



Prof. Dr. Matthias Schott

- Proving theorems

- Note: this and the next section (satisfiability) are based on Chapter 7 of the textbook ("Logical Agents")

# Contents

- Agents that Think Rationally
- The Wumpus World
- A Primer on Logic
- Propositional Logic: Syntax and Semantics
- Logical Entailment
- Logical Derivation (Resolution)

# Agents that Think Rationally

- Until now, the focus has been on agents that act rationally.
- Often, however, rational action requires rational (logical) thought on the agent's part.
- To that purpose, portions of the world must be represented in a knowledge base, or KB.
  - A KB is composed of sentences in a language with a truth theory (logic)
  - We (being external) can interpret sentences as statements about the world. (semantics)
  - Through their form, the sentences themselves have a causal influence on the agent's behavior. (syntax)
- Interaction with the KB through Ask and Tell (simplified):
  - Ask  $(KB, \alpha) = \text{yes}$  exactly when  $\alpha$  follows from the KB
  - Tell  $(KB, \alpha) = KB'$  so that  $\alpha$  follows from  $KB'$
  - Forget  $(KB, \alpha) = KB'$  non-monotonic (will not be discussed)

### 3 Levels

- In the context of knowledge representation, we can distinguish three levels [Newell 1990]:
- Knowledge level: Most abstract level. Concerns the total knowledge contained in the KB. For example, the automated DB information system knows that a trip from Freiburg to Basel SBB with an ICE costs 24.70Eur.
- Logical level: Encoding of knowledge in a formal language. Price (Freiburg , Basel , 24.70)
- Implementation level: The internal representation of the sentences, for example:
  - As a string "Price(Freiburg, Basel, 24.70)"
  - As a value in a matrix
- When Ask and Tell are working correctly, it is possible to remain on the knowledge level.
  - Advantage: very comfortable user interface. The user has his/her own mental model of the world (statements about the world) and communicates it to the agent (Tell).

- represent its background knowledge
- store its observations
- store its executed actions
- ... derive actions

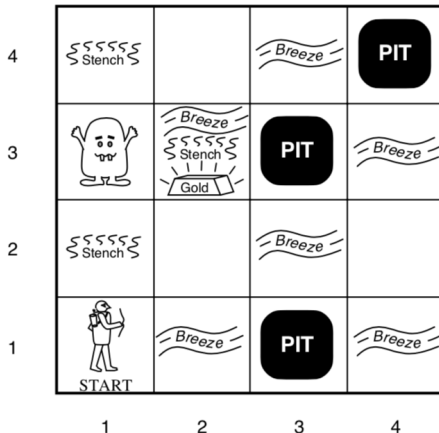
```

function KB-AGENT(percept) returns an action
  persistent: KB, a knowledge base
             t, a counter, initially 0, indicating time
  TELL (KB, MAKE-PERCEPT-SENTENCE(percept, t))
  action  $\leftarrow$  ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE(action, t))
   $t \leftarrow t + 1$ 
return action

```

# The Wumpus World (1): Illustration

This is just one sample configuration.



# The Wumpus World (2)

- A 4x4 grid
- In the square containing the wumpus and in the directly adjacent squares, the agent perceives a stench.
- In the squares adjacent to a pit, the agent perceives a breeze.
- In the square where the gold is, the agent perceives a glitter.
- When the agent walks into a wall, it perceives a bump.
- When the wumpus is killed, its scream is heard everywhere.
- Percepts are represented as a 5-tuple, e.g., [Stench , Breeze , Glitter , None , None ]
  - means that it stinks, there is a breeze and a glitter, but no bump and no scream. The agent cannot perceive its own location, cannot look in adjacent square.



# The Wumpus World (3)

- Actions: Go forward, turn right by  $90^\circ$ , turn left by  $90^\circ$ , pick up an object in the same square (grab), shoot (there is only one arrow), leave the cave (only works in square  $[1, 1]$ ).
- The agent dies if it falls down a pit or meets a live wumpus.
- Initial situation: The agent is in square  $[1, 1]$  facing east. Somewhere exists a wumpus, a pile of gold and 3 pits.
- Goal: Find the gold and leave the cave.

