Logic Background (1A)

Young W. Lim
2/4/19

This work is based https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-825-techniques-in-artificial-intelligence-sma-5504-fall-2002/lecture-notes/

Proposition : etymology

From Old French, from Latin prōpositiō

(“a proposing, design, theme, case”).

The content of an assertion

that may be taken as being **true** or **false**

and is considered abstractly without reference to

the linguistic sentence that constitutes the assertion.

Predicate : etymology

From Middle French predicate (French prédicat),

from post-classical Late Latin praedicatum ("thing said of a subject"),

a noun use of the neuter past participle of praedicare ("proclaim")

From Latin predicātus, perfect passive participle of praedicō,

from prae + dicō ("declare, proclaim"), from dicō ("say, tell").

Proposition

In Aristotelian logic a **proposition** is
a particular kind of sentence,
one which affirms or denies
a **predicate** of a subject.

In formal logic a **proposition** is considered as
objects of a formal language.
A formal language begins with different types of symbols.

Predicate

(grammar) The part of the sentence (or clause)

which states something about

the subject or the object of the sentence.

The dog	barked very loudly
subject	*predicate*

(logic) A term of a statement,

where the statement may be true or false

depending on whether the thing referred to

by the values of the statement's variables

has the property signified by that (predicative) term.

Propositional Logic

propositional logic includes only

- **operators** (connectives)
- propositional **constants**

as symbols in its language.

the **propositions** in this language are

- propositional **constants**

  considered **atomic** propositions

- **composite** propositions

  recursive application of operators to propositions

Predicate Logic

predicate logic include

- variables
- operators (connectives)
- predicate
- function symbols
- quantifiers

as symbols in their languages.

Logic

a formal language

syntax – legal expressions

semantics – the meaning of legal expressions

proof system – a way of manipulating syntactic expressions
to get another syntactic expressions

two kinds of inferences

multiple percepts → conclusions

current state, operators → next state properties

Propositional Logic

sentences (wffs : well formed formulas)

t and **f** are sentences

propositional variables are sentences (**a**, **b**, **c**, …)

if **a** and **b** are sentences, the followings are also sentences

$(\mathbf{a})$, $\neg\mathbf{a}$, $\mathbf{a} \wedge \mathbf{b}$, $\mathbf{a} \vee \mathbf{b}$, $\mathbf{a} \Rightarrow \mathbf{b}$, $\mathbf{a} \Leftrightarrow \mathbf{b}$

Precedence of Connectives

$\neg$	highest
$\wedge$	
$\vee$	
$\Rightarrow$	
$\Leftrightarrow$	lowest

$$A \vee B \wedge C \quad = \quad A \vee (B \wedge C)$$

$$A \wedge B \Rightarrow C \vee D \quad = \quad (A \wedge B) \Rightarrow (C \vee D)$$

$$A \Rightarrow B \vee C \Leftrightarrow D \quad = \quad (A \Rightarrow (B \vee C)) \Leftrightarrow D$$

Semantics

meaning of a sentence : **true** or **false**

interpretation : an assignment of **true** or **false** to the propositional **variables**

$\models_i \varphi$	: sentence φ is **true** in the interpretation **i**
$\not\models_i \varphi$	: sentence φ is **false** in the interpretation **i**

$\varphi = A \lor B \land C$	i : A=T, B=T, C=T	⇒	$\models_i \varphi$
$\varphi = A \lor B \land C$	i : A=F, B=F, C=F	⇒	$\not\models_i \varphi$

