CMSC 471: Intro to AI

Propositional Logic

Many slides courtesy Tim Finin

Propositional logic syntax

- Users specify:
  - Set of propositional symbols (or variables) (e.g., P, Q) whose values can be **True** or **False**
  - What each *means*, e.g.: P: "*It's hot*", Q: "*It's humid*"
- A sentence (well formed formula) is defined as:
  - Any symbol is a sentence
  - If S is a sentence, then **¬S** is a sentence
  - If S is a sentence, then **(S)** is a sentence
  - If S and T are sentences, then so are **(S ∨ T), (S ∧ T), (S → T)** and **(S ↔ T)**
  - A finite number of applications of the rules

Examples of PL sentences

- **Q**
  "It's humid"
- **Q → P**
  "If it's humid, then it's hot"
- **(P ∧ Q) → R**
  "If it's hot and it's humid, then it's raining"
- We're free to choose better symbols, e.g.:
  Hot for "It's hot"
  Humid for "It's humid"
  Raining for "It's raining"

Truth tables

Used to define meaning of logical connectives

[redacted]

Truth tables for the five logical connectives

P	$\neg P$
True	False
True	False
False	True
False	True

“not”

Truth tables

Used to define meaning of logical connectives

[redacted]

Truth tables for the five logical connectives

P	Q	$\neg P$	$P \wedge Q$
True	True	False	True
True	False	False	False
False	False	True	False
False	True	True	False

"and"

Truth tables

Used to define meaning of logical connectives

[redacted]

Truth tables for the five logical connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$
True	True	False	True	True
True	False	False	False	True
False	False	True	False	False
False	True	True	False	True

(inclusive)
“or”

Truth tables

Used to define meaning of logical connectives

Truth tables for the five logical connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \to Q$
True	True	False	True	True	True
True	False	False	False	True	False
False	False	True	False	False	True
False	True	True	False	True	True

implication
of q from p

Truth tables

Used to define meaning of logical connectives

Truth tables for the five logical connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
True	True	False	True	True	True	True
True	False	False	False	True	False	False
False	False	True	False	False	True	True
False	True	True	False	True	True	False

Bidirectional implication (aka, equivalence)
$(P \rightarrow Q) \wedge (Q \rightarrow P)$

Distribution of Negation

¬ ¬

P	Q	$\neg P$	$P \vee Q$	$\neg P \wedge \neg Q$	$P \wedge Q$	$\neg P \vee \neg Q$
True	True	False	True	False	True	False
True	False	False	True	False	False	True
False	False	True	False	True	False	True
False	True	True	True	False	False	True

Examples

- What's the truth table of

$$\neg P \lor Q$$

P	Q	$\neg P$	$P \lor Q$
True	True	False	True
True	False	False	True
False	False	True	False
False	True	True	True

Examples

- What's the truth table of

$$\neg \boldsymbol{P} \lor \boldsymbol{Q}$$

$\boldsymbol{P}$	$\boldsymbol{Q}$	$\neg \boldsymbol{P}$	$\boldsymbol{P} \lor \boldsymbol{Q}$	$\neg \boldsymbol{P} \lor \boldsymbol{Q}$
True	True	False	True	True
True	False	False	True	False
False	False	True	False	True
False	True	True	True	True

Some terms

- Given the truth values of all symbols in a sentence, it can be ***evaluated*** to determine its **truth value** (True or False)
- We consider a **Knowledge Base** (KB) to be a set of sentences that are all True
- A **model** for a KB is a **possible world** – an assignment of truth values to propositional symbols that makes each KB sentence true

More terms

- A **valid sentence** or **tautology:** one that's **True** under all interpretations, no matter what the world is actually like or what the semantics is. Example: "It's raining or it's not raining" $(P \vee \neg P)$
- An **inconsistent sentence** or **contradiction:** a sentence that's **False** under all interpretations. The world is never like what it describes, as in "It's raining and it's not raining." $(P \wedge \neg P)$

