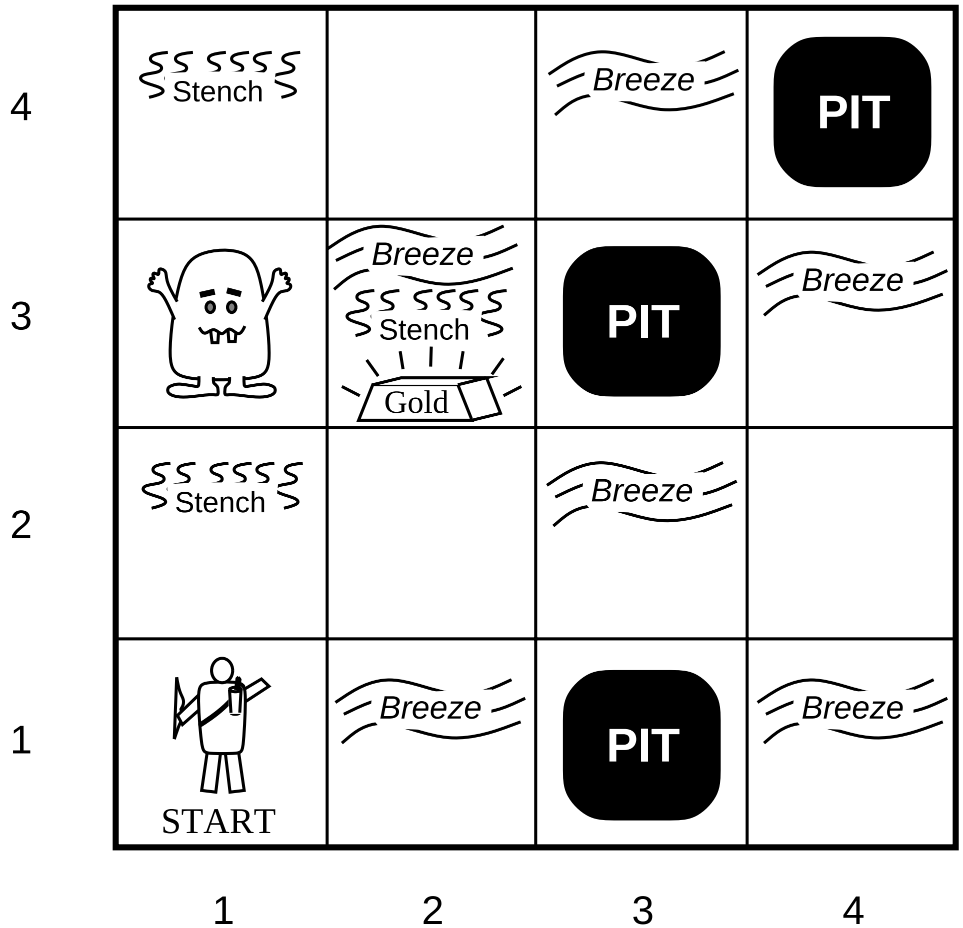
Artificial Intelligence

Knowledge Representation and Reasoning (Logic)

Nilsson - Chapters 13, 14, 15

Russell and Norvig - Chapters 7, 8

“wumpus world”

