Artificial Intelligence

8. Propositional Reasoning, Part I: Principles

How to Think About What is True or False

Jörg Hoffmann, Daniel Fiser, Daniel Höller, Sophia Saller

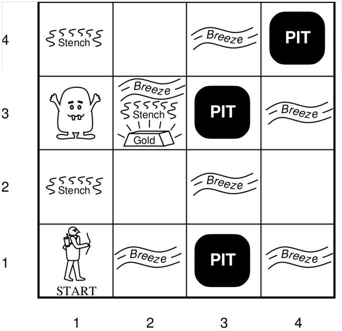
SAARLAND UNIVERSITY
COMPUTER SCIENCE

Summer Term 2022

Agenda

1. Introduction
2. Propositional Logic
3. Resolution
4. Killing a Wumpus
5. Conclusion

The Wumpus World



- **Actions:** *GoForward*, *TurnRight* (by 90°), *TurnLeft* (by 90°), *Grab* object in current cell, *Shoot* arrow in direction you're facing (you got exactly one arrow), *Leave* cave if you're in cell [1,1].

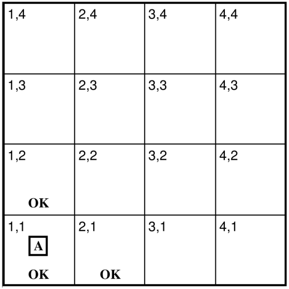
  → Fall down *Pit*, meet live *Wumpus*: Game Over.
- **Initial knowledge:** You're in cell [1,1] facing east. There's a Wumpus, and there's gold.
- **Goal**: Have the gold and be outside the cave.

Percepts: $[Stench, Breeze, Glitter, Bump, Scream]$

- Cell adjacent (i.e. north, south, west, east) to Wumpus: $Stench$ (else: $None$).
- Cell adjacent to Pit: $Breeze$ (else: $None$).
- Cell that contains gold: $Glitter$ (else: $None$).
- You walk into a wall: $Bump$ (else: $None$).
- Wumpus shot by arrow: $Scream$ (else: $None$).

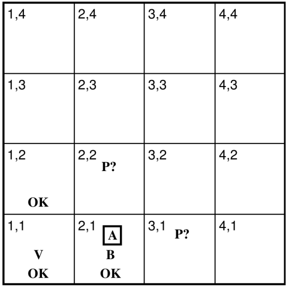
Reasoning in the Wumpus World

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



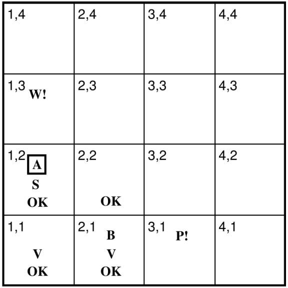
1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

(1) Initial state



1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2 P?	3,2	4,2
1,1 V OK	2,1 A B OK	3,1 P?	4,1

(2) One step to right



1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

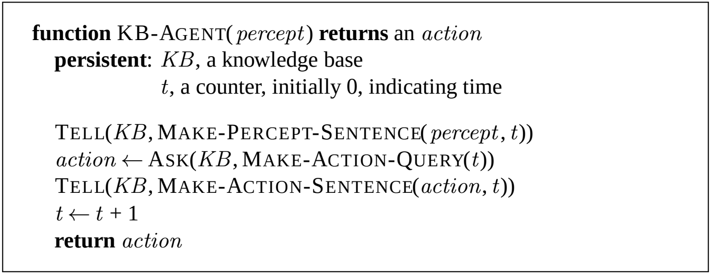
(3) Back, and up to [1,2]

→ The Wumpus is in [1,3]! How do we know? Because in [2,1] we perceived no Stench, the Stench in [1,2] can only come from [1,3].

→ There's a Pit in [3,1]! How do we know? Because in [1,2] we perceived no Breeze, the Breeze in [2,1] can only come from [3,1].

Agents that Think Rationally

Think Before You Act!



```
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 ← ASK(KB, MAKE-ACTION-QUERY(t))
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))
    t ← t + 1
    return action
```

→ "Thinking" = Reasoning about knowledge represented using logic.

Logic: Basic Concepts

Representing Knowledge:

- Syntax: What are legal statements (formulas) φ in the logic?

  E.g., "P" and "$P \rightarrow Q$".

- Semantics: Which formulas φ are true under which interpretation I, written $I \models \varphi$?

  E.g., $I := \{P = 1, Q = 0\}$. Then $I \models P$ but $I \not\models P \rightarrow Q$.

