CS 480

Introduction to Artificial Intelligence

October 6, 2022

Announcements / Reminders

- Midterm Exam: October 13th!
  - Online section: please make arrangements. Contact Mr. Charles Scott (scott@iit.edu) if in doubt
- Written Assignment #02:
  - due: October 16th, 11:00 PM CST
- Quiz #02:
  - due: October 9th, 11:00 PM CST
- Programming Assignment #01:
  - due: October 15th, 11:00 PM CST
- Please follow the Week 07 To Do List instructions
- Spring Semester midterm course evaluation is still UP.
  - Please fill it out if you can. It means a lot to me.
- Grading TA assignment:

https://docs.google.com/spreadsheets/d/1ExS0bKnGt_fdf4LHa3YS1qRA7-Iq4xqXVjfSAPMaGVk/edit?usp=sharing

Plan for Today

- **Entailment**
- **Proof by Resolution**

Inference: The idea

The idea:

Given everything (expressed as sentences) that we know, can we infer if some query (another sentence) is true or not (satisfied or not)?

(Automated) Proof System

In AI we are interested in taking existing knowledge (sentences in KB) and from that:

- **deriving new knowledge (new sentences)**
- **answering questions (query sentences)**

In Propositional Logic this means showing that some sentence Q follows from a Knowledge Base KB

where:

- Q **- some query sentence**
- KB **- knowledge base (a sentence made of sentences)**

Propositional Logic: An Argument

An argument A **in propositional logic has the following form:**

A: P1

P2

...

PN

PREMISES

$\therefore$ C

CONCLUSION

Premises are taken for granted (assumed to be true**).**

Logical Entailment

A set of sentences (called premises) logically entails a sentence (called a conclusion) if and only if every truth assignment that satisfies the premises also satisfies the conclusion.

PREMISES □ CONCLUSION

Logical Entailment

Definition: A sentence KB entails a sentence Q (or Q follows from KB) if every model of KB is also a model of Q. We write:

$$KB \models Q$$

In other words:

- **For every interpretation in which** KB **is** true**,** Q **is also** true
- **"Whenever** KB **is** true**, Q is also** true**"**

Entailment: Deriving Conclusions

You can prove if:

$$\mathrm{KB} \models \mathrm{Q}$$

is true **in a number of ways:**

- **Model checking (enumeration)**
- **Truth table proof (enumeration)**
- **Proof by resolution**

You can also prove related sentences:

- **prove that** $\mathrm{KB} \wedge \neg \mathrm{Q}$ **is unsatisfiable (by contradiction)**
- **prove that** $\mathrm{KB} \Rightarrow \mathrm{Q}$ **is a tautology**

Entailment Proofs

$KB \equiv (p \Rightarrow q) \wedge (p \Rightarrow \neg r) \wedge (\neg p \Rightarrow \neg r) \quad | \quad Q \equiv \neg r$

Prove that $(KB \Rightarrow Q) \Leftrightarrow T$
(Show that $KB \Rightarrow Q$ is a tautology)

Prove that $(KB \wedge \neg Q) \Leftrightarrow \bot$
(Show that $KB \wedge \neg Q$ is a contradiction)

Proof by model checking
Show that all models that are true for Q are also true for KB

Model	p	q	r	$p \Rightarrow q$	$q \Rightarrow \neg r$	$\neg p \Rightarrow \neg r$	KB	Q	$KB \Rightarrow Q$	$KB \wedge \neg Q$
M1	true	true	true	true	false	true	false	false	true	false
M2	true	true	false	true	true	true	true	true	true	false
M3	true	false	true	false	true	true	false	false	true	false
M4	true	false	false	false	true	true	false	true	true	false
M5	false	true	true	true	false	false	false	false	true	false
M6	false	true	false	true	true	true	true	true	true	false
M7	false	false	true	true	true	false	false	false	true	false
M8	false	false	false	true	true	true	true	true	true	false

Entailment Proofs

$KB \equiv (p \Rightarrow q) \wedge (p \Rightarrow \neg r) \wedge (\neg p \Rightarrow \neg r)$ | $Q \equiv \neg r$

Prove that $(KB \Rightarrow Q) \Leftrightarrow T$
(Show that $KB \Rightarrow Q$ **is a tautology)**

Prove that $(KB \wedge \neg Q) \Leftrightarrow \bot$
(Show that $KB \wedge \neg Q$ **is a contradiction)**

Proof by model checking
Show that all models that are true for Q are also true for KB

$KB \Rightarrow Q$ **is true for all models, so KB entails Q**

Model	p	q	r	p⇒q	q⇒¬r	¬p⇒¬r	KB	Q	KB ⇒ Q	KB ∧ ¬ Q
M1	true	true	true	true	false	true	false	false	true	false
M2	true	true	false	true	true	true	true	true	true	false
M3	true	false	true	false	true	true	false	false	true	false
M4	true	false	false	false	true	true	false	true	true	false
M5	false	true	true	true	false	false	false	false	true	false
M6	false	true	false	true	true	true	true	true	true	false
M7	false	false	true	true	true	false	false	false	true	false
M8	false	false	false	true	true	true	true	true	true	false

Entailment Proofs

$KB \equiv (p \Rightarrow q) \wedge (p \Rightarrow \neg r) \wedge (\neg p \Rightarrow \neg r)$ | $Q \equiv \neg r$

Prove that $(KB \Rightarrow Q) \Leftrightarrow T$
(Show that $KB \Rightarrow Q$ is a tautology)

$KB \Rightarrow Q$ **is true for all models, so KB entails Q**

Prove that $(KB \wedge \neg Q) \Leftrightarrow \perp$
(Show that $KB \wedge \neg Q$ is a contradiction)

$KB \wedge \neg Q$ **is false for all models, so KB entails Q**

Proof by model checking
Show that all models that are true for Q are also true for KB

Model	p	q	r	$p \Rightarrow q$	$q \Rightarrow \neg r$	$\neg p \Rightarrow \neg r$	KB	Q	$KB \Rightarrow Q$	$KB \wedge \neg Q$
M1	true	true	true	true	false	true	false	false	true	false
M2	true	true	false	true	true	true	true	true	true	false
M3	true	false	true	false	true	true	false	false	true	false
M4	true	false	false	false	true	true	false	true	true	false
M5	false	true	true	true	false	false	false	false	true	false
M6	false	true	false	true	true	true	true	true	true	false
M7	false	false	true	true	true	false	false	false	true	false
M8	false	false	false	true	true	true	true	true	true	false

Entailment Proofs

KB $\equiv$ (p $\Rightarrow$ q) $\wedge$ (p $\Rightarrow \neg$ r) $\wedge$ ($\neg$ p $\Rightarrow \neg$ r) | Q $\equiv \neg$ r

Prove that (KB $\Rightarrow$ Q) $\Leftrightarrow$ T
(Show that KB $\Rightarrow$ Q is a tautology)

KB $\Rightarrow$ Q **is** true **for all models, so** KB **entails** Q

Prove that (KB $\wedge \neg$ Q) $\Leftrightarrow \bot$
(Show that KB $\wedge \neg$ Q is a contradiction)

KB $\wedge \neg$ Q **is** false **for all models, so** KB **entails** Q

Proof by model checking
Show that all models that are true **for** Q **are also** true **for** KB

