Propositional and First-Order Logic

Chapter 7.4–7.8, 8.1–8.3, 8.5

Some material adopted from notes by Andreas Geyer-Schulz and Chuck Dyer

Logic roadmap overview

- Propositional logic
  - Problems with propositional logic
- First-order logic
  - Properties, relations, functions, quantifiers, …
  - Terms, sentences, wffs, axioms, theories, proofs, …
  - Extensions to first-order logic
- Logical agents
  - Reflex agents
  - Representing change: situation calculus, frame problem
  - Preferences on actions
  - Goal-based agents

Disclaimer

> "Logic, like whiskey, loses its beneficial effect when taken in too large quantities."
>
> \- *Lord Dunsany*

Propositional Logic: Review

Big Ideas

- Logic is a great knowledge representation language for many AI problems
- **Propositional logic** is the simple foundation and fine for many AI problems
- **First order logic** (FOL) is much more expressive as a knowledge representation (KR) language and needed for many AI problems
- **Variations** on FOL are common: horn logic, higher order logic, three-valued logic, probabilistic logic, fuzzy logic, etc.

Propositional logic syntax

- **Logical constants**: true, false
- **Propositional symbols**: P, Q, ... (aka **atomic sentences**)
- **Parentheses**: (…)
- **Sentences** are build with **connectives**:

$\land$	and	[conjunction]
$\lor$	or	[disjunction]
$\Rightarrow$	implies	[implication/conditional/if]
$\Leftrightarrow$	is equivalent	[biconditional/iff]
$\neg$	not	[negation]

- **Literal**: atomic sentence or their negation: P, $\neg$P

Propositional logic syntax

- Simplest logic language in which a user specifies
  - Set of propositional symbols (e.g., P, Q)
  - 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 **$\neg$S** is a sentence
  - If S is a sentence, then **(S)** is a sentence
  - If S and T are sentences, then so are **(S $\vee$ T), (S $\wedge$ T), (S $\rightarrow$ T),** and **(S $\leftrightarrow$ T)**
  - Sentence result from a finite number of applications of the rules

Examples of PL sentences

- $(P \wedge Q) \rightarrow R$

  "If it is hot and humid, then it is raining"
- $Q \rightarrow P$

  "If it is humid, then it is hot"
- Q

  "It is humid."
- We're free to choose better symbols, e.g.:

  Hot = "It is hot"

  Humid = "It is humid"

  Raining = "It is raining"

Some terms

- The meaning or **semantics** of a sentence determines its **interpretation**
- Given the truth values of all symbols in a sentence, it can be ***evaluated*** to determine its **truth value** (True or False)
- A **model** for a KB is a *possible world* – an assignment of truth values to propositional symbols that makes each KB sentence true

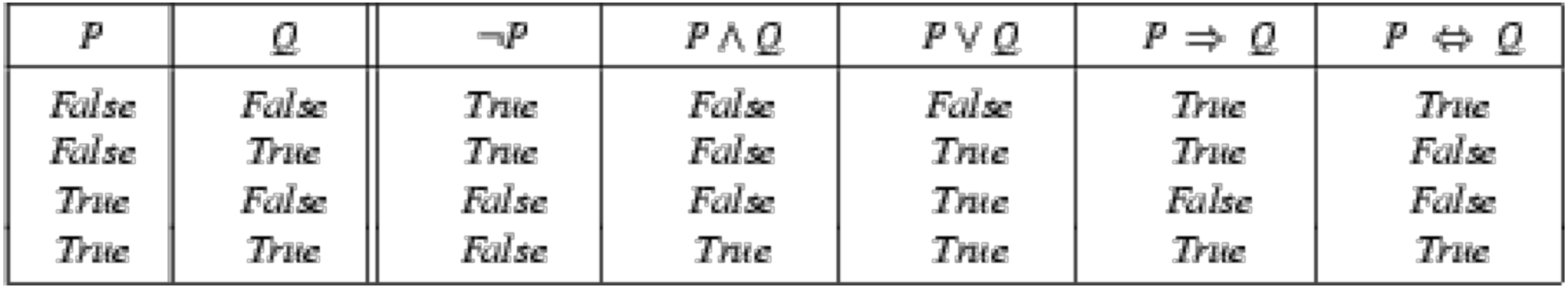
More terms

- A **valid sentence** or **tautology** is 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”
- An **inconsistent sentence** or **contradiction** is 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 entails Q**, written P |= Q, means that whenever P is True, so is Q
  - In all models in which P is true, Q is also true