Reasoning about Knowledge:

- Entailment: Which ψ are entailed by φ, written $\varphi \models \psi$, meaning that, for all I s.t. $I \models \varphi$, we have $I \models \psi$? E.g., $P \wedge (P \rightarrow Q) \models Q$.

- Deduction: Which statements ψ can be derived from φ using a set $\mathcal{R}$ of inference rules (a calculus), written $\varphi \vdash_{\mathcal{R}} \psi$?

  E.g., if our only rule is $\dfrac{\varphi_1, \varphi_1 \rightarrow \psi}{\psi}$ then $P \wedge (P \rightarrow Q) \vdash_{\mathcal{R}} Q$.

  → Calculus soundness: whenever $\varphi \vdash_{\mathcal{R}} \psi$, we also have $\varphi \models \psi$.

  Calculus completeness: whenever $\varphi \models \psi$, we also have $\varphi \vdash_{\mathcal{R}} \psi$.

General Problem Solving using Logic

(some new problem)

model problem in logic ↦ use off-the-shelf reasoning tool

(its solution)

- “Any problem that can be formulated as reasoning about logic.”
- *Very* successful using propositional logic and modern solvers for SAT! (Propositional satisfiability testing, **Chapter 9**.)

Propositional Logic and Its Applications

→ Propositional logic = canonical form of knowledge + reasoning.

- Syntax: Atomic propositions that can be either true or false, connected by "and, or, not".
- Semantics: Assign value to every proposition, evaluate connectives.

Applications: Despite its simplicity, widely applied!

- Product configuration (e.g., Mercedes). Check consistency of customized combinations of components.
- Hardware verification (e.g., Intel, AMD, IBM, Infineon). Check whether a circuit has a desired property p.
- Software verification: Similar.
- CSP applications (cf. **Chapter 6**): Propositional logic can be (successfully!) used to formulate and solve CSP problems.

→ **Chapter 9** gives an example for verification.

Our Agenda for This Topic

→ Our treatment of the topic "Propositional Reasoning" consists of Chapters 8 and 9.

- **This Chapter:** Basic definitions and concepts; resolution.

  → Sets up the framework. Resolution is the quintessential reasoning procedure underlying most successful solvers.

- **Chapter 9:** The Davis-Putnam procedure and clause learning; practical problem structure.

  → State-of-the-art algorithms for reasoning about propositional logic, and an important observation about how they behave.

Our Agenda for This Chapter

- **Propositional Logic:** What's the syntax and semantics? How can we capture deduction?

  → Formalizes this logic.

- **Resolution:** How does resolution work? What are its properties?

  → Formally introduces the most basic reasoning method.

- **Killing a Wumpus:** How can we use all this to figure out where the Wumpus is?

  → Coming back to our introductory example.

Syntax of Propositional Logic

→ Atoms Σ in propositional logic = Boolean variables.

Definition (Syntax). *Let Σ be a set of atomic propositions. Then:*

1. $\bot$ *and* $\top$ *are Σ-formulas.* (“False”, “True”)
2. *Each $P \in \Sigma$ is a Σ-formula.* (“Atom”)
3. *If φ is a Σ-formula, then so is $\neg\varphi$.* (“Negation”)

If φ and ψ are Σ-formulas, then so are:

4. $\varphi \wedge \psi$ (“Conjunction”)
5. $\varphi \vee \psi$ (“Disjunction”)
6. $\varphi \rightarrow \psi$ (“Implication”)
7. $\varphi \leftrightarrow \psi$ (“Equivalence”)

Example: Wumpus-[2,2] → Stench-[2,1].

Notation: Atoms and negated atoms are called literals. Operator precedence: $\neg > \dots$ (we'll be using brackets except for negation).

Semantics of Propositional Logic

Definition (Semantics). *Let Σ be a set of atomic propositions. An interpretation of Σ, also called a truth assignment, is a function $I : \Sigma \mapsto \{1, 0\}$. We set:*

$$\begin{array}{lll}
I \models \top & & \\
I \not\models \bot & & \\
I \models P & \textit{iff} & P^I = 1 \\
I \models \neg\varphi & \textit{iff} & I \not\models \varphi \\
I \models \varphi \wedge \psi & \textit{iff} & I \models \varphi \text{ and } I \models \psi \\
I \models \varphi \vee \psi & \textit{iff} & I \models \varphi \text{ or } I \models \psi \\
I \models \varphi \rightarrow \psi & \textit{iff} & \textit{if } I \models \varphi\textit{, then } I \models \psi \\
I \models \varphi \leftrightarrow \psi & \textit{iff} & I \models \varphi \textit{ if and only if } I \models \psi
\end{array}$$

If $I \models \varphi$, we say that I satisfies φ, or that I is a model of φ. The set of all models of φ is denoted by $M(\varphi)$.

