CS 480

Introduction to Artificial Intelligence

February 6, 2024

Announcements / Reminders

- Please follow the Week 04 To Do List instructions (if you haven't already):
- Quiz #03: due on Sunday (02/11/24) at 11:59 PM CST
  - New quiz will be posted on Monday!
- Written Assignment #01 due on Tuesday (02/06/24) at 11:59 PM CST
  - New written assignment will be posted this week!
- Programming Assignment #01 due on Sunday (02/18/24) at 11:59 PM CST

Plan for Today

- **Logical Agents and Reasoning**

Knowledge-based Agents

- Central component: Knowledge Base (KB)
- Knowledge Base is a set of sentences
- All Sentences are expressed in knowledge representation language
- Sentences can be:
  - given (axioms)
  - derived
  - used for inference
- KB can have background knowledge

Propositional Logic: Laws/Theorems

Equivalence	Law / Theorems
$p \lor q \Leftrightarrow q \lor p$ $p \land q \Leftrightarrow q \land p$	Commutative laws
$(p \lor q) \lor r \Leftrightarrow p \lor (q \lor r)$ $(p \land q) \land r \Leftrightarrow p \land (q \land r)$	Associative laws
$p \land (q \lor r) \Leftrightarrow (p \land q) \lor (p \land r)$ $p \lor (q \land r) \Leftrightarrow (p \lor q) \land (p \lor r)$	Distributive laws
$\neg (p \land q) \Leftrightarrow \neg q \lor \neg p$ $\neg (p \lor q) \Leftrightarrow \neg q \land \neg p$	De Morgan's laws
$p \land (p \lor q) \Leftrightarrow p$ $p \lor (p \land q) \Leftrightarrow p$	Absorption laws
$\neg (\neg p) \Leftrightarrow p$	Double Negation law (involution)
$p \land p \Leftrightarrow p$ $p \lor p \Leftrightarrow p$	Idempotent laws
$p \lor \neg p \Leftrightarrow T$	Law of Excluded Middle (Negation law)
$p \land \neg p \Leftrightarrow \bot$	Contradiction (Negation law)
$p \land T \Leftrightarrow p$ $p \lor \bot \Leftrightarrow p$	Identity laws
$p \land \bot \Leftrightarrow \bot$ $p \lor T \Leftrightarrow T$	Domination laws
$\neg p \lor q \Leftrightarrow p \Rightarrow q$	Implication law
$p \Rightarrow q \Leftrightarrow \neg q \Rightarrow \neg p$	Contraposition law
$(p \land q) \lor (\neg q \land \neg p) \Leftrightarrow (p \Leftrightarrow q)$ $(p \Rightarrow q) \land (q \Rightarrow p) \Leftrightarrow (p \Leftrightarrow q)$	Equivalence law

Propositional Logic and KB-Agents

Propositional Logic: Syntax	**Propositional Logic: Semantics**	Propositional Logic: Inference and Proof Systems	KB-Agents: Inference algorithms

Interpretation

The truth value assignment to propositional sentences is called an interpretation (an assertion about their truth in some possible world / model).

Definition: A mapping $I : \Sigma \rightarrow \{\text{true}, \text{false}\}$**, which assigns a truth value to every proposition variable, is called an interpretation.**

Sentence: $(p \vee q) \wedge (\neg q \vee r)$

Interpretation i: $p^i = \text{true}$, $q^i = \text{false}$, $r^i = \text{true}$

Evaluation

Evaluation is the process of determining the truth values of compound/complex sentences given a truth assignment for the truth values of proposition constants/atomic sentences. Consider the following truth assignment i**:**

$$p^i = \text{true},\ q^i = \text{false},\ r^i = \text{true}$$ Assignment

Let's evaluate the following complex sentence $(p \lor q) \land (\neg q \lor r)$**:**

$$(p \lor q) \land (\neg q \lor r) \rightarrow (\text{true} \lor \text{false}) \land (\neg \text{false} \lor \text{true})$$ Subsitute

$$\mathbf{(true \lor false)} \land (\neg \text{false} \lor \text{true})$$ Disjunction

$$\text{true} \land (\neg \mathbf{false} \lor \text{true})$$ Negation

$$\text{true} \land \mathbf{(true \lor true)}$$ Disjunction

$$\mathbf{true} \land \mathbf{true}$$ Conjunction

$$\text{true}$$ Interpretation

Sentence Classes

SATISFIABLE

A sentence is satisfiable if it is true for AT LEAST ONE interpretation.

In plain English:
"You can find AT LEAST one assignment of logical values of true and false to individual propositional variables that will make this sentence true."

Example:

$$p \Rightarrow q$$

p	q	$p \Rightarrow q$
true	true	**true**
true	false	false
false	true	**true**
false	false	**true**

(LOGICALLY) VALID/TAUTOLOGY

A sentence is (logically) valid if it is true for ALL interpretations. Also called a tautology.

In plain English:
"This sentence is ALWAYS true regardless of value assignment to individual propositional variables."

Example:

$$p \vee \neg p$$

p	$\neg p$	$p \wedge \neg p$
true	false	**true**
true	false	**true**
false	true	**true**
false	true	**true**

UNSATISFIABLE/CONTRADICTION

A sentence is unsatisfiable if it is NOT true for ANY interpretation. Also called a contradiction.

In plain English:
"This sentence is ALWAYS false regardless of value assignment to individual propositional variables."

Example:

$$p \wedge \neg p$$

p	$\neg p$	$p \wedge \neg p$
true	false	**false**
true	false	**false**
false	true	**false**
false	true	**false**

Complex Sentence: Truth Table

Consider a complex sentence R built with N propositional variables $p_1, p_2, p_3, \ldots, p_{N-1}, p_N$ and logical connectives ($\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$). Each truth assignment is a different possible world.

2^N Possible Worlds (Models)	N Propositional Variables						Complex sentence R	2^N Interpretations of R
	p_1	p_2	p_3	…	p_{N-1}	p_N		
	true	true	true	…	true	true	false	
	true	true	true	…	true	false	true	
	true	true	false	…	false	true	false	
	…	…	…	…	…	…	…	
	false	false	true	…	true	false	true	
	false	false	true	…	false	true	true	
	false	false	false	…	false	false	false	

Sentence: Syntactic / Semantic Levels

Each propositional logic "exists" on two levels:

- **Syntactic: where a sentence is just a legal arrangement of language symbols. Sentence**

$$(p \vee q) \wedge (\neg q \vee r)$$

WITHOUT interpretation HAS NO MEANING

  - **we can manipulate symbols, but we CANNOT reason**

- **Semantic: where a sentence has a truth value is assigned to it (interpretation):**

$(p \vee q) \wedge (\neg q \vee r)$ **where** $p^i = \text{true}, q^i = \text{false}, r^i = \text{true}$

HAS MEANING (through interpretation) $\rightarrow$ **it is** true

Sentence: Syntactic / Semantic Levels

Each propositional logic "exists" on two levels:

- **Syntactic: where a sentence is just a legal arrangement of language symbols. Sentence**

$$(\text{cool} \vee \text{funny}) \Rightarrow \text{popular}$$

WITHOUT interpretation HAS NO MEANING

  - **we can't tell if a given person is** popular **here**

- **Semantic: where a sentence has a truth value is assigned to it (interpretation):**

$(\text{cool} \vee \text{funny}) \Rightarrow \text{popular}$ **where** $\text{cool} = \text{true}$, $\text{funny} = \text{false}$

HAS MEANING $\rightarrow$ **we can deduce that a person is** popular

Sentence Semantical Equivalence

Two propositional logic sentences F and G are called <u>semantically</u> equivalent if they take on the same interpretation for all truth value assignments. If that is the case $F \equiv G$.

Example: sentence $\neg a \lor b$ is equivalent to sentence $a \Rightarrow b$. Proof with a truth table:

a	b	$\neg a$	$\neg a \lor b$	$\Leftrightarrow$	$a \Rightarrow b$
true	true	false	**true**		**true**
true	false	false	**false**	$\equiv$	**false**
false	true	true	**true**		**true**
false	false	true	**true**		**true**

Sentence Classes

SATISFIABLE

A sentence is satisfiable if it is true for AT LEAST ONE interpretation.

In plain English:
"You can find AT LEAST one assignment of logical values of true and false to individual propositional variables that will make this sentence true."

Example:

$p \Rightarrow q$

p	q	$p \Rightarrow q$
true	true	**true**
true	false	false
false	true	**true**
false	false	**true**

(LOGICALLY) VALID/TAUTOLOGY

A sentence is (logically) valid if it is true for ALL interpretations.
Also called a tautology.