# The Wumpus World (4)

$[1, 2]$  and  $[2, 1]$  are safe:

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK			
1,1	2,1	3,1	4,1
<b>A</b> OK	OK		

(a)

**A** = Agent  
**B** = Breeze  
**G** = Glitter, Gold  
**OK** = Safe square  
**P** = Pit  
**S** = Stench  
**V** = Visited  
**W** = Wumpus

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK	<b>P?</b>		
1,1	2,1	3,1	4,1
<b>V</b> OK	<b>A</b> <b>B</b> OK	<b>P?</b>	

(b)

# The Wumpus World (5)

The wumpus is in [1, 3]!

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 <b>A</b> S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

(a)

**A** = Agent  
**B** = Breeze  
**G** = Glitter, Gold  
**OK** = Safe square  
**P** = Pit  
**S** = Stench  
**V** = Visited  
**W** = Wumpus

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 <b>A</b> S G B	3,3 P?	4,3
1,2 S V OK	2,2 V OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

(b)

# Syntax and Semantics

## Syntax and Semantics

# Syntax and Semantics

- Knowledge bases consist of sentences
- Sentences are expressed according to the syntax of the representation language
  - Syntax specifies all the sentences that are well-formed
  - E.g., in ordinary arithmetic, syntax is pretty clear:
    - $x + y = 4$  is a well-formed sentence  $x4y+ =$  is not a well-formed sentence
- A logic also defines the semantics or meaning of sentences
  - Defines the truth of a sentence with respect to each possible world
  - E.g., specifies that the sentence  $x + y = 4$  is true in a world in which  $x = 2$  and  $y = 2$ , but not in a world in which  $x = 1$  and  $y = 1$

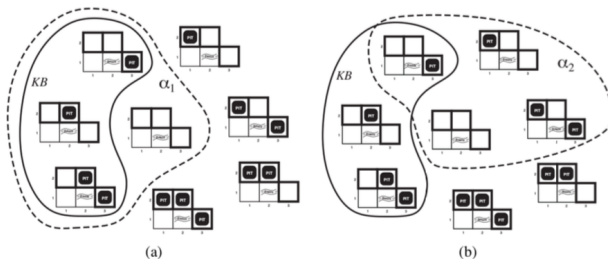
# Logical Entailment

- If a sentence  $\alpha$  is true in a possible world  $m$ , we say that  $m$  satisfies  $\alpha$  or  $m$  is a model of  $\alpha$
- We denote the set of all models of  $\alpha$  by  $M(\alpha)$
- Logical entailment:
  - When does a sentence  $\beta$  logically follow from another sentence  $\alpha$ ?
  - in symbols  $\alpha \models \beta$
  - $\alpha \models \beta$  if and only if (iff) in every model in which  $\alpha$  is true,  $\beta$  is also true,
  - i.e.,  $\alpha \models \beta$  iff  $M(\alpha) \subseteq M(\beta)$
  - -  $\alpha$  is here a stronger assertion than  $\beta$ ; it rules out more possible worlds
  - Example in arithmetic: sentence  $x = 0$  entails sentence  $xy = 0$ .  $x = 0$  rules out the possible world  $\{x = 1, y = 0\}$ , whereas  $xy = 0$  does not rule out that world

# Example in the Wumpus World (1)

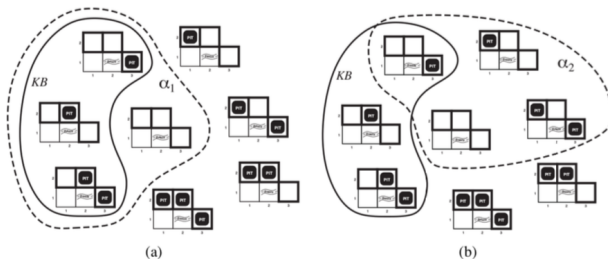
## Setup:

- Agent detected nothing in  $[1, 1]$  and a breeze in  $[2, 1]$
- These percepts, plus the rules of the wumpus world, make up the agent's KB
- Let's reason about three variables: whether  $[1, 2]$ ,  $[2, 2]$ , and  $[3, 1]$  contain pits
- KB is false in any possible world that contradicts what the agent knows
  - E.g., in possible worlds in which  $[1, 2]$  contains a pit (no breeze in  $[1, 1]$ )
  - E.g., when neither  $[2, 2]$  nor  $[3, 1]$  have a pit (breeze in  $[2, 1]$ )