M(KB) $\subseteq$ M(Q) **so** KB **entails** Q

Model	p	q	r	p⇒q	q⇒¬r	¬p⇒¬r	KB	Q	KB ⇒ Q	KB ∧ ¬ Q
M1	true	true	true	true	false	true	false	false	true	false
M2	true	true	false	true	true	true	true	true	true	false
M3	true	false	true	false	true	true	false	false	true	false
M4	true	false	false	false	true	true	false	true	true	false
M5	false	true	true	true	false	false	false	false	true	false
M6	false	true	false	true	true	true	true	true	true	false
M7	false	false	true	true	true	false	false	false	true	false
M8	false	false	false	true	true	true	true	true	true	false

Model Checking: Q is Satisfiable

$$KB \equiv P1 \wedge P2 \wedge P3 \mid Q \equiv .....$$

If $M(KB) \subseteq M(Q)$ Q **follows** KB**, otherwise it does NOT.**

Model	p	q	r	P1	P2	P3	KB	Q
M1	true	true	true	…	…	…	…	false
M2	true	true	false	…	…	…	…	true
M3	true	false	true	…	…	…	…	false
M4	true	false	false	…	…	…	…	false
M5	false	true	true	…	…	…	…	false
M6	false	true	false	…	…	…	…	false
M7	false	false	true	…	…	…	…	false
M8	false	false	false	…	…	…	…	false

Model Checking: Q is a Contradiction

$KB \equiv P1 \wedge P2 \wedge P3$ **|** $Q \equiv$ **.....**

Regardless of $M(KB) \subseteq M(Q)$ Q **will NOT follow** KB**.**

Model	p	q	r	P1	P2	P3	KB	Q
M1	true	true	true	...	...	...	...	**false**
M2	true	true	false	...	...	...	...	**false**
M3	true	false	true	...	...	...	...	**false**
M4	true	false	false	...	...	...	...	**false**
M5	false	true	true	...	...	...	...	**false**
M6	false	true	false	...	...	...	...	**false**
M7	false	false	true	...	...	...	...	**false**
M8	false	false	false	...	...	...	...	**false**

Model Checking: Q is a Tautology

$KB \equiv P1 \land P2 \land P3$ **|** $Q \equiv$ **.....**

Regardless of $M(KB) \subseteq M(Q)$ Q **WILL follow** KB**.**

Model	p	q	r	P1	P2	P3	KB	Q
M1	true	true	true	…	…	…	…	true
M2	true	true	false	…	…	…	…	true
M3	true	false	true	…	…	…	…	true
M4	true	false	false	…	…	…	…	true
M5	false	true	true	…	…	…	…	true
M6	false	true	false	…	…	…	…	true
M7	false	false	true	…	…	…	…	true
M8	false	false	false	…	…	…	…	true

What Does It Mean?

Some queries Q can be proven to follow KB or not without interpreting KB and Q. For example:

- **If Q is a tautology, Q will ALWAYS follow from KB no matter what KB is**
- **If Q is a contradiction, Q will NEVER follow from KB no matter what KB is**

This can be decided at the syntax level through deduction.

Again: Tautology Proved by Deduction

Laws/theorems in propositional logic can be used to prove additional theorems through a process known as deduction:

Prove that $((\neg m \vee n) \wedge \neg n) \Rightarrow \neg m$	**is a tautology:**
$((\neg m \wedge \neg n) \vee (n \wedge \neg n)) \Rightarrow \neg m$	by Distributive law $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$
$((\neg m \wedge \neg n) \vee \bot) \Rightarrow \neg m$	by Negation law (contradiction) $p \wedge \neg p \Leftrightarrow \bot$
$(\neg m \wedge \neg n) \Rightarrow \neg m$	by Identity law $p \vee \bot \Leftrightarrow p$
$\neg(\neg m \wedge \neg n) \vee \neg m$	by Implication law $\neg p \vee q \Leftrightarrow p \Rightarrow q$
$(\neg\neg m \vee \neg\neg n) \vee \neg m$	by De Morgan's law $\neg (p \wedge q) \Leftrightarrow \neg q \vee \neg p$
$(m \vee n) \vee \neg m$	by Double Negation law $\neg (\neg p) \Leftrightarrow p$
$m \vee (n \vee \neg m)$	by Associative law $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
$m \vee (\neg m \vee n)$	by Commutative law $p \vee q \Leftrightarrow q \vee p$
$(m \vee \neg m) \vee n$	by Associative law $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
$\mathsf{T} \vee n$	by Law of Excluded Middle $p \vee \neg p \Leftrightarrow \mathsf{T}$
$n \vee \mathsf{T}$	by Commutative law $p \vee q \Leftrightarrow q \vee p$
T	by Domination Law $p \vee \mathsf{T} \Leftrightarrow \mathsf{T}$