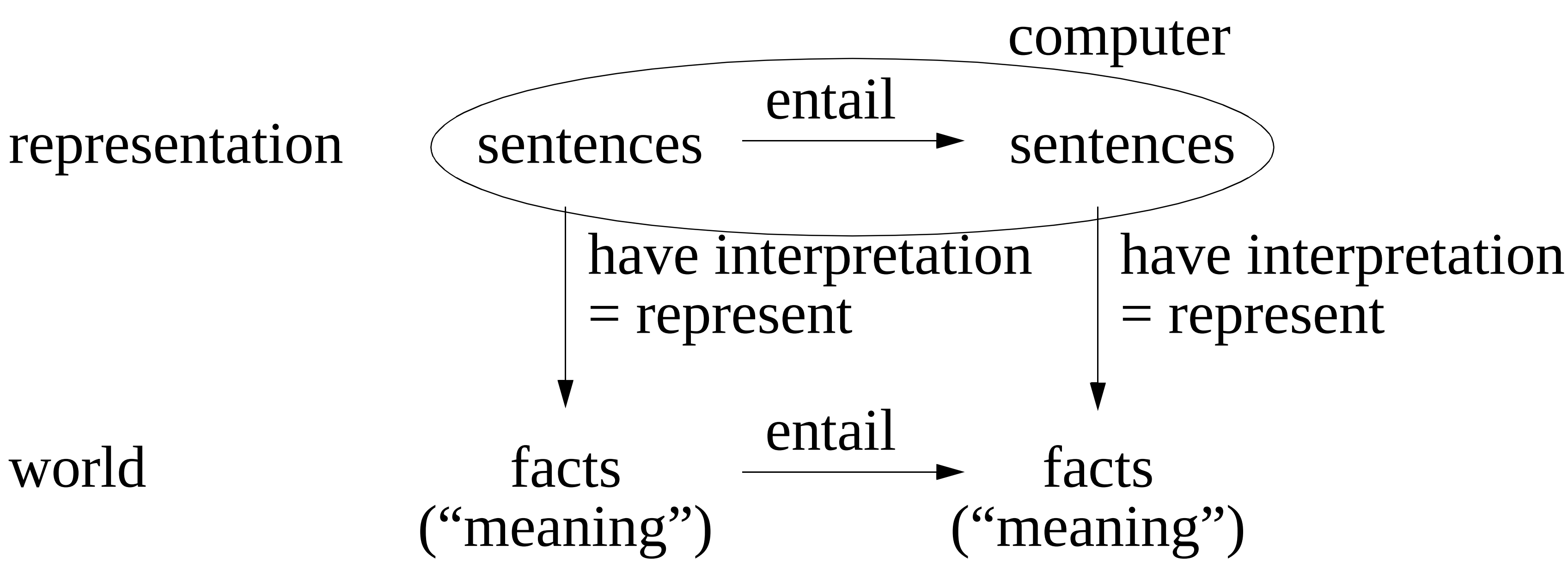


knowledge representation languages should be
expressive, concise, unambiguous, independent of context, effective

Agents are given knowledge about the world.

Knowledge Representation:
How can facts about the world be represented?

Reasoning:
How can the agent infer new facts from the given ones?


computer
representation
sentences
entail
sentences
have interpretation
= represent
have interpretation
= represent
world
facts
("meaning")
entail
facts
("meaning")

arithmetic

syntax

Is “1 > 2” a well-formed formula?

Is “x + * y 5 >” a well-formed formula?

semantics

When is “x + 2 = 5” true?

propositional logic
= sentences represent whole propositions

"2 is prime."	P
"I ate breakfast today."	Q

syntax
= how a sentence looks like

Sentence -> AtomicSentence | ComplexSentence

AtomicSentence -> T(RUE) | F(ALSE) | Symbols

Symbols -> P | Q | R | ...

$\neg$ negation

ComplexSentence -> (Sentence) | NOT Sentence |
Sentence Connective Sentence

Connective -> AND | OR | IMPLIES | EQUIV(ALENT)

conjunction $\wedge$ disjunction $\vee$ implication $\Rightarrow$ equivalence $\Leftrightarrow$

Precedence: NOT AND OR IMPLIES EQUIVALENT

syntax
= how a sentence looks like

The constants T(RUE) and F(ALSE) are sentences.
The symbols P, Q, R, ... are sentences.
If S is a sentence, then (S) is a sentence.
If S is a sentence, then NOT S is a sentence.
If S1 and S2 are sentences, then S1 AND S2 is a sentence.
If S1 and S2 are sentences, then S1 OR S2 is a sentence.
If S1 and S2 are sentences, then S1 IMPLIES S2 is a sentence.
If S1 and S2 are sentences, then S1 EQUIV S2 is a sentence.

Examples: P, T, P IMPLIES NOT T, P AND (Q OR S)

semantics
= what a sentence means

interpretation:
assigns each symbol a truth value, either t(rue) or f(alse)

the truth value of T(RUE) is t(rue)
the truth value of F(ALSE) is f(alse)

truth tables ("compositional semantics")
the meaning of a sentence is a function of the meaning of its parts

A	B	NOT A	A AND B	A OR B	A IMPLIES B	A EQUIV B
t	t	f	t	t	t	t
t	f		f	t	f	f
f	t	t	f	t	t	f
f	f		f	f	t	t

examples

either I go to the movies or I go swimming

(inclusive versus exclusive OR - our OR is inclusive)

A	B	A OR B (inclusive OR)	A XOR B (exclusive OR)
t	t	t	f
t	f	t	t
f	t	t	t
f	f	f	f

Read “A IMPLIES B” as “if A then B”

Read “A EQUIV B” as “if and only if A then B” or “A and B have the same truth value”

examples

2 is prime implies that 2 is even
(implication does not imply causality)

2 is odd implies that 3 is even
(false implies everything)

rewrite rules

A and NOT NOT A have identical truth tables.

