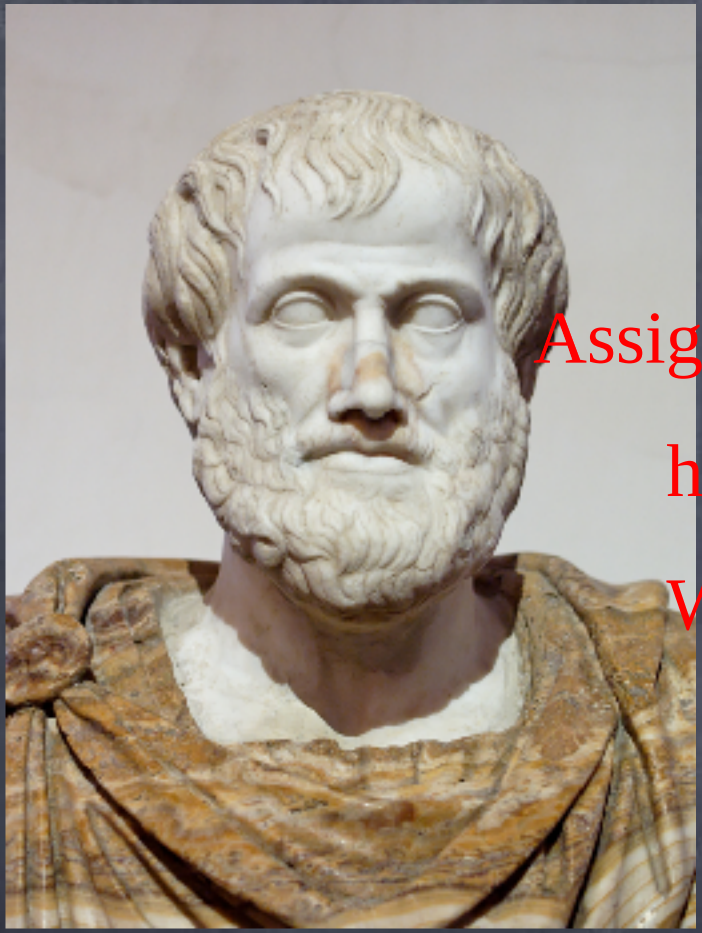
Please put away all electronic devices

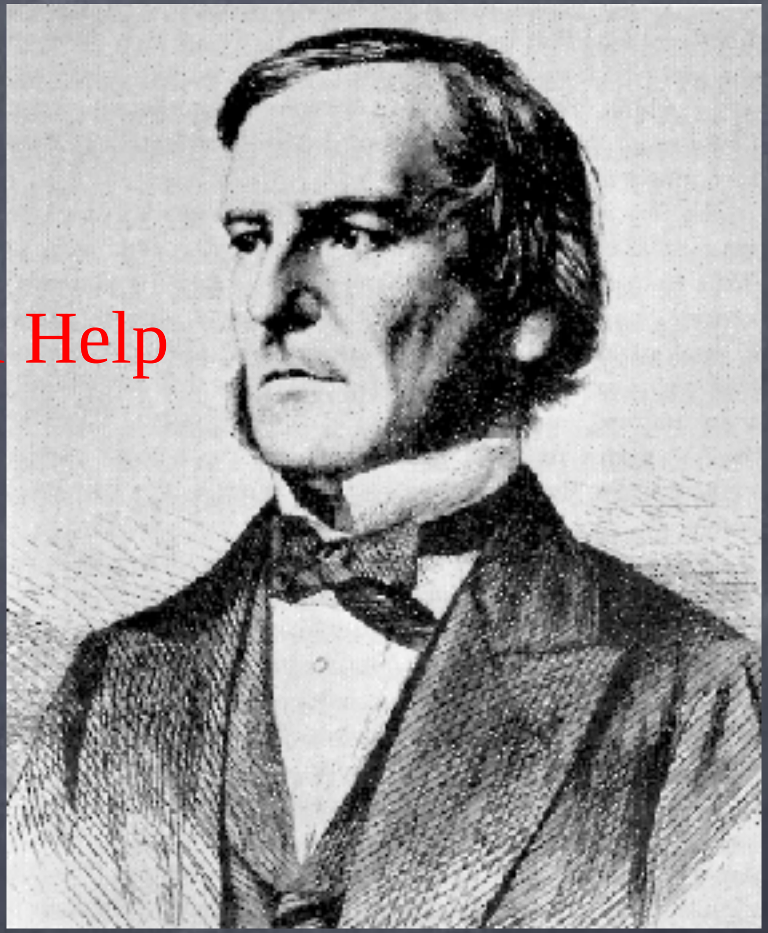
Boolean CSP

- All variables must be Booleans
  - Domains all $\{ \ true, false \ \}$

- Constraints: Identify possible combinations of the boolean variables

Propositional Logic





Aristole

(384BC - 332BC)

George Boole

(1815-1864)

Propositional Logic

- Propositions: things that are true or false
- Connectives: combine propositions into larger propositions
- Sentences: statements about the world (can be true or false)
    - Boolean functions of Boolean variables

Truth Table

$$L_{1,1} \wedge (W_{1,2} \vee W_{2,1})$$

$L_{1,1}$	$W_{1,2}$	$W_{2,1}$	$W_{1,2} \vee W_{2,1}$	$L_{1,1} \wedge (W_{1,2} \vee W_{2,1})$
F	F	F	F	F
F	F	T	T	F
F	T	F	T	F
F	T	T	T	F
T	F	F	F	F
T	F	T	T	T
T	T	F	T	T
T	T	T	T	T

Propositional Logic

- Possible worlds
- Models
- Satisfiability
- Unsatisfiable

Background Knowledge

$B_{1,1} \Leftrightarrow P_{1,2} \vee P_{2,1}$

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

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

...

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

$S_{1,2} \Leftrightarrow W_{1,1} \vee W_{2,2} \vee W_{3,1}$

$S_{2,2} \Leftrightarrow W_{1,2} \vee W_{2,3} \vee W_{3,2} \vee W_{2,1}$

...