Proving Entailment: Two Levels

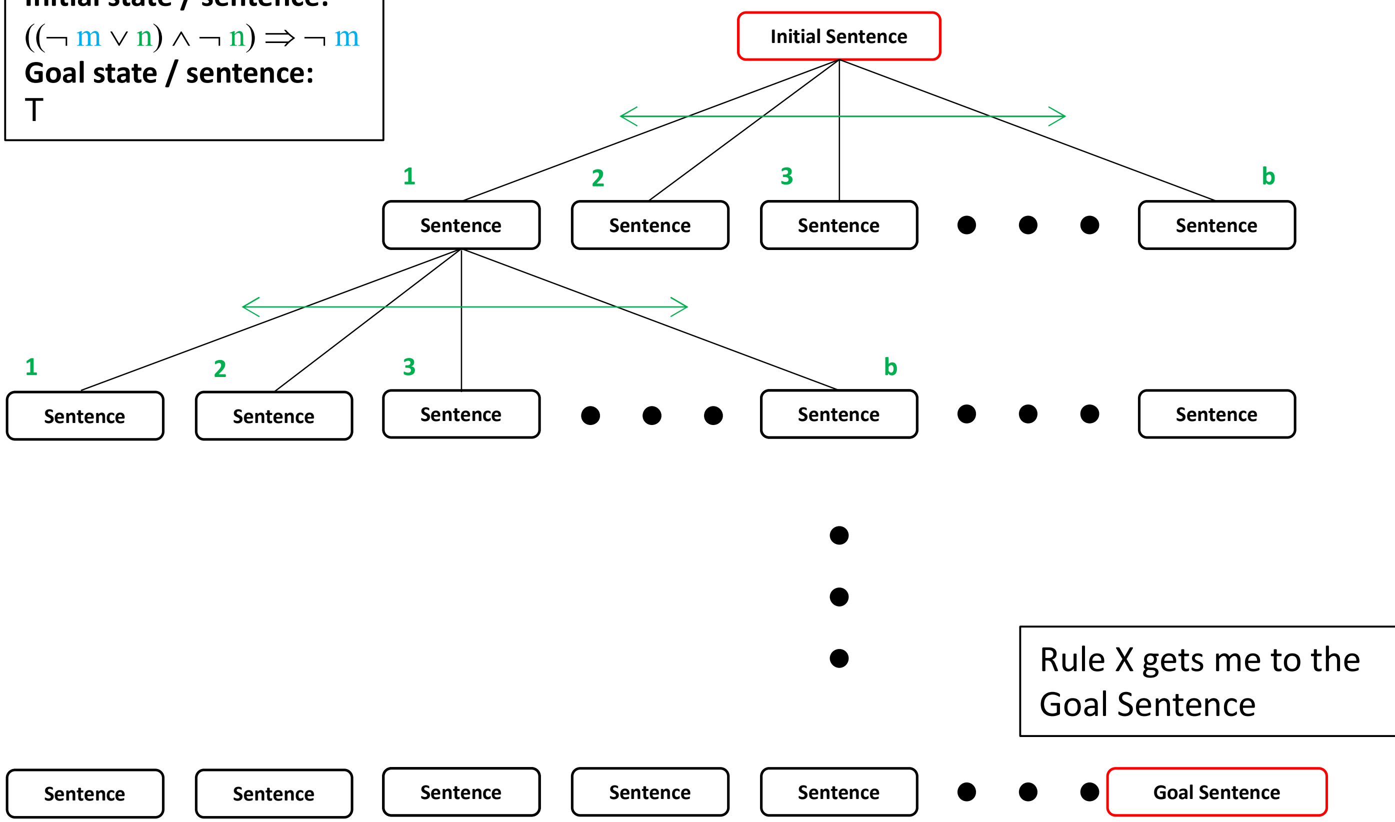
Syntax level

$$KB \xrightarrow{\textbf{Derivation}} Q$$

I was able to check entailment at this level for a particular problem, but will it always work?

Deduction as Search

Initial state / sentence:
$((\neg m \lor n) \land \neg n) \Rightarrow \neg m$
Goal state / sentence:
T