The implies connective: $P \rightarrow Q$

$\rightarrow$ is a *logical connective*

- $P \rightarrow Q$ is a **logical sentence** and has a truth value, i.e., is either **True** or **False**
- If the sentence is in a KB, it can be used by a rule (*Modes Ponens*) to infer that Q is True if P is True in the KB
- Note: $P \rightarrow Q$ is equivalent to $\sim P \vee Q$

Examples

- What's the truth table of

$$\neg P \vee Q$$

P	Q	$\neg P$	$P \vee Q$	$\neg P \vee Q$	$P \to Q$
True	True	False	True	True	True
True	False	False	True	False	False
False	False	True	False	True	True
False	True	True	True	True	True

- What's the truth table of

$$(P \vee Q) \wedge \neg Q) \to P?$$

(Work it out on your own)

$P \rightarrow Q$

When is $P \rightarrow Q$ true? Check all that apply

- ❑ P=Q=true
- ❑ P=Q=false
- ❑ P=true, Q=false
- ❑ P=false, Q=true

$P \rightarrow Q$

When is *P→Q* true? Check all that apply

- ☑ P=Q=true
- ☑ P=Q=false
- ☐ P=true, Q=false
- ☑ P=false, Q=true

- We can get this from the truth table for $\rightarrow$
- Note: in FOL it's much harder to prove that a conditional true, e.g., prime(x) $\rightarrow$ odd(x)

you must prove it's true for every possible value of x

Knowledge Bases (KBs)

- **Literal**: a Boolean variable
- **Clause**: a disjunction of literals
  - If $l_1, \ldots, l_N$ are literals, then $l_1 \vee \cdots \vee \; l_N$ is a clause
  - Clauses don't need to contain *all* literals
- **Definite clause** (aka Strict Horn clause): a *body* implies a *head*
  - Form: $a_1 \wedge a_2 \wedge \cdots \wedge a_M \rightarrow h$
  - Body: $a_1 \wedge a_2 \wedge \cdots \wedge a_M$
  - Head: h

Representing Knowledge Bases (KBs)

- A conjunction of **definite clauses**
- **Conjunctive Normal Form (CNF):** A conjunction of **disjunctions**

Q: Is
$A \vee B \vee \neg C$
a definite clause?

Representing Knowledge Bases (KBs)

- A conjunction of **definite clauses**
- **Conjunctive Normal Form (CNF):** A conjunction of **disjunctions**

Q: Is
$A \vee B \vee \neg C$
a definite clause?

A: No. Can you turn it into one?

Models for a KB

KB: [P∨Q, P→R, Q→R]

P	Q	R	s1	s2	s3
F	F	F	X	✓	✓
F	F	T	X	✓	✓
F	T	F	✓	✓	X
F	T	T	✓	✓	✓
T	F	F	✓	X	✓
T	F	T	✓	✓	✓
T	T	F	✓	X	X
T	T	T	✓	✓	✓

Models for a KB

KB: [P∨Q, P→R, Q→R]

What are the sentences?

P	Q	R	s1	s2	s3
F	F	F	X	✓	✓
F	F	T	X	✓	✓
F	T	F	✓	✓	X
F	T	T	✓	✓	✓
T	F	F	✓	X	✓
T	F	T	✓	✓	✓
T	T	F	✓	X	X
T	T	T	✓	✓	✓

Models for a KB

KB: [P∨Q, P→R, Q→R]

What are the sentences?

s1: P∨Q
s2: P→R
s3: Q→R

P	Q	R	s1	s2	s3
F	F	F	X	✓	✓
F	F	T	X	✓	✓
F	T	F	✓	✓	X
F	T	T	✓	✓	✓
T	F	F	✓	X	✓
T	F	T	✓	✓	✓
T	T	F	✓	X	X
T	T	T	✓	✓	✓

Models for a KB

KB: [P∨Q, P→R, Q→R]

What are the sentences?

s1: P∨Q
s2: P→R
s3: Q→R

What are the propositional variables, symbols or literals?

