CSC 442: ARTIFICIAL INTELLIGENCE

LEC 09: PROPOSITIONAL LOGIC

We desire to create an intelligent agent,
one which is capable of problem solving.

WHAT WE'VE LEARNED SO FAR:

- a problem can be represented as a tuple
- search algorithms can be used for problem solving
- search algorithms can be compared according to complexity, completeness, and optimality.

WHAT WE'VE LEARNED SO FAR:

- tree-based problems seem easier to search than graph-based problems
- for small-enough problems, we saw how to find the best solutions
- we even saw how to solve (some) problems where we take turns with an adversary
- in some problems we have additional heuristics we can use to guide the search
- still up against a fundamental problem: exponential time requirements

WHAT WE'VE LEARNED SO FAR:

- Still up against a fundamental problem:
  - exponential time requirements
- For very large problems, we saw techniques to search for "good enough" solutions;
- This included local search as well as heuristic minimax.

WHAT WE'VE LEARNED SO FAR:

- All of these problems relied on the fundamental state-space model:
  - an initial state
  - a goal/utility/cost function
  - a transition model
- last time, we saw how for some problems we can dramatically simplify

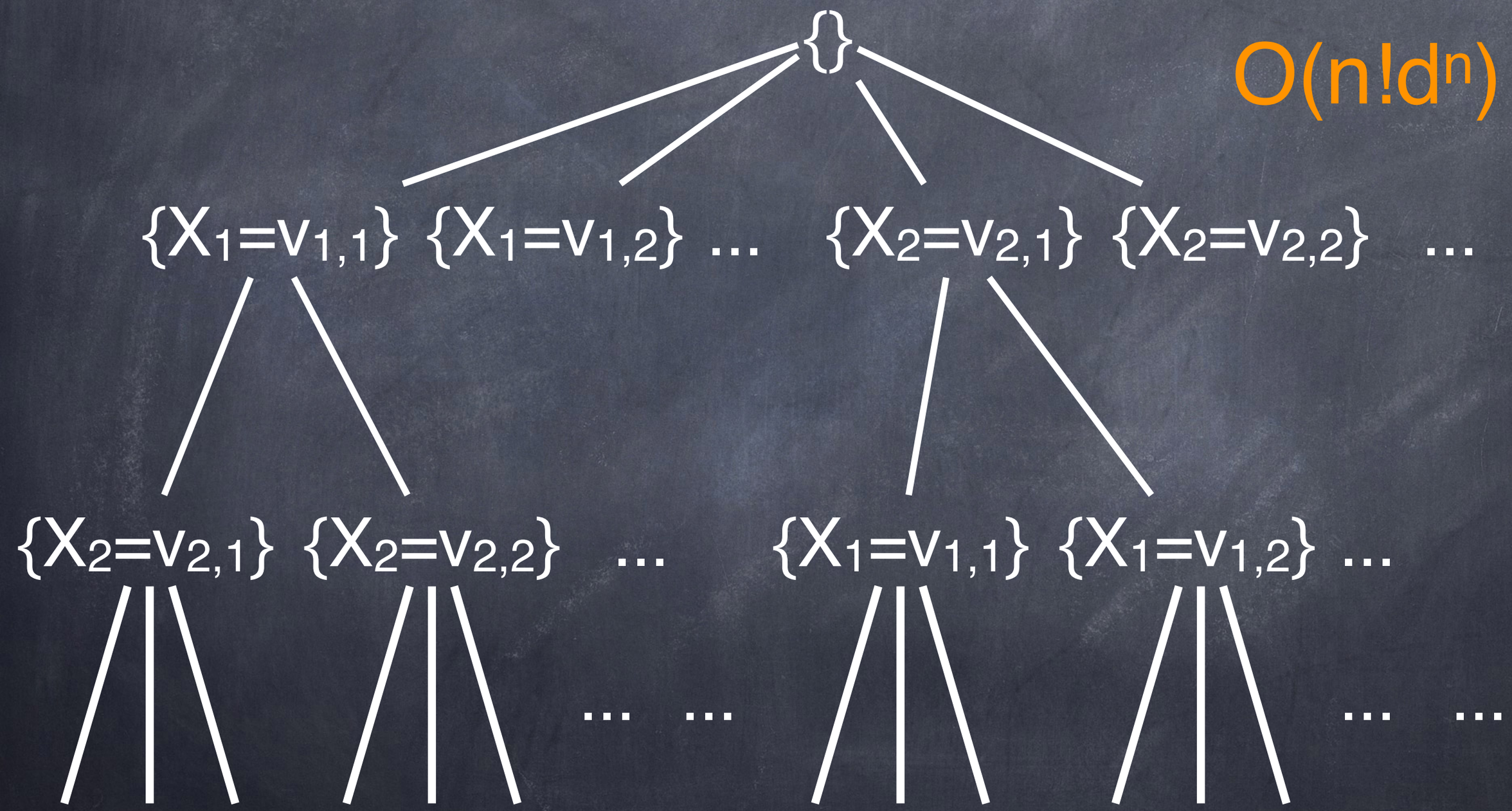
WHAT WE'VE LEARNED SO FAR:

- Last time, we saw how for some problems we can dramatically simplify them by imposing a constraint-space problem structure.

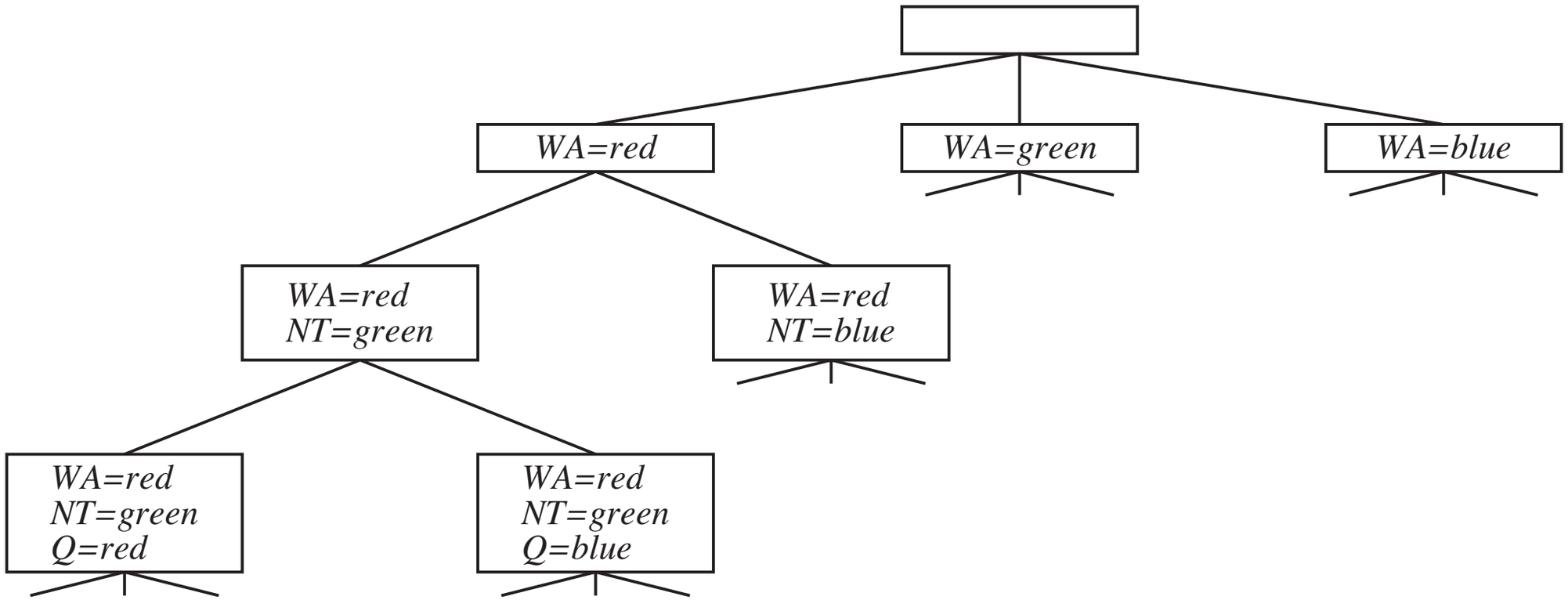
Constraint Satisfaction Problem (CSP)