In plain English:
"This sentence is ALWAYS true regardless of value assignment to individual propositional variables."

Example:

$p \vee \neg p$

p	$\neg p$	$p \wedge \neg p$
true	false	**true**
true	false	**true**
false	true	**true**
false	true	**true**

UNSATISFIABLE/CONTRADICTION

A sentence is unsatisfiable if it is NOT true for ANY interpretation.
Also called a contradiction.

In plain English:
"This sentence is ALWAYS false regardless of value assignment to individual propositional variables."

Example:

$p \wedge \neg p$

p	$\neg p$	$p \wedge \neg p$
true	false	**false**
true	false	**false**
false	true	**false**
false	true	**false**

Propositional Logic and KB-Agents

Propositional Logic: Syntax	Propositional Logic: Semantics	**Propositional Logic: Inference and Proof Systems**	KB-Agents: Inference algorithms

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)**

Inference: Real-life Example

If it is raining, I will need an umbrella. It is raining. Therefore, I will need an umbrella.

Inference: Real-life Example

If it is raining, then I will need an umbrella. It is raining. Therefore, I will need an umbrella.

Inference: Real-life Example

If it is raining, then I will need an umbrella. It is raining. Therefore, I will need an umbrella.

Propositional Logic: An Argument

An argument A in propositional logic has the following form:

$$
\begin{array}{ll}
A: & P1 \\
 & P2 \\
 & \ldots \\
 & PN \\
\hline
 & \therefore\ C
\end{array}
$$

PREMISES

CONCLUSION

An argument A is said to be valid if the implication formed by taking the conjunction of the premiseses (antecedent) and the conclusion C (consequent),

$(P1 \wedge P2 \wedge P3 \wedge \ldots \wedge PN) \Rightarrow C$ **is a tautology.**

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).

Inference: Real-life Example

If it is raining, then I will need an umbrella.

It is raining.

Therefore, I will need an umbrella.

Inference: Real-life Example

If it is raining, then I will need an umbrella.

It is raining. PREMISES

Therefore, I will need an umbrella. CONCLUSION

Inference: Real-life Example

If ______, then _____________.

___________.

Therefore, _____________.

PREMISES

CONCLUSION

Inference: Real-life Example

If __p__, then ______q______. ← PREMISES

_____p_____.

Therefore, ______q_____. ← CONCLUSION

p = "It is raining."

q = "I will need an umbrella."

PREMISE1 = "If it is raining, then I will need an umbrella."

PREMISE2 = "It is raining."

CONCLUSION = "I will need an umbrella."

Inference: Real-life Example

If p then q is true

and p is true.

Therefore, q is true.

PREMISES

CONCLUSION

p = “It is raining.”

q = “I will need an umbrella.”

PREMISE1 = “If it is raining, then I will need an umbrella.”

PREMISE2 = “It is raining.”

CONCLUSION = “I will need an umbrella.”

Inference Rules: Modus Ponens

$p \Rightarrow q$

p

PREMISES

$\therefore q$

CONCLUSION

p = "It is raining."

q = "I will need an umbrella."

PREMISE1 = $p \Rightarrow q$

PREMISE2 = p

CONCLUSION = q

Inference Rules: Modus Ponens

$p \Rightarrow q$ ← PREMISES

p

$\therefore q$ ← CONCLUSION

p = "It is raining."

q = "I will need an umbrella."

PREMISE1 = $p \Rightarrow q$

PREMISE2 = p

CONCLUSION = q

IF PREMISES ARE TRUE, THEREFORE THE CONCLUSION MUST ALSO BE TRUE

Inference: Modus Ponens

$p \Rightarrow q$ ← PREMISES

p

$\therefore q$ ← CONCLUSION

p = "It is raining."

q = "I will need an umbrella."

PREMISE1 = $p \Rightarrow q$

PREMISE2 = p

CONCLUSION = q

PROPOSITIONAL VARIABLES		IMPLICATION
p	q	$p \Rightarrow q$
true	true	true
true	false	false
false	true	true
false	false	true

Inference Rules: Modus Ponens

$p \Rightarrow q$

p

PREMISES

$\therefore q$

CONCLUSION

p = "It is raining."

q = "I will need an umbrella."

PREMISES = PREMISE1 AND PREMISE2 = $(p \Rightarrow q) \wedge p$

CONCLUSION = q

Inference Rules: Modus Ponens

$p \Rightarrow q$

p

PREMISES

$\therefore q$

CONCLUSION

PROPOSITIONAL VARIABLES		INDIVIDUAL PREMISE		PREMISES	CONCLUSION
p	q	P1: $p \Rightarrow q$	P2: p	P1 $\wedge$ P2: $(p \Rightarrow q) \wedge p$	q
true	true	true	true	true	true
true	false	false	true	false	false
false	true	true	false	false	true
false	false	true	false	false	false

Inference Rules: Modus Ponens

$p \Rightarrow q$

p

PREMISES

$\therefore q$

CONCLUSION

PROPOSITIONAL VARIABLES		INDIVIDUAL PREMISE		PREMISES	CONCLUSION
p	q	P1: $p \Rightarrow q$	P2: p	P1 $\wedge$ P2:$(p \Rightarrow q) \wedge p$	q
true	true	true	true	true	true
true	false	false	true	false	false
false	true	true	false	false	true
false	false	true	false	false	false

Inference Rules: Modus Ponens

$p \Rightarrow q$

p

PREMISES

$\therefore q$

CONCLUSION

IF PREMISES ARE TRUE, THEREFORE THE CONCLUSION MUST ALSO BE TRUE

INDIVIDUAL PREMISE		PREMISES	CONCLUSION
P1: $p \Rightarrow q$	P2: p	P1 $\wedge$ P2:$(p \Rightarrow q) \wedge p$	q
true	true	true	true
false	true	false	false
true	false	false	true
true	false	false	false

Inference Rules: Summary

Rules of Inference:

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

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

Tautological forms:

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

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

Argument Validity: Truth Table Proof

$p \Rightarrow q$

$q \Rightarrow \neg r$

$\neg p \Rightarrow \neg r$

$\therefore \neg r$

p	q	r	P1: $p \Rightarrow q$	P2: $q \Rightarrow \neg r$	P3: $\neg p \Rightarrow \neg r$	$P1 \wedge P2 \wedge P3$	$(P1 \wedge P2 \wedge P3) \Rightarrow \neg r$
true	true	true	true	false	true	false	**true**
true	true	false	true	true	true	true	**true**
true	false	true	false	true	true	false	**true**
true	false	false	false	true	true	false	**true**
false	true	true	true	false	false	false	**true**
false	true	false	true	true	true	true	**true**
false	false	true	true	true	false	false	**true**
false	false	false	true	true	true	true	**true**

Argument Validity: Truth Table Proof

$p \Rightarrow q$

$q \Rightarrow \neg r$

$\neg p \Rightarrow \neg r$

$\therefore \neg r$

$$A \Leftrightarrow ((p \Rightarrow q) \wedge (q \Rightarrow \neg r) \wedge (\neg p \Rightarrow \neg r) \Rightarrow \neg r)$$

An argument A is valid if it is a tautology.

p	q	r	P1:$p \Rightarrow q$	P2:$q \Rightarrow \neg r$	P3:$\neg p \Rightarrow \neg r$	$P1 \wedge P2 \wedge P3$	$(P1 \wedge P2 \wedge P3) \Rightarrow \neg r$
true	true	true	true	false	true	false	**true**
true	true	false	true	true	true	true	**true**
true	false	true	false	true	true	false	**true**
true	false	false	false	true	true	false	**true**
false	true	true	true	false	false	false	**true**
false	true	false	true	true	true	true	**true**
false	false	true	true	true	false	false	**true**
false	false	false	true	true	true	true	**true**

Argument Validity: Truth Table Proof

$p \Rightarrow q$

$q \Rightarrow \neg r$

$\neg p \Rightarrow \neg r$

$\therefore \neg r$

$A \Leftrightarrow ((P1) \wedge (P2) \wedge (P3) \Rightarrow \neg r)$

An argument A is valid if it is a tautology.

Argument A is valid, because it is a tautology

(always true regardless of p, q, r truth assignments)

p	q	r	P1:p⇒q	P2:q⇒¬r	P3:¬ p ⇒ ¬ r	P1∧P2∧P3	A
true	true	true	true	false	true	false	**true**
true	true	false	true	true	true	true	**true**
true	false	true	false	true	true	false	**true**
true	false	false	false	true	true	false	**true**
false	true	true	true	false	false	false	**true**
false	true	false	true	true	true	true	**true**
false	false	true	true	true	false	false	**true**
false	false	false	true	true	true	true	**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.

$$\text{PREMISES} \models \text{CONCLUSION}$$

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:

$$\mathrm{KB} \vDash \mathrm{Q}$$

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:

$$KB \models Q$$

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

Model / “Possible World”

A model (a “possible world) is a single truth assignment / interpretation.