P	Q	R	s1	s2	s3
F	F	F	X	✓	✓
F	F	T	X	✓	✓
F	T	F	✓	✓	X
F	T	T	✓	✓	✓
T	F	F	✓	X	✓
T	F	T	✓	✓	✓
T	T	F	✓	X	X
T	T	T	✓	✓	✓

Models for a KB

KB: [P∨Q, P→R, Q→R]

What are the sentences?

s1: P∨Q
s2: P→R
s3: Q→R

What are the propositional variables, symbols or literals?

P, Q, R

P	Q	R	s1	s2	s3
F	F	F	X	✓	✓
F	F	T	X	✓	✓
F	T	F	✓	✓	X
F	T	T	✓	✓	✓
T	F	F	✓	X	✓
T	F	T	✓	✓	✓
T	T	F	✓	X	X
T	T	T	✓	✓	✓

Models for a KB

KB: [P∨Q, P→R, Q→R]

What are the sentences?

s1: P∨Q
s2: P→R
s3: Q→R

What are the propositional variables, symbols or literals?

P, Q, R

What are the candidate models?

1) Consider all **eight** possible assignments of T|F to P, Q, R
2) Check if each sentence is consistent with the model

P	Q	R	s1	s2	s3
F	F	F	X	✓	✓
F	F	T	X	✓	✓
F	T	F	✓	✓	X
F	T	T	✓	✓	✓
T	F	F	✓	X	✓
T	F	T	✓	✓	✓
T	T	F	✓	X	X
T	T	T	✓	✓	✓

Models for a KB

KB: [P∨Q, P→R, Q→R]

What are the sentences?

s1: P∨Q
s2: P→R
s3: Q→R

What are the propositional variables, symbols or literals?

P, Q, R

What are the candidate models?

1) Consider all **eight** possible assignments of T|F to P, Q, R
2) Check if each sentence is consistent with the model

P	Q	R	s1	s2	s3
F	F	F	X	✓	✓
F	F	T	X	✓	✓
F	T	F	✓	✓	X
F	T	T	✓	✓	✓
T	F	F	✓	X	✓
T	F	T	✓	✓	✓
T	T	F	✓	X	X
T	T	T	✓	✓	✓

- Only 3 models are consistent with KB
- R true in **all** of them
- Therefore, R is true and can be added to KB

A simple example

The KB

P

Q ∨ ¬ R

The KB has 2 sentences.

The KB has 3 variables.

Models for the KB

P	Q	R	KB
T	T	F	T
T	T	T	T
T	F	F	T
T	F	T	F
F	T	F	F
F	T	T	F
F	F	T	F
F	F	F	F

A simple example

The KB

P

$Q \vee \neg R$

The KB has 2 sentences.

The KB has 3 variables.

The KB has 3 models. Each model has a value for every variable in the KB such every sentence evaluates to true.

Models for the KB

P	Q	R	KB
T	T	F	T
T	T	T	T
T	F	F	T
T	F	T	F
F	T	F	F
F	T	T	F
F	F	T	F
F	F	F	F

Another simple example

The KB

$P \wedge Q$

$R \wedge \neg P$

The KB has 2 sentences.

The KB has 3 variables.

Models for the KB

P	Q	R

The KB has no models. There is no assignment of True or False to every variable that makes every sentence in the KB true

Finite CSP to Logic

- Let X be a variable with domain $\{a_1, a_2, \ldots, a_D\}$
- Replace X with D different **indicator variables**
  - X_1 is true iff $X = \mathrm{a}_1$
  - X_2 is true iff $X = \mathrm{a}_2$
  - …
  - X_D is true iff $X = \mathrm{a}_D$
- Add pairwise constraints. For $i < j$:
  - $\neg X_i \vee \neg X_j$
- At least one must be "on"
  - $X_1 \vee X_2 \vee \cdots \vee X_D$

Reasoning with Propositional Logic

- There are many ways to approach reasoning with propositional logic
- We’ll look at one, resolution refutation, that can be extended to first order logic
- Later, we will look other approaches that are special to propositional logic

Reasoning / Inference

- **Logical inference** creates new sentences that logically follow from a set of sentences (KB)
- It can also detect if a KB is inconsistent, i.e., has sentences that entail a **contradiction**
- An inference rule is **sound** if every sentence it produces from a KB logically follows from the KB
  - i.e., inference rule creates no contradictions
- An inference rule is **complete** if it can produce every expression that logically follows from (is entailed by) the KB
  - Note analogy to complete search algorithms