$OK_{1,1} \Leftrightarrow \neg(P_{1,1} \vee W_{1,1})$

$OK_{1,2} \Leftrightarrow \neg(P_{1,2} \vee W_{1,2})$

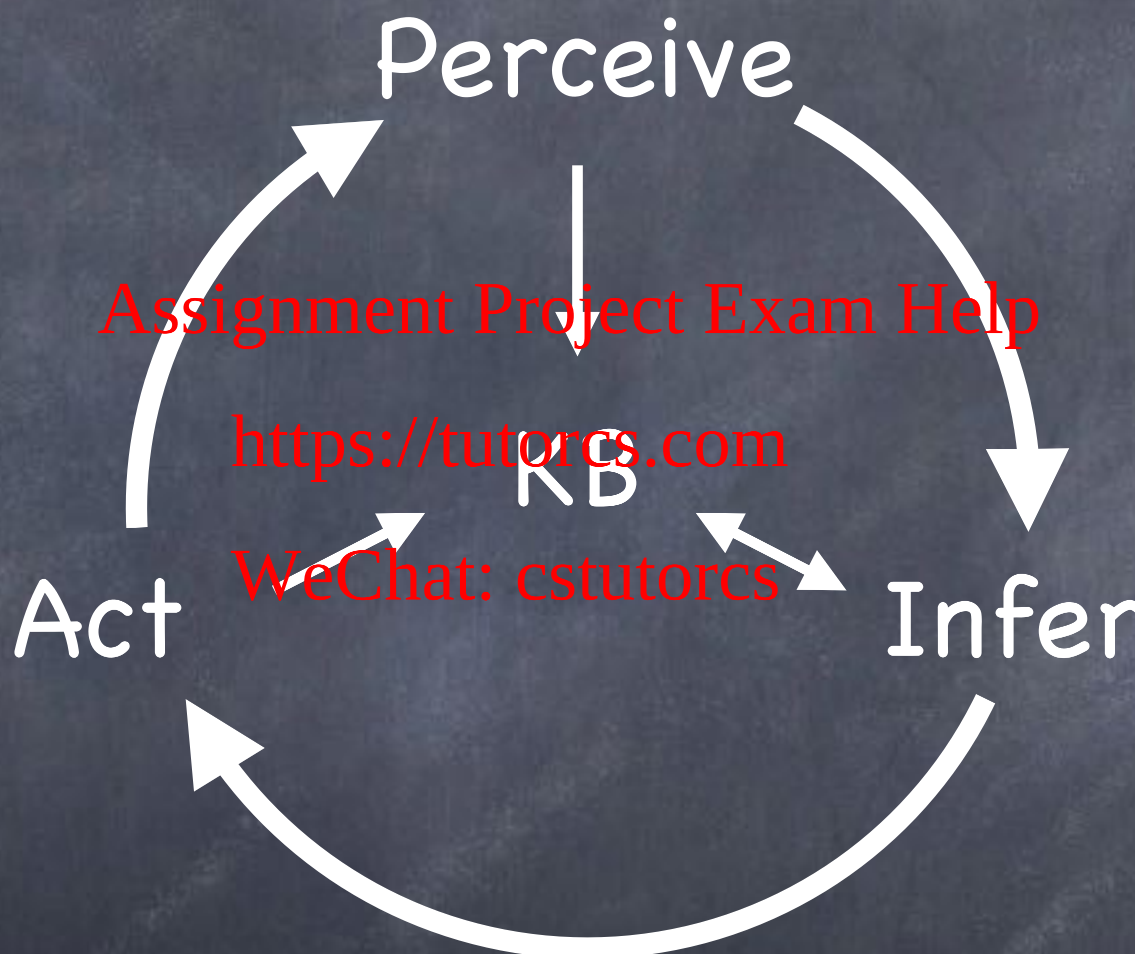
$OK_{2,1} \Leftrightarrow \neg(P_{2,1} \vee W_{2,1})$

...

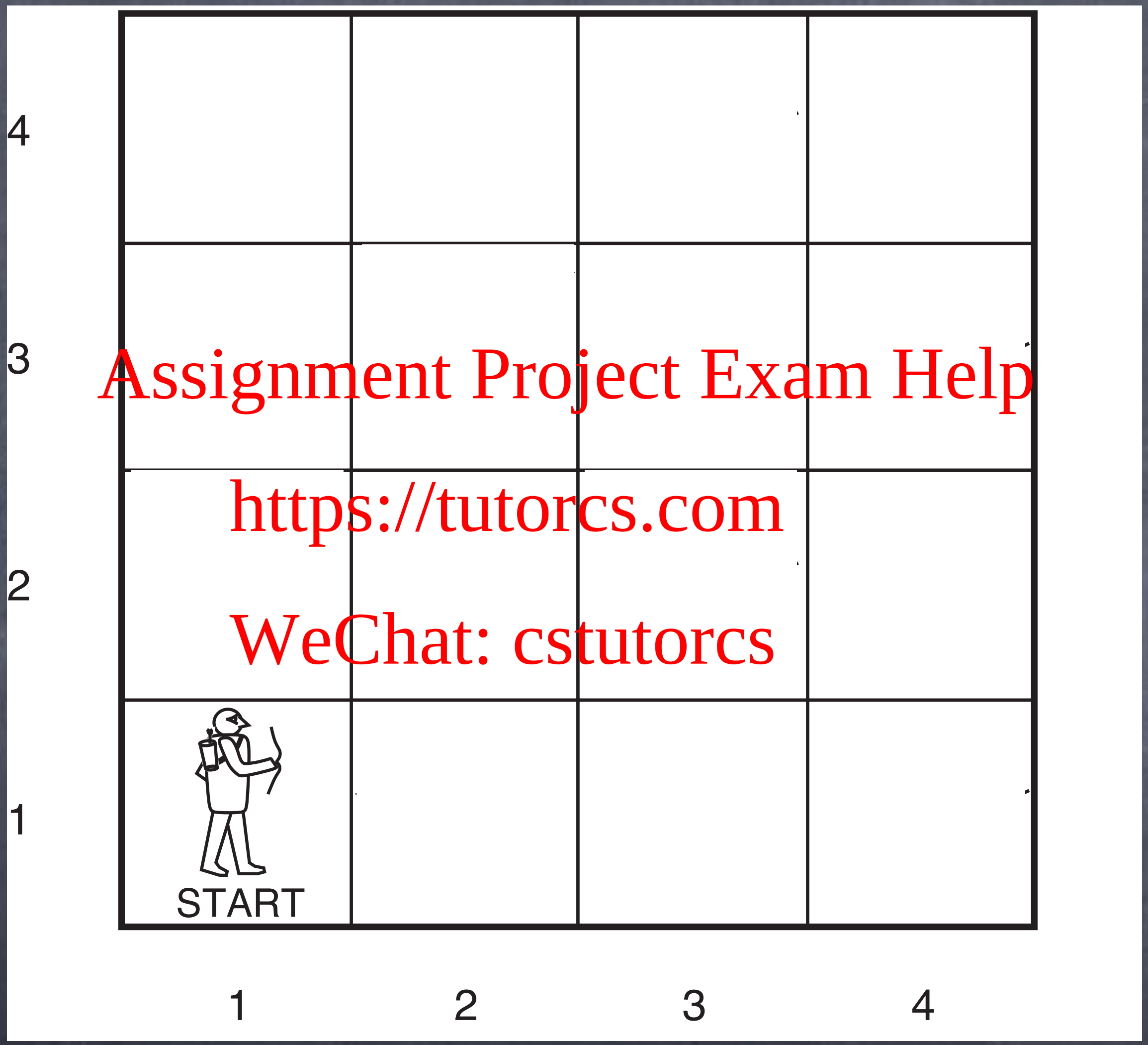
$W_{1,1} \vee W_{1,2} \vee \ldots \vee W_{3,4} \vee W_{4,4}$

$\neg(W_{1,1} \wedge W_{1,2}),\ \neg(W_{1,1} \wedge W_{1,3}),\ \ldots,\ \neg(W_{3,4} \wedge W_{4,4})$

Knowledge-Based Agents



Perception



$$\neg B_{1,1},\ \neg S_{1,1}$$

Inference

- Given what I know...
  - What should I do?