If a sentence U **is** true **in model** K**,** K **satisfies** U.

$M(U)$**: set of ALL models of** U **(that satisfy** U**)**

Now:

$$KB \models Q \text{ if and only if } M(KB) \subseteq M(Q)$$

$KB \models Q$ **is** true **if and only if in EVERY model in which** KB **is** true**,** Q **is also** true**.**

Logical Entailment with Truth Table

$p \Rightarrow q$ KB

$p \Rightarrow \neg r$

$\neg p \Rightarrow \neg r$

$\therefore \neg r$ Q

$KB \Leftrightarrow (p \Rightarrow q) \wedge (p \Rightarrow \neg r) \wedge (\neg p \Rightarrow \neg r)$

$Q \Leftrightarrow \neg r$

Model	p	q	r	P1:$p \Rightarrow q$	P2:$q \Rightarrow \neg r$	P3:$\neg p \Rightarrow \neg r$	KB	Q
M1	true	true	true	true	false	true	false	false
M2	true	true	false	true	true	true	true	true
M3	true	false	true	false	true	true	false	false
M4	true	false	false	false	true	true	false	true
M5	false	true	true	true	false	false	false	false
M6	false	true	false	true	true	true	true	true
M7	false	false	true	true	true	false	false	false
M8	false	false	false	true	true	true	true	true

Entailment: Model Checking

KB

$p \Rightarrow q$

$p \Rightarrow \neg r$

$\neg p \Rightarrow \neg r$

$\therefore \neg r$ Q

$KB \Leftrightarrow (p \Rightarrow q) \wedge (p \Rightarrow \neg r) \wedge (\neg p \Rightarrow \neg r) \mid Q \Leftrightarrow \neg r$

Models where KB **is** true**:** M(KB) = {M2, M6, M8}

Models where Q **is** true**:** M(Q) = {M2, M4, M6, M8}

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

Entailment: Model Checking

KB

$p \Rightarrow q$

$p \Rightarrow \neg r$

$\neg p \Rightarrow \neg r$

$\therefore \boxed{\neg r}$ Q

$KB \Leftrightarrow (p \Rightarrow q) \wedge (p \Rightarrow \neg r) \wedge (\neg p \Rightarrow \neg r) \mid Q \Leftrightarrow \neg r$

Models where KB **is** true: M(KB) = {M2, M6, M8}

Models where Q **is** true: M(Q) = {M2, M4, M6, M8}

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

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

$KB \Rightarrow Q$ is a Tautology Proof

$p \Rightarrow q$

$p \Rightarrow \neg r$

$\neg p \Rightarrow \neg r$

$\therefore \neg r$

$KB \Rightarrow Q$ **is** true **for ALL models / interpretations**

$KB \Rightarrow Q$ **is a tautology**

p	q	r	P1:$p \Rightarrow q$	P2:$q \Rightarrow \neg r$	P3:$\neg p \Rightarrow \neg r$	KB	$KB \Rightarrow Q$
true	true	true	true	false	true	false	**true**
true	true	false	true	true	true	true	**true**
true	false	true	false	true	true	false	**true**
true	false	false	false	true	true	false	**true**
false	true	true	true	false	false	false	**true**
false	true	false	true	true	true	true	**true**
false	false	true	true	true	false	false	**true**
false	false	false	true	true	true	true	**true**

Enumeration: Issues

Consider a complex sentence R built with N propositional variables $p_1, p_2, p_3, \ldots, p_{N-1}, p_N$ and logical connectives ($\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$). Each truth assignment is a different possible world.

2^N **Possible Worlds (Models)**

N Propositional Variables						Complex sentence R
p_1	p_2	p_3	…	p_{N-1}	p_N	
true	true	true	…	true	true	false
true	true	true	…	true	false	true
true	true	false	…	false	true	false
…	…	…	…	…	…	…
false	false	true	…	true	false	true
false	false	true	…	false	true	true
false	false	false	…	false	false	false

2^N **Interpretations of R**

Logical Entailment

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

$$KB \models Q$$

One more way to look at it:

If KB entails Q,

- **"the truth of KB guarantees truth of Q"**
- **"the falsity of KB guarantees falsity of Q"**

Entailment: Deriving Conclusions

You can prove that:

$$KB \models Q$$

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

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

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

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

Prove that $(KB \wedge \neg Q) \Leftrightarrow \perp$
(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	$p \Rightarrow q$	$q \Rightarrow \neg r$	$\neg p \Rightarrow \neg r$	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 $(KB \wedge \neg Q) \Leftrightarrow \perp$
(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 $(KB \wedge \neg Q) \Leftrightarrow \perp$
(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

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

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

Model Checking as a Search Problem

Model checking can be considered a search problem. Searching a truth table for models in which KB entails Q (Q follows from KB). It is a $O(2^N)$ problem.

N Propositional Variables						
p_1	p_2	p_3	…	p_{N-1}	p_N	KB $\models$ Q
true	true	true	…	true	true	false
true	true	true	…	true	false	true
true	true	false	…	false	true	false
…	…	…	…	…	…	…
false	false	true	…	true	false	true
false	false	true	…	false	true	true
false	false	false	…	false	false	false

2^N Possible Worlds (Models)

2^N Interpretations

Can we do better?
Can we automate the process?

Java Programmers! Difference?

& vs. && operator?

as in:

```
if (a & b) vs if(a && b)
```

Java Programmers! Difference?

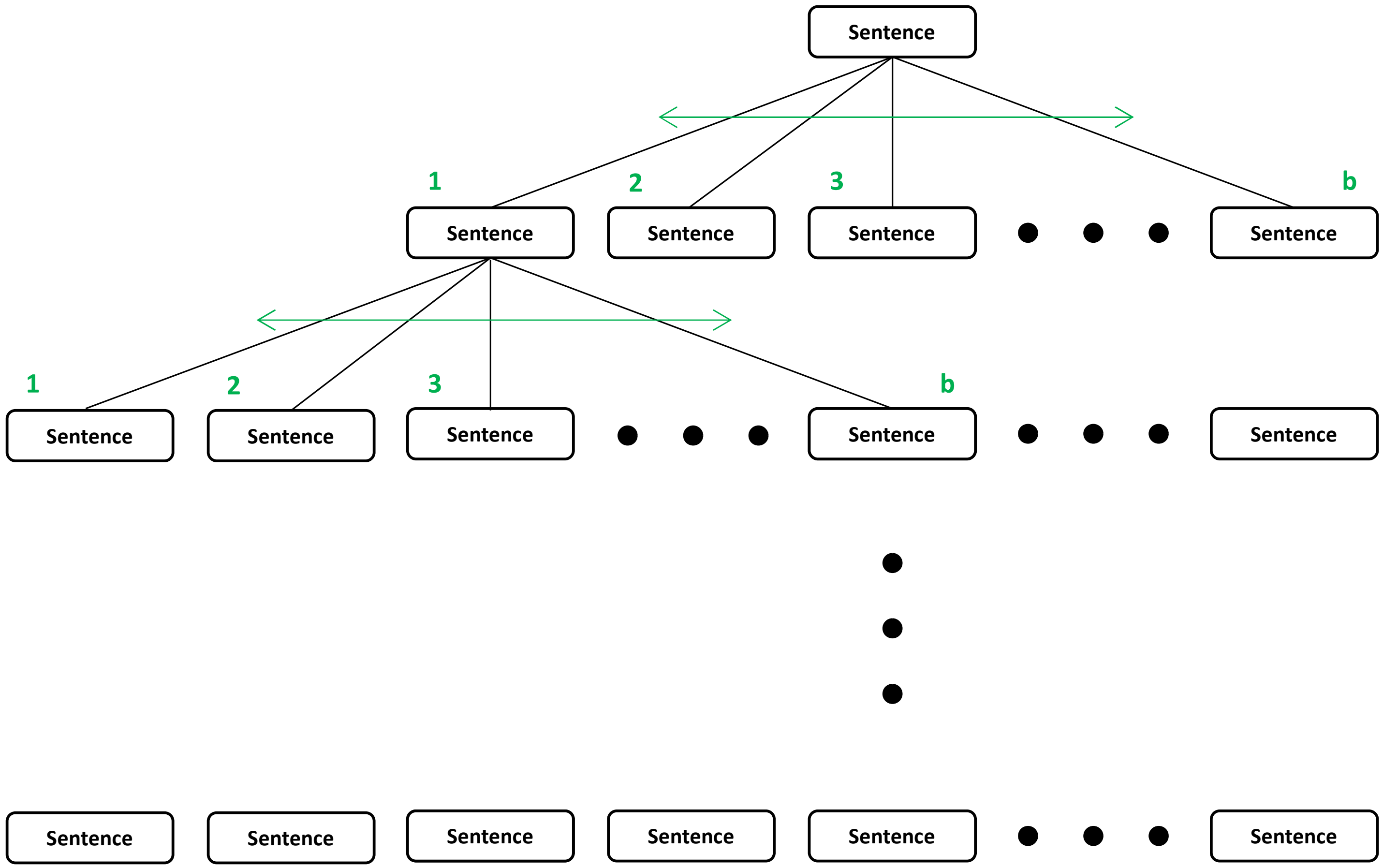
& vs. && operator?