Sound rules of inference

Examples of sound rules of inference

Each can be shown to be sound using a truth table

RULE	PREMISE	CONCLUSION
Modus Ponens	$A, A \rightarrow B$	B
And Introduction	A, B	$A \wedge B$
And Elimination	$A \wedge B$	A
Double Negation	$\neg\neg A$	A
Unit Resolution	$A \vee B, \neg B$	A
Resolution	$\mathbf{A \vee B, \neg B \vee C}$	$\mathbf{A \vee C}$

Resolution

- **Resolution** is a valid inference rule producing a new clause implied by two clauses containing *complementary literals*

  Literal: atomic symbol or its negation, i.e., P, ~P
- Amazingly, this is the **only** interference rule needed to build a sound & complete theorem prover
  - Based on proof by contradiction, usually called resolution refutation
- The resolution rule was discovered by Alan Robinson (CS, U. of Syracuse) in the mid 1960s

Some Standard Tautologies

- Identity:
  - $A \wedge T$ <=> P
  - $A \vee F$ <=> P
- Domination:
  - $A \vee T$ <=> T
  - $A \wedge F$ <=> F
- Distributive :
  - $(A\vee (B \wedge C)) \leftrightarrow (A\vee B)\wedge(A\vee C)$
  - $(A \wedge(B \vee C)) \leftrightarrow (A \wedge B) \vee(A \wedge C)$
- De Morgan:
  - $\sim(A \vee B) \leftrightarrow \sim A \wedge \sim B$
  - $\sim(A \wedge B) \leftrightarrow \sim A \vee \sim B$

- $(A\rightarrow B) \leftrightarrow (\sim A \vee B)$

Resolution of KB

- A KB is a set of sentences all of which are true, i.e., a conjunction of sentences
- To use resolution, put KB into *conjunctive normal form* (CNF)
  - Each sentence is a disjunction of one or more literals (positive or negative atoms)
- Every KB can be put into CNF, by rewriting its sentences using standard tautologies, e.g.:
  - $P \rightarrow Q \equiv \sim P \vee Q$
  - $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R) \equiv (P \vee Q) , (P \vee R)$

Resolution Example

- KB: [$P \rightarrow Q$, $Q \rightarrow R \wedge S$]
- KB: [$P \rightarrow Q$, $Q \rightarrow R$, $Q \rightarrow S$]
- KB in CNF: [$\sim P \vee Q$, $\sim Q \vee R$, $\sim Q \vee S$]
- Resolve KB[0] and KB[1] producing:

  $\sim P \vee R$ *(i.e., $P \rightarrow R$)*
- Resolve KB[0] and KB[2] producing:

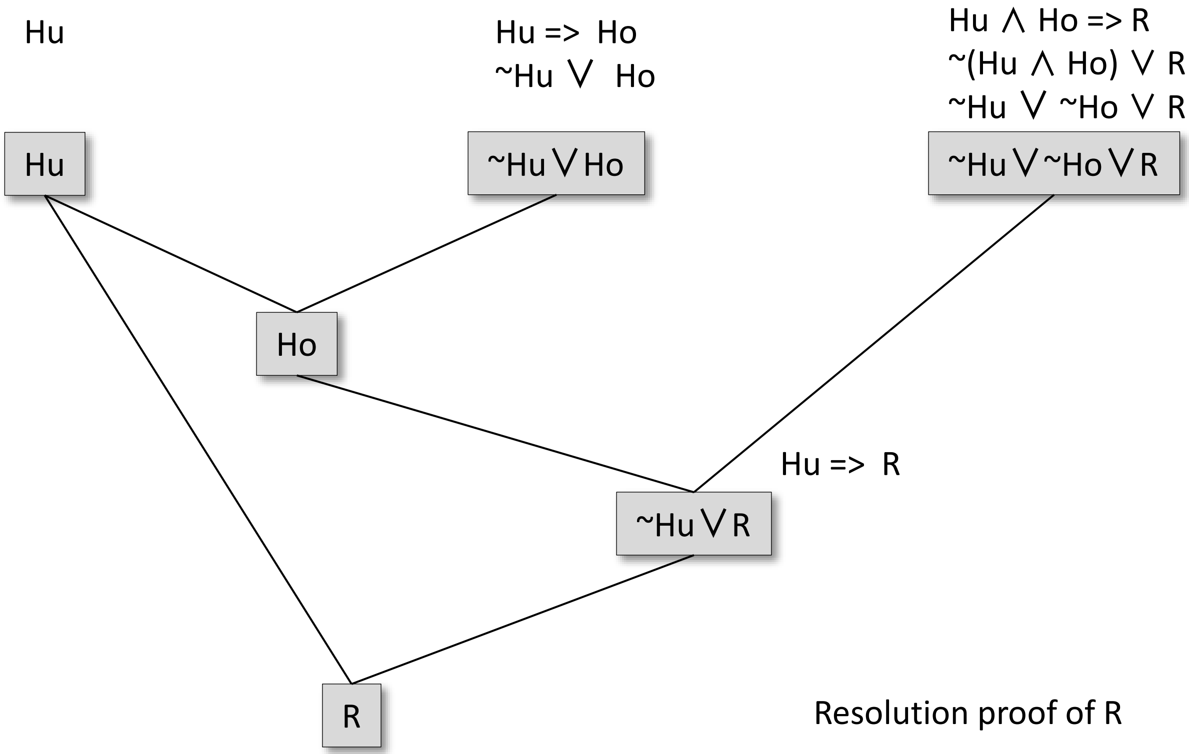
  $\sim P \vee S$ *(i.e., $P \rightarrow S$)*
- New KB: [$\sim P \vee Q$, $\sim Q \vee R$, $\sim Q \vee S$, $\sim P \vee R$, $\sim P \vee S$]

Proving it's raining with rules

- A **proof** is a sequence of sentences, where each is a premise (i.e., a given) or is derived from earlier sentences in the proof by an inference rule
- Last sentence is the **theorem** (also called goal or query) that we want to prove
- The *weather problem* using traditional reasoning

1 Hu	premise	"It's humid"
2 $Hu \rightarrow Ho$	premise	"If it's humid, it's hot"
3 Ho	modus ponens(1,2)	"It's hot"
4 $(Ho \wedge Hu) \rightarrow R$	premise	"If it's hot & humid, it's raining"
5 $Ho \wedge Hu$	and introduction(1,3)	"It's hot and humid"
6 R	modus ponens(4,5)	"It's raining"

Proving it’s raining with resolution



Resolution proof of R

A simple proof procedure

This procedure generates new sentences in a KB

1. Convert all sentences in the KB to CNF[1]
2. Find all pairs of sentences with complementary literals[2] that have not yet been resolved
3. If there are no pairs stop else resolve each pair, adding the result to the KB and go to 2

- Is it sound?
- Is it complete?
- Will it always terminate?

1: a KB in conjunctive normal form is a set of disjunctive sentences

2: a literal is a variable or its negation

Propositional Resolution

- It is sound!
- It's not *generatively complete* in that it can't derive all clauses that follow from the KB
  - The issues are not serious limitations, though
  - Example: if the KB includes P and includes Q we won't derive P ^ Q
- It will always terminate
- But generating all clauses that follow can take a long time and many may be useless

Refutation proofs

- Common use case: we have a question/goal (e.g, **P**) and want to know if it's true
- A refutation proof is a common approach:
  - We start with a KB with all true facts
  - Add negation of what we want to prove to KB (e.g., **~P**)
  - Try to find a contradiction
  - If proof ever produces one, it must be due to adding **~P**, so goal is proven
- Procedure easy to focus & control, so is tends to be more efficient