Truth tables

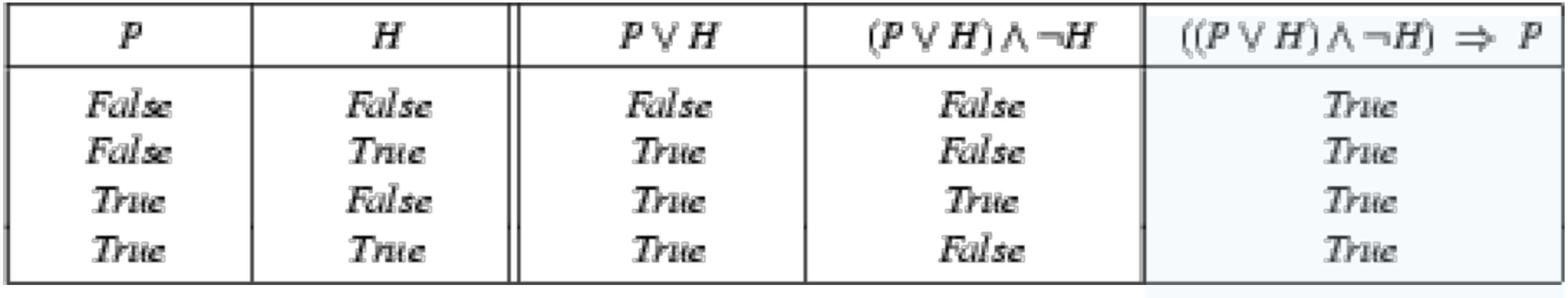
- Used to define meaning of logical connectives
- and to determine when a complex sentence is true given values of its symbols

Truth tables for the five logical connectives



P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	*False*	*True*	*False*	*False*	*True*	*True*
False	*True*	*True*	*False*	*True*	*True*	*False*
True	*False*	*False*	*False*	*True*	*False*	*False*
True	*True*	*False*	*True*	*True*	*True*	*True*

Example of a truth table used for a complex sentence



P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
False	*False*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
True	*True*	*True*	*False*	*True*

The implies connective: P → Q

- → is a *logical connective*
- So *P*→*Q* is a **logical sentence** and has a truth value, i.e., is either true or false
- If we add this sentence to a KB, it can be used by an inference rule, *Modes Ponens*, to derive/infer/prove Q if P is also in the KB
- Given a KB where P=True and Q=True, we can derive/infer/prove that P→Q is True
- Note: P→ Q is equivalent to ~P∨Q

$P \to Q$

- When is $P \to Q$ true? Check all that apply
  - ☐ P=Q=true
  - ☐ P=Q=false
  - ☐ P=true, Q=false
  - ☐ P=false, Q=true

P → 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 →
- Note: in FOL it's much harder to prove that a conditional true, e.g., prime(x) → odd(x)

Models for a KB

- KB: [PvQ, P→R, Q→R]
- What are the sentences?

  s1: PVQ
  s2: P→R
  s3: Q→R
- What are the propositional variables?

  P, Q, R
- What are the candidate models?

1) Consider all possible assignments of T|F to P, Q, R
2) Check truth tables for consistency, eliminating any row that does not make every KB sentence true

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: [PvQ, P→R, Q→R]
- What are the sentences?

  s1: PVQ
  s2: P→R
  s3: Q→R

- What are the propositional variables?

  P, Q, R

- What are the candidate models?

  1) Consider all possible assignments of T|F to P, Q, R
  2) Check truth tables for consistency, eliminating any row that does not make every KB sentence true

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 consistent with KB
- R is true in all of them
- Therefore R is true and can be added to the KB

Inference rules

- **Logical inference** creates new sentences that logically follow from a set of sentences (KB)
- An inference rule is **sound** if every sentence X 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	**$A \vee B, \neg B \vee C$**	**$A \vee C$**

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

Resolution

- 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, it's just a matter of rewriting its sentences using standard tautologies, e.g.:
  - $P \rightarrow Q \equiv \sim P \vee Q$

Resolution Example

Tautologies

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

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

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

- 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:

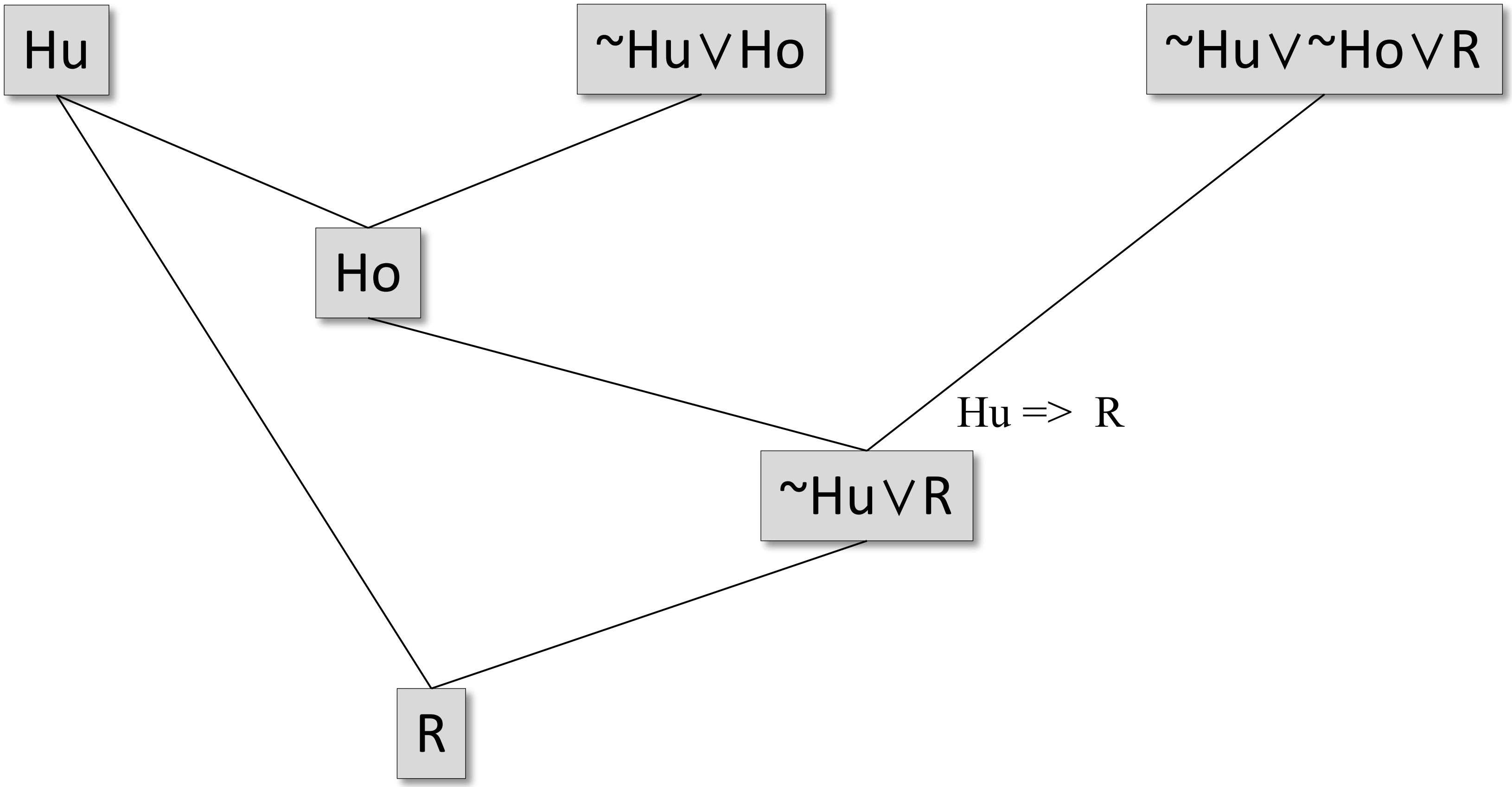
  $\sim P \vee S$ *(i.e., $P \rightarrow S$)*
- New KB: $[\sim P \vee Q\ ,\ \sim Q \vee R,\ \sim Q \vee S,\ \sim P \vee R,\ \sim P \vee S]$

Proving it’s raining with resolution

- 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 (2)



Hu

Hu => Ho

Hu ∧ Ho => R
~(Hu ∧ Ho) ∨ R
~Hu ∨ ~Ho ∨ R

Hu

~Hu∨Ho

~Hu∨~Ho∨R

Ho

Hu => R

~Hu∨R

R

A simple proof procedure

This procedure generates new sentences from a KB

1. Convert all sentences in the KB to CNF
2. Find all pairs of sentences in KB with complementary literals 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?