What if I used?

$$KB \equiv PREMISE1 \;\&\&\; PREMISE2 \;\&\&\; \ldots \;\&\&\; PREMISEN$$

What's the benefit?

Truth Table Enumeration as Search



Depth: 0
No assignment

Depth: 1
p_1: value assigned
partial assignment

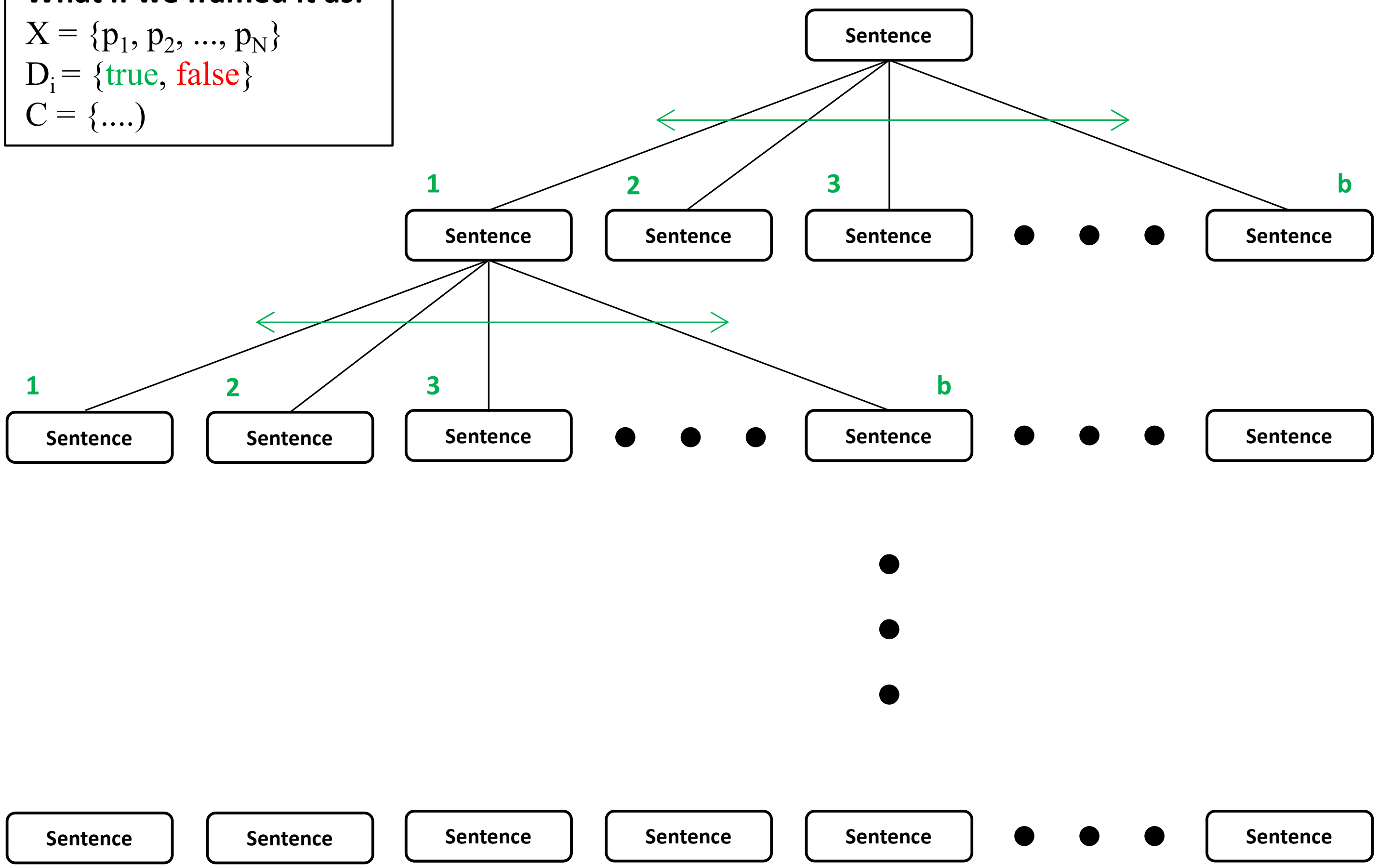
Depth: 2
p_2: value assigned
partial assignment

Depth: N
p_N: value assigned
complete assignment

Some truth assignments will quickly become false.
Not all propositional variables p_i need their values assigned to know that

Truth Table Enumeration as Search

What if we framed it as:
$X = \{p_1, p_2, \ldots, p_N\}$
$D_i = \{$true, false$\}$
$C = \{\ldots.)$



Depth: 0
No assignment

Depth: 1
p_1: **value assigned**
partial assignment

Depth: 2
p_2: **value assigned**
partial assignment

•
•
•

Depth: N
p_N: **value assigned**
complete assignment

Some truth assignments will quickly become false.
Not all propositional variables p_i need their values assigned to know that

Truth Table Enumeration: Pseudocode

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

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

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

Returns true if EITHER "top" OR "bottom" recursive call returns true.

Evaluation

Evaluation is the process of determining the truth values of compound/complex sentences given a truth assignment for the truth values of proposition constants/atomic sentences. Consider the following truth assignment i:

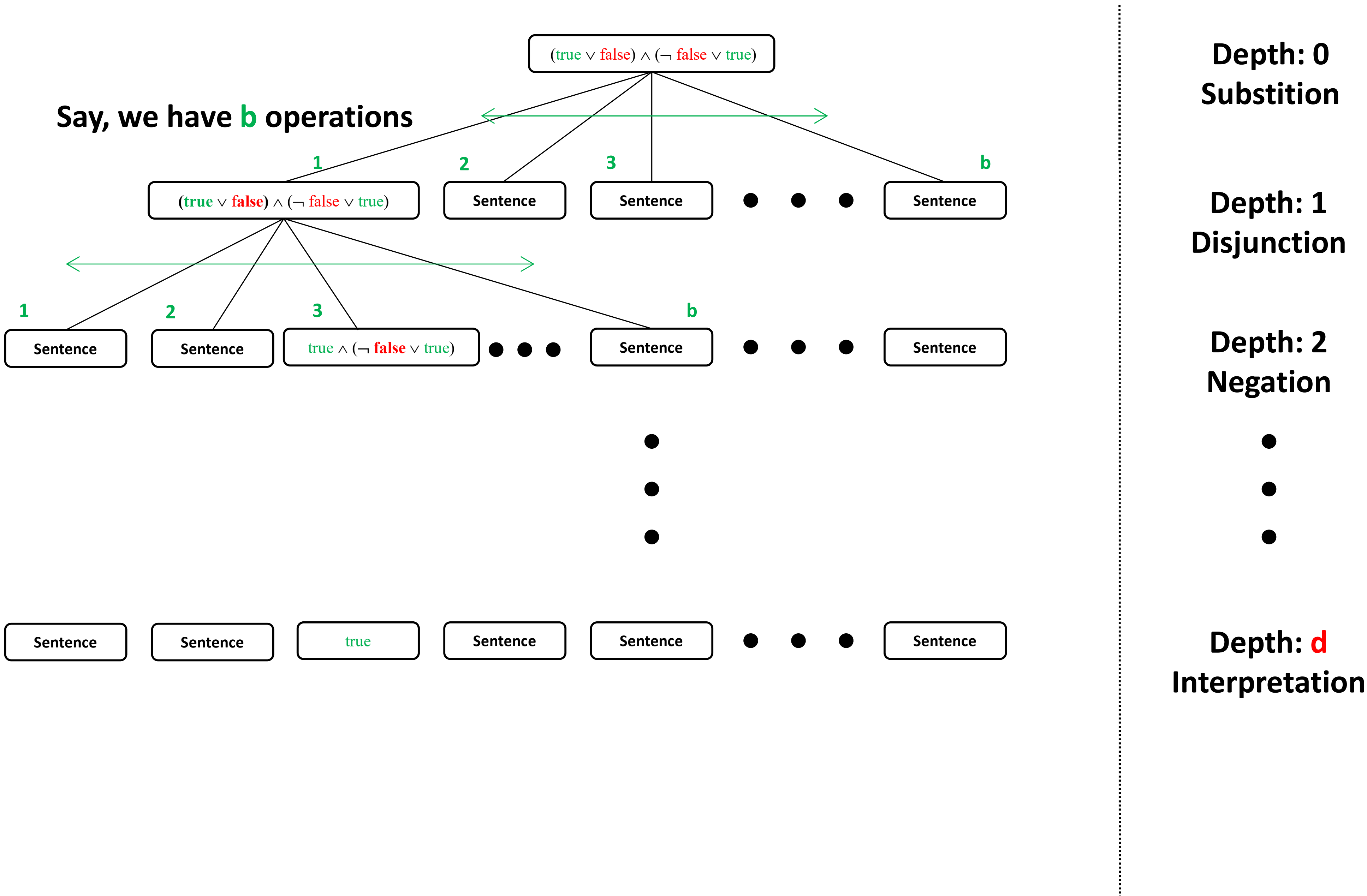
$$p^i = \text{true},\ q^i = \text{false},\ r^i = \text{true}$$ Assignment