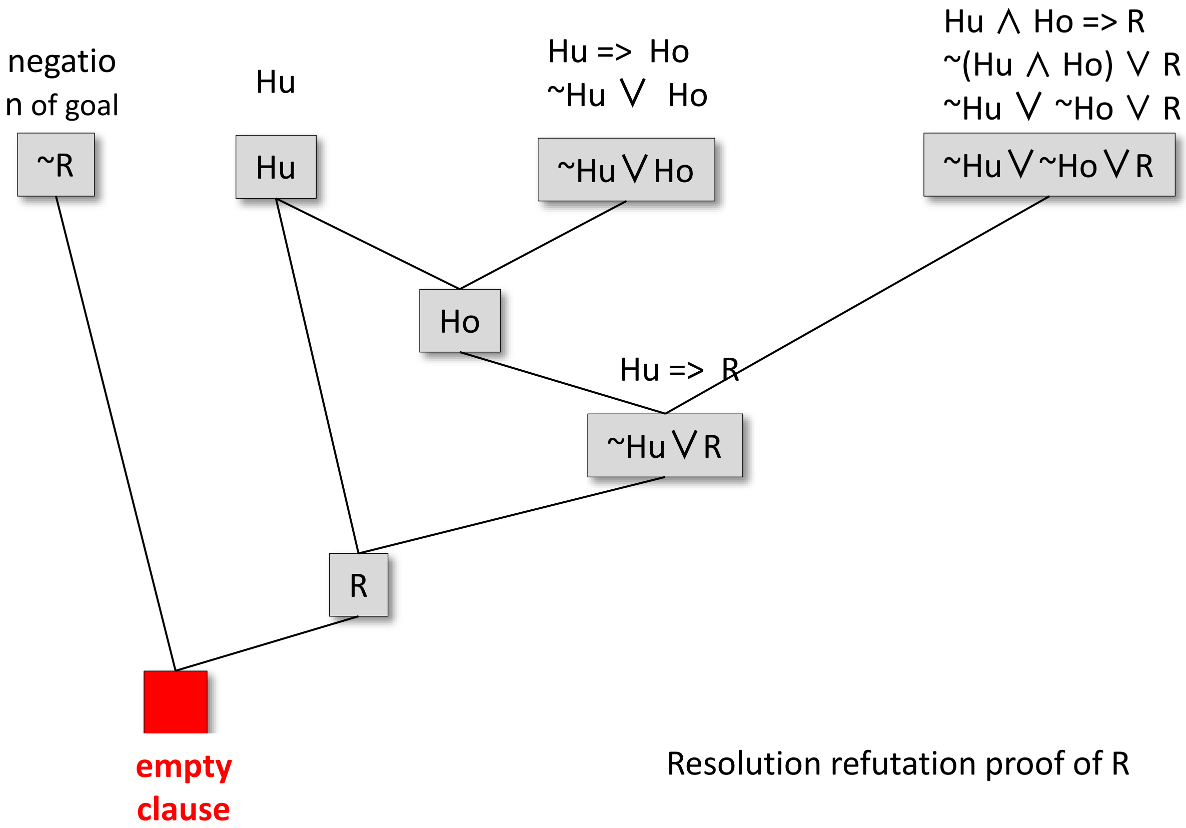
Resolution refutation

Procedure tries to prove a goal **P**

1. Add negation of goal to the KB, **~P**
2. Convert all sentences in KB to CNF
3. Find pairs of sentences with complementary literals that have not yet been resolved
4. If there are no pairs stop else resolve each pair, adding the result to the KB and go to 2

- If we get an empty clause (i.e., a contradiction) then **P** follows from the KB
  - e.g., resolving **X** with **~X** results in an empty clause
- If not, conclusion can't be proved from the KB

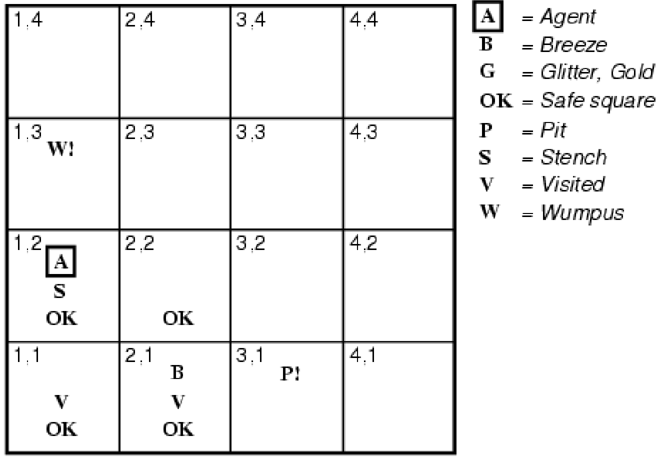
Proving it’s raining with refutation resolution