A	A		A	NOT NOT A
t	t		t	t
f	f		f	f

Therefore, we can say A instead of NOT NOT A, and NOT NOT A instead of A without changing the truth value of any propositional ssentence.

For example, A AND B and A AND NOT NOT B have identical truth tables and are therefore equivalent.

A	B	A AND B	A AND NOT NOT B
t	t	t	t
t	f	f	f
f	t	f	f
f	f	f	f

rewrite rules

NOT NOT A = A
NOT (A AND B) = NOT A OR NOT B
NOT (A OR B) = NOT A AND NOT B
A AND (B OR C) = (A AND B) OR (A AND C)
A IMPLIES B = NOT A OR B
A EQUIV B = (A IMPLIES B) AND (B IMPLIES A)

and many more

deriving formulas

A	B	A XOR B	(= exclusive OR)
t	t	f	A AND B
t	f	t	A AND NOT B
f	t	t	NOT A AND B
f	f	f	NOT A AND NOT B

(A AND NOT B) OR (NOT A AND B)

problem with propositional logic

$$\frac{\text{All cats are mammals.}\quad \text{Bob is a cat.}}{\text{Bob is a mammal.}}$$

first-order logic (= FOL)
= sentences refer to objects, their relationships, and their properties
= superset of propositional logic

constants	objects	Bob, CS3361, October 13
functions		MotherOf, ColorOf
predicates	relations	IsMotherOf
	properties	IsBlue

syntax
= how a sentence looks like

Sentence ->	AtomicSentence | Sentence Connective Sentence | Quantifier Variable Sentence | NOT Sentence (Sentence)
AtomicSentence ->	T(RUE) | F(ALSE) | Predicate (Term, ..., Term) | **Predicate** | Term = Term
Term ->	Constant | Variable | Function (Term, ..., Term)
Connective ->	AND ($\wedge$) | OR ($\vee$) | IMPLIES ($\Rightarrow$) | EQUIV ($\Leftrightarrow$)
Quantifier ->	FORALL ($\forall$) | EXISTS ($\exists$)
Constant ->	A | ... | Z | Bob | CS3361 | ...
Variable ->	a | ... | z
Predicate ->	IsMotherOf | IsBlue | ...
Function ->	MotherOf | ColorOf | ...

Precedence: NOT AND OR IMPLIES EQUIVALENT sentences refer to facts

terms refer to objects

predicated can be understood as sets

IsMotherOf(x,y)

(Sue, Bill)	(Sue, Tom)
(Martha, Irene)	(Anna, Heidi)

= (equal) is a predicate with a predefined meaning

=(x,y)

(Sue, Sue)	(Tom, Tom)	...
(Martha, Martha)	(Anna, Anna)	

functions are just a convenient way of using predicates

MotherOf(y) = x iff IsMotherOf(x,y)

universal quantifier (“for all”): FORALL ($\forall$)

existential quantifier (“there exists”): EXISTS ($\exists$)

every variable must be introduced by a quantifier before it is used

Everything is blue.

FORALL x IsBlue(x)

=

IsBlue(Sven) AND IsBlue(Tom’s Mailbox) AND IsBlue(Sven’s TV) AND ...

At least one thing is blue

EXISTS x IsBlue(x)

=

IsBlue(Sven) OR IsBlue(Tom’s Mailbox) OR IsBlue(Sven’s TV) OR ...

universal quantifier (“for all”): FORALL (∀)
existential quantifier (“there exists”): EXISTS (∃)

Not everything is blue.
NOT FORALL x IsBlue(x)
=
NOT(IsBlue(Sven) AND IsBlue(Tom’s Mailbox) AND ...)
=
NOT(IsBlue(Sven)) OR NOT(IsBlue(Tom’s Mailbox)) OR ...
=
EXISTS x NOT IsBlue(x)

Everything is blue.
FORALL x IsBlue(x)
= NOT NOT FORALL x IsBlue(s)
= NOT EXISTS x NOT IsBlue(s)

universal quantifier (“for all”): FORALL ($\forall$)

existential quantifier (“there exists”): EXISTS ($\exists$)

Nothing is blue.