- X: Set of variables $\{ X_1, ..., X_n \}$
- D: Set of domains $\{ D_1, ..., D_n \}$
  - Each D_i : set of values $\{ v_1, ..., v_k \}$
- C: Set of constraints $\{ C_1, ..., C_m \}$
- Solution: Assign to each X_i a value from D_i such that all the C_i are satisfied

Search for CSPs



$O(n!d^n)$



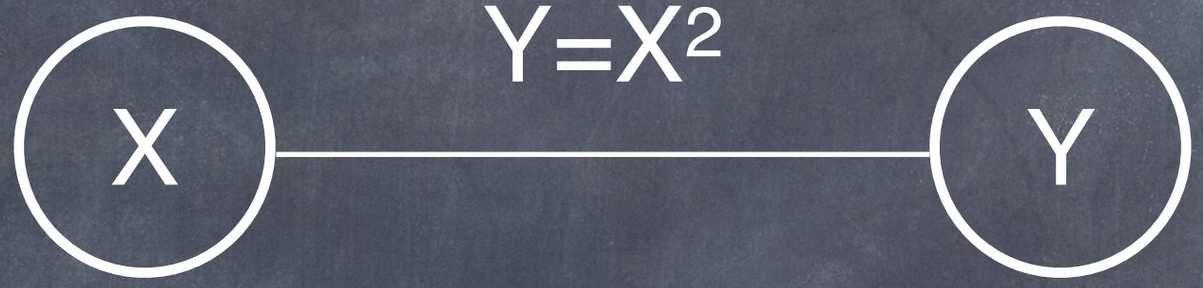
n levels (one per variable), at most d nodes per level:

$$O(d^n)$$

Arc Consistency

X_m is arc-consistent w.r.t. X_n if
for every value in the domain D_m,
there is some value in the domain D_n
that satisfies the binary constraint on the arc (X_m, X_n)