Semantics of Propositional Logic: Examples

Example

Formula: $\varphi = [(P \vee Q) \leftrightarrow (R \vee S)] \wedge [\neg(P \wedge Q) \wedge (R \wedge \neg S)]$

$\rightarrow$ For I with $I(P) = 1, I(Q) = 1, I(R) = 0, I(S) = 0$, do we have $I \models \varphi$? No: $(P \vee Q)$ is true but $(R \vee S)$ is false, so the left-hand side of the conjunction is false and the overall formula is false.

Example

Formula: $\varphi =$ Wumpus-[2,2] $\rightarrow$ Stench-[2,1]

$\rightarrow$ For I with $I($Wumpus-[2,2]$) = 0$, $I($Stench-[2,1]$) = 1$, do we have $I \models \varphi$? Yes: $\varphi = \psi_1 \rightarrow \psi_2$ is true iff either ψ_1 is false, or ψ_2 is true (i.e., $\psi_1 \rightarrow \psi_2$ has the same models as $\neg\psi_1 \vee \psi_2$).

Terminology

Knowledge Base, Models

A Knowledge Base (KB) is a set of formulas. An interpretation is a model of KB if $I \models \varphi$ for all $\varphi \in$ KB.

→ Knowledge Base = set of formulas, interpreted as a conjunction.

Satisfiability

A formula φ is:

- satisfiable if there exists I that satisfies φ.
- unsatisfiable if φ is not satisfiable.
- falsifiable if there exists I that doesn't satisfy φ.
- valid if $I \models \varphi$ holds for all I. We also call φ a tautology.

Equivalence

Formulas φ and ψ are equivalent, $\varphi \equiv \psi$, if $M(\varphi) = M(\psi)$.

Entailment

Remember (slide 5)? Does our knowledge of the cave entail a definite Wumpus position?

→ We don't know everything; what can we conclude from the things we *do* know?

Definition (Entailment). *Let Σ be a set of atomic propositions. We say that a set of formulas KB entails a formula φ, written KB $\models \varphi$, if φ is true in all models of KB, i.e., $M(\bigwedge_{\psi \in KB}) \subseteq M(\varphi)$. In this case, we also say that φ follows from KB.*

→ The following theorem is simple, but will be crucial later on:

Contradiction Theorem. *KB $\models \varphi$ if and only if KB $\cup \{\neg\varphi\}$ is unsatisfiable.*

Proof. "⇒": Say KB $\models \varphi$. Then for any I where $I \models$ KB we have $I \models \varphi$ and thus $I \not\models \neg\varphi$. "⇐": Say KB $\cup \{\neg\varphi\}$ is unsatisfiable. Then for any I where $I \models$ KB we have $I \not\models \neg\varphi$ and thus $I \models \varphi$.

→ Entailment can be tested via satisfiability.

The Truth Table Method

Want: Determine whether φ is satisfiable, valid, etc.

Method: Build the truth table, enumerating all interpretations of Σ.

Example

Is $\varphi = ((P \vee H) \wedge \neg H) \rightarrow P$ valid?

P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$(P \vee H) \wedge \neg H \rightarrow P$
0	0	0	0	1
0	1	1	0	1
1	0	1	1	1
1	1	1	0	1

→ Yes. φ is true for all possible combinations of truth values.

→ Is this a good method for answering these questions? No! For N propositions, the truth table has 2^N rows. [Satisfiability (validity) testing is **NP**-hard (**co-NP**-hard), but that pertains to *worst-case* behavior.]

Questionnaire

There are three persons, Stefan (S), Nicole (N), and Jochen (J). 1. Their hair colors are contained in black (bla), red (red), and green (gre). 2a. Their study subjects are contained in informatics (inf), physics (phy), chinese (chi) (or combinations thereof); 2b. at least one studies informatics. 3. Persons with red or green hair do not study informatics. 4. Neither the physics nor the chinese students have black hair. 5. Of the two male persons, one studies physics, and the other studies chinese.

Question!

Who studies informatics?