Resolution refutation proof of R

Hunt the Wumpus domain

- Some atomic propositions:
  - A12 = agent is in call (1,2)
  - S12 = There's a stench in cell (1,2)
  - B34 = There's a breeze in cell (3,4)
  - W22 = Wumpus is in cell (2,2)
  - V11 = We've visited cell (1,1)
  - OK11 = cell (1,1) is safe
  - …
- Some rules:
  - $\neg S22 \rightarrow \neg W12 \wedge \neg W23 \wedge \neg W32 \wedge \neg W21$
  - $S22 \rightarrow W12 \vee W23 \vee W32 \vee W21$
  - $B22 \rightarrow P12 \vee P23 \vee P32 \vee P21$
  - $W22 \rightarrow S12 \wedge S23 \wedge S32 \wedge W21$
  - $W22 \rightarrow \neg W11 \wedge \neg W21 \wedge \ldots \neg W44$
  - $A22 \rightarrow V22$
  - $A22 \rightarrow \neg W11 \wedge \neg W21 \wedge \ldots \neg W44$
  - $V22 \rightarrow OK22$



If there's no stench in cell 2,2 then the Wumpus isn't in cell 21, 23 32 or 21

Hunt the Wumpus domain

- Eight symbols for each cell, i.e.: A11, B11, G11, OK11, P11, S11, V11, W11
- Lack of variables requires giving similar rules for each cell!
- Ten rules for each:

A11 → …

V11 → …

P11 → …

¬P11 → …

W11 → …

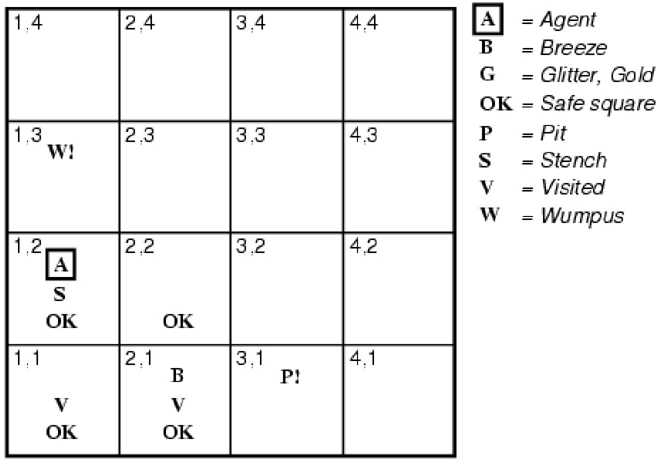
¬W11 → …

S11 → …

¬S11 → …

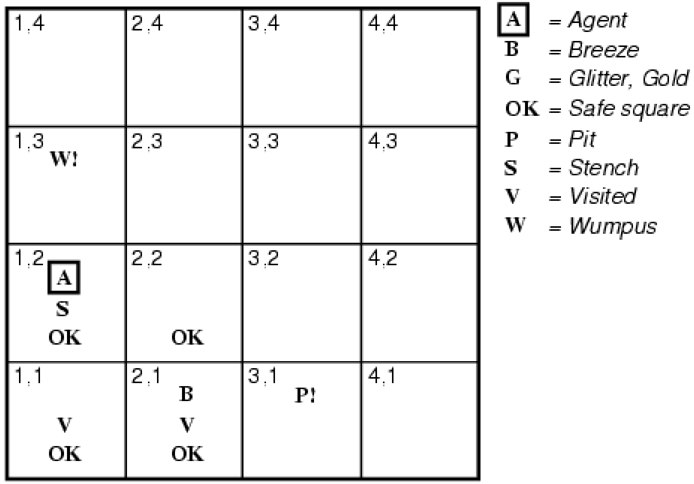
B11 → …

¬B11 → …



- 8 symbols for 16 cells => 128 symbols
- 2^{128} possible models ☹
- Must do better than brute force

