命题逻辑

1 KBA(Knowledge based agent)与逻辑

2 模型，有效性，可满足性，蕴涵，推理过程

3 如何证明KB蕴涵a

模型检验; 逻辑等价;

推理规则 [归结推理（文字、子句、归结）]

KBA与逻辑

what will we learn?

- We design (knowledge based) agents that can form representations of a complex world, use a process of inference to derive new representation about the world, and use these new representations to deduce what to do.

What is the central component of knowledge based agent?

Knowledge Based Agents {7.1}

- The knowledge base (KB) is the central component
  - KB is a set of sentences(语句) representing assertions(断言) about the world
  - Sentences are represented with a knowledge representation language
- Two operations on KBs
  - Tell and Ask
  - Both may involve inferencing(推理), deriving new sentences from old

Types of Knowledge {7.1}

- Procedural(过程式), e.g.: functions
  - Such knowledge can only be used in one way -- by executing it
- Declarative (陈述式), e.g.: constraints
  - It can be used to perform many different sorts of inferences

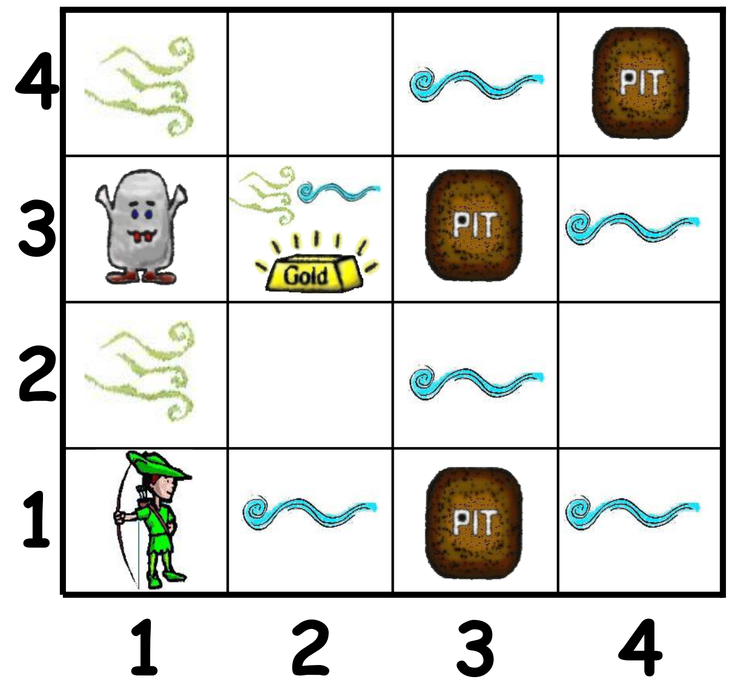
- Logic is a Declarative language to :
  - Assert sentences representing facts that hold in a world W (these sentences are given the value true)
  - Deduce the true/false values to sentences representing other aspects of W

logic is a declarative language{7.1}

- Propositional logic sentence
  - $A \wedge B \Rightarrow C$
- First-order predicate logic sentence
  - $(\forall x)(\exists y)$ Mother(y,x)

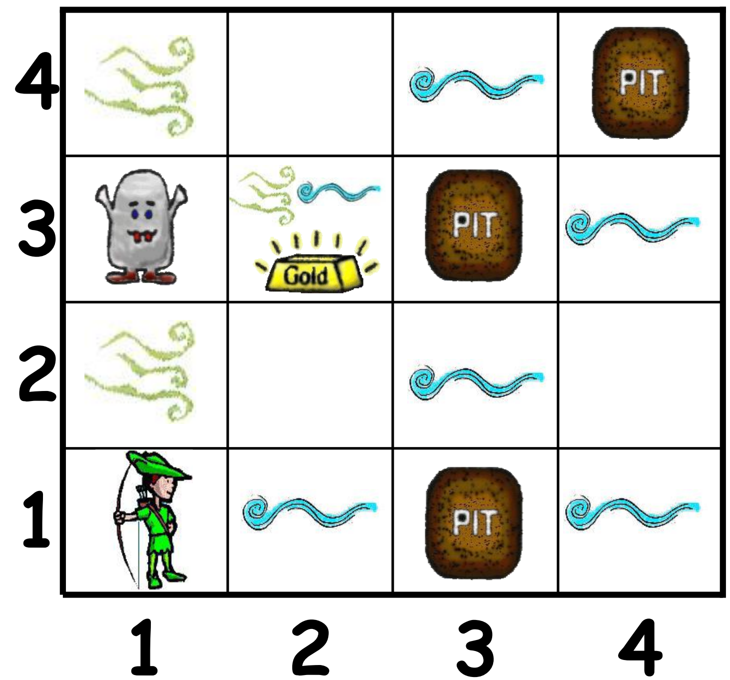
Wumpus World {7.2}

- The **wumpus world** is a cave consisting of rooms connected by passageways. Lurking somewhere is the wumpus, which eats anyone who enters its room. The wumpus can be shot by an agent, but the agent has only one arrow. Some rooms contain pits that will trap anyone who wanders into these rooms (except for the wumpus, which is too big). The only mitigating feature of this bleak environment is the possibility of finding a heap of gold.
- What is the PEAS of this world?