FORALL x NOT IsBlue(x)

=

NOT(IsBlue(Sven)) AND NOT(IsBlue(Tom’s Mailbox)) AND ...

=

NOT(IsBlue(Sven) OR IsBlue(Tom’s Mailbox) OR ...)

=

NOT EXISTS x IsBlue(x)

At least one thing is blue.

EXISTS x IsBlue(x)

= NOT NOT EXISTS x IsBlue(x)

= NOT FORALL x NOT IsBlue(x)

rewrite rules

Make sure that you convince yourself of the following rewrite rules
(both that they make intuitively sense and that they are correct)

FORALL x P(x) = NOT EXISTS x NOT P(x)
EXISTS x P(x) = NOT FORALL x NOT P(x)
FORALL x FORALL y P(x,y) = FORALL y FORALL x P(x,y)
EXISTS x EXISTS y P(x,y) = EXISTS y EXISTS x P(x,y)
NOT FORALL x (P(x) IMPLIES Q(x)) = EXISTS x (P(x) AND NOT Q(x))
NOT EXISTS x (P(x) AND Q(x)) = FORALL x (P(x) IMPLIES NOT Q(x))

and many more

universal quantifier (“for all”): FORALL ($\forall$)
existential quantifier (“there exists”): EXISTS ($\exists$)

Every mailbox is blue
FORALL x (IsMailbox(x) IMPLIES IsBlue(x))

Everything is a blue mailbox (probably not what you mean)
FORALL x (IsMailbox(x) AND IsBlue(x))

At least one mailbox is blue
EXISTS x (IsMailbox(x) AND Blue(x))

At least one thing is not a mailbox or blue (probably not what you mean)
EXISTS x (IsMailbox(x) IMPLIES IsBlue(x))

universal quantifier (“for all”): FORALL ($\forall$)

existential quantifier (“there exists”): EXISTS ($\exists$)

Everyone has at least one mother

FORALL y EXISTS x IsMotherOf(x,y)

Everyone shares at least one mother (= has the same mother).

EXISTS x FORALL y IsMotherOf(x,y)

The translations above are actually not quite correct.

The correct translation is

Everyone has at least one mother

FORALL y (IsPerson(y) IMPLIES EXISTS x IsMotherOf(x,y))

It is unnecessary (but not incorrect) to say

ORALL y (IsPerson(y) IMPLIES EXISTS x (IsPerson(x) AND IsMotherOf(x,y))

examples

Sue is Bill’s mother.
Sue is Bill’s and Tom’s mother.
Bill is smart, and so is Tom.
Either Bill or Tom is smart, but not both.
Sue has kids (= Sue has at least one kid).
Sue has no kids.
Sue has one kid.
Sue has at most one kid.
Sue has at least two kids.
All of Sue’s kids are smart.
At least one of Sue’s kids is smart.
None of Sue’s kids are smart, neither is Sue.
Sue loves Bill.
Noone (= no person) loves Bill.
Everybody (= every person) loves somebody. - ambiguos
Somebody (= some person) loves everybody. - ambiguous
Everyone (= every person) has exactly one mother.
If two people have the same mother
then they are either both smart or neither of them is smart.

semantics
= what a sentence means

propositional logic

interpretation (= “world” = model)
assigns
each propositional symbol a truth value
(every row in a truth table corresponds to one interpretation)

first-order logic

interpretation (= “world” = “model”):
assigns
each constant a meaning (= an object),
each function a meaning,
each predicate a meaning.

sentence is

valid (= a tautology)
if it holds for all interpretations
(e.g. all rows of the truth table are true for a propositional sentence)

satisfiable
if it holds for at least one interpretation
(e.g. at least one row of the truth table is true for a propositional sentence)

unsatisfiable (= a contradiction)
if it holds for no interpretation
(e.g. all rows of the truth table are false for a propositional sentence)

valid? satisfiable? unsatisfiable?

2 is prime.

2 is prime or 2 is not prime.