Let's evaluate the following complex sentence $(p \vee q) \wedge (\neg q \vee r)$:

$(p \vee q) \wedge (\neg q \vee r) \rightarrow (\text{true} \vee \text{false}) \wedge (\neg \text{false} \vee \text{true})$	Subsitute
$(\textbf{true} \vee \textbf{false}) \wedge (\neg \text{false} \vee \text{true})$	Disjunction
$\text{true} \wedge (\neg \textbf{false} \vee \text{true})$	Negation
$\text{true} \wedge (\textbf{true} \vee \textbf{true})$	Disjunction
$\textbf{true} \wedge \textbf{true}$	Conjunction
true	Interpretation

There is a path of operations that leads from substition to the final interpretation.

Sentence Evaluation as Searching



Deduction / Proof

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

Prove that $((\neg m \lor n) \land \neg n) \Rightarrow \neg m$ **is a tautology:**

$((\neg m \land \neg n) \lor (n \land \neg n)) \Rightarrow \neg m$	by Distributive law $p \land (q \lor r) \Leftrightarrow (p \land q) \lor (p \land r)$
$((\neg m \land \neg n) \lor \bot) \Rightarrow \neg m$	by Negation law (contradiction) $p \land \neg p \Leftrightarrow \bot$
$(\neg m \land \neg n) \Rightarrow \neg m$	by Identity law $p \lor \bot \Leftrightarrow p$
$\neg(\neg m \land \neg n) \lor \neg m$	by Implication law $\neg p \lor q \Leftrightarrow p \Rightarrow q$
$(\neg\neg m \lor \neg\neg n) \lor \neg m$	by De Morgan's law $\neg (p \land q) \Leftrightarrow \neg q \lor \neg p$
$(m \lor n) \lor \neg m$	by Double Negation law $\neg (\neg p) \Leftrightarrow p$
$m \lor (n \lor \neg m)$	by Associative law $(p \lor q) \lor r \Leftrightarrow p \lor (q \lor r)$
$m \lor (\neg m \lor n)$	by Commutative law $p \lor q \Leftrightarrow q \lor p$
$(m \lor \neg m) \lor n$	by Associative law $(p \lor q) \lor r \Leftrightarrow p \lor (q \lor r)$
$\mathrm{T} \lor n$	by Law of Excluded Middle $p \lor \neg p \Leftrightarrow \mathrm{T}$
$n \lor \mathrm{T}$	by Commutative law $p \lor q \Leftrightarrow q \lor p$
T	by Domination Law $p \lor \mathrm{T} \Leftrightarrow \mathrm{T}$

There is a path of operations to get from the beginning to the end

Deduction / Proof as Search

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 $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$
$((\neg m \wedge \neg n) \vee \perp) \Rightarrow \neg m$	by Negation law (contradiction) $p \wedge \neg p \Leftrightarrow \perp$
$(\neg m \wedge \neg n) \Rightarrow \neg m$	by Identity law $p \vee \perp \Leftrightarrow p$
$\neg(\neg m \wedge \neg n) \vee \neg m$	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 $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
$\mathrm{T} \vee n$	by Law of Excluded Middle $p \vee \neg p \Leftrightarrow \mathrm{T}$
$n \vee \mathrm{T}$	by Commutative law $p \vee q \Leftrightarrow q \vee p$
T	by Domination Law $p \vee \mathrm{T} \Leftrightarrow \mathrm{T}$

How were the laws chosen for each step?

There is a path of operations to get from the beginning to the end

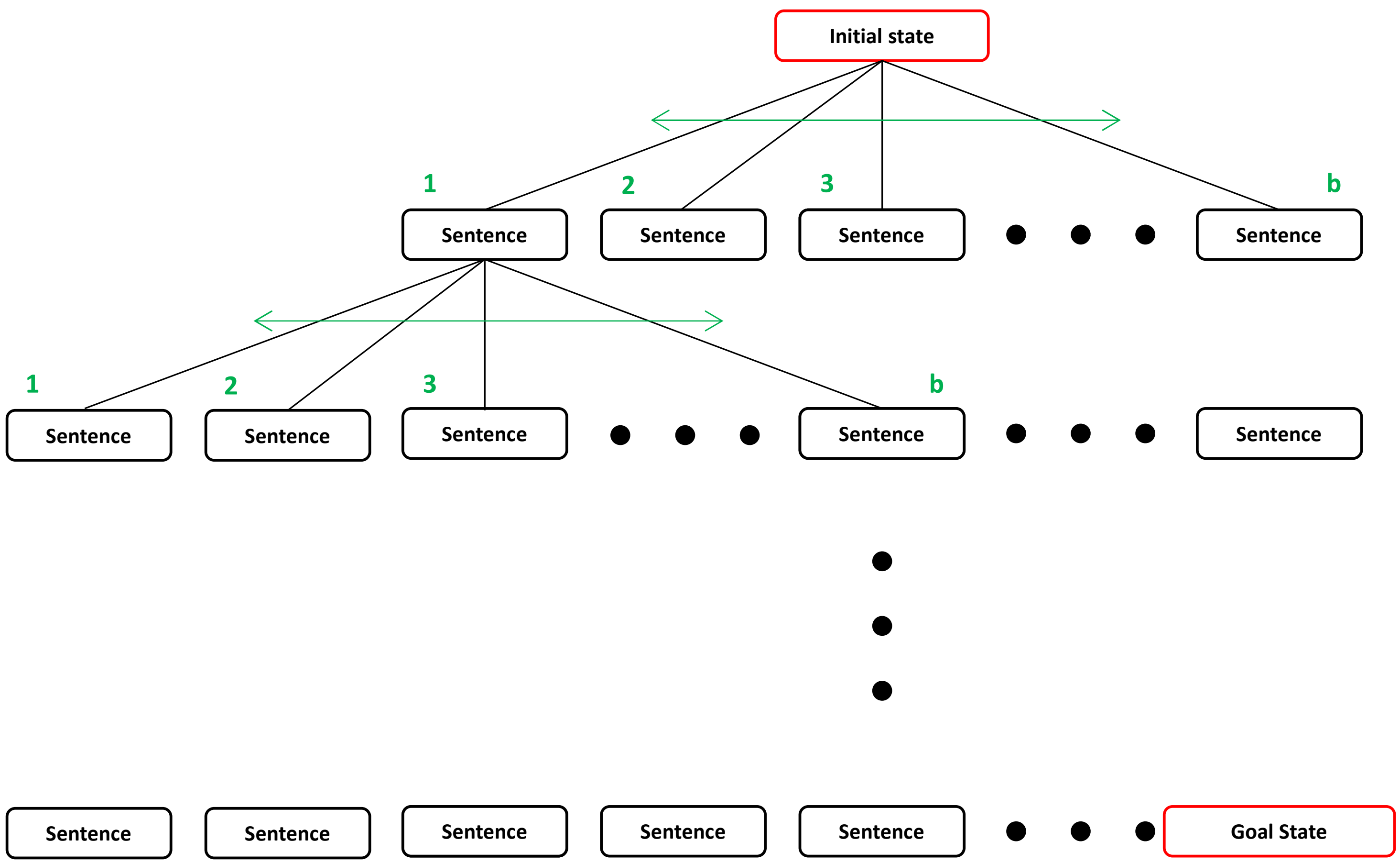
Proof as Search

Search algorithms can be used to find a sequence f steps that consitute a proof.