# Example in the Wumpus World (2)

- Consider two possible sentences:
  - $\alpha_1$  =: "There is no pit in  $[1, 2]$ " (true in models in dashed area below, left)
  - $\alpha_2$  =: "There is no pit in  $[2, 2]$ " (true in models in dashed area below, right)
- $KB \models \alpha_1$ 
  - By inspection: in every model in which KB is true,  $\alpha_1$  is also true
- $KB \not\models \alpha_2$ 
  - In some models, in which KB is true,  $\alpha_2$  is false





# Entailment and Inference

- The inference algorithm we just used was exhaustive enumeration
  - Logical entailment:  $KB \models \alpha$
  - Inference: we can derive  $\alpha$  with an inference algorithm  $i$ . This is written as:  
 $KB \vdash_i \alpha$
- We want to have inference algorithms that derive only sentences that are entailed (soundness) and all of them (completeness)

# Propositional Logic: Syntax and Semantics

## Propositional Logic: Syntax and Semantics

# Declarative Languages

Before a system that is capable of learning, thinking, planning, explaining, ... can be built, one must find a way to express knowledge.

We need a precise, declarative language.

- Declarative
  - We state what we want to compute, not how
  - System believes  $P$  if and only if (iff) it considers  $P$  to be true
- Precise: We must know,
  - which symbols represent sentences,
  - what it means for a sentence to be true, and
  - when a sentence follows from other sentences.

One possibility: Propositional Logic

# Basics of Propositional Logic (1)

- Propositions: The building blocks of propositional logic are indivisible, atomic statements (atomic propositions), e.g.,
  - "The block is red", expressed, e.g., by the symbol " $B_{red}$ "
  - "The wumpus is in  $[1, 3]$ ", expressed, e.g., by the symbol " $W_{1,3}$ "
  - and the logical connectives "and", "or", and "not", which we can use to build formulae.

# Basics of Propositional Logic (2)

We are interested in knowing the following:

- When is a proposition true?
- When does a proposition follow from a knowledge base (KB)?
  - Symbolically:  $KB \models \phi$
- Can we (syntactically) define the concept of derivation,
  - Symbolically:  $KB \vdash \phi$
- And can we make sure that  $\models$  and  $\vdash$  are equivalent?

→ Meaning and implementation of Ask

# Syntax of Propositional Logic

Countable alphabet  $\Sigma$  of atomic propositions:  $P, Q, R, W_{1,3}, \dots$

- Logical formulae:
  - $P \in \Sigma$  atomic formula
  - $\perp$  falseness
  - $\top$  truth
  - $\neg\phi$  negation
  - $\phi \wedge \psi$  conjunction (and)
  - $\phi \vee \psi$  disjunction (or)
  - $\phi \Rightarrow \psi$  implication ( $\neg\phi \vee \psi$ )
  - $\phi \Leftrightarrow \psi$  equivalence ( $\phi \Rightarrow \psi$  and  $\psi \Rightarrow \phi$ )
- Operator precedence:  $\neg > \wedge > \vee > \Rightarrow > \Leftrightarrow$  (use brackets when necessary)
- Atom: atomic formula ( $P, \perp, \top$ )
- Literal: (possibly negated) atomic formula
- Clause: disjunction of literals ( $A \vee B \vee C$ )

# Semantics: Intuition

- Atomic propositions can be true (T) or false (F).
- The truth of a formula follows from the truth of its atomic propositions (truth assignment or interpretation) and the connectives.
- Example:  $(P \vee Q) \wedge R$ 
  - If P and Q are false and R is true, the formula is false
  - If P and R are true, the formula is true regardless of what Q is.