(A): Stefan

(B): Nicole

(C): Jochen

(D): Nobody

→ You can solve this using propositional logic. For every $x \in \{S, N, J\}$ we know that: 1. $bla(x) \vee red(x) \vee gre(x)$; 2a. $inf(x) \vee phy(x) \vee chi(x)$; 3. $inf(x) \rightarrow \neg red(x) \wedge \neg gre(x)$; 4. $phy(x) \rightarrow \neg bla(x)$ and $chi(x) \rightarrow \neg bla(x)$. Further, 2b. $inf(S) \vee inf(N) \vee inf(J)$ and 5. $(phy(S) \wedge chi(J)) \vee (chi(S) \wedge phy(J))$. For every $x \in \{S, N, J\}$, 1. and 3. entail (*) $inf(x) \rightarrow bla(x)$. 4. and 5. together entail $\neg bla(S) \wedge \neg bla(J)$, which with (*) entails $\neg inf(S) \wedge \neg inf(J)$. With 2b., the latter entails $inf(N)$.

Normal Forms

The two quintessential normal forms: (there are others as well)

- A formula is in conjunctive normal form (CNF) if it consists of a conjunction of disjunctions of literals:

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

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

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

→ Every formula has equivalent formulas in CNF and DNF.

Transformation to Normal Form

CNF Transformation (DNF Transformation: Analogously)

Exploit the equivalences:

1. $(\varphi \leftrightarrow \psi) \equiv [(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)]$ (Eliminate "$\leftrightarrow$")
2. $(\varphi \rightarrow \psi) \equiv (\neg\varphi \vee \psi)$ (Eliminate "$\rightarrow$")
3. $\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)$ and $\neg(\varphi \vee \psi) \equiv (\neg\varphi \wedge \neg\psi)$ and $\neg\neg\varphi \equiv \varphi$ (Move "$\neg$" inwards)
4. $[(\varphi_1 \wedge \varphi_2) \vee (\psi_1 \wedge \psi_2)] \equiv [(\varphi_1 \vee \psi_1) \wedge (\varphi_2 \vee \psi_1) \wedge (\varphi_1 \vee \psi_2) \wedge (\varphi_2 \vee \psi_2)]$ (Distribute "$\vee$" over "$\wedge$")

Example: $((P \vee H) \wedge \neg H) \rightarrow P$ (Blackboard).

→ Note: The formula may grow exponentially! ("Distribute" step)

→ However: Given φ, we can in polynomial time construct a CNF ψ that is satisfiable if and only if φ is. (Proof omitted)

Questionnaire

Question!

A CNF formula is . . .

(A): Valid iff at least one disjunction is valid.

(B): Valid iff every disjunction is valid.

(C): Satisfiable if at least one disjunction is satisfiable.

(D): Satisfiable if every disjunction is satisfiable.

→ (A): No, other parts of the global conjunction may be false under any one given interpretation.

→ (B): Yes: The CNF is a conjunction of valid formulas, so is valid itself. (Compare the CNF transformation of the example formula on slide 21).

→ (C): No since we need *all* disjuncts to be satisfied together.

→ (D): No since we need all disjuncts to be satisfied together *by the same interpretation*.

Deduction

Remember (slide 5)? Our knowledge of the cave entails a definite Wumpus position! → But how to find out about this? Deduction!

Basic Concepts in Deduction

- **Inference rule**: Rule prescribing how we can infer new formulas.

  → For example, if the KB is $\{\dots,(\varphi \to \psi),\dots,\varphi,\dots\}$ then ψ can be deduced using the inference rule $\frac{\varphi, \varphi \to \psi}{\psi}$.
- **Calculus**: Set $\mathcal{R}$ of inference rules.
- **Derivation**: φ can be derived from KB using $\mathcal{R}$, KB $\vdash_{\mathcal{R}} \varphi$, if starting from KB there is a sequence of applications of rules from $\mathcal{R}$, ending in φ.
- **Soundness**: $\mathcal{R}$ is sound if all derivable formulas do follow logically: if KB $\vdash_{\mathcal{R}} \varphi$, then KB $\models \varphi$.
- **Completeness**: $\mathcal{R}$ is complete if all formulas that follow logically are derivable: if KB $\models \varphi$, then KB $\vdash_{\mathcal{R}} \varphi$.

→ If $\mathcal{R}$ is sound and complete, then to check whether KB $\models \varphi$, we can check whether KB $\vdash_{\mathcal{R}} \varphi$.

Resolution: Quick Facts

Input: A CNF formula ψ.

Method: Calculus consisting of a single rule, allowing to produce disjunctions using fewer variables. We write $\psi \vdash \varphi$ if φ can be derived from ψ using resolution.