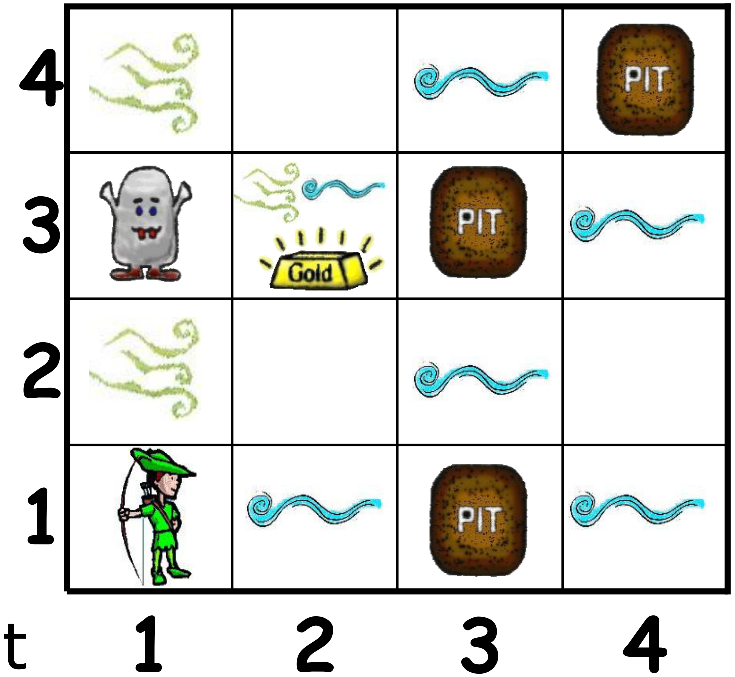
PEAS of Wumpus World {7.2}

- **Performance measure**: gold +1000, death -1000, -1 per step, -10 for using the arrow
- **Environment**: A 4×4 grid of rooms. Start at [1,1], facing to the right. The locations of the gold and the wumpus are chosen randomly (and uniformly) from the squares other than the start square. In addition, each square other than the start can be a pit, with probability 0.2.



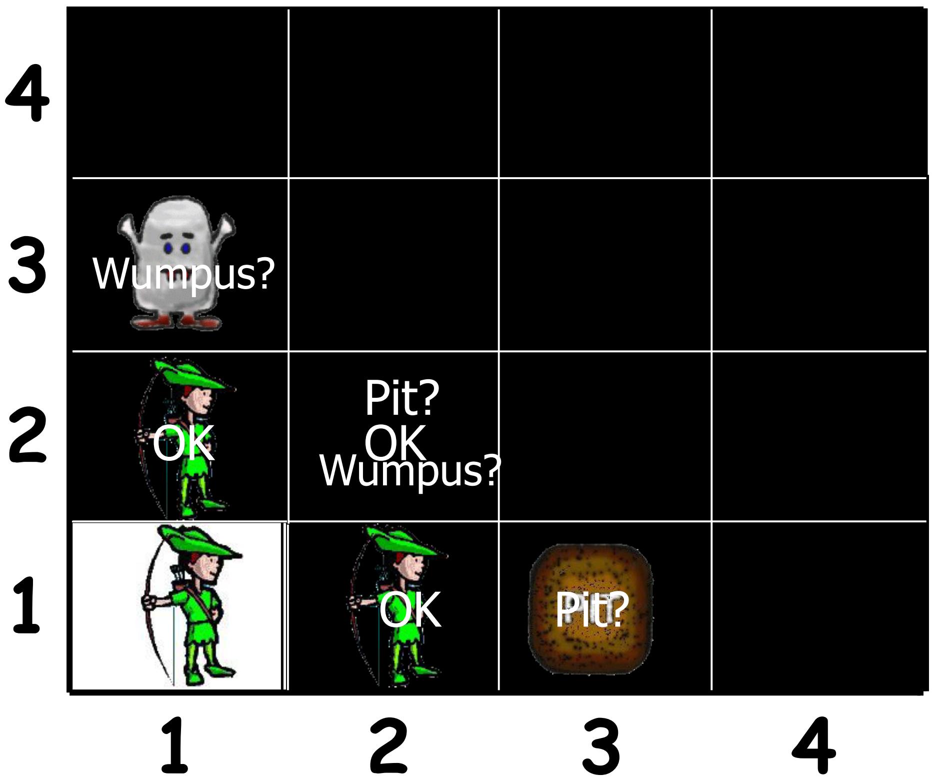
PEAS of Wumpus World {7.2}

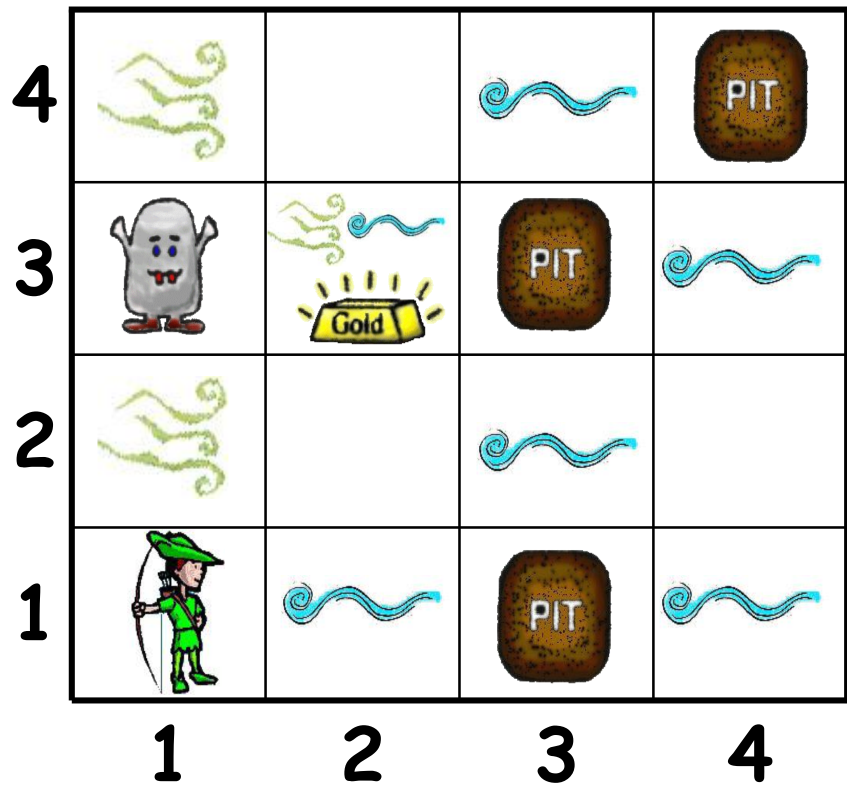
- **Actuators**
  - Left turn
  - Right turn
  - Forward
  - Grab
  - Shoot
- **Sensors**
  - Stench(臭气) , in,4-adjacent
  - Breeze(微风), 4-adjacent
  - Glitter(闪光), in
  - Bump， walking into a wall
  - Scream, wumpus killed, anywhere



Exploring the Wumpus World {7.2}

- Agent needs to know which actions are safe.
- reasoning
- Wumpus world can be solved using logic.