Just define the proof problem as a search problem:

- **INITIAL STATE: initial knowledge base (sentence)**
- **ACTIONS: the set of all language rules**
- **RESULT: resulting sentence after applying a rule**
- **GOAL: a sentence that we are trying to prove**

Deduction / Proof as Search



Depth: 0
Pick a rule/law

Depth: 1
Pick a rule/law

Depth: 2
Pick a rule/law

•
•
•

Depth: N
Pick a rule/law

Deduction / Proof

Laws/theorems in propositional logic can be used to prove additional theorems througha 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 $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
$\mathrm{T} \vee n$	by Law of Excluded Middle $p \vee \neg p \Leftrightarrow \mathrm{T}$
$n \vee \mathrm{T}$	by Commutative law $p \vee q \Leftrightarrow q \vee p$
T	by Domination Law $p \vee \mathrm{T} \Leftrightarrow \mathrm{T}$

Initial state

Goal state

There is a path of operations to get from the beginning to the end

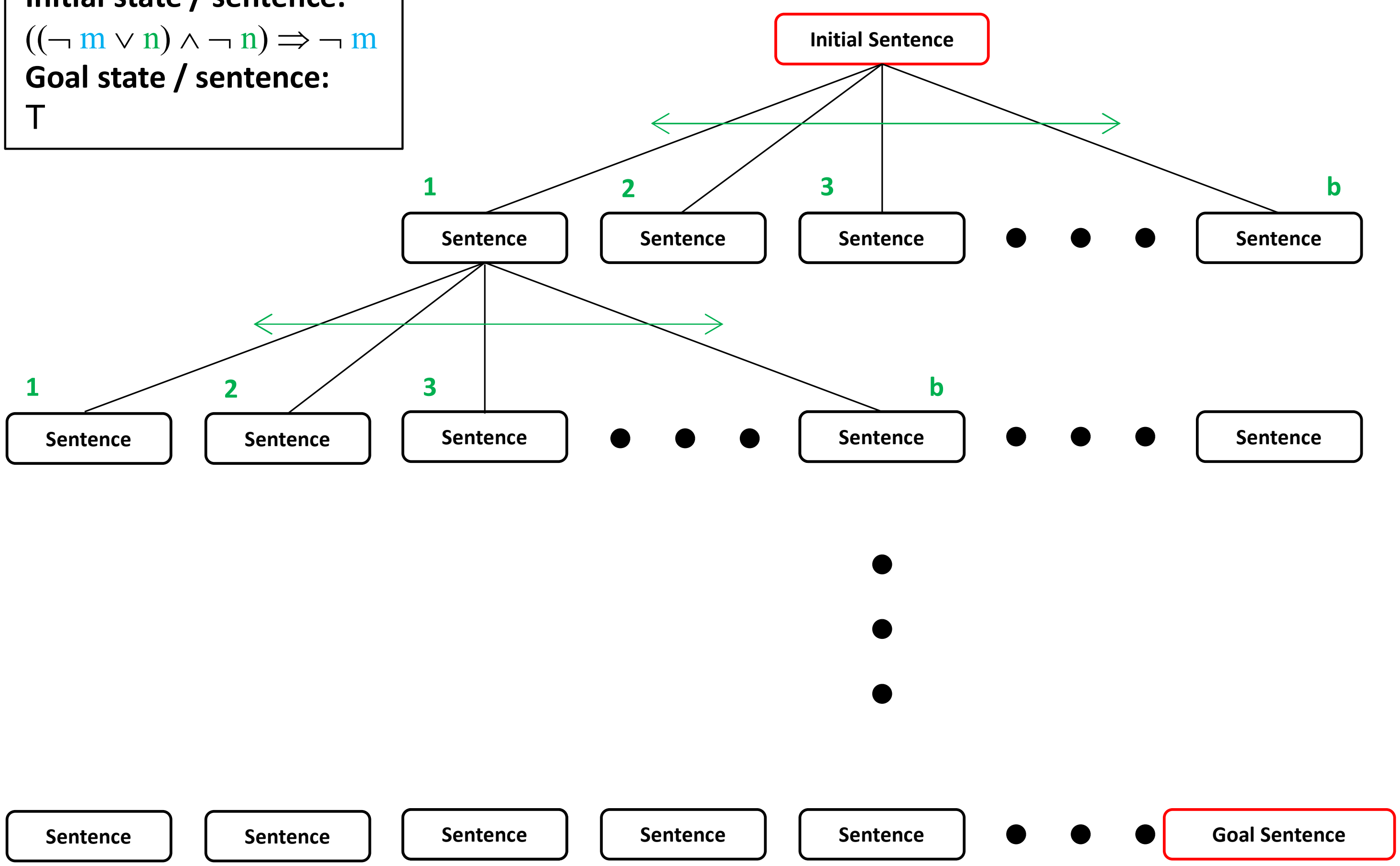
Deduction as Search

Initial state / sentence:

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

Goal state / sentence:

T



Depth: 0
Pick a rule/law

Depth: 1
Pick a rule/law

Depth: 2
Pick a rule/law

•
•
•

Depth: N
Pick a rule/law

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

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

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 <u>can be decided at the syntax level</u> 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 $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
$\mathrm{T} \vee n$	by Law of Excluded Middle $p \vee \neg p \Leftrightarrow \mathrm{T}$
$n \vee \mathrm{T}$	by Commutative law $p \vee q \Leftrightarrow q \vee p$
T	by Domination Law $p \vee \mathrm{T} \Leftrightarrow \mathrm{T}$

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.