Output: Can an impossible φ (the empty disjunction) be derived? "Yes"/"No", where "yes" happens iff ψ is unsatisfiable.

→ So how do we check whether KB $\models \varphi$?

Proof by contradiction (cf. slide 17): Run resolution on $\psi :=$ CNF-transformation$(\text{KB} \cup \{\neg\varphi\})$. By the contradiction theorem, ψ is unsatisfiable iff KB $\models \varphi$.

→ Deduction can be reduced to proving unsatisfiability: "Assume, to the contrary, that KB holds but φ does not hold; then derive False".

Resolution: Conventions

→ For the remainder of this chapter, we assume that the input is a set Δ of clauses: (The same will be assumed in **Chapter 9**)

Terminology and Notation

- A literal l is an atom or the negation thereof (e.g., $P, \neg Q$); the negation of a literal is denoted $\overline{l}$ (e.g., $\overline{\neg Q} = Q$).
- A clause C is a disjunction of literals. We identify C with the set of its literals (e.g., $P \vee \neg Q$ becomes $\{P, \neg Q\}$).
- We identify a CNF formula ψ with the set Δ of its clauses (e.g., $(P \vee \neg Q) \wedge R$ becomes $\{\{P, \neg Q\}, \{R\}\}$).
- The empty clause is denoted $\Box$.

→ An interpretation I satisfies a clause C iff there exists $l \in C$ such that $I \models l$. I satisfies Δ iff, for all $C \in \Delta$, we have $I \models C$.

Resolution Conventions: Rim Cases

It's normally simple . . .

- E.g., I with $I(P) = 0, I(Q) = 0, I(R) = 0$ does not satisfy $\Delta = \{\{P, \neg Q\}, \{R\}\}$.

. . . but can be confusing in the "rim cases":

- Does there exist I so that $I \models \Box$? No, there exists no literal $l \in \Box$ that we can satisfy.
- With $\Delta = \{\Box\}$, does there exist I so that $I \models \Delta$? No, because we can't satisfy $\Box$.
- With $\Delta = \{\}$, does there exist I so that $I \models \Delta$? Yes, because any I satisfies all clauses $C \in \Delta$ (trivial as there is no clause in Δ).

The Resolution Rule

Definition (Resolution Rule). *Resolution uses the following inference rule (with exclusive union $\dot\cup$ meaning that the two sets are disjoint):*

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

If Δ contains parent clauses of the form $C_1 \dot\cup \{l\}$ and $C_2 \dot\cup \{\bar{l}\}$, the rule allows to add the resolvent clause $C_1 \cup C_2$. l and $\bar{l}$ are called the resolution literals.

Example: $\{P, \neg R\}$ resolves with $\{R, Q\}$ to $\{P, Q\}$.

Lemma. *The resolvent follows from the parent clauses.*

Proof. If $I \models C_1 \dot\cup \{l\}$ and $I \models C_2 \dot\cup \{\bar{l}\}$, then I must make at least one literal in $C_1 \cup C_2$ true.

Theorem (Soundness). *If $\Delta \vdash D$, then $\Delta \models D$.* (Direct from Lemma.)

→ What about the other direction? Is the resolvent *equivalent* to its parents?
No, because to satisfy the resolvent it is enough to satisfy one of C_1, C_2. E.g.: Setting $I(P) = 0$, $I(Q) = 1$, and $I(R) = 1$, we satisfy $\{P, Q\}$ but do not satisfy $\{P, \neg R\}$.

Using Resolution: A Simple Example

Input: KB $= \{Q \rightarrow \neg P, \neg P \rightarrow (\neg Q \vee \neg R \vee \neg S), \neg Q \rightarrow \neg S, \neg R \rightarrow \neg S\}$

$\phi = \neg S$

Question: Do we have KB $\models \phi$?

Step 1: Transform KB and $\neg\phi$ into CNF.

KB $\equiv (\neg Q \vee \neg P) \wedge (P \vee \neg Q \vee \neg R \vee \neg S) \wedge (Q \vee \neg S) \wedge (R \vee \neg S)$