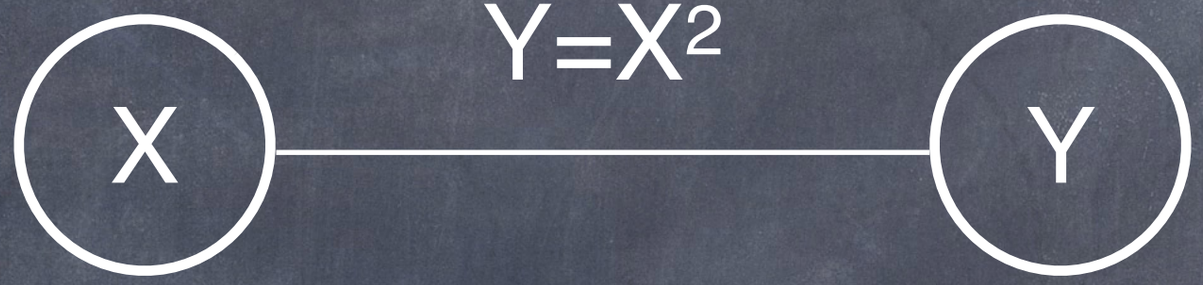
Arc Consistency



{ 0,1,2,3,4,5,6,7,8,9 } { 0,1,2,3,4,5,6,7,8,9 }

possible assignments: 10x10 = 100

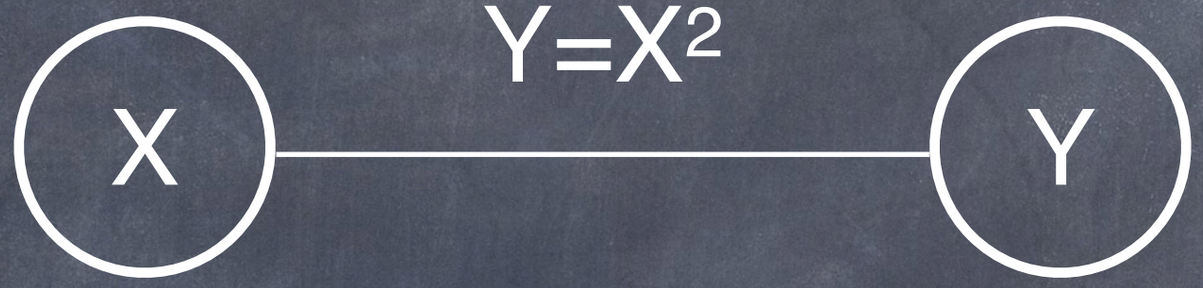
Arc Consistency



{ 0,1,2,3,4,5,6,7,8,9 }

{ 0,1,2,3,4,5,6,7,8,9 }

Arc Consistency

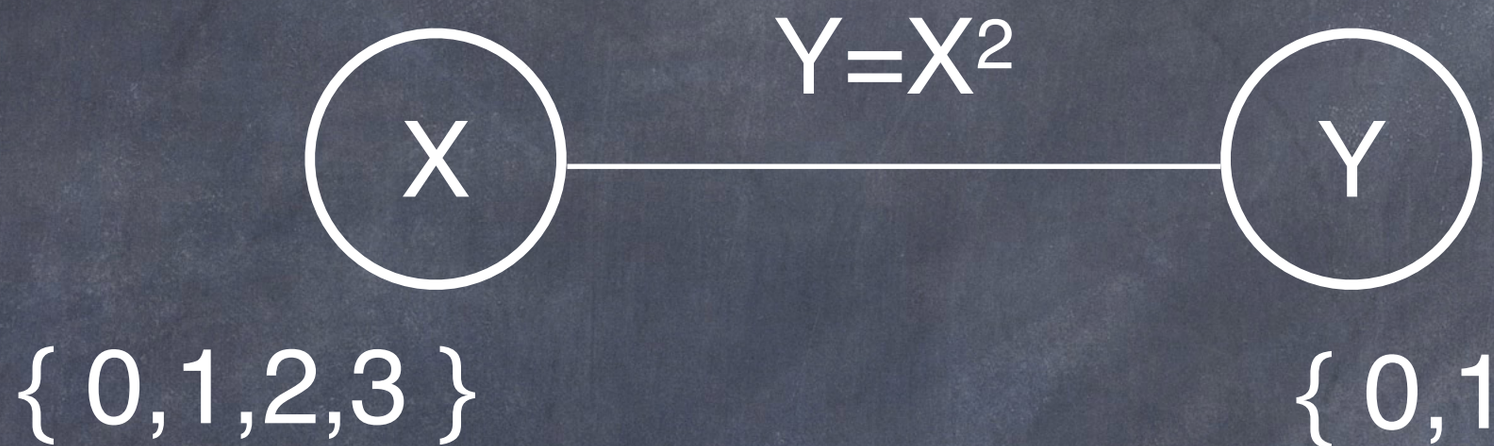


{ 0,1,2,3 }

{ 0,1,2,3,4,5,6,7,8,9 }

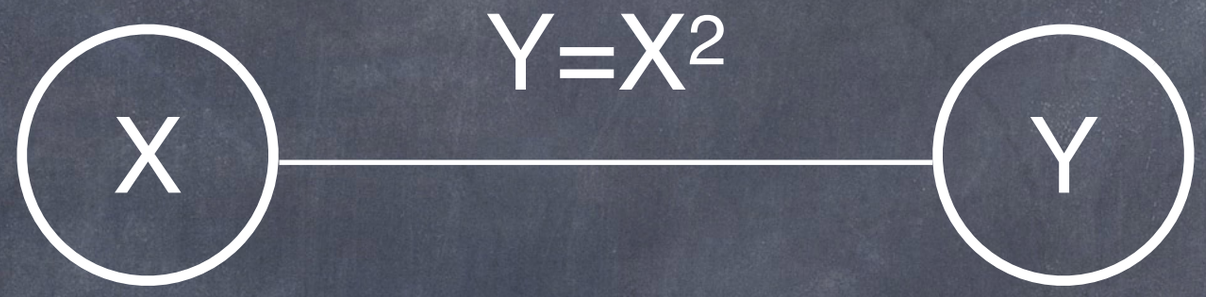
X arc-consistent with respect to Y

Arc Consistency



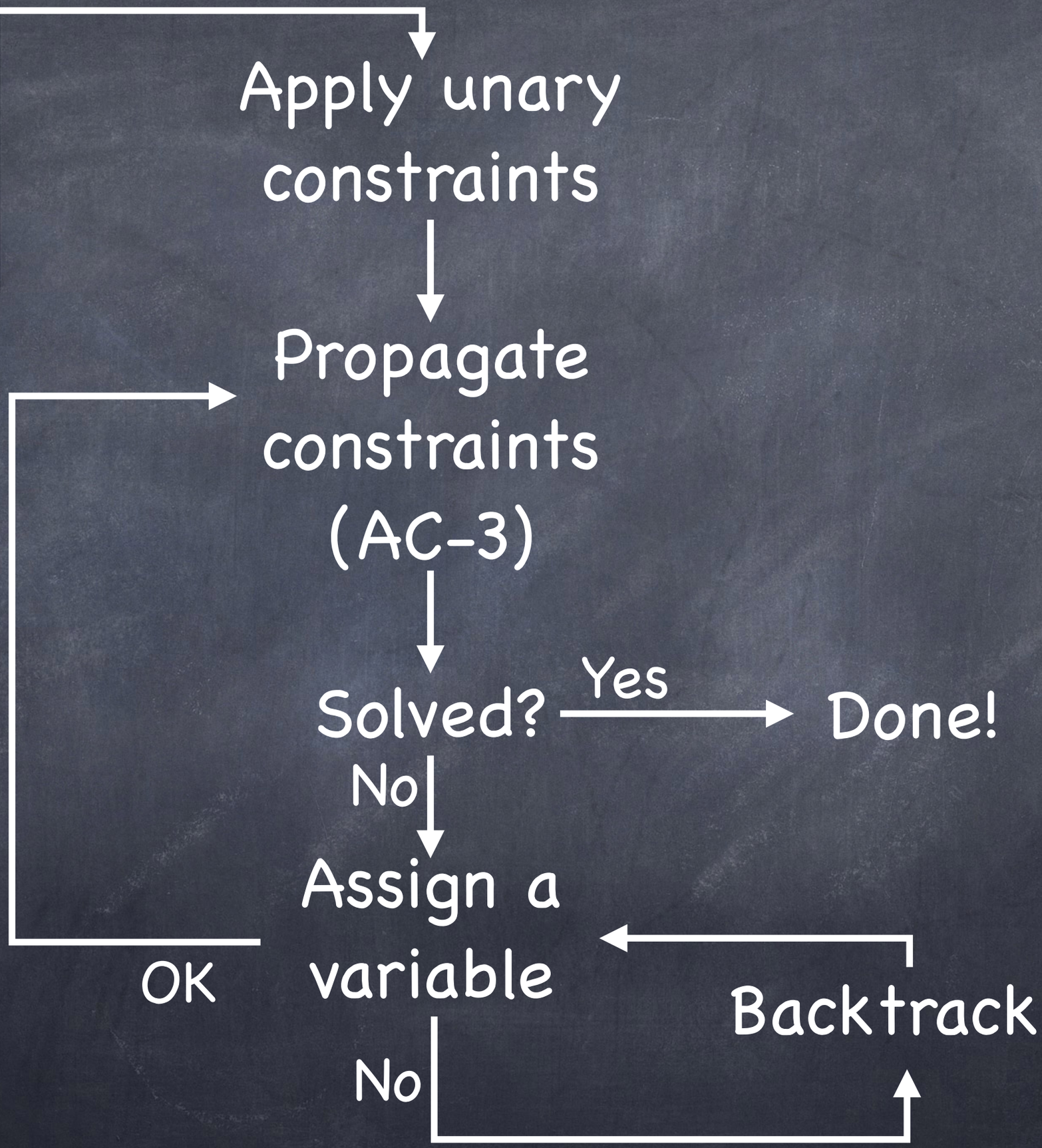
Y arc-consistent with respect to X

Arc Consistency



possible assignments: 4×4 = 16

CSP:
- Variables
- Domains
- Constraints



NEW TOPIC — PROPOSITIONAL LOGIC

- Constraints, domains, and variables are a kind of **knowledge representation**.

- Let's look at a different kind of representation:
  - **Propositional Logic**

ABOUT PROPOSITIONAL LOGIC

- Propositional logic is a **knowledge representation language**, with history dating back to 3rd century BC.
- A propositional logic system has a **knowledge base** which is a database of propositions.
- The available actions are to **tell** the knowledge base a new proposition, or to **ask** the agent if a proposition is **entailed** by the knowledge base.

ADVANTAGES OF PROPOSITIONAL LOGIC

- Propositional logic is important because it provides a mechanism for knowledge representation which is *factorable* (unlike simple states) and amenable to **inference**.

. . . BY THE WAY

- Logical inference has long been associated with intelligence because of its applicability to domains of human interest – medicine, law, philosophy, science, etc.

- Let's take a closer look at what makes a sentence from propositional logic a **well-formed formula**, also called the **syntax** of propositional logic.

- The fundamental building blocks of propositional logic are **propositions** – statements which must be either true or false.

- Propositions often correspond to whole sentences.

- Example propositions:
  - My name is Adam.
  - There is a monster in the next room.
  - 5 is a prime number.
  - Bella is a dog.

- The previous examples represent **atomic propositions**, because they are either true or false on their own, and do not have components.

- Atomic propositions are written with capital letters:
  - A – My name is Adam.
  - M – There is a monster in the next room.
  - P_5 – 5 is a prime number.
  - B – Bella is a dog.

- The language of propositional logic also allows formation of **complex propositions** by combining simpler propositions with **logical connectives**.

COMPLEX PROPOSITIONS

- My name is Adam **and** I am 35 years old.
- There is a monster in the adjacent room **or** there is gold in the adjacent room.
- Four is **not** a prime number.
- If there is a monster in the adjacent room **or** there is gold in the adjacent room, then today will be exciting.