Resolution refutation

1. Add negation of goal to the KB
2. Convert all sentences in KB to CNF
3. Find all pairs of sentences in KB 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 derived an empty clause (i.e., a contradiction) then the conclusion follows from the KB
- If we did not, the conclusion cannot be proved from the KB

Problems with Propositional Logic

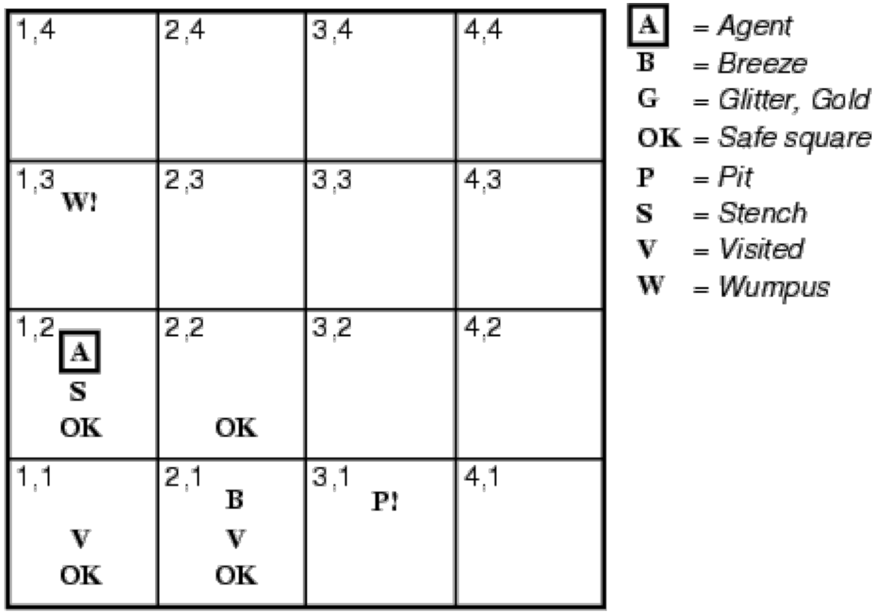
Propositional logic: pro and con

- **Advantages**
  - –Simple KR language good for many problems
  - –Lays foundation for higher logics (e.g., FOL)
  - –Reasoning is decidable, though NP complete; efficient techniques exist for many problems
- **Disadvantages**
  - –Not expressive enough for most problems
  - –Even when it is, it can very “un-concise”

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 height tall”)
- Generalizations, patterns, regularities hard to represent (e.g., “all triangles have 3 sides”)
- First-Order Logic (FOL) represents this information via **relations**, **variables** & **quantifier**s, e.g.,
  - *Every elephant is gray:* $\forall$ x (elephant(x) $\rightarrow$ gray(x))
  - *There is a black swan:* $\exists$ x (swan(X) ^ black(X))

Hunt the Wumpus domain



- Some atomic propositions:

  A12 = AGENT IS IN CELL (“1,2)

  S12 = There is a stench in cell (1,2)

  B34 = There is 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:

  $\neg S22 \rightarrow \neg W12 \wedge \neg W23 \wedge \neg W32 \wedge \neg W21$

  $S22 \rightarrow W12 \vee W23 \vee W32 \vee W21$

  $B22 \rightarrow P12 \vee P23 \vee P32 \vee P21$

  $W22 \rightarrow S12 \wedge S23 \wedge S23 \wedge W21$

  $W22 \rightarrow \neg W11 \wedge \neg W21 \wedge \ldots \neg W44$

  $A22 \rightarrow V22$

  $A22 \rightarrow \neg W11 \wedge \neg W21 \wedge \ldots \neg W44$

  $V22 \rightarrow OK22$

Hunt the Wumpus domain

- Eight variables 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 (I think) for each