# Semantics: Formally

A truth assignment of the atoms in  $\Sigma$ , or an interpretation  $I$  over  $\Sigma$ , is a function

$$I : \Sigma \rightarrow \{T, F\}$$

Interpretation  $I$  satisfies a formula  $\phi$  ( $I \models \phi$ ):

- $I \models \top$
- $I \not\models \perp$
- $I \models P$  iff  $P^I = T$
- $I \not\models \neg\phi$  iff  $I \models \phi$
- $I \models (\phi \wedge \psi)$  iff  $I \models \phi$  and  $I \models \psi$
- $I \models (\phi \vee \psi)$  iff  $I \models \phi$  or  $I \models \psi$
- $I \models (\phi \Rightarrow \psi)$  iff if  $I \models \phi$ , then  $I \models \psi$
- $I \models (\phi \Leftrightarrow \psi)$  iff if  $I \models \phi$  if and only if  $I \models \psi$

$I$  satisfies  $\phi$  ( $I \models \phi$ ) or  $\phi$  is true under  $I$ , when  $I(\phi) = T$ .

$I$  can be seen as a 'possible world'



# Example

$$I : \begin{cases} P \mapsto T \\ Q \mapsto T \\ R \mapsto F \\ S \mapsto F \\ \dots \end{cases}$$

$$\phi = ((P \vee Q) \Leftrightarrow (R \vee S)) \wedge (\neg(P \wedge Q) \wedge (R \wedge \neg S))$$

Question:  $I \models \phi$ ?

# Wumpus World in Propositional Logic

Symbols:  $B_{1,1}, B_{1,2}, \dots, B_{2,1}, \dots, S_{1,1}, \dots, P_{1,1}, \dots, W_{1,1}, \dots$

- Meaning:  $B = Breeze$ ,  $B_{i,j}$  = there is a breeze in  $(i, j)$  etc.

Facts and Rules:

- $R_1: B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
- $R_2: B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- $R_3: B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3})$
- ...
- $F_1: \neg P_{1,1}$
- $F_2: \neg B_{1,1}$  (no percept in  $(1, 1)$ )
- $F_3: B_{2,1}$  (percept)
- $F_4: \neg B_{1,2}$  (no percept)
- ...

# Logical Entailment

## Logical Entailment

# Logical Entailment

- An interpretation  $I$  is called a model of  $\phi$  if  $I \models \phi$ .
- An interpretation is a model of a set of formulae if it fulfills all formulae of the set.
- A formula  $\phi$  is
  - satisfiable if there exists  $I$  that satisfies  $\phi$ ,
  - unsatisfiable if  $\phi$  is not satisfiable,
  - falsifiable if there exists  $I$  that doesn't satisfy  $\phi$ , and
  - valid (a tautology) if  $I \models \phi$  holds for all  $I$ .
- Question: how are these related to each other?
  - Two formulae are logically equivalent ( $\phi \equiv \psi$ ) if  $(I \models \phi \text{ iff } I \models \psi)$  holds for all  $I$ .
- An interpretation is a model of a set of formulae if it fulfills all formulae of the set.

# The Truth Table Method

How can we decide if a formula is satisfiable, valid, etc.?

- $\rightarrow$  generate a truth table
- Example: Is  $\phi = ((P \vee H) \wedge \neg H) \Rightarrow P$  valid?

$P$	$H$	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
$F$	$F$	$F$	$F$	$T$
$F$	$T$	$T$	$F$	$T$
$T$	$F$	$T$	$T$	$T$
$T$	$T$	$T$	$F$	$T$

Since the formula is true for all possible combinations of truth values (satisfied under all interpretations),  $\phi$  is valid.

- Satisfiability, falsifiability, unsatisfiability likewise.

# Logical Derivation (Resolution)

## Logical Derivation (Resolution)

# Logical Derivation (Resolution)

## Logical Implications

- Goal: Find an algorithmic way to derive new knowledge out of a knowledge base
  - Transform KB into a standardized representation
  - define rules that syntactically modify formulae while keeping semantic correctness
- Normal Forms
  - A formula is in conjunctive normal form (CNF) if it consists of a conjunction of disjunctions of literals  $l_{i,j}$ , i.e., if it has the following form:

$$\bigwedge_{i=1}^n \left( \bigvee_{j=1}^{m_i} l_{i,j} \right)$$

- A formula is in disjunctive normal form (DNF) if it consists of a disjunction of conjunctions of literals

$$\bigvee_{i=1}^n \left( \bigwedge_{j=1}^{m_i} l_{i,j} \right)$$

# Logical Derivation (Resolution)

- A formula is in conjunctive normal form (CNF) if it consists of a conjunction of disjunctions of literals  $l_{i,j}$ , i.e., if it has the following form:

$$\bigwedge_{i=1}^n (\bigvee_{j=1}^{m_i} l_{i,j})$$

- A formula is in disjunctive normal form (DNF) if it consists of a disjunction of conjunctions of literals

$$\bigvee_{i=1}^n (\bigwedge_{j=1}^{m_i} l_{i,j})$$

- For every formula, there exists at least one equivalent formula in CNF and one in DNF.
- A formula in DNF is satisfiable iff one disjunct is satisfiable.
  - Checking satisfiability of DNF formula takes linear time.
- A formula in CNF is valid iff every conjunct is valid.
  - Checking validity of CNF formula takes linear time.