FORALL x Eats(Lion,x)
“the lion eats everything”

AND

FORALL x FORALL y (EATS(x,y) IMPLIES Hunts(x,y))
if x eats y then x hunts y

AND

FORALL x FORALL y (Hunts(x,y) IMPLIES NOT Hunts(y,x))
if x hunts y, then y doesn’t hunt x

(1) FORALL x Eats(Lion,x)
"the lion eats everything"

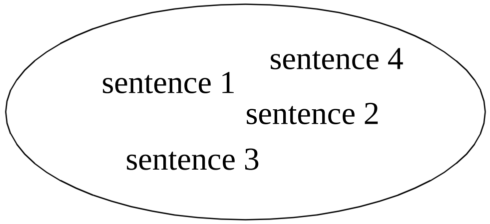
(2) FORALL x FORALL y (EATS(x,y) IMPLIES Hunts(x,y))
if x eats y then x hunts y

(3) FORALL x FORALL y (Hunts(x,y) IMPLIES NOT Hunts(y,x))
if x hunts y, then y doesn't hunt x

This knowledge base is unsatisfiable.

(1) entails: (4) Eats(Lion,Lion) - "the lion eats itself"
(2) and (4) entail: (5) Hunts(Lion, Lion) - "the lion hunts itself"
(3) and (5) entail: (6) NOT Hunts(Lion, Lion) - "the lion does not hunt itself

However, (5) and (6) cannot be true at the same time.



sentence 1 AND sentence 2 AND sentence 3 AND sentence 4

knowledge base $\models$ sentence (read: entails)

= whenever an interpretation makes the knowledge base true,
it also makes the sentence true as well
(advantage: the computer does not need to know the interpretation)

example:

does “heads, I win; tails, you lose”
entail
“I win”

(note: background knowledge is important!)

IN THE FOLLOWING, WE ONLY DEAL WITH PROPOSITIONAL LOGIC

for first order logic, truth tables do not work since there are infinitely many interpretations

for first order logic, resolution might not terminate if a sentence is not entailed by the knowledge base
(= resolution is semi-decidable for first order logic)

method 1: truth table

HE = heads comes up, TA = tails comes up, I = IW win, UL = you lose

heads, I win	HE IMPLIES IW
tails, you lose	TA IMPLIES UL
either heads or tails (background knowledge)	HE EQUIV NOT TA
I win or you lose (background knowledge)	IW EQUIV UL

HE	TA	IW	UL	HE IMPLIES IW	TA IMPLIES UL	HE EQUIV NOT TA	IW EQUIV UL
t	t	t	t	t	t	f	t
t	t	t	f	t	f	f	f
t	t	f	t	f	t	f	f
t	t	f	f	f	f	f	t
t	f	t	t	t	t	t	t
t	f	t	f	t	t	t	f
t	f	f	t	f	t	t	f
t	f	f	f	f	t	t	t
f	t	t	t	t	t	t	t
f	t	t	f	t	f	t	f
f	t	f	t	t	t	t	f
f	t	f	f	t	f	t	t
f	f	t	t	t	t	f	t
f	f	t	f	t	t	f	f
f	f	f	t	t	t	f	f
f	f	f	f	t	t	f	t

each row corresponds to one interpretation

inference procedures

$$\text{knowledge base} \quad \vdash \quad \text{sentence}$$

= the inference procedure can infer the sentence from the knowledge base

the inference procedure is **sound**
if knowledge base $\models$ sentence
whenever knowledge base $\vdash$ sentence

logical inference
(deduction)

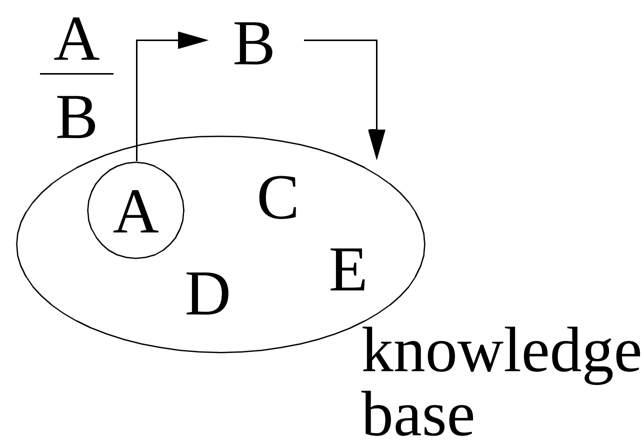
the inference procedure is **complete**
if knowledge base $\vdash$ sentence
whenever knowledge base $\models$ sentence

inference procedures

= the repeated application of inference rules

$$A \vdash B$$

$$\frac{A}{B}$$



local reasoning

why does it work?