Depth: 0
Pick a rule/law

Depth: 1
Pick a rule/law

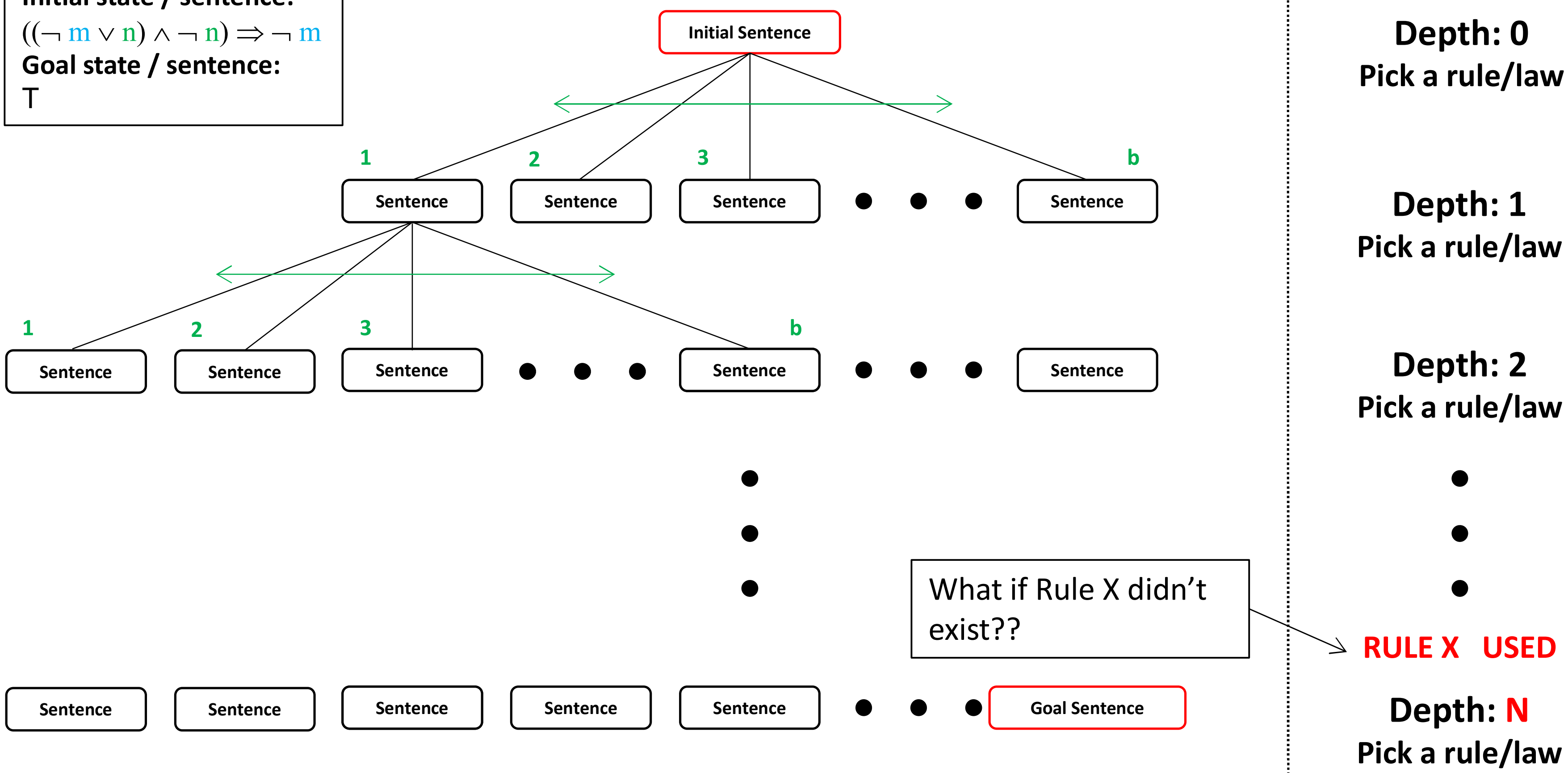
Depth: 2
Pick a rule/law

RULE X USED

Depth: N
Pick a rule/law

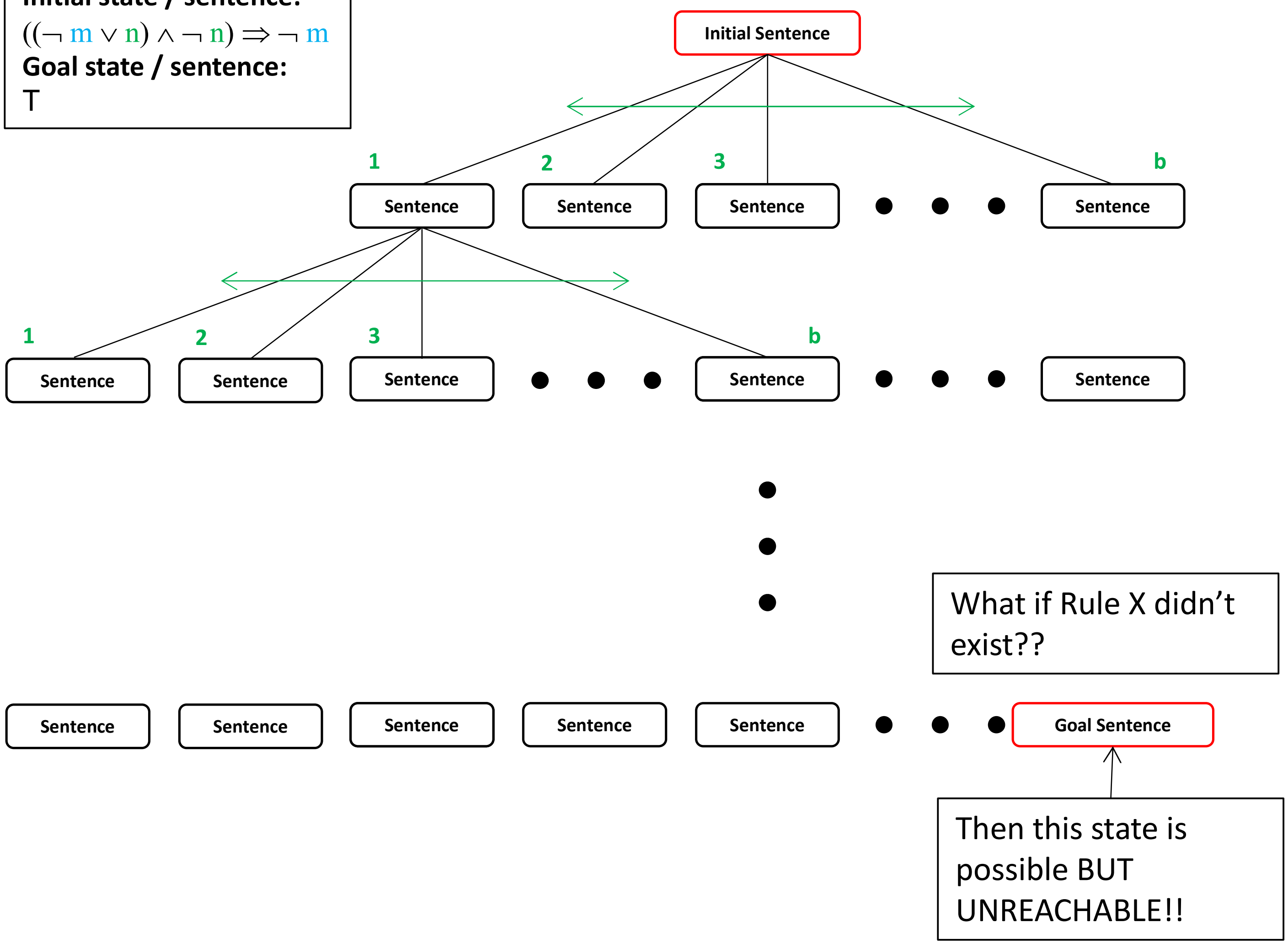
Deduction as Search

Initial state / sentence:
$((\neg m \vee n) \wedge \neg n) \Rightarrow \neg m$
Goal state / sentence:
T



Deduction as Search

Initial state / sentence:
$((\neg m \lor n) \land \neg n) \Rightarrow \neg m$
Goal state / sentence:
T



Depth: 0
Pick a rule/law

Depth: 1
Pick a rule/law

Depth: 2
Pick a rule/law

●
●
●

RULE X USED

Depth: N
Pick a rule/law

Propositional Logic Calculus

Syntactic proof systems are called calculi.

To ensure that a calculus DOES NOT generate errors, two properties need to be satisfied:

- **A calculus is SOUND if every derived proposition follows semantically**
- **A calculus is COMPLETE if all semantic consequences can be derived**

Propositional Logic Calculus

Soundness:

The calculus does NOT produce any FALSE consequences

Completeness:

A complete calculus ALWAYS find a proof if the sentence to be proved follows from the knowledge base

If a calculus is sound and complete, then syntactic derivation and semantic entailment are two equivalent relations.

Entailment: Two Levels

Syntax level

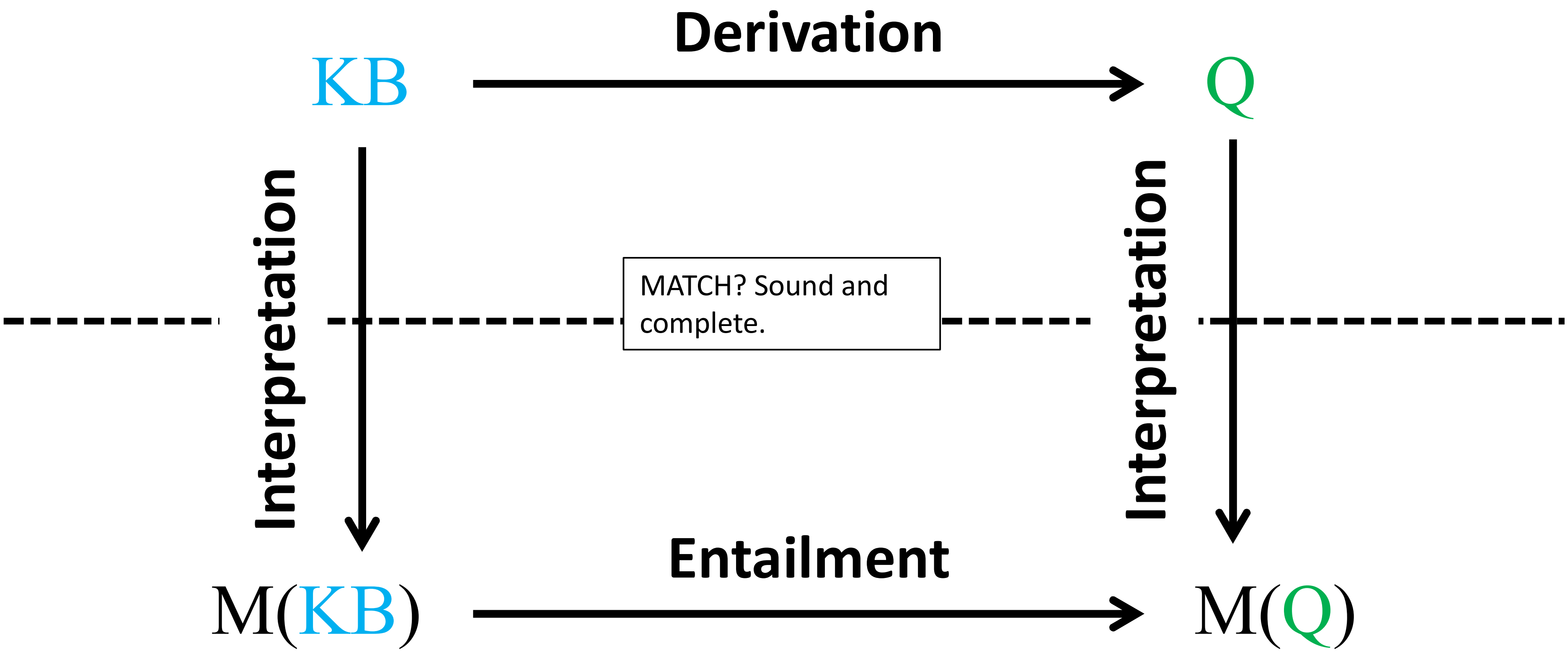
$$KB \xrightarrow{\text{Derivation}} Q$$

$$M(KB) \xrightarrow{\text{Entailment}} M(Q)$$

Semantic level

Proving Entailment: Two Levels

Syntax level



Semantic level

Inference

Bottom line:

An inference system has to be sound and complete.

Resolution rule is. Couple it with a complete search algorithm and an inference system is in place.