# Producing CNF

- 1. Eliminate  $\Rightarrow$  and  $\Leftrightarrow$ :  $\alpha \Rightarrow \beta \rightarrow (\neg\alpha \vee \beta)$  etc.
- 2. Move  $\neg$  inwards:  $\neg(\alpha \wedge \beta) \rightarrow (\neg\alpha \vee \neg\beta)$  etc. (De Morgan's laws)
- 3. Distribute  $\vee$  over  $\wedge$ :  $((\alpha \wedge \beta) \vee \gamma) \rightarrow (\alpha \vee \gamma) \wedge (\beta \vee \gamma)$
- 4. Simplify:  $\alpha \vee \alpha \rightarrow \alpha$  etc.
- The result is a conjunction of disjunctions of literals (CNF)
- An analogous process converts any formula to an equivalent formula in DNF.
  - During conversion, formulae can expand exponentially.
  - Note: Conversion to CNF formula can be done polynomially if only satisfiability should be preserved

# Logical Implication: Intuition

- A set of formulae (a KB) usually provides an incomplete description of the world, i.e., it leaves the truth values of certain propositions open.
- Example:  $KB = \{(P \vee Q) \wedge (R \vee \neg P) \wedge S\}$  is definitive with respect to  $S$ , but leaves  $P$ ,  $Q$ ,  $R$  open (although they cannot take on arbitrary values).
- Models of the KB:

$P$	$Q$	$R$	$S$
$F$	$T$	$F$	$T$
$F$	$T$	$T$	$T$
$T$	$F$	$T$	$T$
$T$	$T$	$T$	$T$

- In all models of the KB,  $Q \vee R$  is true, i.e.,  $Q \vee R$  follows logically from KB.

# Logical Implication: Formal

- The formula  $\phi$  follows logically from a KB iff  $\phi$  is true in all models of the KB (symbolically  $KB \models \phi$ ):
  - $KB \models \phi$  iff  $I \models \phi$  for all models  $I$  of KB
- Note: The  $\models$  symbol is a meta-symbol
- Question: Can we determine  $KB \models \phi$  without considering all interpretations (the truth table method)?
- Some properties of logical implication relationships:
  - Deduction theorem:  $KB \cup \{\phi\} \models \psi$  iff  $KB \models (\phi \Rightarrow \psi)$
  - Contraposition theorem:  $KB \cup \{\phi\} \models \neg\psi$  iff  $KB \cup \{\psi\} \models \neg\phi$
  - Contradiction theorem:  $KB \cup \{\phi\}$  is unsatisfiable iff  $KB \models \neg\phi$

# Proof of the Deduction Theorem

Deduction theorem:  $KB \cup \{\phi\} \models \psi$  iff  $KB \models \phi \Rightarrow \psi$

- " $\Rightarrow$ " Assumption:  $KB \cup \{\phi\} \models \psi$ , i.e., every model of  $KB \cup \phi$  is also a model of  $\psi$ 
  - Let  $I$  be any model of KB. If  $I$  is also a model of  $\phi$ , then it follows that  $I$  is also a model of  $\psi$ .
  - This means that  $I$  is also a model of  $\phi \Rightarrow \psi$ , i.e.,  $KB \models \phi \Rightarrow \psi$ .
- " $\Leftarrow$ " Assumption:  $KB \models (\phi \Rightarrow \psi)$ . Let  $I$  be any model of KB that is also a model of  $\phi$ , i.e.,  $I \models KB \cup \phi$ 
  - From the assumption,  $I$  is also a model of  $\phi \Rightarrow \psi$  and thereby also of  $\psi$ , i.e.,  $KB \cup \phi \models \psi$ .