monotonicity!

in propositional logic it holds that $\dfrac{\text{knowledge base that includes A}}{\text{B}}$ if $\dfrac{A}{B}$

proof by contradiction

we want to show:

“heads, I win; tails, you lose” entails “I win;” in other words, whenever an interpretation makes “heads, I win; tails, you lose” true, it also makes “I win” true

instead, we show:

there is no interpretation that makes “heads, I win; tails, you lose” and “NOT(I win)” true at the same time; in other words “heads, I win; tails, you lose” and “NOT(I win)” entails FALSE.

resolution

$$\frac{\text{P OR Q, NOT Q OR R}}{\text{P OR R}}$$

(read: “(P OR Q) AND (NOT Q OR R)” entails “P OR R”)

P and R can be arbitrary sentences
afterwards we remove duplicates from the resulting disjunct

examples:

$$\frac{\text{A OR C OR NOT E, B OR NOT C OR NOT D OR F}}{\text{A OR B OR NOT D OR NOT E OR F}}$$

$$\frac{\text{A OR B, NOT A OR B}}{\text{B}}$$

$$\frac{\text{P OR Q, NOT Q OR R}}{\text{P OR R}}$$

$$\frac{\text{NOT P IMPLIES Q, Q IMPLIES R}}{\text{NOT P IMPLIES R}}$$

proof by contradiction

we want to show:

“heads, I win; tails, you lose” entails “I win;” in other words, whenever an interpretation makes “heads, I win; tails, you lose” true, it also makes “I win” true

instead, we show:

there is no interpretation that makes “heads, I win; tails, you lose” and “NOT(I win)” true at the same time; in other words “heads, I win; tails, you lose” and “NOT(I win)” entails FALSE.

to do that, we add the negated sentence “NOT(I win)” to the knowledge base “heads, I win; tails, you lose” and use resolution to derive false (“empty”) - if we can do that, “heads, I win; tails you lose” entails “I win.”

transform into conjunctive normal form (CNF)

Add the negated sentence to the knowledge base.

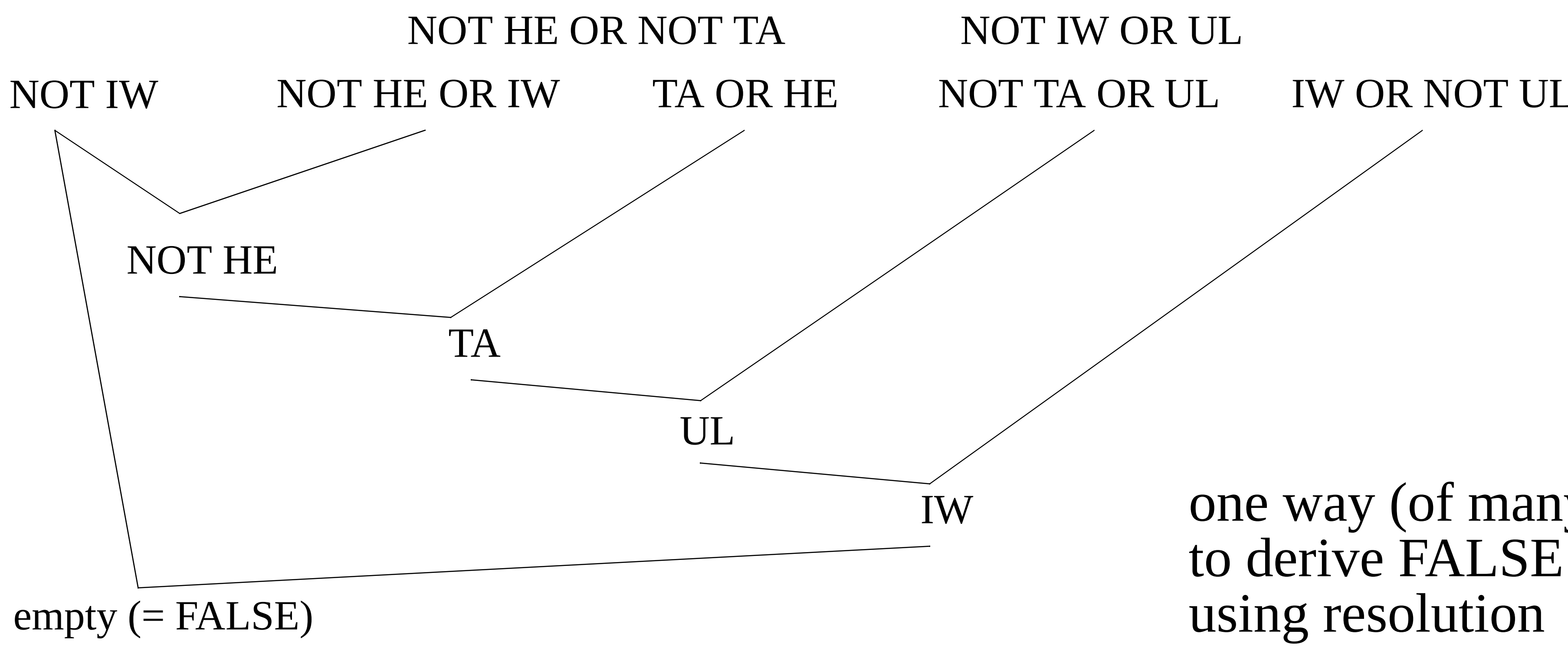
For each sentence in the knowledge base, do:

1. Eliminate EQUIV
    - rewrite A EQUIV B as A IMPLIES B AND B IMPLIES A
2. Eliminate IMPLIES
    - rewrite A IMPLIES B as NOT A OR B
3. Move NOT inward
    - rewrite NOT(A AND B) as NOT A OR NOT B
    - rewrite NOT(A OR B) as NOT A AND NOT B
    - rewrite NOT(NOT A) as A
3. Distribute AND over OR
    - rewrite A AND B OR C as (A OR C) AND (B OR C)
    - rewrite A OR B AND C as (A OR B) AND (A OR C)
4. Flatten nested conjuncts and disjuncts
    - rewrite A AND (B AND C) as A AND B AND C
    - rewrite (A AND B) AND C as A AND B AND C
    - rewrite A OR (B OR C) as A OR B OR C
    - rewrite (A OR B) OR C as A OR B OR C
5. replace the original sentence with all of the obtained disjuncts
    - example: for A OR B AND C, add both A OR B and C

HE = heads comes up, TA = tails comes up, IW = I win, UL = you lose

heads, I win	HE IMPLIES IW
tails, you lose	TA IMPLIES UL
either heads or tails (background knowledge)	HE EQUIV NOT TA
I win or you lose (background knowledge)	IW EQUIV UL
I don’t win (negated sentence)	NOT IW

knowledge base after steps 1-5:



one way (of many)
to derive FALSE
using resolution

resolution is sound and complete

If a sentence is entailed by the knowledge base,
resolution will eventually produce FALSE.
This requires a good search strategy.
There are several heuristics for good search strategies out there,
including resolving with unit clauses whenever possible
(to reduce the length of existing clauses)
or resolving with clauses derived from the sentence to be shown
to be entailed by the knowledge base
(because FALSE cannot be produced from a satisfiable knowledge
base alone).

If a sentence is not entailed by the knowledge base,
then resolution will eventually not be able to produce new clauses,
without having produced FALSE at that time.
(This statement is true only for propositional logic.)
This requires an exhaustive enumeration of all producable clauses
and thus can take a long time.

resolution is sound and complete

think about this:

why can't we just use resolution to derive the sentence we are interested in (instead of adding its negation and derive false)?

answer

the empty knowledge base (KB) entails "P OR NOT P"

but resolution cannot be used to derive "P OR NOT P" from the empty KB resolution is sound and complete

think about this:

why can’t we just use modus ponens instead of resolution to derive the sentence we are interested in?

modus ponens

$$\frac{P,\ P \text{ IMPLIES } Q}{Q}$$

$$\frac{\text{it rains, it rains implies the grass is wet}}{\text{the grass is wet}}$$

answer

“P IMPLIES Q” and “NOT P IMPLIES Q” entails “Q”

but modus ponens cannot show that