See AIMA 7.7

Inference

Given what I know… Is there no pit in room [2,1]?

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

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

$\neg P_{2,1}$?

$\neg B_{1,1}$

Entailment

- α entails β : $\alpha \vDash \beta$
  - Every model of α is also a model of β
  - Whenever α is true, so is β
  - β is true in every world consistent with α
  - $Models(\alpha) \subseteq Models(\beta)$
  - β logically follows from α

Model Checking

- Given knowledge α and query β
  - For every possible world w
    - If α is satisfied by w
      - If β is not satisfied by w
        - Conclude that $\alpha \not\models \beta$
  - Conclude that $\alpha \models \beta$

Model Checking

- Given knowledge α and query β
  - For every possible world w
    - If α is satisfied by w
      - If β is not satisfied by w
        - Conclude that $\alpha \not\models \beta$
  - Conclude that $\alpha \models \beta$

AIMA Fig. 7.10

Model Checking

Possible Worlds

$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$B_{1,1}$				
F	F	F	F				
F	F	F	T				
F	F	T	F				
...							
T	T	F	T				
T	T	T	F				
T	T	T	T				

Model Checking

Knowledge

$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$B_{1,1}$	R_1	R_2	$\neg B_{1,1}$	$\neg P_{2,1}$
F	F	F	F	T	T	T	
F	F	F	T	T	F	F	
F	F	T	F	T	F	T	
$\ldots$				$\ldots$			
T	T	F	T	F	T	F	
T	T	T	F	F	F	T	
T	T	T	T	F	T	F	

Model Checking

Knowledge

$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$B_{1,1}$	R_1	R_2	$\neg B_{1,1}$	$\neg P_{2,1}$
F	F	F	F	T	T	T	
F	F	F	T	T	F	F	
F	F	T	T	T	F	T	
…				…			
T	T	F	T	F	T	F	
T	T	T	F	F	F	T	
T	T	T	T	F	T	F	

Model Checking

Knowledge

$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$B_{1,1}$	R_1	R_2	$\neg B_{1,1}$	$\neg P_{2,1}$
F	F	F	F	T	T	T	
F	F	F	T	T	F	F	
F	F	T	T	T	F	T	
$\dots$				$\dots$			
T	T	F	T	F	T	F	
T	T	T	F	F	F	T	
T	T	T	T	F	T	F	

Model Checking

Query

$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$B_{1,1}$	R_1	R_2	$\neg B_{1,1}$	$\neg P_{2,1}$
F	F	F	F	T	T	T	T
F	F	F	T	T	F	F	
F	F	T	T	T	F	T	
...				...			
T	T	F	T	F	T	F	
T	T	T	F	F	F	T	
T	T	T	T	F	T	F	

Entailment

Given what I know… Is there no pit in room [2,1]?

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

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

$\neg B_{1,1}$

$\neg P_{2,1}$?

$$KB \vDash \neg P_{2,1}$$

Entailment

Given what I know... Is there no pit in room [1,2]?

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

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

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

$\neg B_{1,1}$

$B_{2,1}$

$\neg P_{1,2}$?

$KB \models \neg P_{1,2}$

$KB \not\models P_{2,2}$

$KB \not\models \neg P_{2,2}$

But...

Model Checking

- Given knowledge α and query β
  - For every possible world w
    - If α is satisfied by w
      - If β is not satisfied by w
        - Conclude that $\alpha \not\models \beta$
  - Conclude that $\alpha \models \beta$

Model Checking

n propositions m sentences, $O(k)$ connectives

$P_{1,1}$	$P_{1,2}$	$\cdots$	$OK_{1,1}$	$OK_{2,1}$	R_1	R_2	$\cdots$	$\neg B_{1,1}$	$\neg S_{1,1}$	$OK_{2,1}$
F	F	...	F	F	F	...	...	...	T	
...					...					
F	F	...	T	T	T	T	...	...	T	T
...					...					
T	T	...	F	T	F	...	...	...	F	
T	T	...	T	F	T	F	...	...	T	
T	T	...	T	T	T	T	...	...	F	

Model Checking

n propositions m sentences, $O(k)$ connectives

$P_{1,1}$	$P_{1,2}$	$\cdots$	$OK_{1,1}$	$OK_{2,1}$	R_1	R_2	$\cdots$	$\neg B_{1,1}$	$\neg S_{1,1}$	$OK_{2,1}$
F	F	...	F	F	$\boxed{\text{F}}$	...	...	...	T	
...					...					
F	F	...	T	T	T	T	...	...	T	T
...					...					
T	T	...	F	T	F	...	...	...	F	
T	T	...	T	F	T	F	...	...	T	
T	T	...	T	T	T	T	...	...	F	

$O(k)$

Model Checking

n propositions $\quad m$ sentences, $O(k)$ connectives

$P_{1,1}$	$P_{1,2}$	$\cdots$	$OK_{1,1}$	$OK_{2,1}$	R_1	R_2	$\cdots$	$\neg B_{1,1}$	$\neg S_{1,1}$	$OK_{2,1}$
F	F	...	F	F	F	...	...	...	T	
...					...					
F	F	...	T	T	T	T	...	...	T	T
...					...					
T	T	...	F	T	F	...	...	...	F	
T	T	...	T	F	T	F	...	...	T	
T	T	...	T	T	T	T	...	...	F	

$O(mk)$

Model Checking

n propositions $\quad m$ sentences, $O(k)$ connectives

$P_{1,1}$	$P_{1,2}$	$\cdots$	$OK_{1,1}$	$OK_{2,1}$	R_1	R_2	$\cdots$	$\neg B_{1,1}$	$\neg S_{1,1}$	$OK_{2,1}$
F	F	…	F	F	F	…	…	…	T	
…					…					
F	F	…	T	T	T	T	…	…	T	T
…					…					
T	T	…	F	T	F	…	…	…	F	
T	T	…	T	F	T	F	…	…	T	
T	T	…	T	T	T	T	…	…	F	

$O(2^n)$

Model Checking

n propositions m sentences, $O(k)$ connectives

$P_{1,1}$	$P_{1,2}$	$\cdots$	$OK_{1,1}$	$OK_{2,1}$	R_1	R_2	$\cdots$	$\neg B_{1,1}$	$\neg S_{1,1}$	$OK_{2,1}$
F	F	...	F	F	F	...	...	...	T	
...					...					
F	F	...	T	T	T	T	...	...	T	T
...					...					
T	T	...	F	T	F	...	...	...	F	
T	T	...	T	F	T	F	...	...	T	
T	T	...	T	T	T	T	...	...	F	

$O(2^n mk)$ Intractable!