Contraposition theorem:  $KB \cup \phi \models \neg\psi$  iff  $KB \cup \psi \models \neg\phi$

$$\text{iff KB} \models (\neg\varphi \vee \neg\psi)$$

$$\text{iff KB} \models (\neg\psi \vee \neg\varphi)$$

$$\text{iff KB} \models \psi \Rightarrow \neg\varphi$$

$$\text{iff } \text{KB} \cup \{\psi\} \models \neg\varphi \quad (2)$$

Note: (1) and (2) are applications of the deduction theorem.

We can often derive new formulae from formulae in the KB. These new formulae should follow logically from the syntactical structure of the KB formulae.

- $\rightarrow$  Inference rules, e.g.,  $\frac{\phi, \phi \Rightarrow \psi}{\psi}$  (here modus ponens)
- Calculus: Set of inference rules (potentially including so-called logical axioms).
- Proof step: Application of an inference rule on a set of formulae.
- Proof: Sequence of proof steps where every newly-derived formula is added, and in the last step, the goal formula is produced.

- $\rightarrow$  Inference rules, e.g.,  $\frac{\phi, \phi \Rightarrow \psi}{\psi}$  (here modus ponens)
- Calculus: Set of inference rules (potentially including so-called logical axioms).
- Proof step: Application of an inference rule on a set of formulae.
- Proof: Sequence of proof steps where every newly-derived formula is added, and in the last step, the goal formula is produced.







# Resolution: Representation

- Assumption: All formulae in the KB are in CNF.
- Equivalently, we can assume that the KB is a set of clauses. E.g.: Replace  $\{(P \vee Q) \wedge (R \vee \neg P) \wedge S\}$  by  $\{\{P, Q\}, \{R, \neg P\}, \{S\}\}$
- Due to commutativity, associativity, and idempotence of  $\vee$ , clauses can also be understood as sets of literals. The empty set of literals is denoted by  $\square$ .
- Set of clauses:  $\Delta$
- Set of literals:  $C, D$
- Literal:  $l$
- Negation of a literal:  $\bar{l}$
- An interpretation  $I$  satisfies  $C$  iff there exists  $l \in C$  such that  $I \models l$ .  $I$  satisfies  $\Delta$  if for all  $C \in \Delta : I \models C$ , i.e.,  $I \not\models \square, I \not\models \{\square\}$ , for all  $I$ .

$$\frac{C_1 \dot{\cup} \{l\}, C_2 \dot{\cup} \{\bar{l}\}}{C_1 \cup C_2}$$

- $C_1 \cup C_2$  are called resolvents of the parent clauses  $C_1 \cup \{l\}$  and  $C_2 \cup \{\bar{l}\}$ .  $l$  and  $\bar{l}$  are the resolution literals.
- Example:  $\{a, b, \neg c\}$  resolves with  $\{a, d, c\}$  to  $\{a, b, d\}$ .
- Note: The resolvent is not equivalent to the parent clauses, but it follows from them!
- Notation:  $R(\Delta) = \Delta \cup \{C \mid C \text{ is a resolvent of two clauses from } \Delta\}$

# The Resolution Algorithm

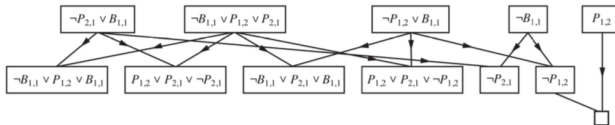
```

function PL-RESOLUTION( $KB, \alpha$ ) returns true or false
  inputs:  $KB$ , the knowledge base, a sentence in propositional logic
            $\alpha$ , 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
   $clauses \leftarrow clauses \cup new$ 

```

Example:





Is resolution also complete, i.e., is

$$\Delta \models \phi \text{ implies } \Delta \vdash \phi$$

valid? Not in general. For example, consider:

$$\{\{a, b\}, \{\neg b, c\}\} \models \{a, b, c\}$$

- However, it can be shown that resolution is refutation-complete:  $\Delta$  is unsatisfiable implies  $\Delta \vdash \square$
- Theorem:  $\Delta$  is unsatisfiable iff  $\Delta \vdash \square$
- With the help of the contradiction theorem, we can show that  $KB \models \phi$ .
- Idea:  $KB \cup \{\neg\phi\}$  is unsatisfiable iff  $KB \models \phi$