LOGICAL CONNECTIVES

- $\wedge$ – conjunction (and)
- $\vee$ – disjunction (or)
- $\neg$ – negation (not)
- $\Rightarrow$ – implication (kind of like if-then)
- $\Leftrightarrow$ – biconditional

$$
\begin{array}{rcl}
Sentence & \rightarrow & AtomicSentence \mid ComplexSentence \\
AtomicSentence & \rightarrow & True \mid False \mid P \mid Q \mid R \mid \ldots \\
ComplexSentence & \rightarrow & (\ Sentence\ ) \mid [\ Sentence\ ] \\
& \mid & \neg\ Sentence \\
& \mid & Sentence \wedge Sentence \\
& \mid & Sentence \vee Sentence \\
& \mid & Sentence \Rightarrow Sentence \\
& \mid & Sentence \Leftrightarrow Sentence \\
\text{OPERATOR PRECEDENCE} & : & \neg, \wedge, \vee, \Rightarrow, \Leftrightarrow
\end{array}
$$

Figure 7.7 A BNF (Backus–Naur Form) grammar of sentences in propositional logic, along with operator precedences, from highest to lowest.

AIMA p244

Next up: propositional semantics.

PROPOSITIONAL SEMANTICS

- The semantics of a language are the meanings of the sentences in the language. For propositional logic, semantics correspond to identifying which propositions are **true** and which are **false**.

PROPOSITIONAL SEMANTICS

- A propositional **model** is an assignment of true or false to each atomic proposition in a given domain.
- So asking if an atomic proposition is true corresponds to dictionary lookup – if we have a model.

PROPOSITIONAL SEMANTICS

- Truth for complex sentences is evaluated compositionally based on specific rules associated with each logical connective.
- E.g., “P and Q” is true if “P” is true AND “Q” is true.
- It’s easier to show these rules with **truth tables**.

PROPOSITIONAL SEMANTICS

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

Figure 7.8 Truth tables for the five logical connectives. To use the table to compute, for example, the value of $P \vee Q$ when P is true and Q is false, first look on the left for the row where P is $true$ and Q is $false$ (the third row). Then look in that row under the $P \vee Q$ column to see the result: $true$.

AIMA p244

PROPOSITIONAL SEMANTICS

- A propositional formula which is necessarily true in all possible worlds (i.e., for all models) is called **logically valid** or a **tautology.**

- A propositional formula is is necessarily false in all possible worlds (i.e., for all models) is called **logically invalid** or **unsatisfiable.**

- A propositional formula which is true in some (but not necessary all) models is called **satisfiable** or **contingent.**

PROPOSITIONAL SEMANTICS

- The fundamental question of propositional logic is that of **entailment** – when the truth of one formula necessarily implies truth of another.

- Entailment is so important that it has special notation, where α entails β is written as:

$$\alpha \models \beta$$

- This means all models of α are also models of β !

- Alternatively, if we write M(x) to represent the set of all models of an arbitrary formula x, then

$$\alpha \models \beta \text{ if and only if } M(\alpha) \subseteq M(\beta)$$

- A **knowledge base (KB)** is a collection of facts which are all asserted to be true (e.g., a giant conjunction). For simplicity, we often write a generic knowledge base using the symbol delta – Δ

- An ideal logical agent would have a knowledge base describing the real world (including common sense!) and then would "ask" questions of the knowledge base by determining entailment.

- We stopped here on Monday, September 30th.

Last time:

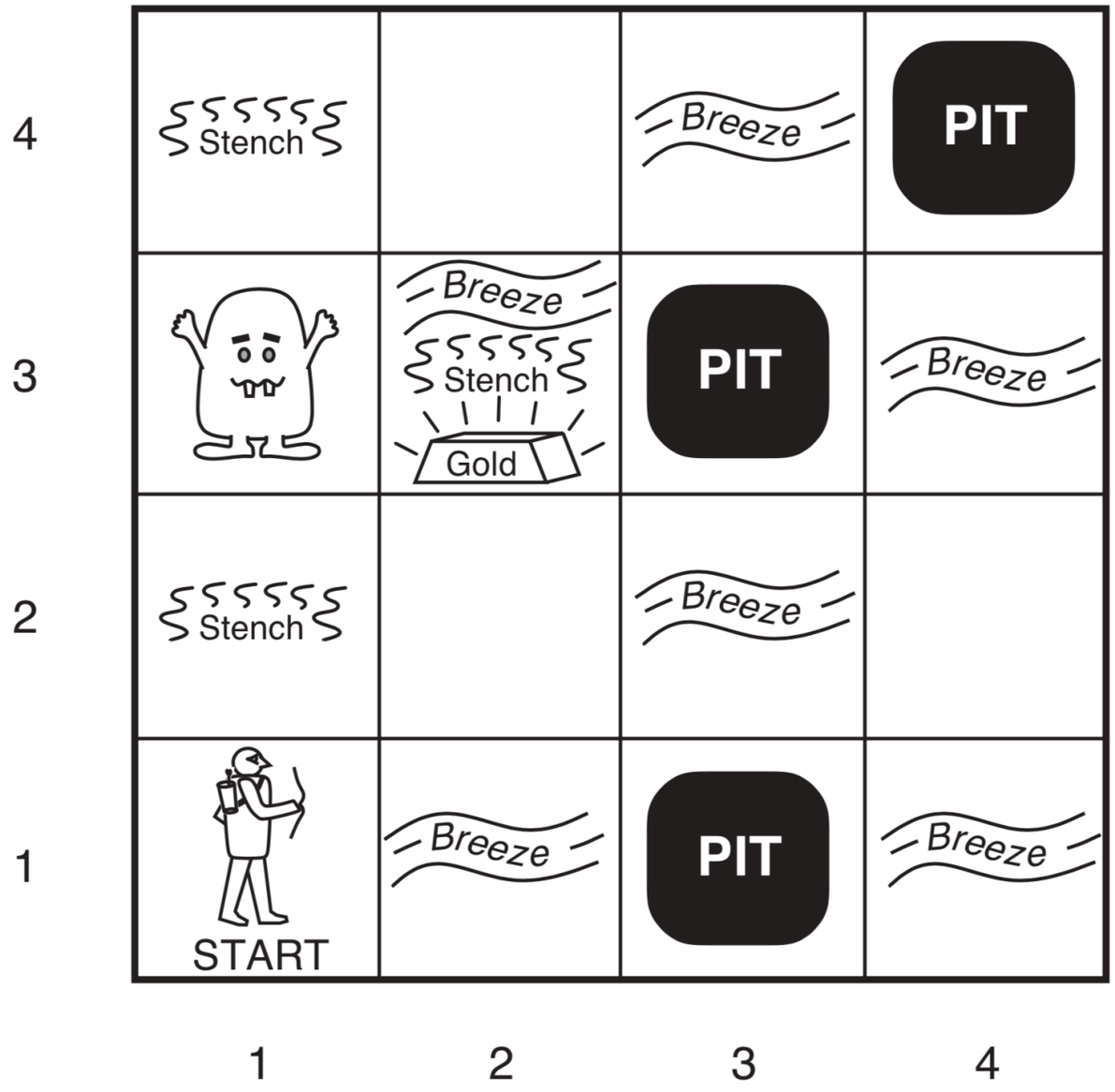
Propositional Logic

- syntax
  - propositions
  - logical connectives
  - well-formed formulas
    - atomic
    - complex
- semantics
  - models
  - compositional truth tables
  - entailment