Proving Entailment: Two Levels

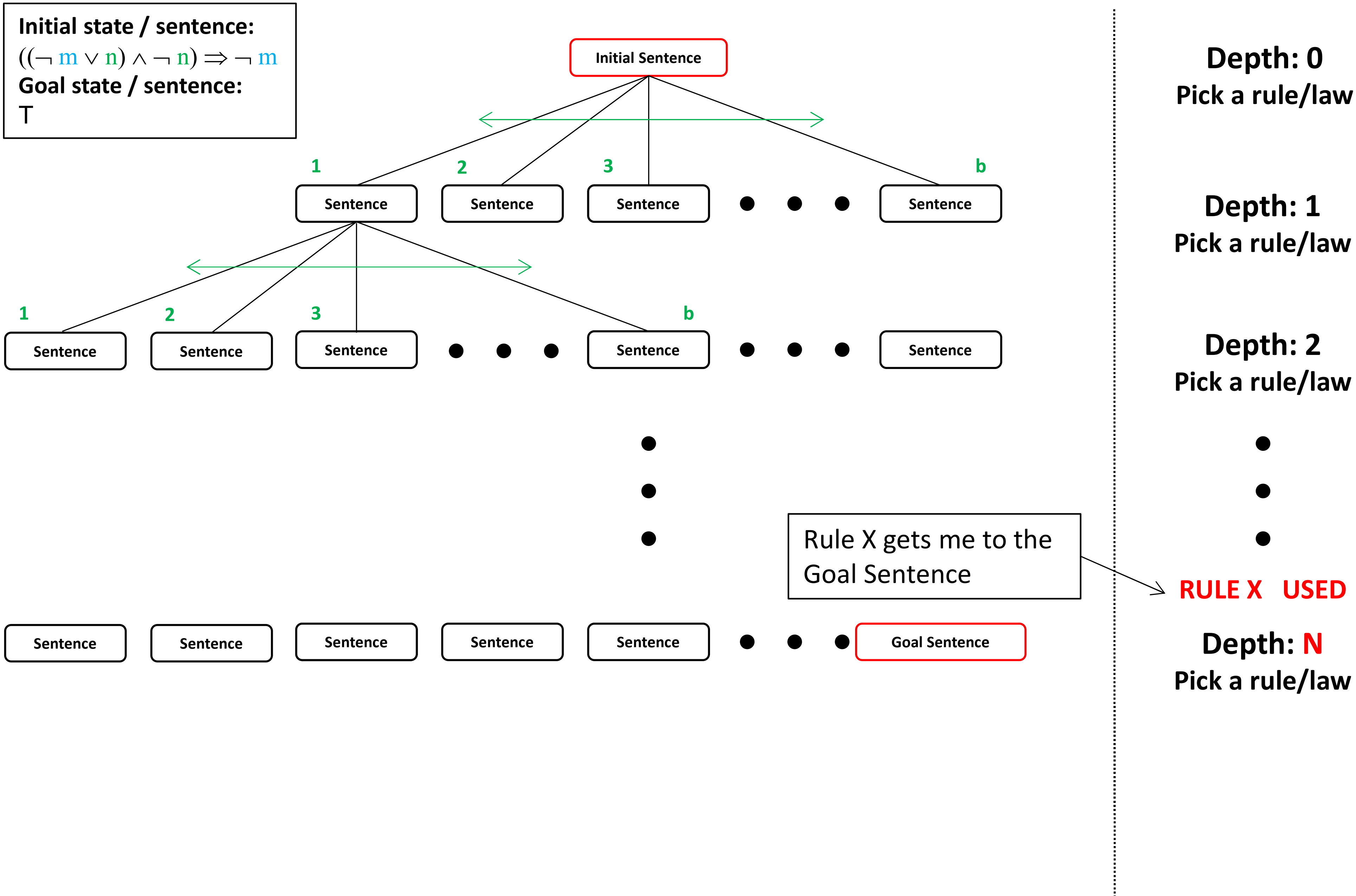
Syntax level

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

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

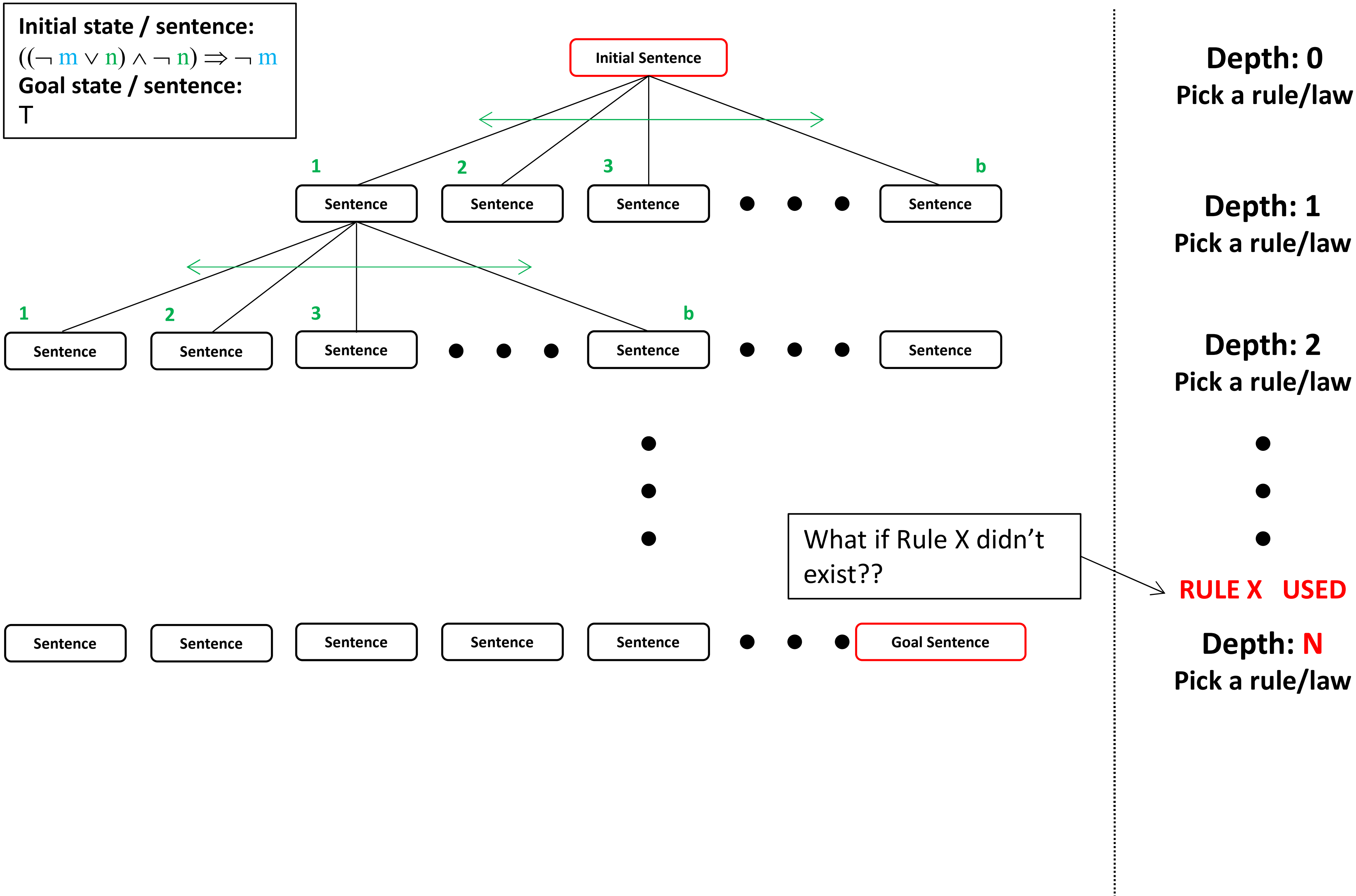
Deduction as Search

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



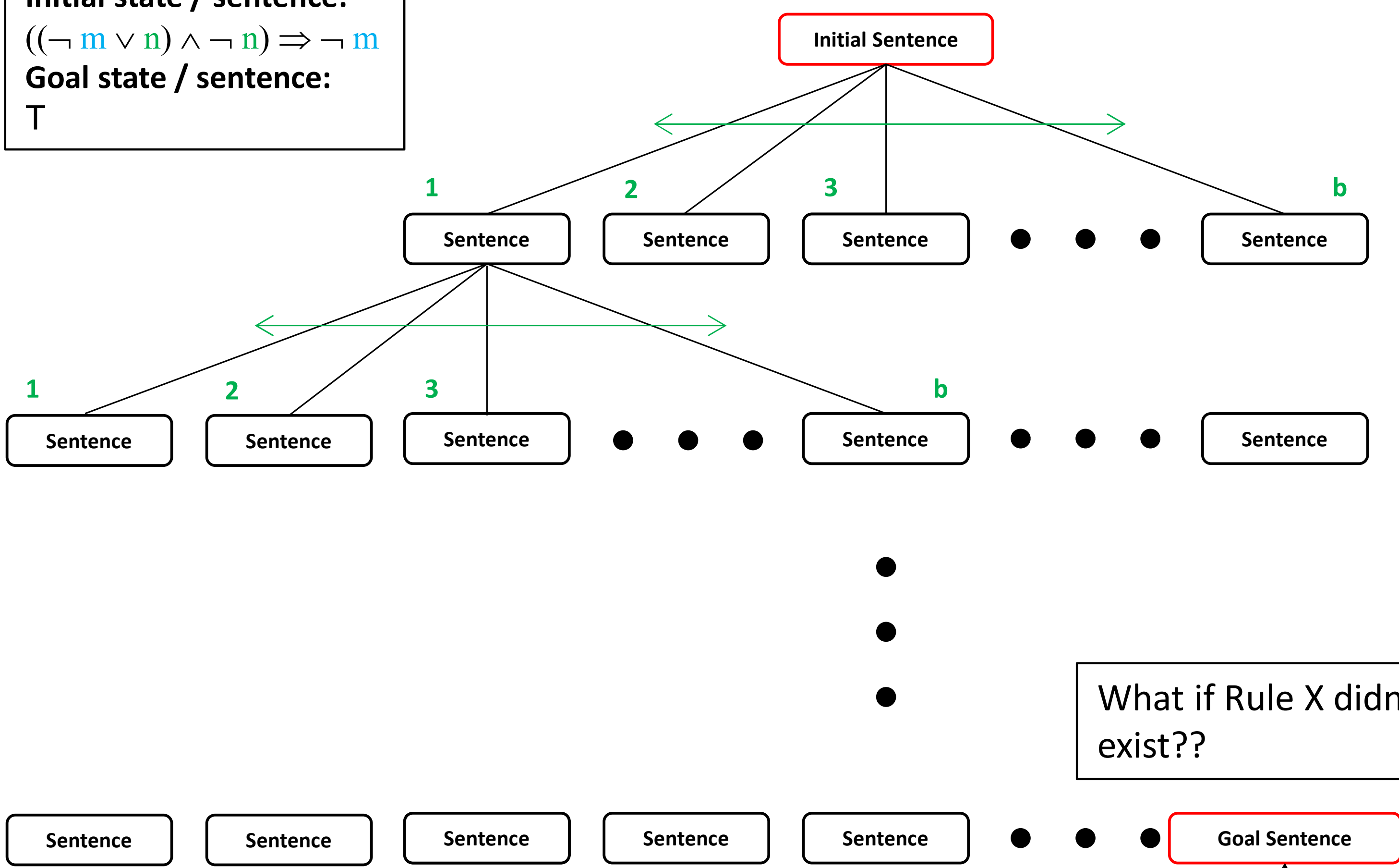
Deduction as Search

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



Deduction as Search

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



What if Rule X didn't exist??