Entailment

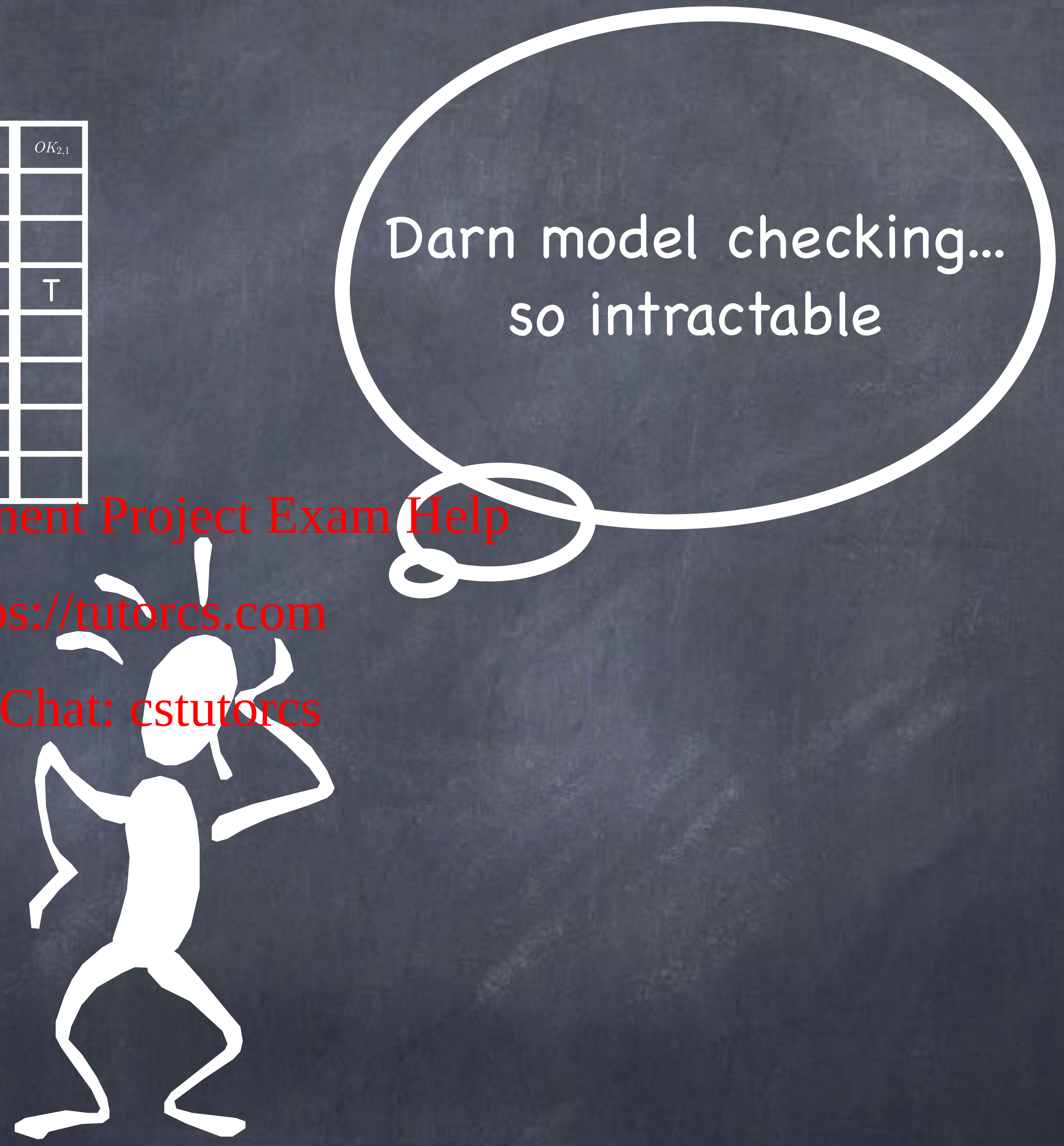
- α entails β : $\alpha \models \beta$
  - Every model of α is also a model of β
  - Whenever α is true, so is β
  - β is true in every world consistent with α
  - $Models(\alpha) \subseteq Models(\beta)$
  - β logically follows from α

co-NP-complete!

Propositional Logic

- Programming language for knowledge
- Factored representation (Boolean CSP)
  - Propositions, connectives, sentences
- Possible worlds, satisfiability, models
- Entailment: $\alpha \vDash \beta$
  - Every model of α is a model of β
  - Model checking intractable!

$P_{1,1}$	$P_{1,2}$	$\cdots$	$OK_{1,1}$	$OK_{2,1}$	R_1	R_2	$\cdots$	$\neg B_1,$	$\neg S_1,$	$OK_{2,1}$
F	F	…	F	F	F	…	…	…	T	
…					…					
F	F	…	T	T	T	T	…	…	T	T
…					…					
T	T	…	F	T	F	…	…	…	F	
T	T	…	T	F	T	F	…	…	T	
T	T	…	T	T	T	T	…	…	F	



Rule: If you know α, then you also know β.

Rule: If you know α, then you also know β.

No model checking!

Rule: If you know α, then you also know β.

No model checking!

Seems impossible...

$$KB = \{\ Hungry \Rightarrow\ Cranky,$$
$$Hungry\ \}$$

$$KB = \{\ Hungry \Rightarrow Cranky,$$
$$Hungry\ \}$$

$$Cranky$$

$$KB = \{\ Hungry \Rightarrow Cranky,$$
$$Hungry\ \}$$

$$Cranky$$

$Hungry$	$Cranky$	$Hungry \Rightarrow Cranky$
F	F	T
F	T	T
T	F	F
T	T	T

$KB = \{\ Hungry \Rightarrow\ Cranky,$

$Hungry\ \}$

$Cranky$

$Hungry$	$Cranky$	$Hungry \Rightarrow\ Cranky$
F	F	T
F	T	T
T	F	F
T	T	T

$$KB = \{\ Hungry \Rightarrow\ Cranky,$$
$$Hungry\ \}$$

$$Cranky$$

$Hungry$	$Cranky$	$Hungry \Rightarrow\ Cranky$
F	F	T
F	T	T
T	F	F
T	T	T

$KB = \{ Hungry \Rightarrow Cranky,$

$Hungry \}$

$Cranky$

$KB \models Cranky$

$Hungry$	$Cranky$	$Hungry \Rightarrow Cranky$
F	F	T
F	T	T
T	F	F
T	T	T

Inference Rule

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

Inference Rule

Premises
(Antecedents)

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

Inference Rule

Premises
(Antecedents)