Inference Rules: Resolution

Rules of Inference:

Modus Ponens	Modus Tollens	Hypothetical Syllogism (Transitivity)	Conjunction
$P \Rightarrow Q$ P	$P \Rightarrow Q$ $\neg Q$	$P \Rightarrow Q$ $Q \Rightarrow R$	P Q
$\therefore Q$	$\therefore \neg P$	$\therefore P \Rightarrow R$	$\therefore P \wedge Q$

Addition	Simplification	Disjunctive Syllogism	Resolution
P	$P \wedge Q$	$P \vee Q$ $\neg P$	$P \vee Q$ $\neg P \vee R$
$\therefore P \vee Q$	$\therefore P$	$\therefore Q$	$\therefore Q \vee R$

Tautological forms:

Modus Ponens: $((P \Rightarrow Q) \wedge P) \Rightarrow Q$ | **Modus Tollens:** $((P \Rightarrow Q) \wedge \neg Q) \Rightarrow \neg P$

Hypothetical Syllogism: $((P \Rightarrow Q) \wedge (Q \Rightarrow R)) \Rightarrow (P \Rightarrow R)$

Disjunctive Syllogism: $((P \vee Q) \wedge \neg P) \Rightarrow \neg Q$

Addition: $P \Rightarrow P \vee Q$ | **Simplification:** $(P \wedge Q) \Rightarrow P$

Conjunction: $(P) \wedge (Q) \Rightarrow (P \wedge Q)$ | **Resolution:** $((P \vee Q) \wedge (\neg P \vee R)) \Rightarrow (Q \vee R)$

Proof by Resolution

Recall that we can show that KB **entails sentence** Q **(or** Q **follows from** KB**):**

$$KB \ \square\ Q$$

by proving that:

$$(KB \wedge \neg Q) \Leftrightarrow \bot$$

(show that $KB \wedge \neg Q$ **is a contradiction / <u>empty clause</u>)**

Resolution: Two Forms of Notation

Resolution
$P \lor Q$
$\neg P \lor R$
$\therefore\ Q \lor R$

Resolution (textbook)
$$\frac{(P \lor Q),\ (\neg P \lor R)}{(Q \lor R)}$$

Resolution: Two Forms of Notation

Resolution

$P \vee Q$

$\neg P \vee R$

$\therefore\ Q \vee R$

Resolution (textbook)

$$\frac{(P \vee Q),\ (\neg P \vee R)}{(Q \vee R)}$$

derived clause (resolvent)

The Empty Clause: $(p \wedge \neg p) \Leftrightarrow \bot$

Symbol	Name	Alternative symbols*	Should be read
$\neg$	Negation	$\sim$, !	not
$\wedge$	(Logical) conjunction	$\bullet$, &	and
$\vee$	(Logical) disjunction	$+$, $\|\|$	or
$\Rightarrow$	(Material) implication	$\rightarrow$, $\supset$	implies
$\Leftrightarrow$	(Material) equivalence	$\leftrightarrow$, $\equiv$, iff	if and only if
T	Tautology	T, 1, $\blacksquare$	truth
$\bot$	Contradiction	F, 0, $\square$	falsum empty clause
$\therefore$	Therefore		therefore

* you can encounter it elsewhere in literature

Conjunctive Normal Form (CNF)

A sentence is in conjunctive normal form (CNF) if and only if consists of conjunction:

$$K_1 \wedge K_2 \wedge \ldots \wedge K_m$$

of clauses. A clause Ki consists of a disjunction

$$(l_{i1} \vee l_{i2} \vee \ldots \vee l_{ini})$$

of literals. Finally, a literal is propositional variable (positive literal) or a negated propositional variable (negative literal).

Conjunctive Normal Form (CNF

Example:

$$(a \vee b \vee \neg c) \wedge (a \vee b \vee \neg c) \wedge (\neg b \vee \neg c)$$

where: a, b, c **are literals.**

Conjunctive Normal Form (CNF)

Example:

Convert $m \Leftrightarrow (n \vee o)$ **into CNF**:

by Equivalence law $(p \Rightarrow q) \wedge (q \Rightarrow p) \;\Leftrightarrow\; (p \Leftrightarrow q)$

$$(m \Rightarrow (n \vee o)) \wedge ((n \vee o) \Rightarrow m)$$

by Implication law $\neg p \vee q \Leftrightarrow p \Rightarrow q$

$$(\neg m \vee (n \vee o)) \wedge (\neg (n \vee o) \vee m)$$

we can remove parentheses

$$(\neg m \vee n \vee o) \wedge (\neg (n \vee o) \vee m)$$

by De Morgan's law $\neg (p \wedge q) \Leftrightarrow \neg q \vee \neg p$

$$(\neg m \vee n \vee o) \wedge ((\neg n \wedge \neg o) \vee m)$$

by Distributive law $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$

$$(\neg m \vee n \vee o) \wedge (\neg n \vee m) \wedge (\neg o \vee m)$$

Conjunctive Normal Form (CNF)

Example:

Sentence $m \Leftrightarrow (n \vee o)$ **converted into CNF:**

literals

$$(\neg m \vee n \vee o) \wedge (\neg n \vee m) \wedge (\neg o \vee m)$$

clauses

CNF Grammar

$$
\begin{aligned}
\mathit{CNFSentence} &\rightarrow \mathit{Clause}_1 \wedge \cdots \wedge \mathit{Clause}_n \\
\mathit{Clause} &\rightarrow \mathit{Literal}_1 \vee \cdots \vee \mathit{Literal}_m \\
\mathit{Fact} &\rightarrow \mathit{Symbol} \\
\mathit{Literal} &\rightarrow \mathit{Symbol} \mid \neg \mathit{Symbol} \\
\mathit{Symbol} &\rightarrow P \mid Q \mid R \mid \ldots \\
\mathit{HornClauseForm} &\rightarrow \mathit{DefiniteClauseForm} \mid \mathit{GoalClauseForm} \\
\mathit{DefiniteClauseForm} &\rightarrow \mathit{Fact} \mid (\mathit{Symbol}_1 \wedge \cdots \wedge \mathit{Symbol}_l) \Rightarrow \mathit{Symbol} \\
\mathit{GoalClauseForm} &\rightarrow (\mathit{Symbol}_1 \wedge \cdots \wedge \mathit{Symbol}_l) \Rightarrow \mathit{False}
\end{aligned}
$$

* **I will:**

- **be using** true **and** false **instead of** True **and** False
- **use lowercase** p, q **for atomic and uppercase** P, Q **for complex**

General Resolution Rule

General resolution rule allows clauses with arbitrary number of literals

$$\frac{(a_1 \vee \ldots \vee a_m \vee b), (\neg\, b \vee c_1 \vee \ldots \vee c_n)}{(a_1 \vee \ldots \vee a_m \vee c_1 \vee \ldots \vee c_n)}$$

where: a_i, b, $\neg\, b$, c_j **are literals.**

Unit Resolution

General resolution rule allows clauses with arbitrary number of literals

initial clauses

$$\frac{(a_1 \vee \ldots \vee a_m \vee \mathbf{b}), (\neg\, \mathbf{b} \vee c_1 \vee \ldots \vee c_n)}{(a_1 \vee \ldots \vee a_m \vee c_1 \vee \ldots \vee c_n)}$$

new clause

Literals $\mathbf{b}$ and $\neg\, \mathbf{b}$ are complimentary. The resolution rule deletes a pair of complimentary literals from two clauses and combines the rest.