- Extended example – The Wumpus World

THE WUMPUS WORLD: A SIMPLE TASK ENVIRONMENT

- P (1000,-1000,-1)
- E (NxN w/flavor)
- A (ULDR, S, P)
- S (S,Z,G,B,R)



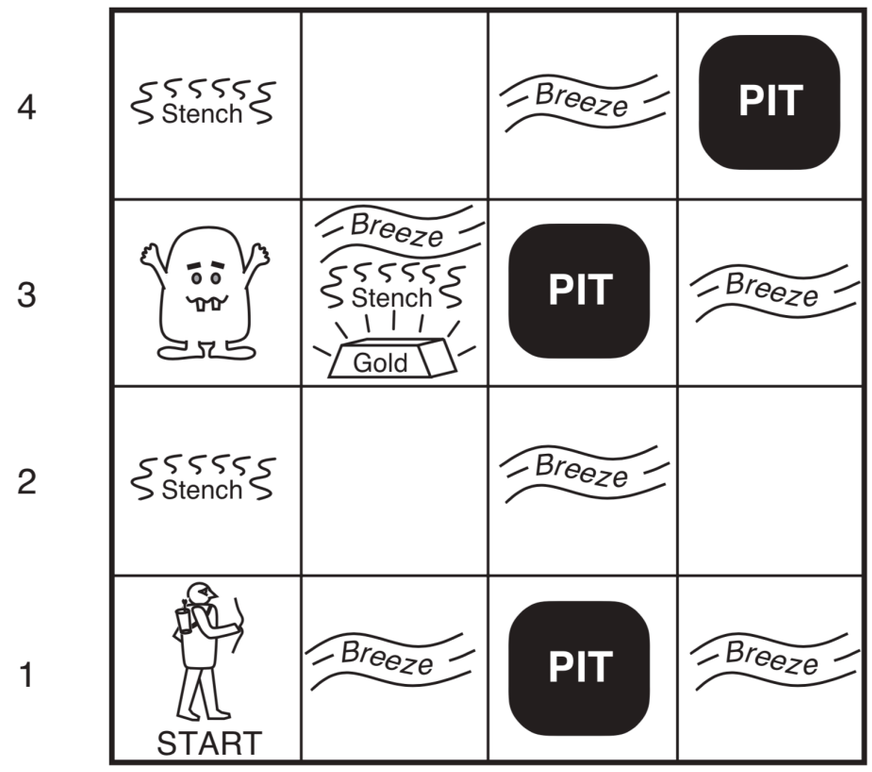
https://www.youtube.com/watch?v=xGVOw8gXl6Y

AIMA Fig 7.2, p238

A WUMPUS WORLD KNOWLEDGE BASE

$OK_{x,y}$ is true if [x,y] is a safe location.

$P_{x,y}$ is true if there is a pit in $[x, y]$.
$W_{x,y}$ is true if there is a wumpus in $[x, y]$, dead or alive.
$B_{x,y}$ is true if the agent perceives a breeze in $[x, y]$.
$S_{x,y}$ is true if the agent perceives a stench in $[x, y]$.



5*(NxN) propositions!

Rules (axioms) derived from *schemas:*

(e.g. pits are breezy, gold glitters, wumpus stinks)

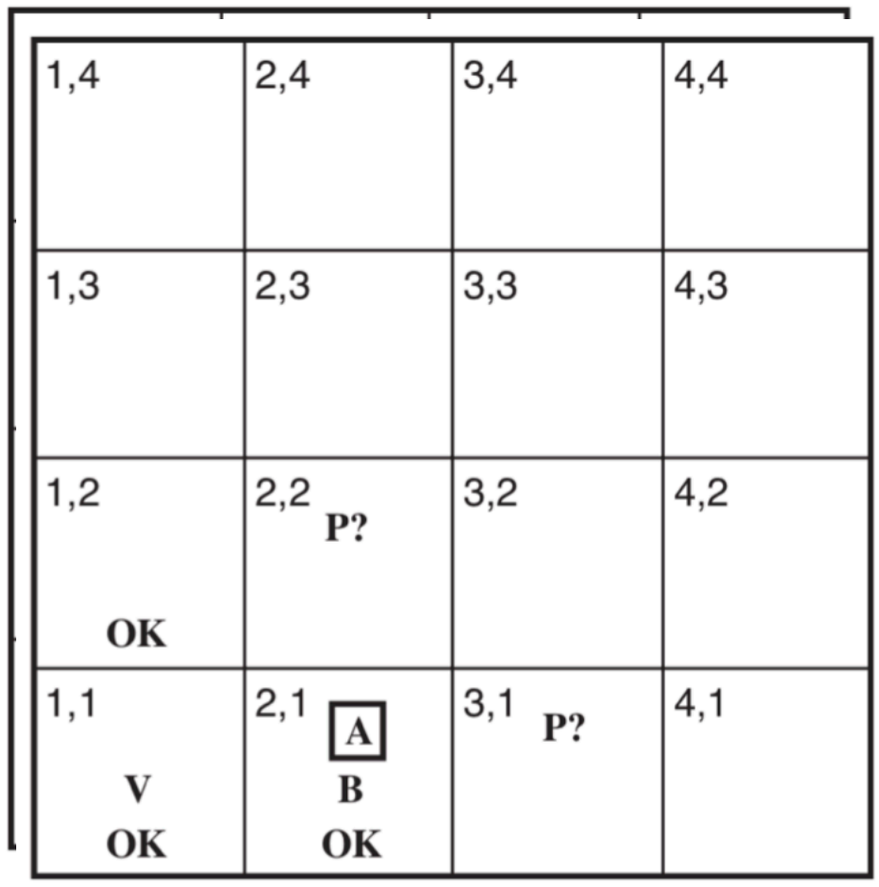
$$R_1 : \quad \neg P_{1,1} .$$
$$R_2 : \quad B_{1,1} \quad \Leftrightarrow \quad (P_{1,2} \vee P_{2,1}) .$$
$$R_3 : \quad B_{2,1} \quad \Leftrightarrow \quad (P_{1,1} \vee P_{2,2} \vee P_{3,1}) .$$
$$R_4 : \quad \neg B_{1,1} .$$
$$R_5 : \quad B_{2,1} .$$