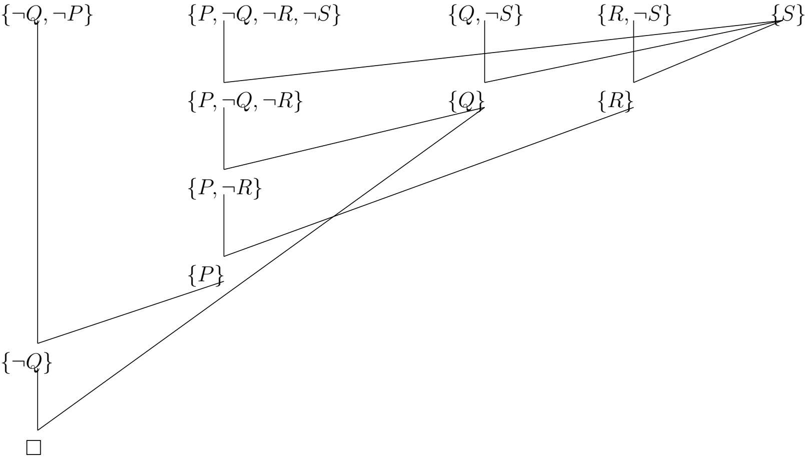
$\neg\phi = S$

Step 2: Write as set of clauses Δ.

$\Delta = \{\{\neg Q, \neg P\}, \{P, \neg Q, \neg R, \neg S\}, \{Q, \neg S\}, \{R, \neg S\}, \{S\}\}$

Using Resolution: A Simple Example

Step 3: Derive □ by applying the resolution rule.



$\{\neg Q, \neg P\}$ $\{P, \neg Q, \neg R, \neg S\}$ $\{Q, \neg S\}$ $\{R, \neg S\}$ $\{S\}$

$\{P, \neg Q, \neg R\}$ $\{Q\}$ $\{R\}$

$\{P, \neg R\}$

$\{P\}$

$\{\neg Q\}$

$\Box$

Using Resolution: A Frequent Mistake

Question: Given clauses $C_1 \dot\cup \{P, Q\}$ and $C_2 \dot\cup \{\neg P, \neg Q\}$, can we resolve them to $C_1 \cup C_2$?

Answer: NO!

Observation 1: Consider $\Delta = \{\{P, Q\}, \{\neg P, \neg Q\}\}$, and assume we were able to resolve as above. Then we could derive the empty clause. However, Δ is satisfiable (e.g. $P := T, Q := F$), so this deduction would be unsound.

Observation 2: The proof of the lemma on slide 28 is not valid for the hypothetical resolution of $C_1 \dot\cup \{P, Q\}$ and $C_2 \dot\cup \{\neg P, \neg Q\}$ to $C_1 \cup C_2$. An interpretation can set, e.g., $P := T, Q := F$, satisfying *both* $\{P, Q\}$ *and* $\{\neg P, \neg Q\}$ *together*, avoiding the need to satisfy either of C_1 or C_2.

Completeness

Is resolution complete? Does $\Delta \models \varphi$ imply $\Delta \vdash \varphi$?

→ No. Example: $\{\{P, Q\}, \{\neg Q, R\}\} \models \{P, R, S\}$ but $\{\{P, Q\}, \{\neg Q, R\}\} \not\vdash \{P, R, S\}$.

Applying resolution to $\{\{P, Q\}, \{\neg Q, R\}\}$, the only new clause we can derive is $\{P, R\}$. This is obviously stronger than $\{P, R, S\}$, but the syntactic operation defined by the resolution calculus cannot make that derivation.

BUT remember: Proof by contradiction!

→ Run resolution on $\psi :=$ CNF-transformation(KB $\cup \{\neg\varphi\}$): By the contradiction theorem, ψ is unsatisfiable iff KB $\models \varphi$.

→ This method *is* complete, see next slide.

Refutation-Completeness

Theorem (Refutation-Completeness). *Δ is unsatisfiable iff $\Delta \vdash \Box$.*

Proof. "If": Soundness. For "only if", we prove that, if $\Delta \not\vdash \Box$, then Δ is satisfiable.

Say $\Sigma = \{P_1, \dots, P_n\}$. Denote by $RC(\Delta) := \{C \mid \Delta \vdash C\}$ the set of all clauses derivable from Δ. Note that $\Delta \vdash \Box$ is the same as $\Box \in RC(\Delta)$.

Consider the algorithm attempting to construct an interpretation I, as follows: Start with $I = \emptyset$; for $i := 1, \dots, n$: (a) let $I' := I \cup \{P_i \mapsto 1\}$; if $RC(\Delta)$ does not contain a clause C s.t. C only uses variables from $\{P_1, \dots, P_i\}$ and is not satisfied by I' (i.e., C is empty under I'), let $I := I'$; else (b) let $I' := I \cup \{P_i \mapsto 0\}$; if $RC(\Delta)$ does not contain a clause C as in (a), let $I := I'$; else (c) fail. (In short: greedy value selection, no backtracking.)