Logic in general {7.3}

- Logics are formal languages for representing information such that conclusions can be drawn
- Syntax defines the form of sentences
- E.g., the language of arithmetic
  - $x+2 \geq y$ is a sentence; $x2+y > \{\}$ is not a sentence
- E.g., propositional logic
  - $P \wedge Q$ is a sentence; $\wedge Q$ is not a sentence

Logic in general {7.3}

- Semantics defines the truth of each sentence w.r.t. each **possible world**
- E.g., the language of arithmetic
  - $x+2 \geq y$ is true in a world where $x = 7, y = 1$
  - $x+2 \geq y$ is false in a world where $x = 0, y = 6$
- E.g., propositional logic
  - $P \wedge Q$ is true in a world where $P=T$, $Q=T$.

Propositional logic: Syntax {7.4.1}

- Propositional logic (命题逻辑) is the simplest logic – illustrates basic ideas
- The proposition symbols P_1, P_2 etc are sentences
  - If S is a sentence, $\neg$S is a sentence (negation,否定式)
  - If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction,合取式)
  - If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (disjunction,析取式)
  - If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication, 蕴含式)
  - If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditiona, 双向蕴含式)

Propositional logic: Semantics {7.4.2}

- Rules for evaluating truth with respect to model m
  - $\neg S$ is true iff S is false
  - $S_1 \wedge S_2$ is true iff S_1 is true and S_2 is true
  - $S_1 \vee S_2$ is true iff S_1 is true or S_2 is true
  - $S_1 \Rightarrow S_2$ is true iff S_1 is false or S_2 is true
  - $S_1 \Leftrightarrow S_2$ is true iff $S_1 \Rightarrow S_2$ is true and $S_2 \Rightarrow S_1$ is true

- Simple recursive process evaluates an sentence.
  - $\neg P_{12} \wedge (P_{22} \vee P_{31})$

模型、有效性、可满足性、蕴涵

Model of a sentence {7.3}

- use the term model in place of "possible world."
- We say *m* is a model of a sentence α (or *m* satisfies α) if α is true in *m*,
  - "x=2,y=2" ______ a model of $x^2+y^2<=16$
  - "x=3,y=3" ______ a model of $x^2+y^2<=16$
  - "P=T,Q=T" ______ a model of $P \wedge Q$.
  - "P=T,Q=F" ______ a model of $P \wedge Q$

Model of a sentence {7.3}

- use the term model in place of "possible world."
- We say *m* is a model of a sentence α (or *m* satisfies α) if α is true in *m*,
  - "x=2,y=2" is a model of $x^2+y^2<=16$
  - "x=3,y=3" is not a model of $x^2+y^2<=16$
  - "P=T,Q=T" is a model of $P \wedge Q$.
  - "P=T,Q=F" is not a model of $P \wedge Q$
- *M*(α) is the set of all models of α
  - $M(P \vee Q)$ = ______________________________

Satisfiability of a sentence {}

- A sentence is *valid* if it is true in all models,
  - $A \vee B$, $x \geq 0$ are__________________________
  - $A \vee \neg A$, $X^2 \geq 0$ are_______________________
- A sentence is *satisfiable* if it is true in some model; A sentence is *unsatisfiable* if it is true in no models
  - $A \wedge \neg A$, $X^2<0$ are________________________
  - $A \vee B$, $X>0$ are___________________________
  - α is satisfiable iff $M(\alpha)$ is______________
  - α is unsatisfiable iff $M(\alpha)$ is______________

Model of a KB {7.3}

- A KB is a set of sentences
- A model m is a model of KB iff it is a model of all sentences in KB, that is, all sentences in KB are true in m.
- KB={P, P $\vee$ R}
  M(KB)=______________________________

Model of a KB {7.3}

- A KB is a set of sentences
- A model m is a model of KB iff it is a model of all sentences in KB, that is, all sentences in KB are true in m.
- KB={P, P $\vee$ R}
  M(KB)={"P=T,R=T","P=T,R=F"}

Satisfiability of a KB {7.3}

- A KB is *satisfiable* iff it admits at least one model (M(KB) is not empty); otherwise it is *unsatisfiable* (M(KB) is empty)
  - KB1 = $\{P, \neg Q \wedge R\}$ is ______________
  - KB2 = $\{\neg P \vee P\}$ is ______________
  - KB3 = $\{P, \neg P\}$ is ______________

Satisfiability of a KB {7.3}

- A KB is *satisfiable* iff it admits at least one model (M(KB) is not empty); otherwise it is *unsatisfiable* (M(KB) is empty)
  - KB1 = {P, $\neg Q \wedge R$} is satisfiable
  - KB2 = {$\neg P \vee P$} is satisfiable
  - KB3 = {P, $\neg P$} is unsatisfiable

Entailment(蘊涵) {7.3}

- Entailment means that one thing follows from another: KB $\models \alpha$
- Knowledge base KB entails sentence α if and only if α is true in all worlds where KB is true
  - $x+y = 4 \models 4 = x+y$? ______ yes/no
  - $x^2+y^2<=4 \models X^2+y^2<=16$? ______ yes/no
  - $P \models P \wedge R$? ______ yes/no
  - Entailment is a relationship between sentences (syntax) that is based on semantics