# Resolution: Overview

- Resolution is a refutation-complete proof process. There are others (Davis-Putnam Procedure, Tableaux Procedure, ...).
- In order to implement the process, a strategy must be developed to determine which resolution steps will be executed and when.
- In the worst case, a resolution proof can take exponential time. This, however, very probably holds for all other proof procedures.
- For CNF formulae in propositional logic, the fastest complete algorithms are indeed based on resolution (combined with backtracking search)

## Where is the Wumpus? The Situation

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 <b>A</b> S OK	2,2  OK	3,2	4,2
1,1  V OK	2,1 B V OK	3,1 P!	4,1

**A** = Agent  
**B** = Breeze  
**G** = Glitter, Gold  
**OK** = Safe square  
**P** = Pit  
**S** = Stench  
**V** = Visited  
**W** = Wumpus

# Where is the Wumpus? Knowledge of the Situation

$B = \text{Breeze}$ ,  $S = \text{Stench}$ ,  $B_{i,j} = \text{there is a breeze in } (i,j)$

- $\neg S_{1,1} \quad \neg B_{1,1}$
- $\neg S_{2,1} \quad B_{2,1}$
- $S_{1,2} \quad \neg B_{1,2}$

Knowledge about the wumpus and smell:

- $R_1 : \neg S_{1,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$
- $R_2 : \neg S_{2,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1}$
- $R_3 : \neg S_{1,2} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,2} \wedge \neg W_{1,3}$
- $R_4 : S_{1,2} \Rightarrow W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$

To show:  $KB \models W_{1,3}$



# Clausal Representation of the Wumpus World

Situational knowledge:  $\neg S_{1,1}, \neg S_{2,1}, S_{1,2}$

Knowledge of rules:

- Knowledge about the wumpus and smell:
  - $R_1 : S_{1,1} \vee \neg W_{1,1}, S_{1,1} \vee \neg W_{1,2}, S_{1,1} \vee \neg W_{2,1}$
  - $R_2 : \dots, S_{2,1} \vee \neg W_{2,2}, \dots$
  - $R_3 : \dots$
  - $R_4 : \neg S_{1,2} \vee W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$
  - ...
- Negated goal formula:  $\neg W_{1,3}$

# Resolution Proof for the Wumpus World

Resolution:

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

$$\rightarrow \neg S_{1,2} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$$

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

$$\rightarrow W_{1,2} \vee W_{2,2} \vee W_{1,1}$$

$$\neg S_{1,1}, S_{1,1} \vee \neg W_{1,1}$$

$$\rightarrow \neg W_{1,1}$$

$$\neg W_{1,1}, W_{1,2} \vee W_{2,2} \vee W_{1,1}$$

$$\rightarrow W_{1,2} \vee W_{2,2}$$

...

$$\neg W_{2,2}, W_{2,2}$$

$$\rightarrow \square$$

# From Knowledge to Action

- We can now infer new facts, but how do we translate knowledge into action?
- Negative selection: Excludes any provably dangerous actions.

$$A_{1,1} \wedge East_A \wedge W_{2,1} \Rightarrow \neg Forward$$

- Positive selection: Only suggests actions that are provably safe.

$$A_{1,1} \wedge East_A \wedge \neg W_{2,1} \Rightarrow Forward$$

- Differences?
  - From the suggestions, we must still select an action.

Although propositional logic suffices to represent the wumpus world, it is rather involved.

- |                           |                            |                |         |
|---------------------------|----------------------------|----------------|---------|
| Prof. Dr. Matthias Schott | Introduction to AI (Ref A) | March 30, 2021 | 52 / 53 |
|---------------------------|----------------------------|----------------|---------|

# Summary

- Rational agents require knowledge of their world in order to make rational decisions.
- With the help of a declarative (knowledge-representation) language, this knowledge is represented and stored in a knowledge base.
- We use propositional logic for this (for the time being).
- Formulae of propositional logic can be valid, satisfiable, or unsatisfiable.
- The concept of logical implication is important.
- Logical implication can be mechanized by using an inference calculus → resolution.
- Propositional logic quickly becomes impractical when the world becomes too large (or infinite).