A11 → …

V11 → …

P11 → …

¬P11 → …

W11 → …

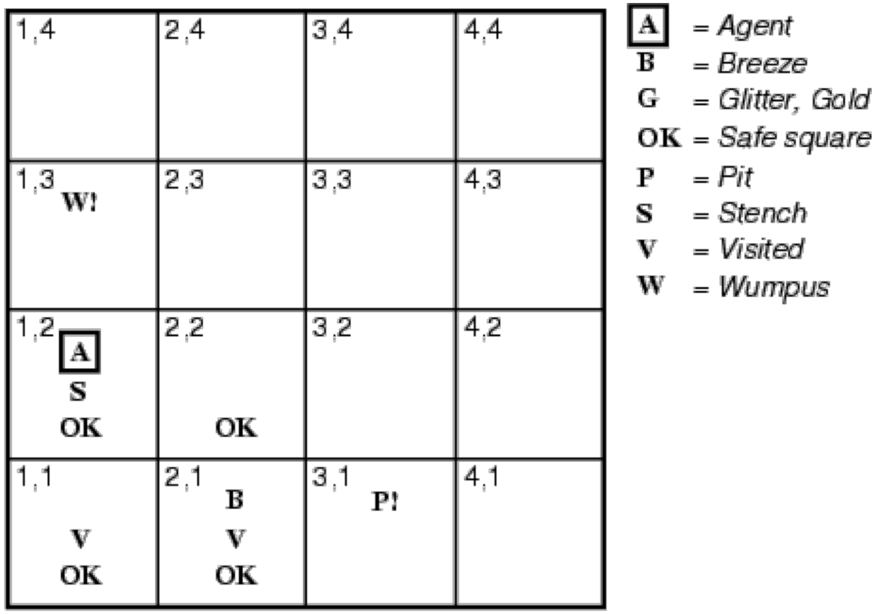
¬W11 → …

S11 → …

¬S11 → …

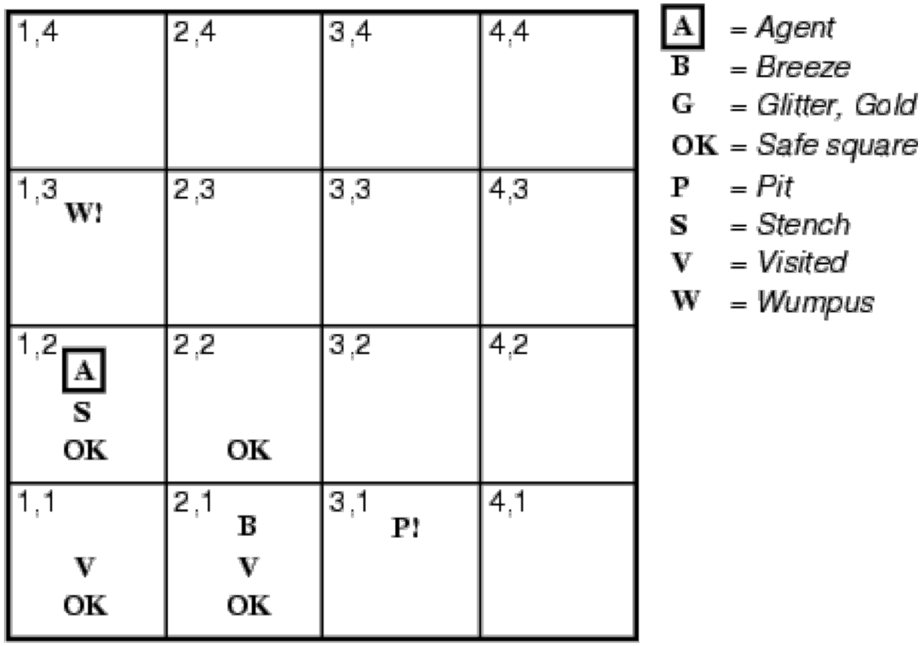
B11 → …

¬B11 → …



- 8 variables for 16 cells
- => 128 variables
- 2^{128} combinations ☹
- 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) $\neg S11 \rightarrow \neg W11 \land \neg W12 \land \neg W21$

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

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

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

Proving W13

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

Apply MP with $\neg S11$ and R1:

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

Apply And-Elimination to this, yielding 3 sentences:

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

Apply MP to ~S21 and R2, then apply And-elimination:

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

Apply MP to S12 and R4 to obtain:

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

Apply Unit Resolution on ($W13 \vee W12 \vee W22 \vee W11$) and $\neg W11$:

$W13 \vee W12 \vee W22$

Apply Unit Resolution with ($W13 \vee W12 \vee W22$) and $\neg W22$:

$W13 \vee W12$

Apply Unit Resolution with ($W13 \vee W12$) and $\neg W12$:

W13

QED

Propositional Wumpus problems

- Lack of variables prevents stating more general rules, like these:
  - $\forall$ 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 the KB over time is 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, t0)
  - Thus we have a separate KB for every time point

Propositional logic summary

- **Inference**: process of deriving new sentences from old
  - **Sound** inference derives true conclusions given true premises
  - **Complete** inference derives all true conclusions from a set of premises
- **Valid sentence:** true in all worlds under all interpretations
- If an implication sentence can be shown to be valid, then, given its premise, its consequent can be derived
- Different logics make different **commitments** about what the world is made of and the kind of beliefs we can have
- **Propositional logic** commits only to existence of facts that may or may not be the case in the world being represented
  - Simple syntax and semantics suffices to illustrate the process of inference
  - Propositional logic can become impractical, even for very small worlds