Entailment(蕴涵) {7.3}

- Entailment means that one thing follows from another: $KB \models \alpha$
- Knowledge base KB entails sentence α if and only if α is true in all worlds where KB is true
  - $x+y = 4 \models 4 = x+y$? (yes)
  - $x^2+y^2<=4 \models X^2+y^2<=16$? (yes)
  - $P \models P \wedge R$? (no)
  - Entailment is a relationship between sentences (syntax) that is based on semantics

Entailment {7.3}

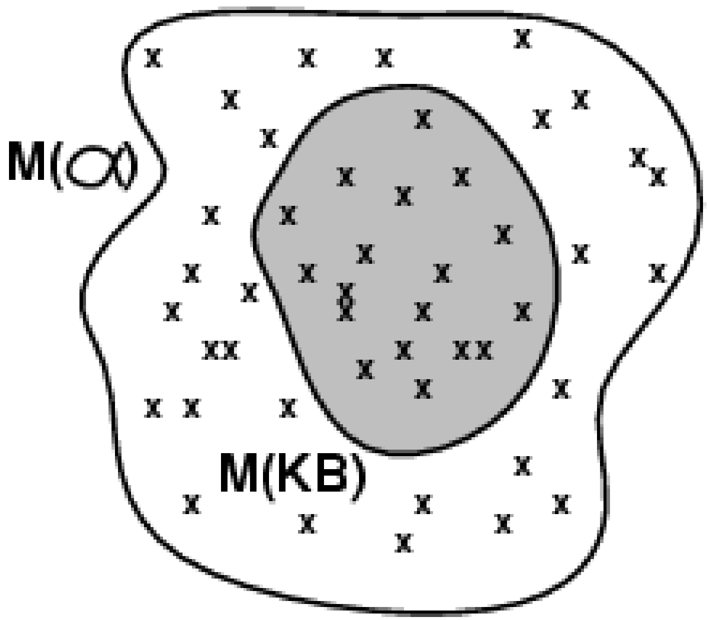
- KB $\models \alpha$
  - iff ...,
  - iff ...,
  - iff ...,
  - iff ...

Entailment {7.3}

- KB $\models \alpha$
  - iff $M(\text{KB})$ ________ $M(\alpha)$,
  - iff $\{\text{KB}, \neg\alpha\}$ is ________________,
  - iff KB $\Rightarrow \alpha$ is _____________ ,
  - iff KB, $\neg\alpha \models$ __________________

Entailment {7.3}

- KB $\models \alpha$
  - iff $M(\text{KB}) \subseteq M(\alpha)$,
  - iff $\{\text{KB}, \neg\alpha\}$ is unsatisfiable,
  - iff $\text{KB} \Rightarrow \alpha$ is valid,
  - iff $\text{KB}, \neg\alpha \models$ False



Inference Procedures {7.3}

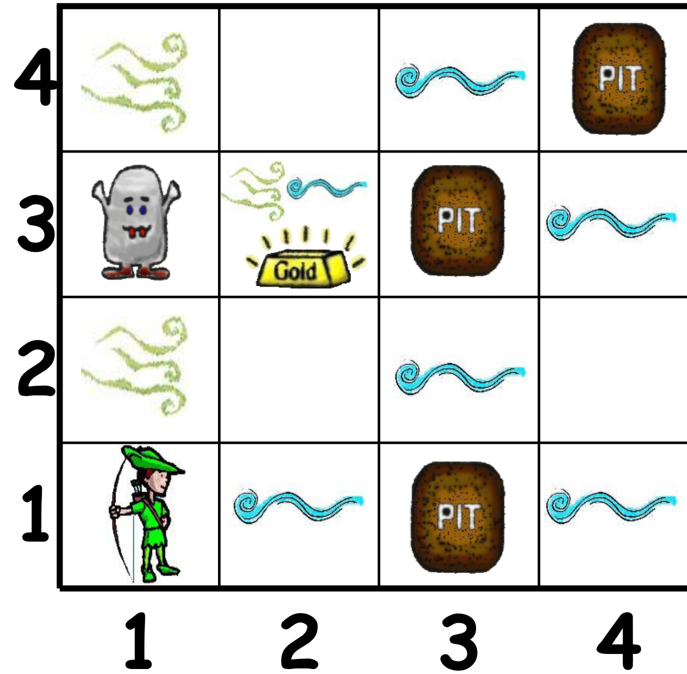
- $KB \vdash_i \alpha$
  - sentence α can be derived from *KB* by procedure *i* (which derives new sentences from old).
- *Soundness(可靠性)*: *i* is sound if whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$ (对于推理过程i从KB推理出的每条语句α，都有$KB \models \alpha$，则i是可靠的)
- *Completeness(完备性)*: *i* is complete if whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$ （对于KB蕴涵的每条语句α，推理过程i都能从KB推理出α，则i是完备的）

如何证明蕴涵

Wumpus world sentences {7.4.3}

- how to represent wumpus world using PL?
- Let $P_{i,j}$ be true if there is a pit in [i, j]
- Let $B_{i,j}$ be true if there is a breeze in [i, j]

- Start
  - R_1: $\neg P_{1,1}$
- Pits cause breezes in adjacent squares
  - R_2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
  - R_3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- Perceiving
  - R_4: $\neg B_{1,1}$
  - R_5: $B_{2,1}$
- is $\neg P_{1,2}$ true????