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

$[None, None, None, None, None]$

Stench, Breeze, Glitter, Scream, Bump

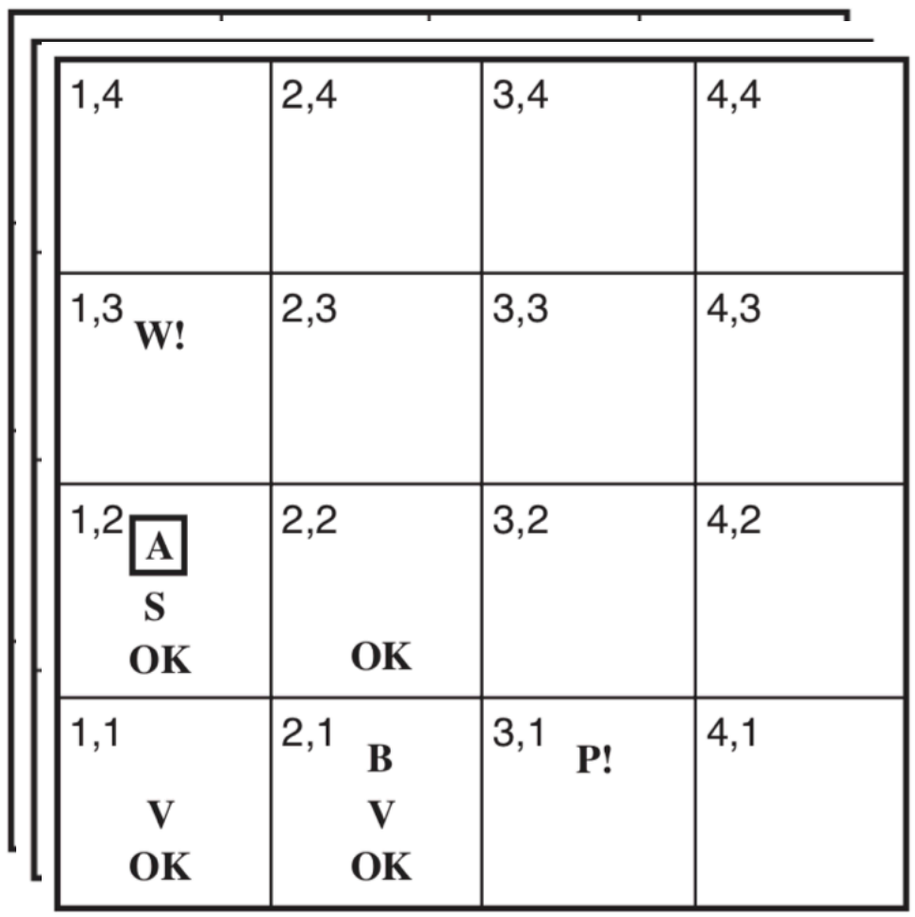
- We must explore (search) environment to find gold.
- The environment is partially observable.
- Some actions are dangerous.
- Can we use logic to solve?



$[None, Breeze, None, None, None]$.

Stench, Breeze, Glitter, Scream, Bump

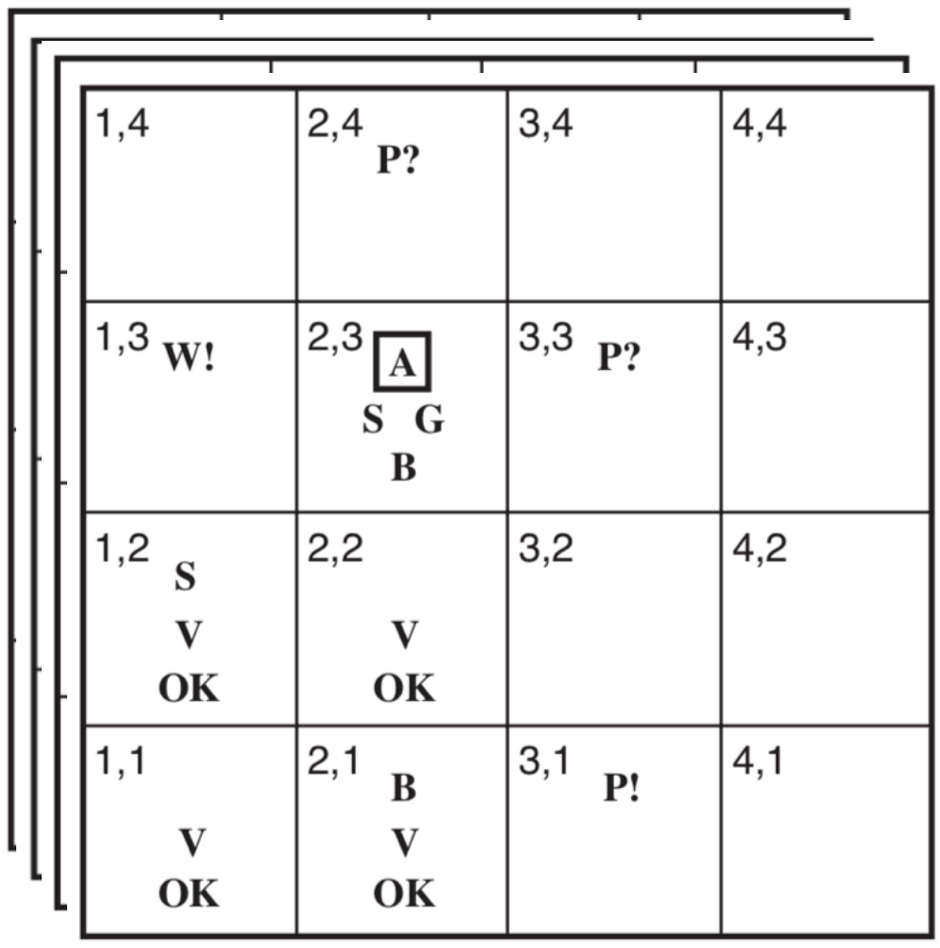
- We must explore (search) environment to find gold.
- The environment is partially observable.
- Some actions are dangerous.
- Can we use logic to solve?



$[Stench, None, None, None, None]$

Stench, Breeze, Glitter, Scream, Bump

- We must explore (search) environment to find gold.
- The environment is partially observable.
- Some actions are dangerous.
- Can we use logic to solve?



$[Stench, Breeze, Glitter, None, None]$

Stench, Breeze, Glitter, Scream, Bump

- We must explore (search) environment to find gold.
- The environment is partially observable.
- Some actions are dangerous.
- Can we use logic to solve?

A TRUTH TABLE FOR THE WUMPUS WORLD

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	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*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
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*

Figure 7.9 A truth table constructed for the knowledge base given in the text. *KB* is true if R_1 through R_5 are true, which occurs in just 3 of the 128 rows (the ones underlined in the right-hand column). In all 3 rows, $P_{1,2}$ is false, so there is no pit in [1,2]. On the other hand, there might (or might not) be a pit in [2,2].

PROPOSITIONAL INFERENCE

- Inference – making implicit knowledge explicit.
- Two major approaches:
  - Model Checking
  - Theorem Proving

INFERENCE VIA MODEL CHECKING

- Simple algorithm based on a brute-force application of the definition of entailment to determine if $\alpha \models \beta$
- Enumerate all 2^k possible worlds (i.e., truth combinations of k atomic propositions).
- If every model of α is also a model of β, then $\alpha \models \beta$. Otherwise, $\alpha \not\models \beta$.
- Works up to about 32 propositions.

INFERENCE VIA MODEL CHECKING

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

  $symbols \leftarrow$ a list of the proposition symbols in KB and α
  return TT-CHECK-ALL($KB, \alpha, symbols, \{\}$)

---

function TT-CHECK-ALL($KB, \alpha, symbols, model$) **returns** *true* or *false*
  if EMPTY?($symbols$) **then**
    if PL-TRUE?($KB, model$) **then return** PL-TRUE?($\alpha, model$)
    else return *true // when KB is false, always return true*
  else do
    $P \leftarrow$ FIRST($symbols$)
    $rest \leftarrow$ REST($symbols$)
    return (TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P = true\}$)
      and
      TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P = false\}$))

Figure 7.10 A truth-table enumeration algorithm for deciding propositional entailment. (TT stands for truth table.) PL-TRUE? returns *true* if a sentence holds within a model. The variable *model* represents a partial model—an assignment to some of the symbols. The keyword "**and**" is used here as a logical operation on its two arguments, returning *true* or *false*.