Unit Resolution

General resolution rule allows clauses with arbitrary number of literals

initial clauses

$$\frac{(a_1 \vee \ldots \vee a_m \vee \mathbf{b}),\ (\neg\, \mathbf{b} \vee c_1 \vee \ldots \vee c_n)}{(a_1 \vee \ldots \vee a_m \vee c_1 \vee \ldots \vee c_n)}$$

derived clause (resolvent)

Literals $\mathbf{b}$ and $\neg\, \mathbf{b}$ are complimentary. The clause ($\mathbf{b} \wedge \neg\, \mathbf{b}$) is a contradiction (an <u>empty clause</u>).

Unit Resolution

General resolution rule allows clauses with arbitrary number of literals

$$\frac{(a_1 \vee \ldots \vee a_m \vee \mathbf{b}), (\neg\, \mathbf{b} \vee c_1 \vee \ldots \vee c_n)}{(a_1 \vee \ldots \vee a_m \vee c_1 \vee \ldots \vee c_n)}$$

Literals b and ¬ b are complimentary. The resolution rule deletes a pair of complimentary literals from two clauses and combines the rest.

Factorization

Ocassionally, unit resolution will produce a new clause with the the following clause ($\mathbf{d} \lor \mathbf{d}$):

$$\frac{(a_1 \lor \ldots \lor a_m \lor \mathbf{d} \lor b), (\neg\, b \lor c_1 \lor \ldots \lor c_n \lor \mathbf{d})}{(a_1 \lor \ldots \lor a_m \lor c_1 \lor \ldots \lor c_n \lor \mathbf{d} \lor \mathbf{d})}$$

Disjunction of multiple copies of literals ($\mathbf{d} \lor \mathbf{d}$) can be replaced by a single literal $\mathbf{d}$. This is called factorization.

Resolution and Factorization

In this example resolution along with factorization will generate a new clause:

$$\frac{(a_1 \lor \ldots \lor a_m \lor \mathbf{d} \lor \mathbf{b}),\ (\neg\, \mathbf{b} \lor c_1 \lor \ldots \lor c_n \lor \mathbf{d})}{(a_1 \lor \ldots \lor a_m \lor c_1 \lor \ldots \lor c_n \lor \mathbf{d})}$$

Clause is ($\mathbf{d} \lor \mathbf{d}$) is replaced by a single literal $\mathbf{d}$. This is called factorization. Contradiction ($\mathbf{b} \land \neg\, \mathbf{b}$) becomes an "empty clause" and is removed.

Proof by Resolution

The process of proving by resolution is as follows:

A. Formalize the problem: “English to Propositional Logic”
B. derive $KB \wedge \neg Q$
C. convert $KB \wedge \neg Q$ into CNF (“standardized”) form
D. Apply resolution rule to resulting clauses. New clauses will be generated (add them to the set if not already present)
E. Repeat (C) until:
    a. no new clause can be added (KB does NOT entail Q)
    b. last two clauses resolve to yield the empty clause (KB entails Q)

Logical Entailment

So far, we have been asking the question:

"Does KB entail Q (does Q follow from KB)?"

KB □ Q

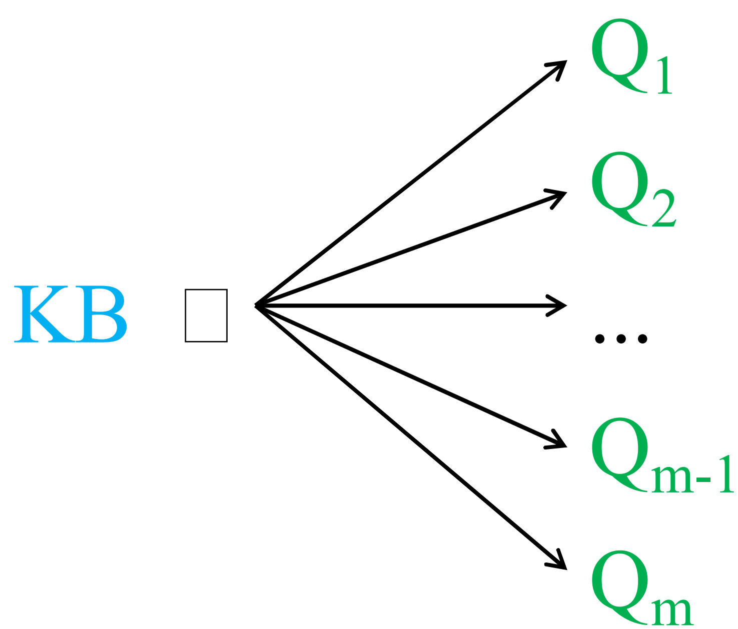
But we could ask the following question:

"Which Qs follow from KB?"

Logical Entailment

But we could ask the following question:

"Which Qs follow from KB?"



Logical Entailment

But we could ask the following question:

"Which Qs follow from KB?"

KB $\models$

- Q_1 follows from KB
- Q_2 does NOT
- …
- Q_{m-1} follows from KB
- Q_m follows from KB

KB Agents

Sentences are physical configurations of the agent, and reasoning is a process of constructing new physical configurations from old ones.

Logical reasoning should ensure that the new configurations represent aspects of the world that actually follow from the aspects that the old configurations represent.

Knowledge-based Agents

function KB-AGENT(*percept*) **returns** an *action*
  persistent: *KB*, a knowledge base
    t, a counter, initially 0, indicating time

KBBEFORE

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*

CURRENTKB

new percept

CURRENTKB ⇔ KBBEFORE ∧ percept

Logical Entailment with KB Agents

But we could ask the following question:

"Which ACTIONs follow from CURRENTKB?"

CURRENTKB □

$ACTION_1$ follows from

$ACTION_2$ does NOT follow

…

$ACTION_{m-1}$ follows from

$ACTION_m$ follows from

Logical Entailment with KB Agents

But we could ask the following question:

"Which ACTIONs follow from CURRENTKB?"

(KBBEFORE $\wedge$ percept) □

$ACTION_1$ follows from

$ACTION_2$ does NOT follow

…

$ACTION_{m-1}$ follows from

$ACTION_m$ follows from

Logical Entailment with KB Agents

Let's try a simpler example with just ONE ACTION to consider. The question is:

"Does ACTION follow from CURRENTKB?"

Test / prove:

(KBBEFORE $\wedge$ percept) □ ACTION follows from

to decide whether to apply ACTION or not.

KB Agent: Should I Stay or Should I Go

Problem: KB Agent wants to cross the street. Traffic comes from left an right. KB agent cannot cross if there is ANY traffic (ignore the STOP sign).

KB Agent: Should I Stay or Should I Go

Sensor

PERCEPT:

Knowledge Base and Inference System

Actuator

ACTION:

KB Agent: Street Crosser

Environment: Street

KB Agent: Should I Stay or Should I Go

KB Agent: Street Crosser

Environment: Street

Proof by Resolution

The process of proving by resolution is as follows:

A. **Formalize the problem: "English to Propositional Logic"**

B. **derive** $KB \wedge \neg Q$

C. **convert** $KB \wedge \neg Q$ **into CNF ("standardized") form**

D. **Apply resolution rule to resulting clauses. New clauses will be generated (add them to the set if not already present)**

E. **Repeat (C) until:**

   a. **no new clause can be added (**KB **does NOT entail** Q**)**

   b. **last two clauses resolve to yield the empty clause (**KB **entails** Q**)**

Street Crosser: Knowledge Base KB

English:

A: "Walk if and only if there is NO traffic coming from the left AND NO traffic coming from the right."

or

B: "DON'T walk if and only if there is traffic coming from the left OR traffic coming from the right."