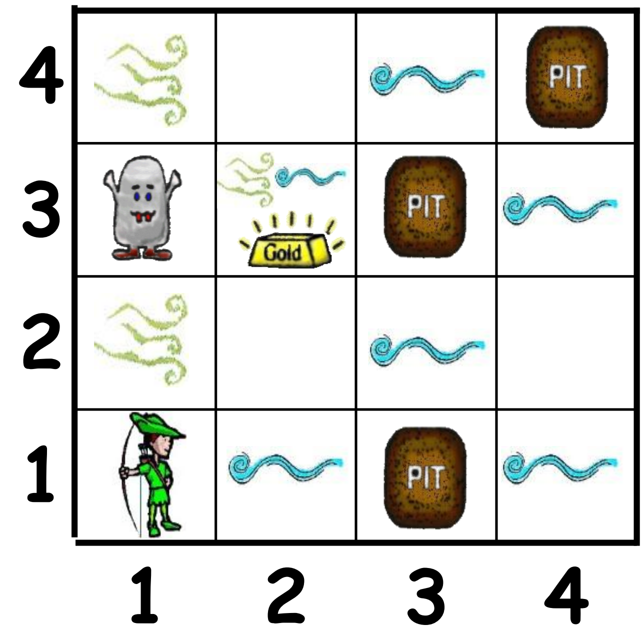


How to prove KB $\models \alpha$

Proof methods

- TELL(KB, R_1)
- …
- TELL(KB, R_5)
- ASK(KB, $\neg P_{1,2}$)

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

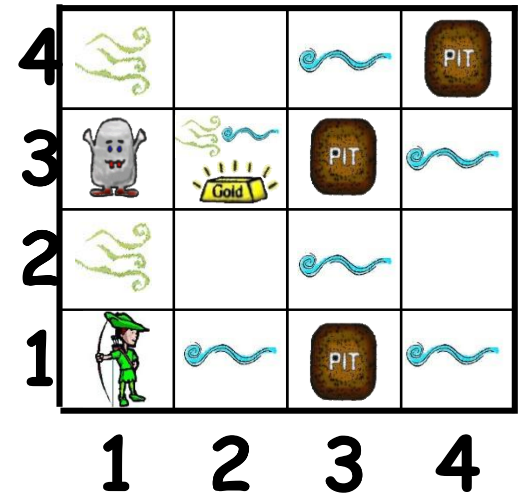


- how to prove KB $\models \neg P_{1,2}$

Proof methods

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

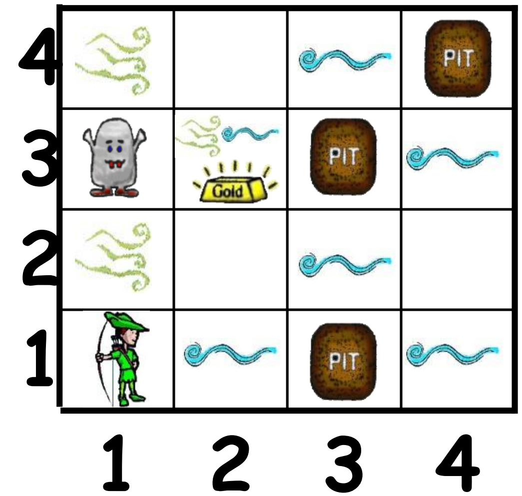
- KB ⊨ α iff ...
- KB ⊨ α iff ...
- KB ⊨ α iff ...
- KB ⊨ α iff ...



Proof methods

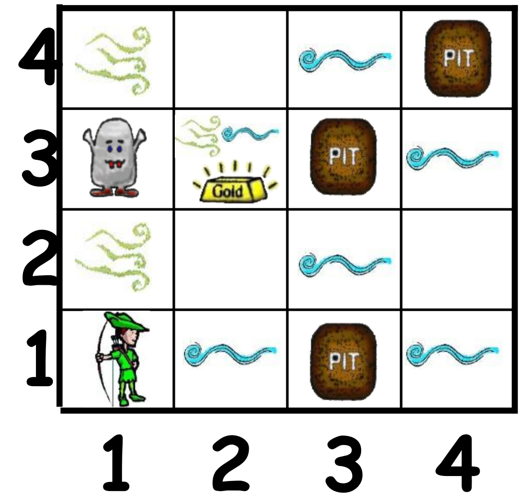
- $KB \models \neg P_{1,2}$?

- KB $\models \alpha$ iff $M(\text{KB}) \subseteq M(\alpha)$
- KB $\models \alpha$ iff (KB $\wedge \neg \alpha$) is unsatisfiable
- KB $\models \alpha$ iff KB $\Rightarrow \alpha$ is valid
- KB $\models \alpha$ iff KB, $\neg \alpha \models$ False



Proof methods

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



- KB $\models \alpha$ iff $M(\text{KB}) \subseteq M(\alpha)$
- KB $\models \alpha$ iff (KB $\wedge \neg \alpha$) is unsatisfiable
- KB $\models \alpha$ iff KB $\Rightarrow \alpha$ is valid
- KB $\models \alpha$ iff KB, $\neg \alpha \models$ False
- to prove KB $\models \alpha$:
- for any m, if m is the model of KB, then m is also the model of α.
- for any m, KB $\wedge \neg \alpha$ is false
- for any m, KB $\Rightarrow \alpha$ is true.
- KB $\Rightarrow \alpha$ is logically equivalent to True.
- From KB, derive α
- From KB, $\neg \alpha$, derive False

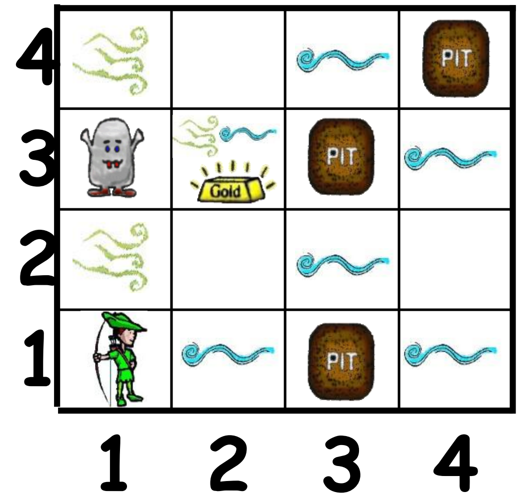
Model checking
truth table enumeration

logical equivalence

inference rules

Proof methods

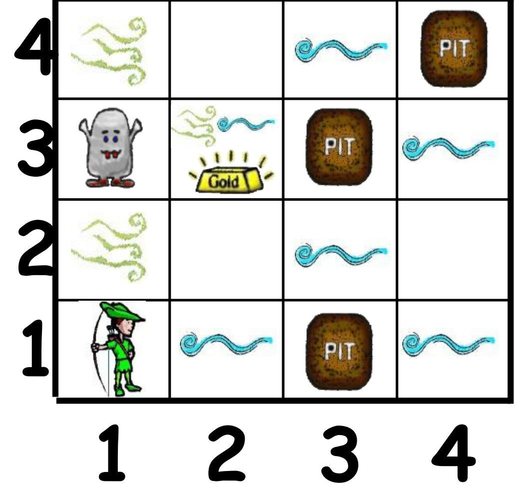
- $KB \mid= \neg P_{1,2}$?