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

Conclusions
(Consequents)

Inference Rule

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

$Hungry$	$Cranky$	$Hungry \Rightarrow Cranky$
F	F	T
F	T	T
T	F	F
T	T	T

Inference Rule

$$\frac{\varphi \Rightarrow \psi,\ \varphi}{\psi}$$

φ	ψ	$\varphi \Rightarrow \psi$
F	F	T
F	T	T
T	F	F
T	T	T

Modus Ponens

"mode that affirms"

$$\text{MP: } \frac{\varphi \Rightarrow \psi,\ \varphi}{\psi}$$

Derivation

$$\{ \; Hungry \Rightarrow Cranky, \; Hungry \; \} \vdash_{\mathrm{MP}} Cranky$$

Derivation

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

Derivation

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

$$\underbrace{\alpha \vdash \beta \vdash \gamma \vdash \ldots \vdash \omega}_{\alpha \vdash \omega}$$

$\alpha \models \beta$

β logically follows from α

Intractable to compute

$\alpha \vdash \beta$

Easier to compute

But does α really follow from β ?

Soundness

If $\alpha \vdash \beta$ then $\alpha \vDash \beta$

Soundness

If $\alpha \vdash \beta$ then $\alpha \models \beta$

Modus Ponens is sound

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

$$\frac{\varphi \Rightarrow \psi,\ \psi}{\varphi}$$

$$\frac{Hungry \Rightarrow Cranky,\ Cranky}{Hungry} \qquad \frac{\varphi \Rightarrow \psi,\ \psi}{\varphi}$$

φ	ψ	$\varphi \Rightarrow \psi$
F	F	T
F	T	T
T	F	F
T	T	T

$$\frac{Hungry \Rightarrow Cranky,\ Cranky}{Hungry} \qquad \frac{\varphi \Rightarrow \psi,\ \psi}{\varphi}$$

φ	ψ	$\varphi \Rightarrow \psi$
F	F	T
F	T	T
T	F	F
T	T	T

$$\frac{Hungry \Rightarrow Cranky,\ Cranky}{Hungry} \qquad \frac{\varphi \Rightarrow \psi,\ \psi}{\varphi}$$

$$\{\ \varphi \Rightarrow \psi,\ \psi\ \} \not\models \varphi$$

φ	ψ	$\varphi \Rightarrow \psi$
F	F	T
F	T	T
T	F	F
T	T	T

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

$$\frac{\varphi \Rightarrow \psi,\ \psi}{\varphi}$$

Unsound!

$$\{\ \varphi \Rightarrow \psi,\ \psi\ \} \not\models \varphi$$

φ	ψ	$\varphi \Rightarrow \psi$
F	F	T
F	T	T
T	F	F
T	T	T

Affirming the Consequent

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

$$\frac{\varphi \Rightarrow \psi,\ \psi}{\varphi}$$

Unsound!

$$\{\ \varphi \Rightarrow \psi,\ \psi\ \} \not\models \varphi$$

φ	ψ	$\varphi \Rightarrow \psi$
F	F	T
F	T	T
T	F	F
T	T	T

$$\frac{Hungry \Rightarrow Cranky, \neg Cranky}{\neg Hungry} \qquad \frac{\varphi \Rightarrow \psi, \neg\psi}{\neg\varphi}$$

$$\frac{Hungry \Rightarrow Cranky,\ \neg Cranky}{\neg Hungry} \qquad \frac{\varphi \Rightarrow \psi,\ \neg \psi}{\neg \varphi}$$

φ	ψ	$\varphi \Rightarrow \psi$	$\neg\varphi$	$\neg\psi$
F	F	T	T	T
F	T	T	T	F
T	F	F	F	T
T	T	T	F	F

$$\frac{Hungry \Rightarrow Cranky, \neg Cranky}{\neg Hungry} \qquad \frac{\varphi \Rightarrow \psi, \neg\psi}{\neg\varphi}$$

φ	ψ	$\varphi \Rightarrow \psi$	$\neg\varphi$	$\neg\psi$
F	F	T	T	T
F	T	T	T	F
T	F	F	F	T
T	T	T	F	F

$$\frac{Hungry \Rightarrow Cranky, \neg Cranky}{\neg Hungry} \qquad \frac{\varphi \Rightarrow \psi, \neg\psi}{\neg\varphi}$$

φ	ψ	$\varphi \Rightarrow \psi$	$\neg\varphi$	$\neg\psi$
F	F	T	T	T
F	T	T	T	F
T	F	F	F	T
T	T	T	F	F

$$\frac{Hungry \Rightarrow Cranky, \neg Cranky}{\neg Hungry}$$

$$\frac{\varphi \Rightarrow \psi, \neg\psi}{\neg\varphi}$$

Sound!

$$\{\ \varphi \Rightarrow \psi, \neg\psi\ \} \models \neg\varphi$$

φ	ψ	$\varphi \Rightarrow \psi$	$\neg\varphi$	$\neg\psi$
F	F	T	T	T
F	T	T	T	F
T	F	F	F	T
T	T	T	F	F

Modus Tollens

$$\text{MT:}\ \frac{Hungry \Rightarrow Cranky,\ \neg Cranky}{\neg Hungry} \qquad \text{MT:}\ \frac{\varphi \Rightarrow \psi,\ \neg\psi}{\neg\varphi}$$

Sound!

$$\{\ \varphi \Rightarrow \psi,\ \neg\psi\ \} \models \neg\varphi$$

φ	ψ	$\varphi \Rightarrow \psi$	$\neg\varphi$	$\neg\psi$
F	F	T	T	T
F	T	T	T	F
T	F	F	F	T
T	T	T	F	F

Equivalences

For any sentences φ and ψ

If $\varphi \equiv \psi$ then

$$\frac{\varphi}{\psi} \quad \text{and} \quad \frac{\psi}{\varphi}$$

are inference rules

Equivalences

AIMA Fig. 7.11

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	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 $\wedge$
$((\alpha \vee \beta) \wedge \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	Associativity of $\vee$
$\neg(\neg\alpha) \equiv \alpha$	Double-negation elimination
$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$	Contrapositive
$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$	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's law
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	De Morgan's law
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	Distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	Distributivity of $\vee$ over $\wedge$