Then this state is possible BUT UNREACHABLE!!

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

Completness:

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

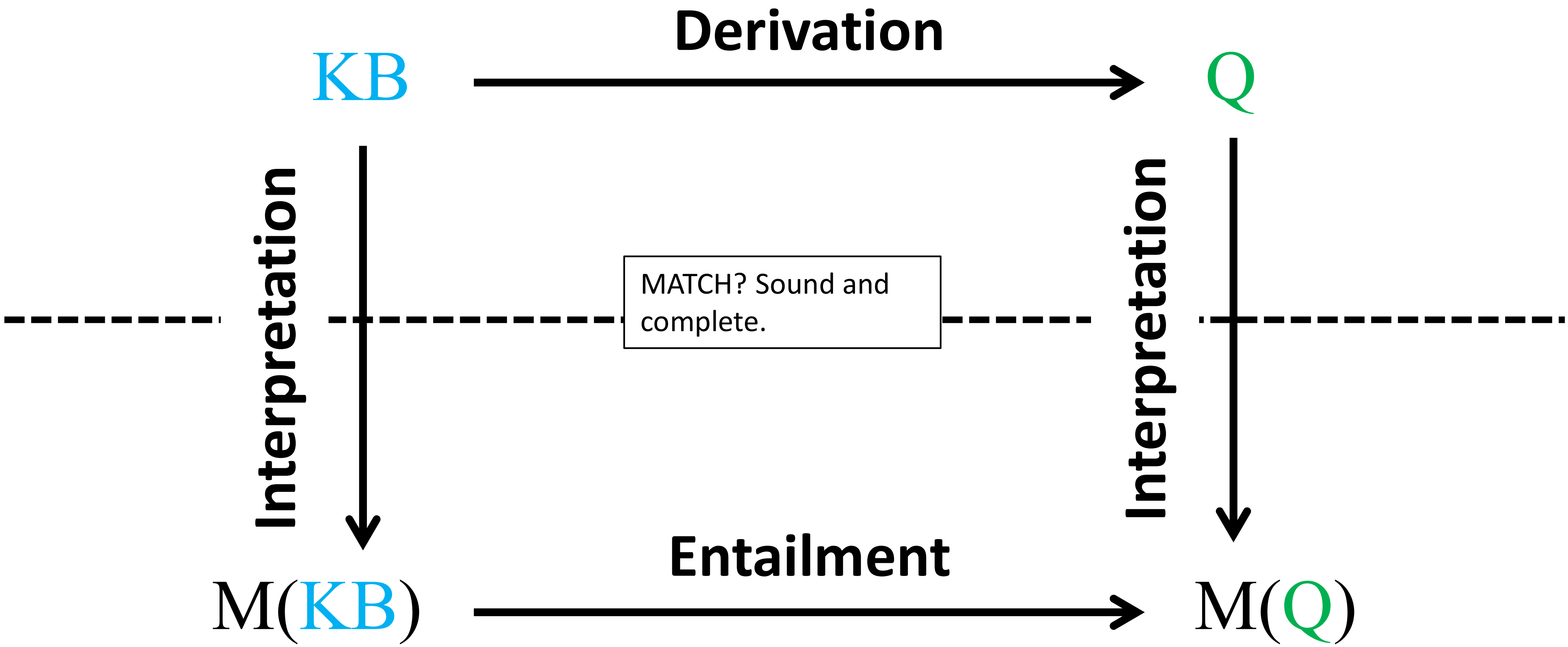
$$KB \xrightarrow{\text{Derivation}} Q$$

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

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):

$$KB \models Q$$

by proving that:

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

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

Resolution: Two Forms of Notation

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

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

Resolution: Two Forms of Notation

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

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

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	$+$, $\vert\vert$	or
$\Rightarrow$	(Material) implication	$\rightarrow$, $\supset$	implies
$\Leftrightarrow$	(Material) equivalence	$\leftrightarrow$, $\equiv$, iff	if and only if
$\top$	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:

$$K_1 \wedge K_2 \wedge \dots \wedge K_m$$

of clauses. A clause Ki consists of a disjunction

$$(l_{i1} \vee l_{i2} \vee \dots \vee l_{in_i})$$

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

literals

$$(\neg m \lor n \lor o) \land (\neg n \lor m) \land (\neg o \lor m)$$

clauses

CNF Grammar

$$
\begin{array}{rcl}
CNFSentence & \rightarrow & Clause_1 \wedge \cdots \wedge Clause_n \\
Clause & \rightarrow & Literal_1 \vee \cdots \vee Literal_m \\
Fact & \rightarrow & Symbol \\
Literal & \rightarrow & Symbol \mid \neg Symbol \\
Symbol & \rightarrow & P \mid Q \mid R \mid \ldots \\
HornClauseForm & \rightarrow & DefiniteClauseForm \mid GoalClauseForm \\
DefiniteClauseForm & \rightarrow & Fact \mid (Symbol_1 \wedge \cdots \wedge Symbol_l) \Rightarrow Symbol \\
GoalClauseForm & \rightarrow & (Symbol_1 \wedge \cdots \wedge Symbol_l) \Rightarrow False
\end{array}
$$

*** 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 b and $\neg b$ are complementary. The resolution rule deletes a pair of complementary 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 b and $\neg$ b are complementary. The clause (b $\wedge$ $\neg$ 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 $\mathbf{b}$ and $\neg\, \mathbf{b}$ are complementary. The resolution rule deletes a pair of complementary literals from two clauses and combines the rest.

Factorization

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

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

Disjunction of multiple copies of literals ($\mathbf{d} \vee \mathbf{d}$) 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:

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

Clause is ($\mathbf{d} \vee \mathbf{d}$) is replaced by a single literal $\mathbf{d}$. This is called factorization. Contradiction ($\mathbf{b} \wedge \neg\, \mathbf{b}$) 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 (D) 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)?"

$$KB \models Q$$

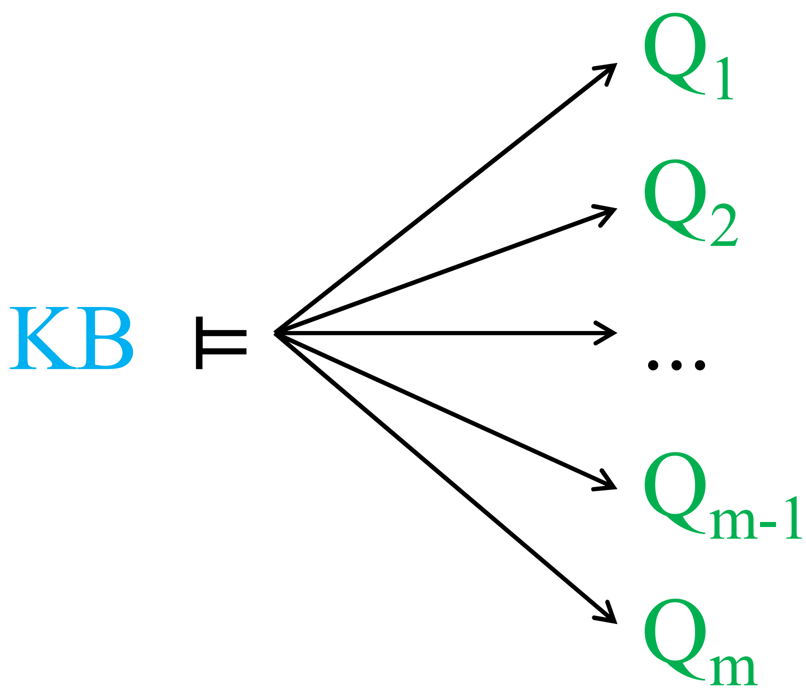
But we could ask the following question:

"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*))
  $t \leftarrow t + 1$
  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?"

$$\text{CURRENTKB} \vDash \begin{cases} \text{ACTION}_1 & \text{follows from} \\ \text{ACTION}_2 & \text{does NOT follow} \\ \dots & \\ \text{ACTION}_{m-1} & \text{follows from} \\ \text{ACTION}_m & \text{follows from} \end{cases}$$

Logical Entailment with KB Agents

But we could ask the following question:

"Which ACTIONs follow from CURRENTKB?"

$$(KBBEFORE \wedge percept) \models \begin{cases} ACTION_1 & \text{follows from} \\ ACTION_2 & \text{does NOT follow} \\ \ldots & \\ ACTION_{m-1} & \text{follows from} \\ ACTION_m & \text{follows from} \end{cases}$$

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:

$$(\mathrm{KBBEFORE} \wedge \mathrm{percept}) \vDash \mathrm{ACTION} \text{ follows from}$$

to decide whether to apply ACTION **or not.**