- KB $\models \alpha$ iff $M(\text{KB}) \subseteq M(\alpha)$
- KB $\models \alpha$ iff (KB $\wedge \neg \alpha$) is unsatisfiable
- KB $\models \alpha$ iff KB $\Rightarrow \alpha$ is valid
- KB $\models \alpha$ iff KB, $\neg \alpha \models$ False

- proof by:
  - Model checking
    - truth table enumeration (always exponential in *n*)
  - logical equivalence
  - inference rules

proof by model checking{7.4.4}



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

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	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*	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
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*

proof by model checking:

- $(A \vee \neg B) \wedge (B \vee \neg C) \Rightarrow A \vee \neg C$ is valid ?

A	B	C	$A \vee \neg B$	$B \vee \neg C$	$A \vee \neg C$	$(A \vee \neg B) \wedge (B \vee \neg C) \Rightarrow A \vee \neg C$
0	0	0	1	1	1	1
0	0	1	1	0		1
0	1	0	0	1		1
0	1	1	0	1		1
1	0	0	1	1	1	1
1	0	1	1	0		1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Logical equivalence{7.5}

- Logical equivalence: Two sentences are logically equivalent iff they are true in same models

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

poof by logical equivalence:

- $(A \vee \neg B) \wedge (B \vee \neg C) \Rightarrow A \vee \neg C$ is valid ?

$$(A \vee \neg B) \wedge (B \vee \neg C) \Rightarrow A \vee \neg C$$
$$\equiv \neg((A \vee \neg B) \wedge (B \vee \neg C)) \vee A \vee \neg C$$
$$\equiv (\neg(A \vee \neg B) \vee \neg(B \vee \neg C)) \vee A \vee \neg C$$
$$\equiv (\neg A \wedge B) \vee (\neg B \wedge C) \vee A \vee \neg C$$
$$\equiv (\neg A \wedge B) \vee A \vee (\neg B \wedge C) \vee \neg C$$
$$\equiv ((\neg A \vee A) \wedge (B \vee A)) \vee ((\neg B \vee \neg C) \wedge (C \vee \neg C))$$
$$\equiv (B \vee A) \vee (\neg B \vee \neg C)$$
$$\equiv T$$

inference rule{7.5.1}

- Application of inference rules
  - Legitimate generation of new sentences from old
  - Proof = a sequence of inference rule applications
  - Can use inference rules as operators in a standard search algorithm
  - **Typically require transformation of sentences into a normal form**

Inference Rule {7.5.1}

- Modus Ponens (假言推理规则) $$\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta}$$

- And-Elimination (消去合取词) $$\frac{\alpha \wedge \beta}{\alpha}$$

- *logical equivalence*

  $\alpha \Leftrightarrow \beta \equiv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$

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

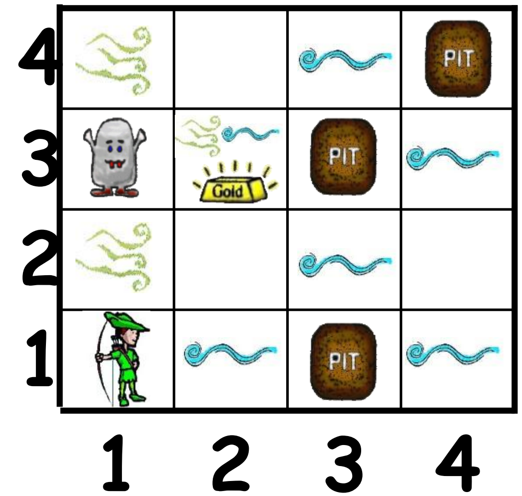
$$\frac{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta}$$

review: Inference procedure{7.3}

- 推理算法i从KB中导出语句α记为： $KB \vdash_i \alpha$
- 推理算法i从KB中导出语句α都是KB蕴涵的语句，该推理算法被称为**可靠的(sound)**或**真值保持的(true-preserving)**
- 如果推理算法可以生成任一蕴涵句，则它是**完备的(complete)**

inference{7.5.1}

- Start
  - $R_1: \neg P_{1,1}$
- Pits cause breezes in adjacent squares
  - $R_2: B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
  - $R_3: B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- Perceiving
  - $R_4: \neg B_{1,1}$
  - $R_5: B_{2,1}$
- $KB \models \neg P_{1,2}$?



- From KB, how to derive $\neg P_{1,2}$ based on the rules:

Modus Ponens

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

And-Elimination

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

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$ commutativity of $\wedge$

$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$ commutativity of $\vee$

$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$ associativity of $\wedge$

$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$ associativity of $\vee$

$\neg(\neg\alpha) \equiv \alpha$ double-negation elimination

$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$ contraposition

$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$ implication elimination

$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$ biconditional elimination

$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ de Morgan

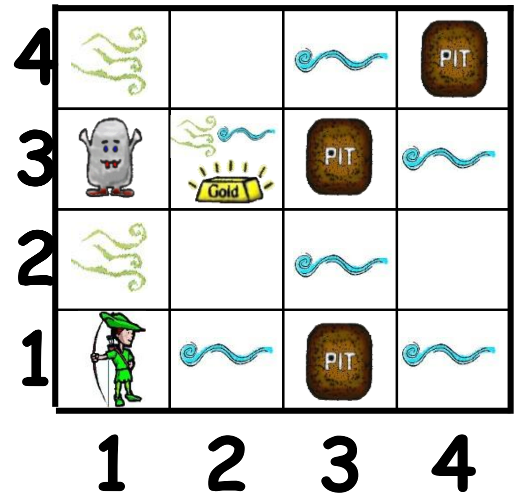
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$ de Morgan

$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$ distributivity of $\wedge$ over $\vee$

$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$ distributivity of $\vee$ over $\wedge$

inference{7.5.1}

- Start
  - $R_1: \neg P_{1,1}$
- Pits cause breezes in adjacent squares
  - $R_2: B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
  - $R_3: B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- Perceiving
  - $R_4: \neg B_{1,1}$
  - $R_5: B_{2,1}$
- $KB \models \neg P_{1,2}$?