Inference Rules

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

And-elimination

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

Double negation

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

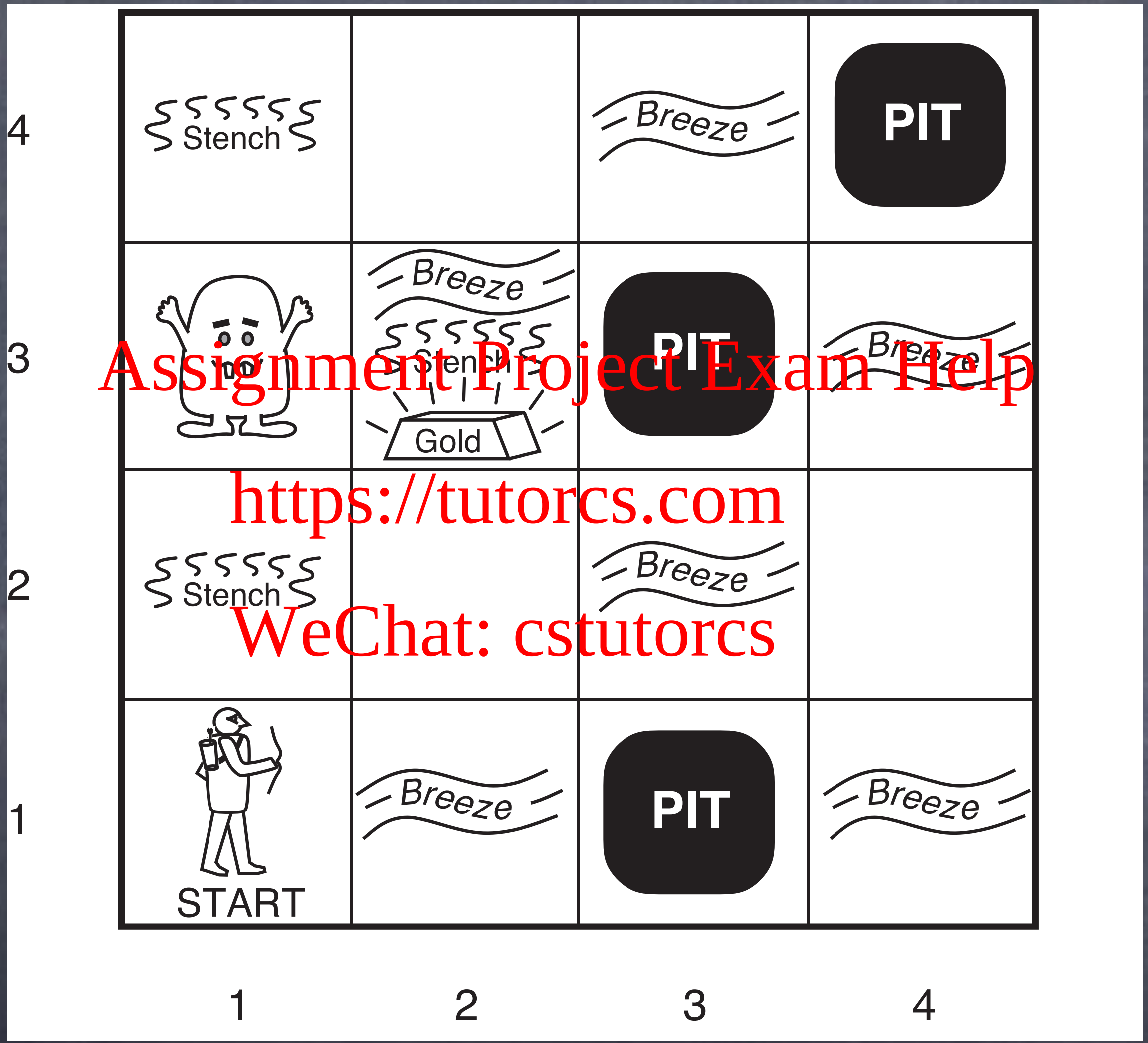
DeMorgan's Laws

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

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


4
3
2
1
Stench
Breeze
PIT
Breeze
Stench
Gold
PIT
Breeze
Stench
Breeze
START
Breeze
PIT
Breeze
1
2
3
4

Proof

- Inference rules produce theorems derived from other sentences
- The sequence of inference rule applications used in the derivation constitutes a proof of the theorem

Proof

Given what I know... Is there no pit in room [1,2]?

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

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

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

$4{:}\ \neg B_{1,1}$

$5{:}\ B_{2,1}$

$\neg P_{1,2}?$

1: $\neg P_{1,1}$
2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
4: $\neg B_{1,1}$
5: $B_{2,1}$

Rule	Premises	Conclusion
Bicond elim	2	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 elim	6	7: $((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
Contrapositive	7	8: $(\neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1}))$
MP	8, 4	9: $\neg(P_{1,2} \vee P_{2,1})$
De Morgan	9	10: $\neg P_{1,2} \wedge \neg P_{2,1}$
And elim	10	11: $\neg P_{1,2}$

Proof

- Each step's premises must be in the KB already

- Each step's conclusion is added to the KB

- The last step derives the query

$$KB \vdash \neg P_{1,2}$$

Proof

- Each step's premises must be in the KB already

- Each step's conclusion is added to the KB

- The last step derives the query

$$KB \vdash \neg P_{1,2}$$

- If all the inference rules are sound

$$KB \models \neg P_{1,2}$$

Proof as Search

- States are sets of sentences (KBs)
- Actions are applying inference rules
  - $Actions(s) = \{\ r_m \mid Match(Premises(r),\ KB){=}m\ \}$
  - $Result(r_m,\ s) = s \cup Subst(m,\ Conclusions(r))$
- Initial state: initial KB
- Goal test: $query \in KB$

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."

Theorem Proving

- States are sets of sentences (KBs)
- Actions are applying inference rules

  - $Actions(s) = \{\ r_m \mid Match(Premises(r),\ KB)=m\ \}$

  - $Result(r_m,\ s) = s \cup Subst(m,\ Conclusions(r))$
- Initial state: initial KB
- Goal test: $query \in KB$

Theorem Proving

- Need a complete search strategy

Theorem Proving

- Need a complete search strategy
- Need a complete set of inference rules

Completeness

If $\alpha \models \beta$ then $\alpha \vdash \beta$

Completeness

If $\alpha \models \beta$ Then $\alpha \vdash \beta$

$$\mathrm{MP}: \frac{\varphi \Rightarrow \psi, \ \varphi}{\psi}$$

Modus Ponens is not complete

Proof

Rule	Premises	Conclusion
Bicond elim	2	6: $(B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1})$
And elim	6	7: $((P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1})$
Contrapositive	7	8: $(\neg B_{1,1} \Rightarrow \neg(P_{1,2} \lor P_{2,1}))$
MP	8, 4	9: $\neg(P_{1,2} \lor P_{2,1})$
De Morgan	9	10: $\neg P_{1,2} \land \neg P_{2,1}$
And elim	10	11: $\neg P_{1,2}$