AIMA pg 248

$$\neg P_{1,1} \wedge (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge (\neg B_{1,1}) \models \neg P_{1,2} \quad ?$$

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

$$\neg P_{1,1} \wedge (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge (\neg B_{1,1}) \models \neg P_{1,2} \quad ?$$

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

$$\neg P_{1,1} \wedge (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge (\neg B_{1,1}) \models \neg P_{1,2} \quad ?$$

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

$$\neg P_{1,1} \wedge (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge (\neg B_{1,1}) \models \neg P_{1,2} \quad ?$$

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

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

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

Therefore only the bottom row is a model of the KB!

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

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

Therefore only the bottom row is a model of the KB!

THEOREM PROVING

- Model checking works, but it doesn't feel like *reasoning*.
- Theorem proving is the action of developing a proof of an entailment, based on a series of smaller inferences.
- Proofs can be shared and explained. (They're more like legal or mathematical **arguments** than brute force checking.)
- Normally, one must **search** for a proof.

INFERENCE AS SEARCH

- Initial State – our knowledge base
- Actions – applicable inference rules
- Transition Model – results of inference rules
- Goal Test – prove or disprove the query
- Path Cost – uniform

INFERENCE RULES

- Inference rules are logical equivalence **schemas** or **templates** which apply to generalized propositions.
- The most famous is **modus ponens:**

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

… but there are many more!

INFERENCE RULES

- We compare inference rules (and systems) by two terms.
  - Soundness
  - Completeness

SOUNDNESS AND COMPLETENESS

- Write $\alpha \vdash \beta$ to mean that from α we can derive β.
- An inference is **sound** if whenever $\alpha \vdash \beta$,
  it is also the case that $\alpha \models \beta$.
- An inference rule (or system) is said to be **complete**,
  if for every entailment $\alpha \models \beta$ it is also the case that $\alpha \vdash \beta$.

What about modus ponens?

... is it sound?

... is it complete?

MODUS PONENS IS SOUND

- Let m be any model of α and a model of $\alpha \Rightarrow \beta$.
- Because m is a model of $\alpha \Rightarrow \beta$, and by the truth conditions of implication, m must either model β or not be a model of α.
- But we assume m is also a model of α!
- Therefore m is also a model of β and the rule is sound.

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

MODUS PONENS IS NOT COMPLETE

- Consider the formula $P \wedge Q$ and the formula P.
- Clearly, $(P \wedge Q) \models P$; however, because these formulas do not match the modus ponens template, the rule does not apply so $(P \wedge Q) \not\vdash P$.
- Therefore modus ponens is not complete.

MODUS PONENS IS NOT COMPLETE

- If we want a complete inference mechanism, we need more rules of inference!

SAMPLE INFERENCE RULES

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

Figure 7.11 Standard logical equivalences. The symbols α, β, and γ stand for arbitrary sentences of propositional logic.

Figure from AIMA Ch 7

- Problem – *too many rules of inference!*
- Solution – choose an **inferentially complete** subset

THE RESOLUTION RULE

- Consider the following potential inference:

$$\frac{\text{A or B or C,} \quad \text{not B}}{\text{A or C}}$$

A or B or C, not B

A or C

A	B	C	A or B or C	not B	A or C
F	F	F	F	T	F
F	F	T	T	T	T
F	T	F	T	F	F
F	T	T	T	F	T
T	F	F	T	T	T
T	F	T	T	T	T
T	T	F	T	F	T
T	T	T	T	F	T

A or B or C, not B

A or C

A	B	C	**A or B or C**	not B	A or C
F	F	F	F	T	F
F	**F**	**T**	**T**	**T**	**T**
F	**T**	**F**	**T**	**F**	**F**
F	**T**	**T**	**T**	**F**	**T**
T	**F**	**F**	**T**	**T**	**T**
T	**F**	**T**	**T**	**T**	**T**
T	**T**	**F**	**T**	**F**	**T**
T	**T**	**T**	**T**	**F**	**T**

A or B or C, not B

A or C

A	B	C	**A or B or C**	**not B**	A or C
F	F	F	F	T	F
F	**F**	**T**	**T**	**T**	**T**
F	T	F	T	F	F
F	T	T	T	F	T
T	**F**	**F**	**T**	**T**	**T**
T	**F**	**T**	**T**	**T**	**T**
T	T	F	T	F	T
T	T	T	T	F	T

A or B or C, not B

A or C

A	B	C	**A or B or C**	**not B**	**A or C**
F	F	F	F	T	F
F	**F**	**T**	**T**	**T**	**T**
F	T	F	T	F	F
F	T	T	T	F	T
T	**F**	**F**	**T**	**T**	**T**
T	**F**	**T**	**T**	**T**	**T**
T	T	F	T	F	T
T	T	T	T	F	T

Sound!

THE RESOLUTION RULE

- Consider the following potential inference:

$$\frac{\text{A or B or C}, \quad \text{not B}}{\text{A or C}}$$

- This is an example of the (unit) **resolution rule:**

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k}$$

where ℓ_i and m are **complementary literals**

GENERAL RESOLUTION

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

ℓ_i , m_j are **complementary literals**

$$\frac{\text{A or B or C,} \quad \text{not B or D}}{\text{A or C or D}}$$

A or B or C, not B or D

A or C or D

A	B	C	D	A or B or C	not B or D	A or C or D
F	F	F	F			
F	F	F	T			
F	F	T	F			
F	F	T	T			
F	T	F	F			
F	T	F	T			
F	T	T	F			
F	T	T	T			
T	F	F	F			
T	F	F	T			
T	F	T	F			
T	F	T	T			
T	T	F	F			
T	T	F	T			
T	T	T	F			
T	T	T	T			

A or B or C, not B or D

A or C or D

A	B	C	D	A or B or C	not B or D	A or C or D
F	F	F	F	F		
F	F	F	T	F		
F	F	T	F	T		
F	F	T	T	T		
F	T	F	F	T		
F	T	F	T	T		
F	T	T	F	T		
F	T	T	T	T		
T	F	F	F	T		
T	F	F	T	T		
T	F	T	F	T		
T	F	T	T	T		
T	T	F	F	T		
T	T	F	T	T		
T	T	T	F	T		
T	T	T	T	T		

A or B or C, not B or D

A or C or D

A	B	C	D	A or B or C	not B or D	A or C or D
F	F	F	F			
F	F	F	T			
F	F	T	F	T		
F	F	T	T	T		
F	T	F	F	T		
F	T	F	T	T		
F	T	T	F	T		
F	T	T	T	T		
T	F	F	F	T		
T	F	F	T	T		
T	F	T	F	T		
T	F	T	T	T		
T	T	F	F	T		
T	T	F	T	T		
T	T	T	F	T		
T	T	T	T	T		

A or B or C, not B or D

A or C or D

A	B	C	D	A or B or C	not B or D	A or C or D
F	F	F	F			
F	F	F	T			
F	F	T	F	T	T	
F	F	T	T	T	T	
F	T	F	F	T	F	
F	T	F	T	T	T	
F	T	T	F	T	F	
F	T	T	T	T	T	
T	F	F	F	T	T	
T	F	F	T	T	T	
T	F	T	F	T	T	
T	F	T	T	T	T	
T	T	F	F	T	F	
T	T	F	T	T	T	
T	T	T	F	T	T	
T	T	T	T	T	T	