Semantic Rules

$\models_i \varphi$: sentence φ is **true** in the interpretation **i**

$\not\models_i \varphi$: sentence φ is **false** in the interpretation **i**

$\models_i \mathbf{t}$ for all **i**

$\not\models_i \mathbf{f}$ for all **i**

$\models_i \neg\varphi$ iff $\not\models_i \varphi$

$\models_i \varphi \wedge \psi$ iff $\models_i \varphi$ **and** $\models_i \psi$

$\models_i \varphi \vee \psi$ iff $\models_i \varphi$ **or** $\models_i \psi$

$\models_i \mathbf{p}$ iff $\mathbf{i(p) = t}$

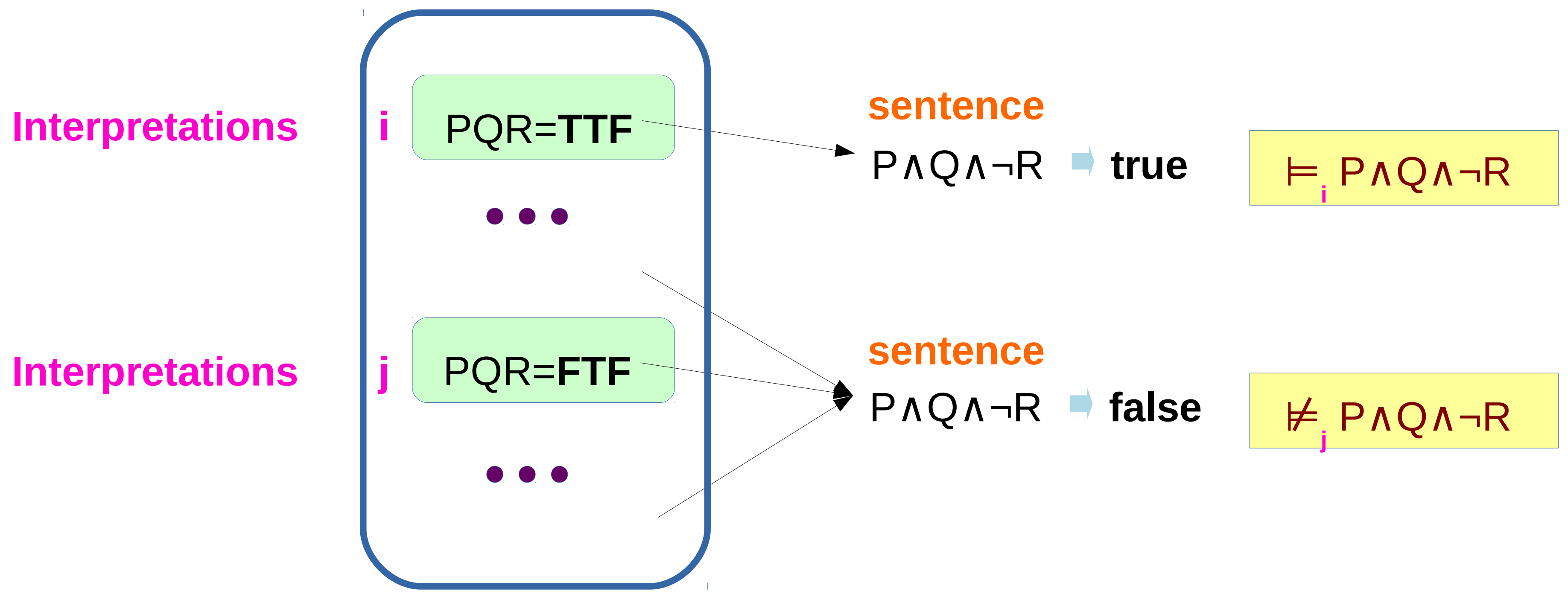
Since **i** is a mapping from variables to truth values, Look P up in **i** and return the truth value assigned to P

Semantic examples

meaning of a sentence : **true** or **false**

interpretation : an assignment of **true** or **false**

to the propositional variables (p, q, r)



Valid, Satisfiable, Unsatisfiable

a sentence is **valid** iff

its truth value is **t** in <u>all</u> interpretations

ex) **t**, ¬**f**, p ∨ ¬p (**tautology**: ⊤)

a sentence is **satisfiable** iff

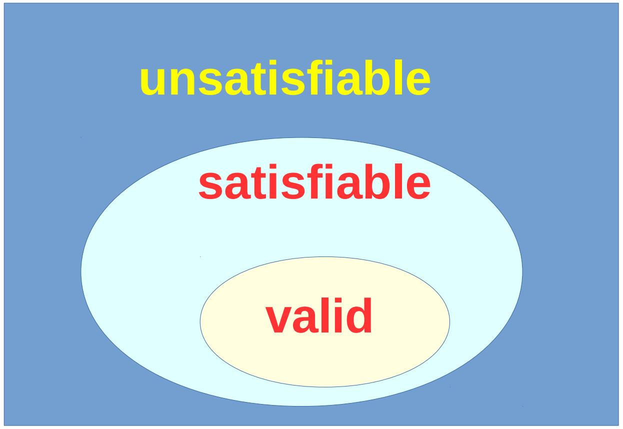
its truth value is **t** in <u>at least one</u> interpretation

ex) p, **t**, ¬p

a sentence is **unsatisfiable** valid iff

its truth value is **f** in <u>all</u> interpretations

ex) **f**, ¬**t**, p∧¬p (**contradiction**: ⊥)