Theorem Proving

- Need a complete search strategy
- Need a complete set of inference rules

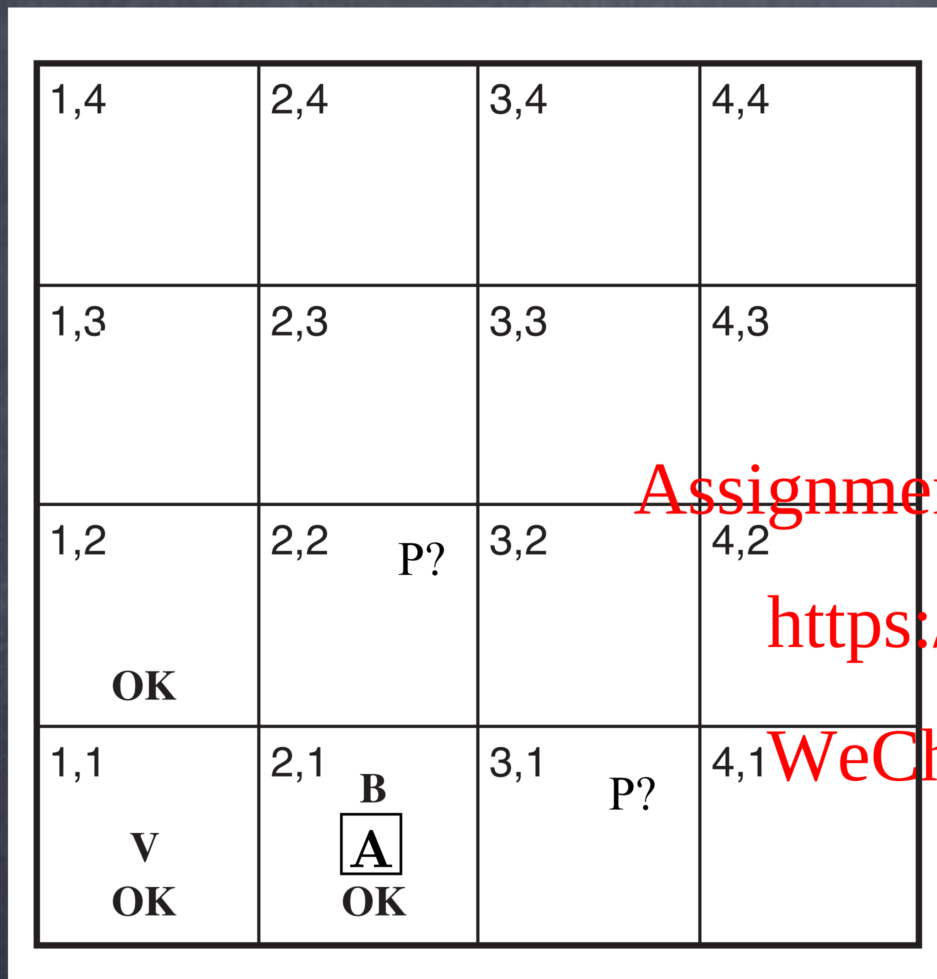
Theorem Proving

- Need a complete search strategy
- Need a complete set of inference rules
  - Or a single complete inference rule

$$Hungry \lor Cranky$$

$$\neg Hungry$$

$$Cranky$$



$$B_{2,1}$$

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

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

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

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

$$P_{3,1}$$

Reasoning By Cases

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

then it must be B

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

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

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

l_i is gone

Literals

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

Literals

Complementary

$m = \neg l_i$
or
$l_i = \neg m$

$$\frac{l_1 \lor \ \ldots \ \lor l_i \lor \ \ldots \ \lor l_k \qquad\qquad m}{l_1 \lor \ \ldots \ \lor l_{i-1} \lor l_{i+1} \lor \ \ldots \ \lor l_k}$$

Literal

Complementary

$m = \neg l_i$
or
$l_i = \neg m$

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

Unit Resolution

Clause

Unit Clause

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

l_1, …, l_k and m are literals

l_i and m are complementary

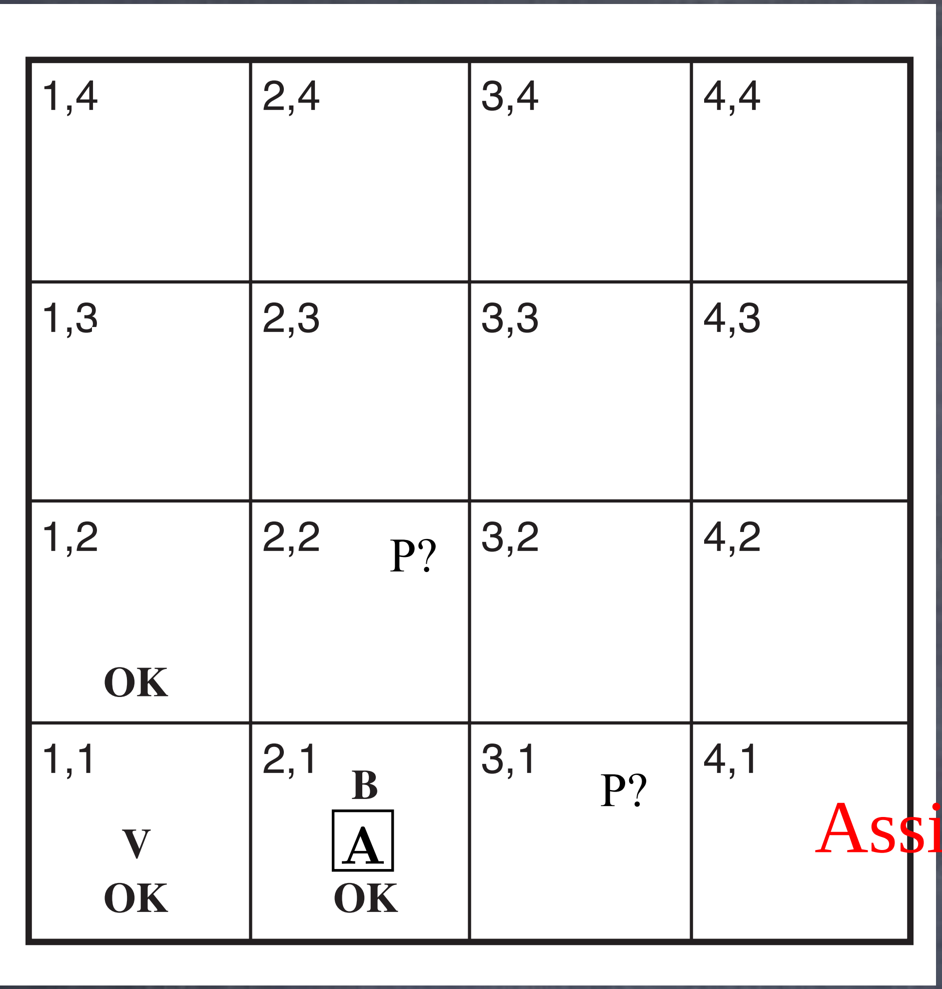
1. $Hungry \lor Cranky$

2. $\neg Hungry$

Q: $Cranky?$

Unit Res: $Hungry$	1,2	3. $Cranky$

$$KB \vdash_{UR} Cranky$$



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

Q: $P_{3,1}$?