A or B or C, not B or D

A or C or D

A	B	C	D	A or B or C	not B or D	A or C or D
F	F	F	F			
F	F	F	T			
F	F	T	F	T	T	
F	F	T	T	T	T	
F	T	F	F	T		
F	T	F	T	T	T	
F	T	T	F	T		
F	T	T	T	T	T	
T	F	F	F	T	T	
T	F	F	T	T	T	
T	F	T	F	T	T	
T	F	T	T	T	T	
T	T	F	F	T		
T	T	F	T	T	T	
T	T	T	F	T	T	
T	T	T	T	T	T	

A or B or C, not B or D

A or C or D

A	B	C	D	A or B or C	not B or D	A or C or D
F	F	F	F			
F	F	F	T			
F	F	T	F	T	T	T
F	F	T	T	T	T	T
F	T	F	F	T		
F	T	F	T	T	T	T
F	T	T	F	T		
F	T	T	T	T	T	T
T	F	F	F	T	T	T
T	F	F	T	T	T	T
T	F	T	F	T	T	T
T	F	T	T	T	T	T
T	T	F	F	T		T
T	T	F	T	T	T	T
T	T	T	F	T	T	T
T	T	T	T	T	T	T

A or B or C, not B or D

A or C or D

A	B	C	D	A or B or C	not B or D	A or C or D
F	F	F	F			
F	F	F	T			
F	F	T	F	T	T	T
F	F	T	T	T	T	T
F	T	F	F	T		
F	T	F	T	T	T	T
F	T	T	F	T		
F	T	T	T	T	T	T
T	F	F	F	T	T	T
T	F	F	T	T	T	T
T	F	T	F	T	T	T
T	F	T	T	T	T	T
T	T	F	F	T		
T	T	F	T	T	T	T
T	T	T	F	T	T	T
T	T	T	T	T	T	T

FACTORING

- In order to obtain completeness, the resulting clauses must also be **factored**, by removing redundant literals.
- E.g., $A \vee A$ should be factored to just A.

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

GENERAL RESOLUTION

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

ℓ_i , m_j are **complementary literals**

RESOLUTION-BASED THEOREM PROVING

- What does it mean to derive the empty clause?
- We must have resolved two clauses:

$$\frac{P\ ,\quad \neg P}{[]}$$

RESOLUTION

- So, that's what the resolution rule looks like, but how is it useful?
- Resolution can be used to derive a **proof by contradiction** of any entailment which follows from a propositional knowledge base.

RESOLUTION

- Resolution is **refutation complete** — if it is possible to derive the empty clause from a KB, then the KB is inconsistent, and if the KB is inconsistent then resolution will eventually derive the empty clause.

PROOF BY CONTRADICTION

$\alpha \models \beta$ if and only if $(\alpha \wedge \neg\beta)$ is unsatisfiable

RESOLUTION-BASED THEOREM PROVING

To show $KB \models \alpha$,

we show that $(KB \wedge \neg\alpha)$ is unsatisfiable.

RESOLUTION-BASED THEOREM PROVING

- We convert $(KB \wedge \neg\alpha)$ to a logically equivalent, normalized form (CNF), then repeatedly apply the resolution rule until one of two things happens:

1. No new clauses can be added.
   In which case $KB \not\models \alpha$.

2. We derive the empty clause.
   In which case $KB \models \alpha$.

A TRAP

- Note that resolution must be applied one literal at a time, otherwise you may make unsound inferences:
- The clauses $P \vee Q$ and $\neg P \vee \neg Q$ can be resolved to produce $P \vee \neg P$ or $Q \vee \neg Q$ but not the empty clause!

RESOLUTION

- But what if we have a knowledge base that doesn't match the resolution rule schema?
- Every propositional knowledge base is equivalent to one in **conjunctive normal form (CNF)** – a conjunction of disjunctions, referred to as **clauses.**

CONVERTING TO CONJUNCTIVE NORMAL FORM (CNF)

1. Eliminate biconditionals.

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

2. Eliminate implications.

$$\alpha \Rightarrow \beta \equiv \neg\alpha \vee \beta$$

3. Move negation inwards.

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

4. Distribute disjunction over conjunction.

$$[(\alpha \wedge \beta) \vee \gamma] \equiv (\alpha \vee \gamma) \wedge (\beta \vee \gamma)$$

- (P v Q) => R
- ~(P v Q) v R
- (~P ^ ~Q) v R
- (~P v R) ^ (~Q v R)

$$KB = R_2 \wedge R_4 = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$$

$$\alpha = \neg P_{1,2}$$

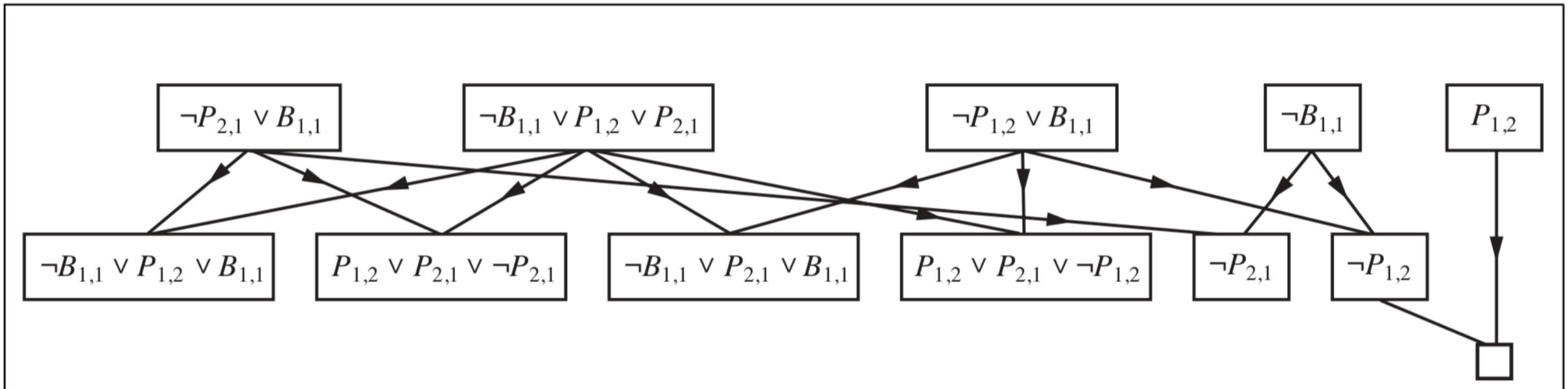


Figure 7.13 Partial application of PL-RESOLUTION to a simple inference in the wumpus world. $\neg P_{1,2}$ is shown to follow from the first four clauses in the top row.

THE LABYRINTH

- You are walking in a labyrinth and all of a sudden you find yourself in front of three possible roads: the road on your left is paved with gold, the one in front of you is paved with marble, while the one on your right is made of small stones. Each street is protected by a treacherous guardian who only speaks in lies.

THE LABYRINTH

- You talk to the guardians and this is what they tell you:
- The guardian of the gold street: “This road will bring you straight to the center. Moreover, if the stones take you to the center, then also the marble takes you to the center.”
- The guardian of the marble street: “Neither the gold nor the stones will take you to the center.”
- The guardian of the stone street: “Follow the gold and you’ll reach the center, follow the marble and you will be lost.”
- Given that you know that all the guardians are liars, can you choose a road being sure that it will lead you to the center of the labyrinth? If this is the case, which road you choose?

THE LABYRINTH

- Given that you know that **all the guardians are liars**, can you choose a road being sure that it will lead you to the center of the labyrinth? If this is the case, which road you choose?