Valid, Satisfiable, Unsatisfiable Examples

smoke ⇒ smoke	valid sentence	
smoke ∨ ¬smoke	valid sentence	
smoke ⇒ fire	satisfiable but not valid	(T ⇒ F) = F
(s ⇒ f) ⇒ (¬s ⇒ ¬f)	satisfiable but not valid	(F ⇒ T) = T, (T ⇒ F) = F
(s ⇒ f) ⇒ (¬f ⇒ ¬s)	valid sentence	contrapositive
b ∨ d ∨ (b ⇒ d)	valid sentence	
b ∨ d ∨ ¬b ∨ d	valid sentence	

Models

semantics :

the relationship between sentences and interpretations

there are some **set** of interpretations

that makes a sentence **true**

→ **models** of a sentence

an interpretations is a **model** of a sentence

if the sentence is **true** in that interpretation

an interpretation ***i*** is a **model** of a sentence φ ***iff*** $\models_i \varphi$

An interpretation can be a model

an **interpretation *i*** is a model of a sentence φ ***iff*** $\vDash_i \varphi$

models

x = John

x = Socrates

x = Baker

a sentence

F(x) ➡ **true**

A Model, an interpretation, a sentence

an interpretation ***i*** is a **model** of a sentence φ ***iff*** $\models_i \varphi$

$$\models_i \varphi$$

an interpretation

a sentence

sentence

φ

interpretation ***i***

T F ...

T

an interpretation ***i***
is a model
of a sentence φ

Models and Interpretations

P	Q	R	P∧(Q∨R)
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	F
F	F	T	F
F	F	F	F

models

3 among 8 interpretations

a set of

interpretation :