Street Crosser: Knowledge Base KB

English and Propositional Logic:

A: "Walk if and only if there is NO traffic coming from the left AND NO traffic coming from the right."

$$\text{walk} \Leftrightarrow (\neg\text{trafficLeft} \wedge \neg\text{trafficRight})$$

or

B: "DON'T walk if and only if there is traffic coming from the left OR traffic coming from the right."

$$\neg\text{walk} \Leftrightarrow (\text{trafficLeft} \vee \text{trafficRight})$$

Street Crosser: Convert KB to CNF

Variant A: $KB \equiv walk \Leftrightarrow (\neg trafficLeft \wedge \neg trafficRight)$

$$(walk \Rightarrow (\neg trafficLeft \wedge \neg trafficRight)) \wedge ((\neg trafficLeft \wedge \neg trafficRight) \Rightarrow walk)$$

by Biconditional Elimination

$$(\neg walk \vee (\neg trafficLeft \wedge \neg trafficRight)) \wedge (\neg(\neg trafficLeft \wedge \neg trafficRight) \vee walk)$$

by Implication Elimination

$$((\neg walk \vee \neg trafficLeft) \wedge (\neg walk \vee \neg trafficRight)) \wedge (\neg(\neg trafficLeft \wedge \neg trafficRight) \vee walk)$$

by Distributivity Rule

$$((\neg walk \vee \neg trafficLeft) \wedge (\neg walk \vee \neg trafficRight)) \wedge ((\neg\neg trafficLeft \vee \neg\neg trafficRight) \vee walk)$$

by De Morgan's Rule

$$((\neg walk \vee \neg trafficLeft) \wedge (\neg walk \vee \neg trafficRight)) \wedge ((trafficLeft \vee trafficRight) \vee walk)$$

by Double Negation Elimination

$$(\neg walk \vee \neg trafficLeft) \wedge (\neg walk \vee \neg trafficRight) \wedge (trafficLeft \vee trafficRight \vee walk)$$

remove extraneous parentheses

Street Crosser: Convert KB to CNF

Variant B: $KB \equiv \neg walk \Leftrightarrow (trafficLeft \lor trafficRight)$

$(\neg walk \Rightarrow (trafficLeft \lor trafficRight)) \land ((trafficLeft \lor trafficRight) \Rightarrow \neg walk)$

by Biconditional Elimination

$(\neg(\neg walk) \lor (trafficLeft \lor trafficRight)) \land (\neg(trafficLeft \lor trafficRight) \lor \neg walk)$

by Implication Elimination

$(walk \lor (trafficLeft \lor trafficRight)) \land (\neg(trafficLeft \lor trafficRight) \lor \neg\, walk)$

by Double Negation Elimination

$(walk \lor (trafficLeft \lor trafficRight)) \land ((\neg trafficLeft \land \neg trafficRight) \lor \neg\, walk)$

by De Morgan's Rule

$(walk \lor (trafficLeft \lor trafficRight)) \land ((\neg trafficLeft \lor \neg\, walk) \land (\neg trafficRight \lor \neg\, walk))$

by Distributivity Rule

$(walk \lor trafficLeft \lor trafficRight) \land (\neg trafficLeft \lor \neg\, walk) \land (\neg trafficRight \lor \neg\, walk)$

remove extraneous parentheses

Should I Go: Proof by Resolution

We have our knowledge base KB **in CNF ready:**

$$KB \equiv (\neg walk \vee \neg trafficLeft) \wedge (\neg walk \vee \neg trafficRight) \wedge (trafficLeft \vee trafficRight \vee walk)$$

Let's rename propositional variables to simplify:

$$KB \equiv (w \vee tL \vee tR) \wedge (\neg tL \vee \neg w) \wedge (\neg tR \vee \neg w)$$

Assume that traffic is coming from both left and right (percepts):

$$PERCEPTS \equiv (tR) \wedge (tL)$$

Let's add (TELL) PERCEPTS **to the Knowledge Base** KB**:**

$$KB_N \equiv KB \wedge PERCEPTS$$

$$KB_N \equiv (w \vee tL \vee tR) \wedge (\neg tL \vee \neg w) \wedge (\neg tR \vee \neg w) \wedge (tR) \wedge (tL)$$

Our query Q **is "Should I (choose action) walk?":**

$Q \equiv w$ **(and in negated form:** $\neg Q \equiv \neg w$**)**

To test / prove entailment I want to prove that $KB_N \wedge \neg Q$ **is true:**

$$KB_N \wedge \neg Q \equiv (w \vee tL \vee tR) \wedge (\neg tL \vee \neg w) \wedge (\neg tR \vee \neg w) \wedge (tR) \wedge (tL) \wedge (\neg w)$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (tR)_4 \wedge (tL)_5 \wedge (\neg w)_6$$

Should I Go: Proof by Resolution

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (tR)_4 \wedge (tL)_5 \wedge (\neg w)_6$$

Should I Go: Proof by Resolution

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (tR)_4 \wedge (tL)_5 \wedge (\neg w)_6$$

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. (tR)
5. (tL)
6. $(\neg w)$

Added clauses:

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (tR)_4 \wedge (tL)_5 \wedge (\neg w)_6$$

Resolution applied to clauses 1 and 6

$$\frac{(w \vee tL \vee tR),\ (\neg w)}{(tL \vee tR)}$$

Produces a new clause ($tL \vee tR$). We can add it to the list as clause (7).

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. (tR)
5. (tL)
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (tR)_4 \wedge (tL)_5 \wedge (\neg w)_6$$

Resolution applied to clauses 2 and 5

$$\frac{(\neg tL \vee \neg w), (tL)}{(\neg w)}$$

Produces a clause ($\neg w$), but we already have it (6). Don't add it to the list.

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. (tR)
5. (tL)
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (tR)_4 \wedge (tL)_5 \wedge (\neg w)_6$$

Resolution applied to clauses 3 and 4

$$\frac{(\neg tR \vee \neg w),\ (tR)}{(\neg w)}$$

Produces a clause ($\neg w$), but we already have it (6). Don't add it to the list.

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. (tR)
5. (tL)
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (tR)_4 \wedge (tL)_5 \wedge (\neg w)_6$$

Resolution applied to clauses 2 and 7

$$\frac{(\neg tL \vee \neg w), (tL \vee tR)}{(\neg w \vee tR)}$$

Produces a new clause ($\neg w \vee tR$). We can add it to the list as clause (8).

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. (tR)
5. (tL)
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$
8. $(\neg w \vee tR)$

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (tR)_4 \wedge (tL)_5 \wedge (\neg w)_6$$

Resolution applied to clauses 1 and 8

$$\frac{(w \vee tL \vee tR), (\neg w \vee tR)}{(tL \vee tR)}$$

Produces a clause ($tL \vee tR$), but we already have it (7). Don't add it to the list.

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. (tR)
5. (tL)
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$
8. $(\neg w \vee tR)$

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (tR)_4 \wedge (tL)_5 \wedge (\neg w)_6$$

Resolution applied to clauses 3 and 8

$$\frac{(\neg tR \vee \neg w), (\neg w \vee tR)}{(\neg w)}$$

Produces a clause ($\neg w$), but we already have it (6). Don't add it to the list.