We denote by I_j the assignment after iteration j.

Observe: With $\Box \in RC(\Delta)$, the algorithm fails at P_1 due to $C := \Box$. With $\Box \notin RC(\Delta)$, if the algorithm succeeds up to P_j, then (*) I_j satisfies all clauses from $RC(\Delta)$ using only variables from $\{P_1, \dots, P_j\}$.

Refutation-Completeness, Proof Continued

Reminder: With $\Box \in RC(\Delta)$, the algorithm fails at P_1 due to $C := \Box$. With $\Box \notin RC(\Delta)$, if the algorithm succeeds up to P_j, then (*) I_j satisfies all clauses from $RC(\Delta)$ using only variables from $\{P_1, \dots, P_j\}$.

Say that $\Box \notin RC(\Delta)$. Then the algorithm succeeds up to P_n, so with (*) returns a satisfying assignment for Δ. Assume failure at P_i to the contrary.

By construction, there must be $C_{(a)}, C_{(b)} \in RC(\Delta)$, using only variables from $\{P_1, \dots, P_i\}$, where (**) $C_{(a)}$ is not satisfied by $I_{i-1} \cup \{P_i \mapsto 1\}$ and $C_{(b)}$ is not satisfied by $I_{i-1} \cup \{P_i \mapsto 0\}$. With (*) applied to $j = i - 1$, $C_{(a)}$ and $C_{(b)}$ must use the variable P_i (or else they would be among the clauses guaranteed to be satisfied by I_{i-1}). With (**), we must have $\neg P_i \in C_{(a)}$ and $P_i \in C_{(b)}$.

Let $C_{(ab)}$ be the resolvent of $C_{(a)}$ and $C_{(b)}$. Then $C_{(ab)} \in RC(\Delta)$. Further, $C_{(ab)}$ uses only variables from $\{P_1, \dots, P_{i-1}\}$. Hence, by (*), I_{i-1} satisfies $C_{(ab)}$. This contradicts (**), showing the claim.

→ If $\Box \notin RC(\Delta)$, then greedy value selection based on $RC(\Delta)$ will necessarily find a satisfying assignment.

Questionnaire

Question!

What are resolvents of $\{P, \neg Q, R\}$ and $\{\neg P, Q, R\}$?

(A): $\{Q, \neg Q, P, R\}$. (B): $\{P, \neg P, R, S\}$.

(C): $\{R\}$. (D): $\{Q, \neg Q, R\}$.

→ (A): No. If we resolve on P then it disappears completely.

→ (B): No. By resolving on Q we get this clause except S, and although the larger clause always is sound as well of course, we are not allowed to deduce it by the rule.

→ (C): No. If we resolve on P then we get both Q and $\neg Q$ into the clause, similar if we resolve on Q.
→ We can resolve on only ONE literal at a time, cf. slide 30.

→ (D): Yes, this is what we get by resolving on P.

Where is the Wumpus? The Situation

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 $\boxed{A}$ S OK	2,2	3,2	4,2
1,1 V OK	2,1 B V OK	3,1	4,1

Where is the Wumpus? Our Knowledge

→ We worry only about the Wumpus and Stench ...

$S_{i,j}$ = Stench in (i,j), $W_{i,j}$ = Wumpus in (i,j).

Propositions whose value we know:

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

Knowledge about the wumpus and smell: From "Cell adjacent to Wumpus: Stench (else: None)", we get, amongst many others:

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

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

And Now Using Resolution Conventions

→ Consider Δ composed of the following clauses:

Propositions whose value we know:

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

Knowledge about the wumpus and smell:

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

Negated goal formula: $\{\neg W_{1,3}\}$

Resolution Proof Killing the Wumpus!

Derivation proving that the Wumpus is in $(1,3)$:

- "Assume the Wumpus is not in $(1,3)$. Then either there's no stench in $(1,2)$, or the Wumpus is in some other neighbor cell of $(1,2)$."

  Parents: $\{\neg W_{1,3}\}$ and $\{\neg S_{1,2}, W_{1,3}, W_{2,2}, W_{1,1}\}$.

  → Resolvent: $\{\neg S_{1,2}, W_{2,2}, W_{1,1}\}$.

- "There's a stench in $(1,2)$, so it must be another neighbor."

  Parents: $\{S_{1,2}\}$ and $\{\neg S_{1,2}, W_{2,2}, W_{1,1}\}$.

  → Resolvent: $\{W_{2,2}, W_{1,1}\}$.