model

2^3 - 1models

T	T	T
T	T	F

T	T	T
T	F	T

T	F	T

Entailment

An interpretation i is a **model** of a sentence φ ***iff*** $\vDash_i \varphi$

A <u>set</u> of <u>sentences</u> KB <u>**entails**</u> a <u>sentence</u> φ

iff <u>**every**</u> **model** of KB is also a **model** of φ

KB: Knowledge Base
a set of sentences

every model of KB

every <u>model</u> of
each sentence in KB

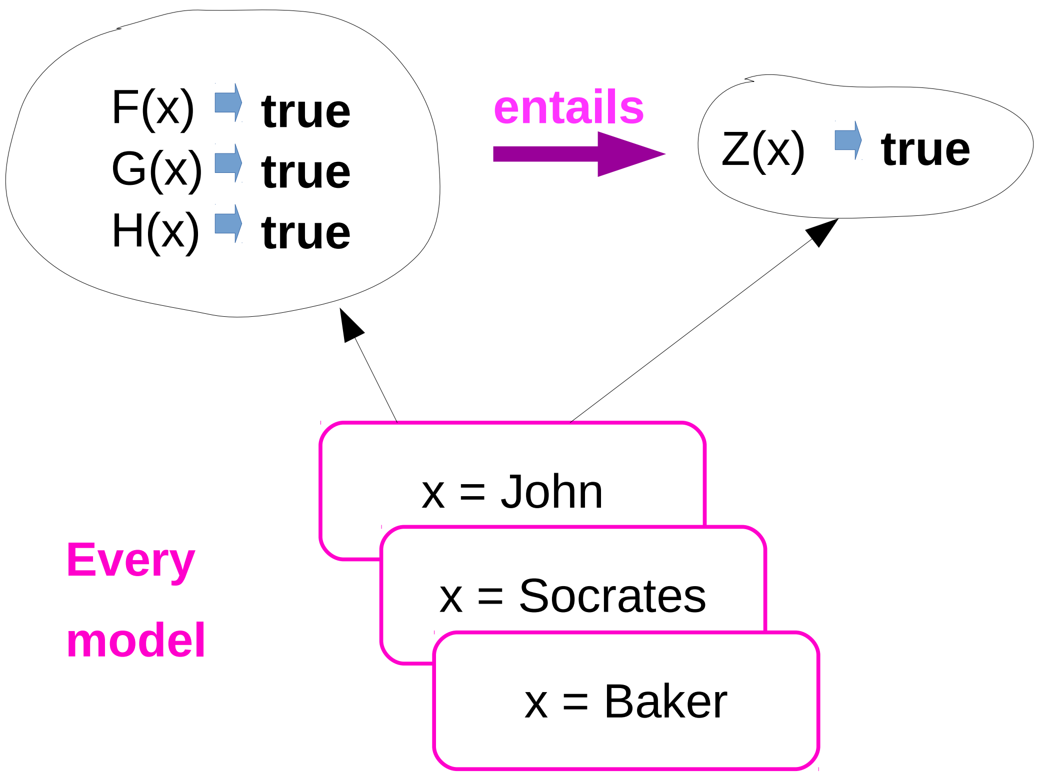
a <u>model</u> of
a sentence φ

KB <u>**entails**</u> φ

Entailment example

A set of sentences KB **entails** a sentence φ

iff **<u>every</u>** **model** of KB is also a **model** of φ



Models and Entailment

A set of sentences KB **entails** a sentence φ

iff **<u>every</u>** **model** of KB is also a **model** of φ

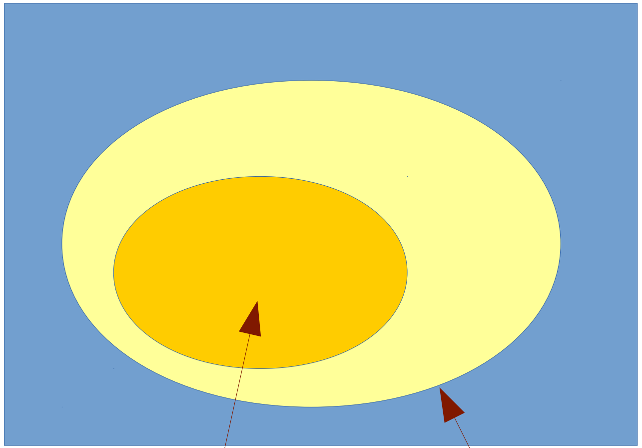
entail

$$KB \models \varphi$$

a set of sentences

a sentence

U: A set of all possible interpretation



A set of all models of KB	⊂	A set of all models of φ

A set of all models of KB	⊂	A set of all models of φ

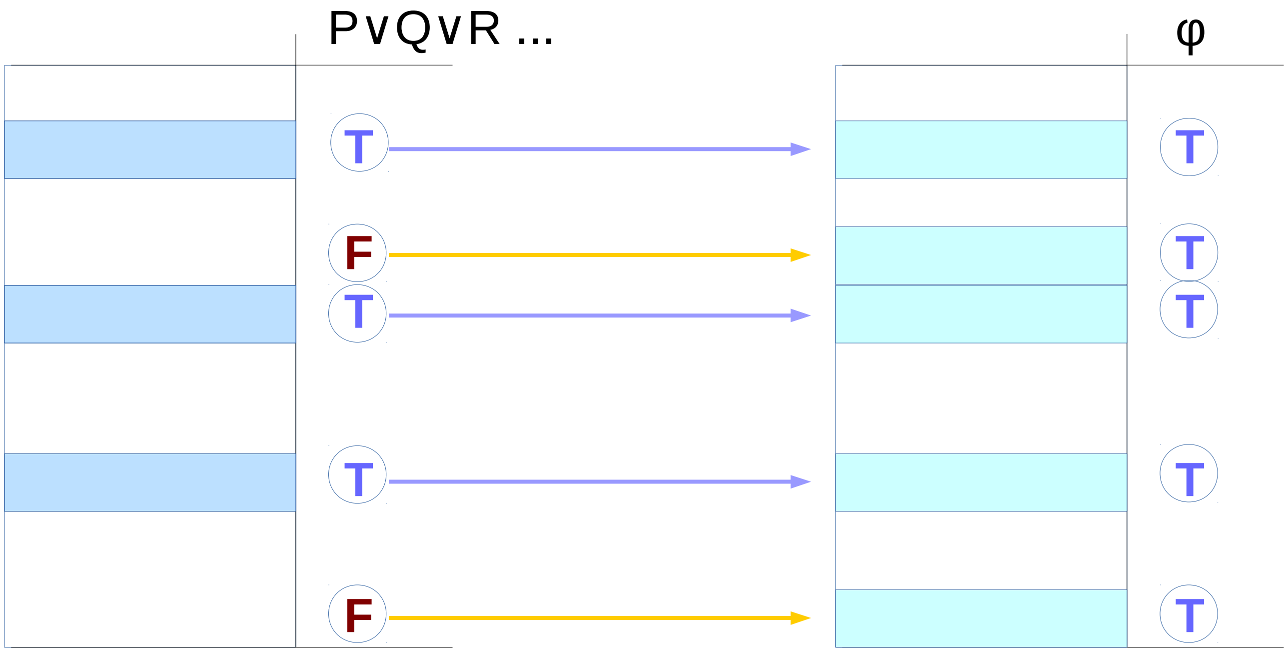
Truth Tables and Entailment

entail

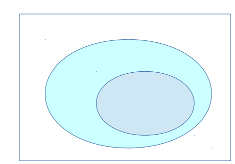
A set of sentences KB **entails** a sentence φ