- $R_6: (B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$ R_2, biconditional-elimination
- $R_7: ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$ R_6 , And-Elimination
- $R_8: (\neg B_{1,1} \Rightarrow \neg (P_{1,2} \vee P_{2,1}))$ R_7, contraposition
- $R_9: \neg (P_{1,2} \vee P_{2,1})$ R_8,R_4 , Modus-ponens
- $R_{10}: \neg P_{1,2} \wedge \neg P_{2,1}$ R_9, De Morgan
- $R_{11}: \neg P_{1,2}$ R10, And-Elimination

inference{7.5.1}

- Car problem:
  - (1) Battery-OK、
  - (2) ¬Empty-Gas-Tank、
  - (3) ¬Car-OK
  - (4) Battery-OK ∧ ¬Empty-Gas-Tank ⇒ Engine-Starts
  - (5) Engine-Starts ∧ ¬Flat-Tire ⇒ Car-OK
- What’s the problem?
  - ______________________

inference{7.5.1}

- Car problem:
  (1) Battery-OK、
  (2) ¬Empty-Gas-Tank、
  (3) ¬Car-OK
  (4) Battery-OK ∧ ¬Empty-Gas-Tank ⇒ Engine-Starts
  (5) Engine-Starts ∧ ¬Flat-Tire ⇒ Car-OK
- What's the problem?
  - Flat-Tire

formalize the inference
as a search problem

inference by searching {7.5.1}

- apply a search algorithm
  - Initial state: ______________________
  - Actions: _________________
  - Result: __________________
  - Goal: ________________________

inference by searching {7.5.1}

- apply a search algorithm
  - Initial state: the initial KB
  - Actions: inference rules
  - Result: add inferred sentences to KB
  - Goal: the sentence we are trying to prove is in the KB.

from KB, derive α

from $\{KB, \neg\alpha\}$, derive False

$\{KB, \neg\alpha\}$ is unsatisfiable

using resolution inference rule

resolution {7.5.2}

- is the following inference rule sound?

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

- resolution rule

Complementary Literals {7.5.2}

- A *literal* is a either an atomic sentence or the negated atomic sentence, e.g.:

$$P, \neg P$$

- Two literals are complementary if one is the negation of the other, e.g.:

$$P \text{ and } \neg P$$

CNF (合取范式){7.5.2}

- Clause
  disjunction of literals
  E.g., $A \vee \neg B$

- Conjunctive Normal Form
  conjunction of disjunctions of literals
  E.g., $(A \vee \neg B) \wedge (B \vee C)$:

Basic intuition, resolve B, $\neg B$ to $A \vee C$

CNF: why? how?

Conversion to CNF {7.5.2}

- Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
- Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.
- Move $\neg$ inwards using de Morgan's rules and double-negation:
- Apply distributivity law ($\wedge$ over $\vee$) and flatten:

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

$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$ contraposition

Resolution(归结) {7.5.2}

- Resolution inference rule (for CNF)

$$\frac{l_1 \vee \ldots \vee l_k, \qquad m_1 \vee \ldots \vee m_n}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n}$$

where l_i and m_j are complementary literals.

$$\frac{P \qquad \neg P \vee Q}{Q} \qquad\qquad \frac{P \vee Q \qquad \neg P \vee Q}{Q}$$

$$\frac{\neg P \vee Q \qquad \neg Q \vee R}{\neg P \vee R} \qquad\qquad \frac{P \qquad \neg P}{NIL}$$

Resolution(归结) {7.5.2}

- sound ?

$$\frac{P \vee Q \quad \neg P \vee \neg Q}{NIL}$$

$$\frac{P \vee Q \quad \neg P \vee \neg Q}{Q \vee \neg Q}$$

$$\frac{P \vee Q \quad \neg P \vee \neg Q}{P \vee \neg P}$$

Resolution(归结) {7.5.2}

- sound ?

$$\frac{P \vee Q \quad \neg P \vee \neg Q}{NIL}$$ NO

$$\frac{P \vee Q \quad \neg P \vee \neg Q}{Q \vee \neg Q}$$ YES

$$\frac{P \vee Q \quad \neg P \vee \neg Q}{P \vee \neg P}$$ YES

resolution{7.5.2}

- How to prove KB $\models \alpha$ using resolution?

- consider:

KB $\models \alpha$ iff (KB $\wedge \neg \alpha$) is unsatisfiable

and

$$\frac{P \qquad \neg P}{NIL}$$

Resolution algorithm {7.5.2}

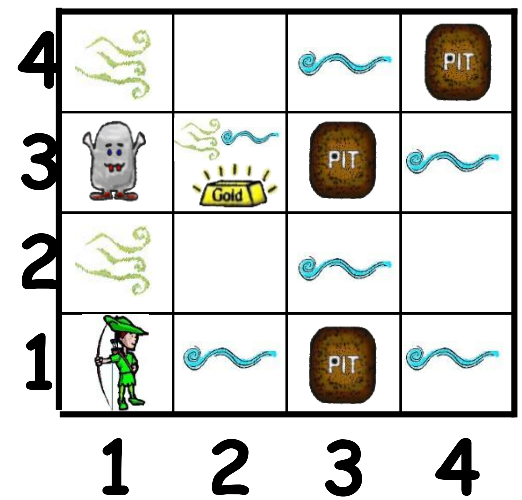
- Proof by contradiction, show $KB \wedge \neg \alpha$ unsatisfiable

function PL-RESOLUTION(KB, α) **returns** *true* or *false*
  inputs: KB, the knowledge base, a sentence in propositional logic
    α, the query, a sentence in propositional logic