- "We've been to $(1,1)$, and there's no Wumpus there, so it can't be $(1,1)$."

  Parents: $\{\neg W_{1,1}\}$ and $\{W_{2,2}, W_{1,1}\}$. → Resolvent: $\{W_{2,2}\}$.

- "There is no stench in $(2,1)$ so it can't be $(2,2)$ either, in contradiction."

  Parents: $\{\neg S_{2,1}\}$ and $\{S_{2,1}, \neg W_{2,2}\}$. → Resolvent: $\{\neg W_{2,2}\}$.

  Parents: $\{\neg W_{2,2}\}$ and $\{W_{2,2}\}$. → Resolvent: $\Box$.

Questionnaire

Question!

Do there exist "failed" Wumpus problems, where we can find a solution without risking death, but resolution is not strong enough for the reasoning required?

(A): Yes (B): No

→ No, because resolution is (refutation-)complete: Everything that can be concluded at all, can be concluded using resolution.

Question!

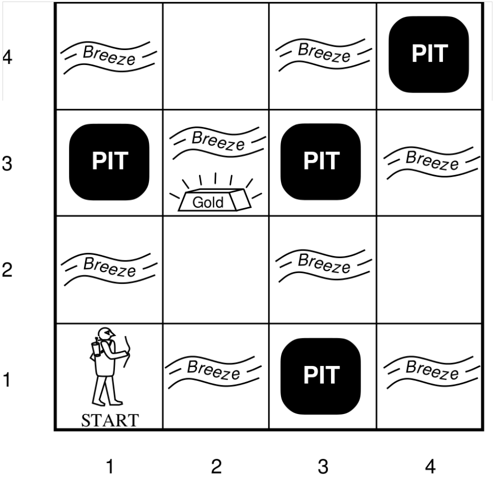
Do there exist "unsafe" Wumpus problems, that are solvable but where we cannot find the solution without risking death?

(A): Yes (B): No

→ Yes: See an example on the next slide.

Answer to 2nd Question from Previous Slide

Yes. For example this one:



Summary

- Sometimes, it pays off to think before acting.
- In AI, “thinking” is implemented in terms of reasoning in order to deduce new knowledge from a knowledge base represented in a suitable logic.
- Logic prescribes a syntax for formulas, as well as a semantics prescribing which interpretations satisfy them. φ entails ψ if all interpetations that satisfy φ also satisfy ψ. Deduction is the process of deriving new entailed formulas.
- Propositional logic formulas are built from atomic propositions, with the connectives “and, or, not”.
- Every propositional formula can be brought into conjunctive normal form (CNF), which can be identified with a set of clauses.
- Resolution is a deduction procedure based on trying to derive the empty clause. It is refutation-complete, and can be used to prove KB $\models \varphi$ by showing that KB $\cup \{\neg\varphi\}$ is unsatisfiable.

Issues with Propositional Logic

Awkward to write for humans: E.g., to model the Wumpus world we had to make a copy of the rules for every cell . . .

$R_1 : \neg S_{1,1} \rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$
$R_2 : \neg S_{2,1} \rightarrow \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1}$
$R_3 : \neg S_{1,2} \rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,2} \wedge \neg W_{1,3}$

Compared to “Cell adjacent to Wumpus: Stench (else: None)”, that is not a very nice description language . . .

Can we design a more human-like logic? Yep:

- Predicate logic: Quantification of variables ranging over objects. → **Chapters 10 and 11**
- . . . and a whole zoo of logics much more powerful still.
- Note: In applications, propositional CNF encodings are generated by computer programs. This mitigates (but does not remove!) the inconveniences of propositional modeling.

Reading

- *Chapter 7: Logical Agents*, Sections 7.1 – 7.5 [Russell and Norvig (2010)].

  Content: Sections 7.1 and 7.2 roughly correspond to my "Introduction", Section 7.3 roughly corresponds to my "Logic (in AI)", Section 7.4 roughly corresponds to my "Propositional Logic", Section 7.5 roughly corresponds to my "Resolution" and "Killing a Wumpus".

  Overall, the content is quite similar. I have tried to add some additional clarifying illustrations. RN gives many complementary explanations, nice as additional background reading.

  I would note that RN's presentation of resolution seems a bit awkward, and Section 7.5 contains some additional material that is imho not interesting (alternate inference rules, forward and backward chaining). Horn clauses and unit resolution (also in Section 7.5), on the other hand, are quite relevant.

References I

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (Third Edition)*. Prentice-Hall, Englewood Cliffs, NJ, 2010.