iff **<u>every</u>** **model** of KB is also a **model** of φ

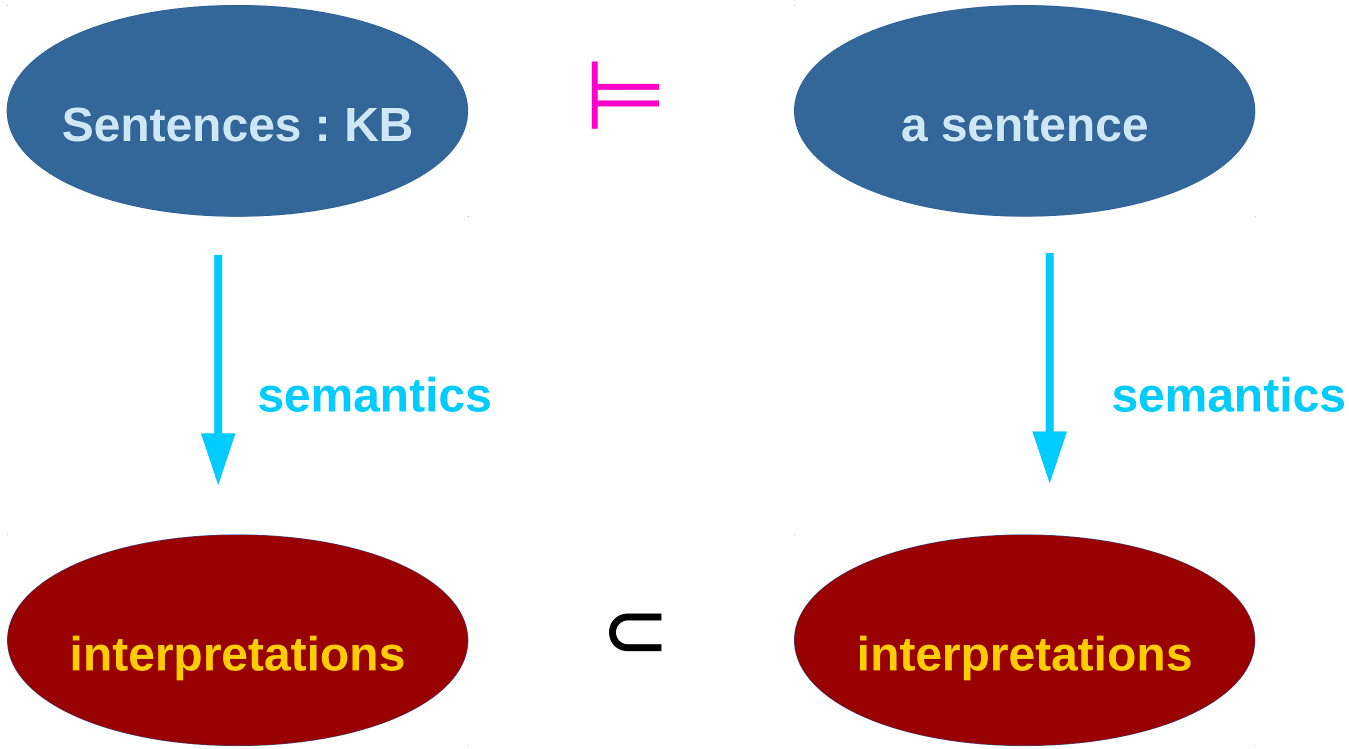
$$KB \models \varphi$$



A set of all models of KB $\subset$ A set of all models of φ



Entailment


Sentences : KB
⊨
a sentence
semantics
semantics
interpretations
⊂
interpretations