After third move



- We can prove that the Wumpus is in (1,3) using these four rules
- See R&N section 7.5

(R1) $\neg S11 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W21$

(R2) $\neg S21 \rightarrow \neg W11 \wedge \neg W21 \wedge \neg W22 \wedge \neg W31$

(R3) $\neg S12 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W22 \wedge \neg W13$

(R4) $S12 \rightarrow W13 \vee W12 \vee W22 \vee W11$

Proving W13: Wumpus is in cell 1,3

Apply **MP** with $\neg$S11 and R1:

$\neg$ W11 $\wedge$ $\neg$ W12 $\wedge$ $\neg$ W21

Apply **AE**, yielding three sentences:

$\neg$ W11, $\neg$ W12, $\neg$ W21

Apply **MP** to ~S21 and R2, then apply **AE**:

$\neg$ W22, $\neg$ W21, $\neg$ W31

Apply **MP** to S12 and R4 to obtain:

W13 $\vee$ W12 $\vee$ W22 $\vee$ W11

Apply **UR** on (W13 $\vee$ W12 $\vee$ W22 $\vee$ W11) and $\neg$W11:

W13 $\vee$ W12 $\vee$ W22

Apply **UR** with (W13 $\vee$ W12 $\vee$ W22) and $\neg$W22:

W13 $\vee$ W12

Apply **UR** with (W13 $\vee$ W12) and $\neg$W12:

W13

QED

(R1) $\neg$S11 $\rightarrow$ $\neg$W11 $\wedge$ $\neg$ W12 $\wedge$ $\neg$ W21

(R2) $\neg$ S21 $\rightarrow$ $\neg$W11 $\wedge$ $\neg$ W21 $\wedge$ $\neg$ W22 $\wedge$ $\neg$ W31

(R3) $\neg$ S12 $\rightarrow$ $\neg$W11 $\wedge$ $\neg$ W12 $\wedge$ $\neg$ W22 $\wedge$ $\neg$ W13

(R4) S12 $\rightarrow$ W13 $\vee$ W12 $\vee$ W22 $\vee$ W11

Rule Abbreviation

MP: modes ponens

AE: and elimination

R: unit resolution

Propositional Wumpus problems

- Lack of variables prevents general rules, e.g.:
  - $\forall$ x, y V(x,y) → OK(x,y)
  - $\forall$ x, y S(x,y) → W(x-1,y) $\vee$ W(x+1,y) …
- Change of KB over time difficult to represent
  - In classical logic; a fact is true or false for all time
  - A standard technique is to index dynamic facts with the time when they're true
    - A(1, 1, 0)   *# agent was in cell 1,1 at time 0*
    - A(2, 1, 1) *# agent was in cell 2,1 at time 1*
  - Thus we have a separate KB for every time point

PL is a weak KR language

- Hard to identify *individuals* (e.g., Mary, 3)
- Can’t directly represent properties of individuals or relations between them (e.g., “Bill age 24”)
- Generalizations, patterns, regularities hard to represent (e.g., “all triangles have 3 sides”)
- First-Order Logic (FOL) represents this informa-tion via **relations**, **variables** & **quantifier**s, e.g.,
  - *John loves Mary: loves(John, Mary)*
  - *Every elephant is gray:* ∀ x (elephant(x) → gray(x))
  - *There is a black swan:* ∃ x (swan(X) ^ black(X))

Propositional logic summary

- **Inference**: deriving new sentences from old
  - **Sound** inference derives true conclusions given true premises
  - **Complete** inference derives all true conclusions from premises
- Different logics make different **commitments** about what world is made of and kinds of beliefs we can have
- **Propositional logic** commits only to existence of facts that may or may not be the case
  - Simple syntax & semantics illustrates inference process
  - Sound, complete and fast proof procedures
  - It can be impractical or cumbersome for many worlds