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. (tR)
5. (tL)
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$
8. $(\neg w \vee tR)$

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (tR)_4 \wedge (tL)_5 \wedge (\neg w)_6$$

Resolution applied to clauses 3 and 7

$$\frac{(\neg tR \vee \neg w),\ (tL \vee tR)}{(\neg w \vee tL)}$$

Produces a new clause ($\neg w \vee tL$). We can add it to the list as clause (9).

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. (tR)
5. (tL)
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$
8. $(\neg w \vee tR)$
9. $(\neg w \vee tL)$

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (tR)_4 \wedge (tL)_5 \wedge (\neg w)_6$$

Resolution applied to clauses 2 and 9

$$\frac{(\neg tL \vee \neg w),\ (\neg w \vee tL)}{(\neg w)}$$

Produces a clause ($\neg w$), but we already have it (6). Don't add it to the list.

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. (tR)
5. (tL)
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$
8. $(\neg w \vee tR)$
9. $(\neg w \vee tL)$

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (tR)_4 \wedge (tL)_5 \wedge (\neg w)_6$$

At this point, we tried to resolve all promising clause pairs, but we have not reached an empty clause $\rightarrow$ KB **does NOT entail** Q.

Given PERCEPTS: $(tR) \wedge (tL)$

we should NOT apply action walk (w) **and stay.**

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. (tR)
5. (tL)
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$
8. $(\neg w \vee tR)$
9. $(\neg w \vee tL)$

Should I Go: Proof by Resolution

We have our knowledge base KB in CNF ready:

$$KB \equiv (\neg walk \lor \neg trafficLeft) \land (\neg walk \lor \neg trafficRight) \land (trafficLeft \lor trafficRight \lor walk)$$

Let's rename propositional variables to simplify:

$$KB \equiv (w \lor tL \lor tR) \land (\neg tL \lor \neg w) \land (\neg tR \lor \neg w)$$

Assume that traffic is NOT coming from both left and right (percepts):

$$PERCEPTS \equiv (\neg tR) \land (\neg tL)$$

Let's add (TELL) PERCEPTS to the Knowledge Base KB:

$$KB_N \equiv KB \land PERCEPTS$$

$$KB_N \equiv (w \lor tL \lor tR) \land (\neg tL \lor \neg w) \land (\neg tR \lor \neg w) \land (\neg tR) \land (\neg tL)$$

Our query Q is "Should I (choose action) walk?":

$Q \equiv w$ **(and in negated form: $\neg Q \equiv \neg w$)**

To test / prove entailment I want to prove that $KB_N \land \neg Q$ is true:

$$KB_N \land \neg Q \equiv (w \lor tL \lor tR) \land (\neg tL \lor \neg w) \land (\neg tR \lor \neg w) \land (\neg tR) \land (\neg tL) \land (\neg w)$$

$$(w \lor tL \lor tR)_1 \land (\neg tL \lor \neg w)_2 \land (\neg tR \lor \neg w)_3 \land (\neg tR)_4 \land (\neg tL)_5 \land (\neg w)_6$$

Should I Go: Proof by Resolution

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (\neg tR)_4 \wedge (\neg tL)_5 \wedge (\neg w)_6$$

Should I Go: Proof by Resolution

Prove:

$$KB_N \land \neg Q \equiv$$

$$(w \lor tL \lor tR)_1 \land (\neg tL \lor \neg w)_2 \land (\neg tR \lor \neg w)_3 \land (\neg tR)_4 \land (\neg tL)_5 \land (\neg w)_6$$

Known clauses:

1. $(w \lor tL \lor tR)$
2. $(\neg tL \lor \neg w)$
3. $(\neg tR \lor \neg w)$
4. $(\neg tR)$
5. $(\neg tL)$
6. $(\neg w)$

Added clauses:

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (\neg tR)_4 \wedge (\neg tL)_5 \wedge (\neg w)_6$$

Resolution applied to clauses 1 and 6

$$\frac{(w \vee tL \vee tR), (\neg w)}{(tL \vee tR)}$$

Produces a new clause ($tL \vee tR$). We can add it to the list as clause (7).

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. $(\neg tR)$
5. $(\neg tL)$
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (\neg tR)_4 \wedge (\neg tL)_5 \wedge (\neg w)_6$$

Resolution applied to clauses 4 and 7

$$\frac{(\neg tR),\ (tL \vee tR)}{(tL)}$$

Produces a new clause (tL). We can add it to the list as clause (8).

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. $(\neg tR)$
5. $(\neg tL)$
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$
8. (tL)

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (\neg tR)_4 \wedge (\neg tL)_5 \wedge (\neg w)_6$$

Resolution applied to clauses 5 and 8

$$\frac{(\neg tL),\ (tL)}{()}$$

Produces an empty clause / contradiction. Stop.

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. $(\neg tR)$
5. $(\neg tL)$
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$
8. (tL)

Proof by Resolution: Example

Prove:

$$KB_N \wedge \neg Q \equiv$$

$$(w \vee tL \vee tR)_1 \wedge (\neg tL \vee \neg w)_2 \wedge (\neg tR \vee \neg w)_3 \wedge (\neg tR)_4 \wedge (\neg tL)_5 \wedge (\neg w)_6$$

At this point, we tried to resolve all promising clause pairs and we reached an empty clause $\rightarrow$ KB entails Q.

Given PERCEPTS: $(\neg tR) \wedge (\neg tL)$

we should apply action walk (w) and go.

Known clauses:

1. $(w \vee tL \vee tR)$
2. $(\neg tL \vee \neg w)$
3. $(\neg tR \vee \neg w)$
4. $(\neg tR)$
5. $(\neg tL)$
6. $(\neg w)$

Added clauses:

7. $(tL \vee tR)$
8. (tL)

Street Crosser Agent: Summary

Applying resolution to all possible $\mathrm{PERCEPTS}$ **and** Q **(only one) combinations and decisions:**

- $\mathrm{PERCEPTS} \equiv (\neg tR) \wedge (\neg tL) \rightarrow$ **WALK**
- $\mathrm{PERCEPTS} \equiv (tR) \wedge (\neg tL) \rightarrow$ **DON'T WALK**
- $\mathrm{PERCEPTS} \equiv (\neg tR) \wedge (tL) \rightarrow$ **DON'T WALK**
- $\mathrm{PERCEPTS} \equiv (tR) \wedge (tL) \rightarrow$ **DON'T WALK**

allowed our agent to:

- **reason and make decisions**
- **learn: percepts $\rightarrow$ decision is new knowledge!**

Knowledge Base: But wait...

If I keep adding multiple new PERCEPTS to the knowledge base KB, for example:

$$\text{PERCEPTS1} \equiv (\neg tR) \wedge (\neg tL)$$

$$\text{PERCEPTS2} \equiv (tR) \wedge (tL)$$

I may end up with a contradiction in my KB, right?

Knowledge-based Agents

function KB-AGENT(*percept*) **returns** an *action*
  persistent: *KB*, a knowledge base
    t, a counter, initially 0, indicating time

KBBEFORE

  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*

CURRENTKB

new percept

CURRENTKB ⇔ KBBEFORE ∧ percept

Knowledge-based Agents

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*))
  $t \leftarrow t + 1$
  return *action*

KBBEFORE

"time stamps"

CURRENTKB

new percept

CURRENTKB ⇔ KBBEFORE ∧ percept