Models and Entailment

A set of sentences KB **entails** a sentence φ

iff **<u>every</u>** **model** of KB is also a **model** of φ

entail

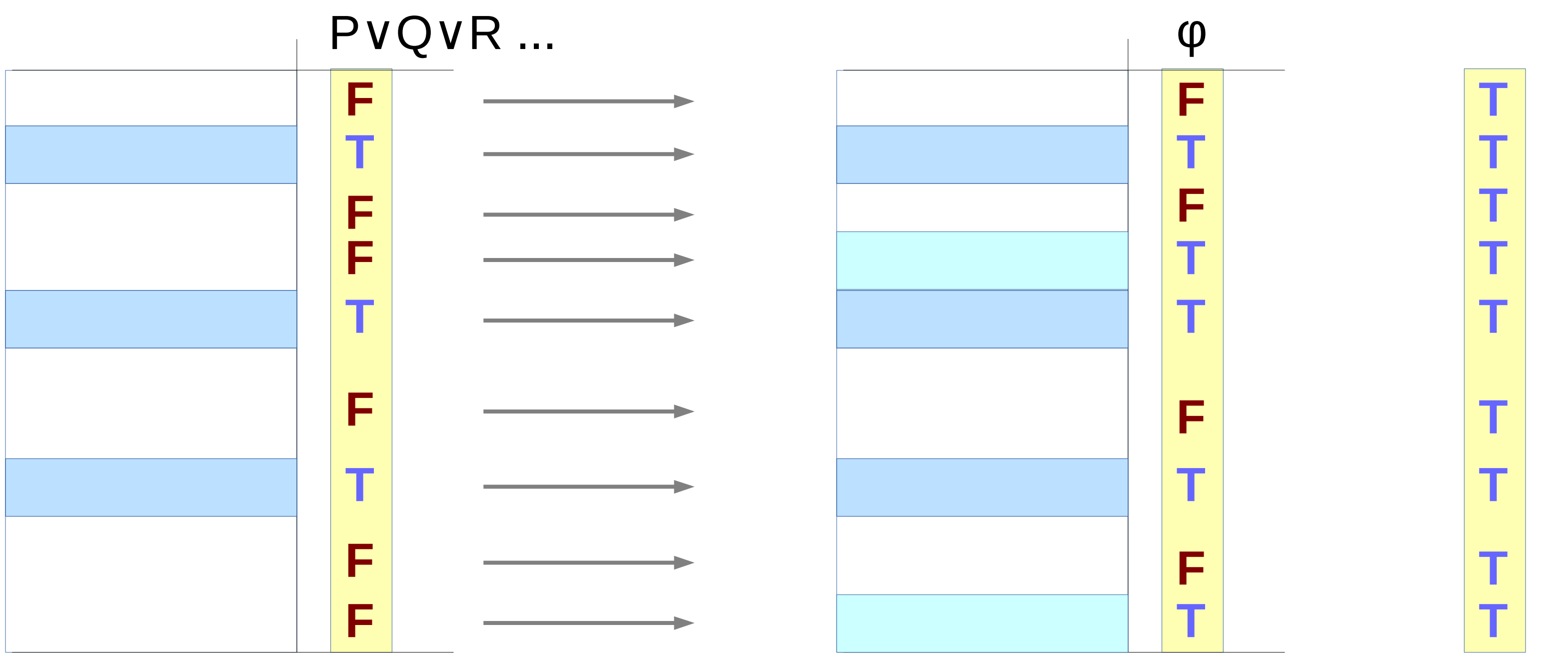
$$KB \models \varphi \quad \text{iff}$$

$$KB \rightarrow \varphi$$

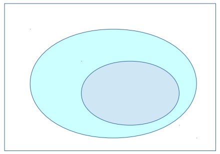
Truth Tables and Entailment

A set of sentences KB **entails** a sentence φ
iff **<u>every</u>** **model** of KB is also a **model** of φ

$\mathrm{KB} \models \varphi$ iff
$\mathrm{KB} \Rightarrow \varphi$



A set of all models of KB ⊂ A set of all models of φ



References

[1] en.wikipedia.org
[2] en.wiktionary.org
[3] U. Endriss, “Lecture Notes : Introduction to Prolog Programming”
[4] http://www.learnprolognow.org/ Learn Prolog Now!
[5] http://www.csupomona.edu/~jrfisher/www/prolog_tutorial
[6] www.cse.unsw.edu.au/~billw/cs9414/notes/prolog/intro.html
[7] www.cse.unsw.edu.au/~billw/dictionaries/prolog/negation.html
[8] http://ilppp.cs.lth.se/, P. Nugues,` An Intro to Lang Processing with Perl and Prolog