  $clauses \leftarrow$ the set of clauses in the CNF representation of $KB \wedge \neg\alpha$
  $new \leftarrow \{\,\}$
  loop do
    for each pair of clauses C_i, C_j **in** $clauses$ **do**
      $resolvents \leftarrow$ PL-RESOLVE(C_i, C_j)
      if $resolvents$ contains the empty clause **then return** *true*
      $new \leftarrow new \cup resolvents$
    if $new \subseteq clauses$ **then return** *false*
    $clauses \leftarrow clauses \cup new$

Logical agent in the Wumpus world

- Start
  - R_1: $\neg P_{1,1}$
- Pits cause breezes in adjacent squares
  - R_2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
  - R_3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- Perceiving
  - R_4: $\neg B_{1,1}$
  - R_5: $B_{2,1}$
- $KB \models \neg P_{1,2}$?
- CNF representation of KB …
- resolution …



$\{P \vee Q, \neg P \vee Q, P \vee \neg Q, \neg P \vee \neg Q\}$ is unsatisfiable

1) $P \vee Q$
2) $\neg P \vee Q$
3) $P \vee \neg Q$
4) $\neg P \vee \neg Q$
5) Q 1,2
6) P 1,3
7) $P \vee \neg P$ 1,4
8) $Q \vee \neg Q$ 1,4
9) $Q \vee \neg Q$ 2,3
10) $P \vee \neg P$ 2,3
11) $\neg P$ 2,4
12) $\neg Q$ 3,4

14) $P \vee Q$ 1,8
15) $P \vee Q$ 1,9
16) $P \vee Q$ 1,10
17) Q 1,11
18) P 1,12
19) Q 2,6
20) $\neg P \vee Q$ 2,7
21) $\neg P \vee Q$ 2,8
22) $\neg P \vee Q$ 2,9
23) $\neg P \vee Q$ 2,10
24) $\neg P$ 2,12
25) P 3,5
26) $P \vee \neg Q$ 3,7

27) $P \vee \neg Q$ 3,8
28) $P \vee \neg Q$ 3,9
29) $P \vee \neg Q$ 3,10
30) $\neg Q$ 3,11
31) $\neg P$ 4,5
32) $\neg Q$ 4,6
33) $\neg P \vee \neg Q$ 4,7
34) $\neg P \vee \neg Q$ 4,8
35) $\neg P \vee \neg Q$ 4,9
36) $\neg P \vee \neg Q$ 4,10
37) Q 5,8
38) Q 5,9
39) NIL 5,12

think: how to improve the efficiency?

Horn子句、限定子句{7.5.3}

- **限定子句**就是受限形式的一种子句，它是指*恰好只含一个正文字*的析取式。例如，子句 $(\neg L_{1,1} \lor \neg Breeze \lor B_{1,1})$是限定子句，而 $(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1})$不是。

- 更一般的形式是**Horn子句**，是指*至多只有一个正文字*的析取式。因此所有限定子句都是Horn子句；没有正文字的析取式也是Horn子句，这些称为**目标子句**。Horn子句在归结下是封闭的：如果对两个Horn子句进行归结，结果依然是Horn子句。

Horn子句、限定子句{7.5.3}

- 每个限定子句可写为蕴含式
- 用Horn子句判定蕴涵需要的时间与知识库大小成*线性*关系
- 使用Horn子句的推理可使用前向连接和反向连接

前向链接{7.5.4}

function PL-FC-ENTAILS?(KB, q) **returns** *true* or *false*
  inputs: KB, the knowledge base, a set of propositional definite clauses
    q, the query, a proposition symbol
  $count \leftarrow$ a table, where $count[c]$ is the number of symbols in c's premise
  $inferred \leftarrow$ a table, where $inferred[s]$ is initially $false$ for all symbols
  $agenda \leftarrow$ a queue of symbols, initially symbols known to be true in KB

  while $agenda$ is not empty **do**
    $p \leftarrow$ POP($agenda$)
    if $p = q$ **then return** *true*
    if $inferred[p] = false$ **then**
      $inferred[p] \leftarrow true$
      for each clause c in KB where p is in c.PREMISE **do**
        decrement $count[c]$
        if $count[c] = 0$ **then** add c.CONCLUSION to $agenda$
  return *false*

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

前向链接{7.5.4}

- *Agenda*记录了已知为真但未被“处理”的符号。
- *count*表记录着每个蕴含式还有多少前提未知。当待办事项表*Agenda*中的一个新符号*p*被处理，对于每个前提中出现*p*的蕴含式而言，它相应的计数值减去1，如果计数变为0，蕴含式的所有前提都已知，将它的结论添加到*Agenda*中。
- 如果一个符号已经在推出的符号集合*inferred*中，则无需再次添加到*Agenda*中。

- 前向链接是**数据驱动**推理的实例——即推理是从已知数据开始的。是可靠的，完备的。

反向链接{7.5.4}

- 反向链接算法从查询开始进行推理。如果查询q已知为真，那么无需进行任何操作。否则，寻找知识库中那些以q为结论的蕴含式。如果其中某个蕴含式的所有前提都能证明为真（通过反向链接），则q为真。与前向链接一样，有效实现的时间复杂度是线性的。

- 反向链接是一种**目标制导的推理**形式。

Summary

- Logical agents apply inference to a knowledge base to derive new information and make decisions
- Basic concepts of logic
  - syntax: formal structure of sentences
  - semantics: truth of sentences wrt models
- Propositional logic lacks expressive power
  - can’t say "pits cause breezes in adjacent squares”
  - except by writing one sentence for each square

Thanks!

next :
chapter 8 first-order logic