Unit Res: $P_{1,1}$	1,2	4. $P_{2,2} \vee P_{3,1}$
Unit Res: $P_{2,2}$	4,3	5. $P_{3,1}$

$$KB \vdash_{UR} \mathrm{P}_{3,1}$$

Unit Resolution

- Sound: if $\alpha \vdash \beta$ then $\alpha \models \beta$
  - Easy to show
- Not complete: if $\alpha \models \beta$ then $\alpha \vdash \beta$
  - Give a counterexample

$$Hungry \lor Cranky$$

$$\neg Hungry \lor Sleepy$$

$$Cranky \lor Sleepy$$

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

l_i and m_j are gone

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

1. $Hungry \vee Cranky$
2. $\neg Sleepy \vee \neg Hungry$
3. $Cranky \vee Sleepy$

Rule	Premises	Conclusion
Resolution: $Hungry$	1,2	4. $Cranky \vee \neg Sleepy$
Resolution: $Sleepy$	3,4	5. $Cranky \vee Cranky$
Factoring	5	6. $Cranky$

$$KB \vdash Cranky$$

Resolution

- Sound: if $\alpha \vdash \beta$ then $\alpha \models \beta$ ✓
  - Easy to show

Resolution

- Sound: if $\alpha \vdash \beta$ then $\alpha \models \beta$ ✔
  - Easy to show
- Not complete: if $\alpha \models \beta$ then $\alpha \vdash \beta$ ✘
  - Give a counterexample

Resolution is Refutation-Complete

- If a set of clauses is unsatisfiable, then resolution can derive the empty clause (□)

AIMA p. 255

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.
(See AIMA)

Challenges for Using Resolution

- Only works on clauses

- Only refutation-complete

Conjunctive Normal Form (CNF)

- Any sentence of propositional logic can be converted into an equivalent conjunction (set) of clauses

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 $\wedge$:
  - $(\alpha \vee (\beta \wedge \gamma)) \rightarrow ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$

AIMA p. 253-254

Challenges for Using Resolution

- Only works on clauses

  - Convert KB & query to clauses (CNF)

- Only refutation-complete

Challenges for Using Resolution

- Only works on clauses

  - Convert KB & query to clauses (CNF)

- Only refutation-complete

Resolution is Refutation-Complete

- If a set of clauses is unsatisfiable, then resolution can derive the empty clause (□)

a clause with zero literals

Entailment and Satisfiability

- If a set of clauses is unsatisfiable, then resolution can derive the empty clause (□)

- $KB \vDash \beta$
  - iff every model of KB is a model of β
  - iff no model of KB is a model of $\neg\beta$
  - iff there are no models of $KB \cup \{ \neg\beta \}$
  - iff $KB \cup \{ \neg\beta \}$ is unsatisfiable

Resolution Refutation

- Convert $KB \cup \{ \neg\beta \}$ 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)
    - $KB \models \beta$

Resolution Refutation

Given what I know... Is there no pit in room [1,2]?

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

$\neg P_{1,2}$?

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

Resolution Refutation

$\mathrm{KB} = \{\ B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1}),\ \neg B_{1,1}\ \}$

Query: $\neg P_{1,2}$?

Add ¬Query to KB

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

Convert to CNF

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

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

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

$\neg B_{1,1} \vee (P_{1,2} \vee P_{2,1}),\ (\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1},\ \neg B_{1,1},\ P_{1,2}$

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

$\neg B_{1,1} \vee P_{1,2} \vee P_{2,1},\ \neg P_{1,2} \vee B_{1,1},\ \neg P_{2,1} \vee B_{1,1},\ \neg B_{1,1},\ P_{1,2}$

Resolution Refutation

1: $\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$

2: $\neg P_{1,2} \vee B_{1,1}$

3: $\neg P_{2,1} \vee B_{1,1}$

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

5: $P_{1,2}$

Premises	Literal	Result
2,4	$B_{1,1}$	6: $\neg P_{1,2}$
5,6	$P_{1,2}$	7: $\square$

Resolution Refutation

$1{:}\ \neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$

$2{:}\ \neg P_{1,2} \vee B_{1,1}$

$3{:}\ \neg P_{2,1} \vee B_{1,1}$

$4{:}\ \neg B_{1,1}$

$5{:}\ P_{1,2}$

$$
\begin{tikzcd}
2{:}\ \neg P_{1,2} \vee B_{1,1} \arrow[dr] & & 4{:}\ \neg B_{1,1} \arrow[dl] & \\
& 6{:}\ \neg P_{1,2} \arrow[dr] & & 5{:}\ P_{1,2} \arrow[dl] \\
& & 7{:}\ \Box &
\end{tikzcd}
$$

Resolution Refutation

- Convert $KB \cup \{ \neg\beta \}$ 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)
    - $KB \models \beta$

Resolution

- Complete when used in a refutation (proof by contradiction)
- Search challenges remaining:
  - Which clauses to resolve?
  - On which complementary literals?

Resolution

- Complete when used in a refutation (proof by contradiction)

- Search challenges remaining:
  - Which clauses to resolve?
  - On which complementary literals?

Intractable!

But wait there's more...

Theorem Proving

- States are sets of sentences (KBs)
- Actions are applying inference rules
  - $Actions(s) = \{\, r_m \mid Match(Premises(r),\, KB) = m \,\}$
  - $Result(r_m,\, s) = s \cup Subst(m,\, Conclusions(r))$
- Initial state: initial KB
- Goal test: $query \in KB$

Inference for Knowledge-Based Agents

- Data-driven (forward chaining)

- Goal-directed (backward chaining)

Forward Chaining

- Given new fact φ (often perception)
  - Add φ to agenda
  - While agenda is not empty
    - Remove sentence α from agenda
    - Add α to KB
    - Apply inference using only rules whose premises include α
      - Add conclusion β to agenda

Forward Chaining

- Reasons forward from new facts
  - Data-driven
- Done by humans to some extent
  - When to stop?

Forward Chaining

- Reasons forward from new facts
  - Data-driven
- Done by humans—to some extent
  - When to stop?
- For KBs using only definite clauses
  - Sound, complete, linear time

AIMA 7.5.4

Backward Chaining

- In order to prove β
    - Find an implication whose conclusion is β
    - Try to prove its premises α (recursively)

Backward Chaining

- Reasons backward from query
  - Goal-directed
- Useful for answering specific questions

Backward Chaining

- Reasons backward from query
  - Goal-directed
- Useful for answering specific questions
- For KBs using only definite clauses
  - Sound, complete, linear time

AIMA 7.5.4

Propositional Theorem Proving

- Inference rules: Soundness, Completeness
- Proof: $\alpha \vdash \beta$

  - Searching for proofs is an alternative to enumerating models; "can be more efficient"

- Resolution is a sound and complete inference rule
  - Works on clauses (CNF); requires refutation
- Forward and backward chaining

For next time:

AIMA 8.0-8.3;

8.1.1-8.1.2 fyi