Automated reasoning 2018

Lecture 2: FOL semantics and theory

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2018-07-20

Logistics

- Send an email to TA (avais _at_ cse . iitb . ac . in) to make the list of participants
- Install Z3

Topic 2.1

FOL - semantics

Semantics : models

Definition 2.1

For signature $\mathbf{S} = (\mathbf{F}, \mathbf{R})$*, a* **S***-model* m *is a*

$$(D_m; \{f_m : D_m^n \rightarrow D_m | f/n \in \mathbf{F}\}, \{P_m \subseteq D_m^n | P/n \in \mathbf{R}\}),$$

where D_m *is a nonempty set. Let* **S***-Mods denotes the set of all* **S***-models.*

Some terminology

- D_m is called domain of m.
- f_m assigns meaning to f under model m.
- Similarly, P_m assigns meaning to P under model m.

Example 2.1 (Running example)

Consider $\mathbf{S} = (\{\cup/2\}, \{\in /2\})$*.*
$m = (\mathbb{N}; \cup_m = max, \in_m = \{(i,j)|i < j\})$ *is a* **S***-model.*

Commentary: Models are also known as interpretations/structures.

Semantics: assignments

Definition 2.2

An *assignment* *is a map* $\nu : \mathbf{Vars} \rightarrow D_m$

Semantics: term value

Definition 2.3

For a model m and assignment ν, we define $m^\nu : T_{\mathbf{S}} \rightarrow D_m$ as follows.

$$m^\nu(x) \triangleq \nu(x) \qquad\qquad x \in \mathbf{Vars}$$

$$m^\nu(f(t_1, \dots, t_n)) \triangleq f_m(m^\nu(t_1), \dots, m^\nu(t_n))$$

Definition 2.4

Let t be a closed term. $m(t) \triangleq m^\nu(t)$ *for any* ν.

Example 2.2

Consider assignment $\nu = \{x \mapsto 2, y \mapsto 3\}$ *and term* $\cup(x, y)$.

$m^\nu(\cup(x, y)) = max(2, 3) = 3$

Semantics: satisfaction relation

Definition 2.5

We define the satisfaction relation $\models$ among models, assignments, and formulas as follows

$$
\begin{array}{lll}
m, \nu \models \top & & \\
m, \nu \models P(t_1, \dots, t_n) & \textit{if} & (m^\nu(t_1), \dots, m^\nu(t_n)) \in P_m \\
m, \nu \models t_1 \approx t_2 & \textit{if} & m^\nu(t_1) = m^\nu(t_n) \\
m, \nu \models \neg F & \textit{if} & m, \nu \not\models F \\
m, \nu \models F_1 \vee F_2 & \textit{if} & m, \nu \models F_1 \textit{ or } m, \nu \models F_2 \\
 & & \textit{skipping other boolean connectives} \\
m, \nu \models \exists x. F & \textit{if} & \textit{there is } u \in D_m : m, \nu[x \mapsto u] \models F \\
m, \nu \models \forall x. F & \textit{if} & \textit{for each } u \in D_m : m, \nu[x \mapsto u] \models F
\end{array}
$$

Exercise 2.1

Consider sentence $F = \exists x. \forall y.\ \neg y \in x$ *(what does it say to you!)*

Use m *and* ν *from previous example. Does* $m, \nu \models F$*?*

Satisfiable, true, valid, and unsatisfiable

We say

- F is *satisfiable* if there are m and ν such that $m, \nu \models F$
- Otherwise, F is called unsatisfiable
- F is *true* in m ($m \models F$) if for all ν we have $m, \nu \models F$
- F is *valid* ($\models F$) if for all ν and m we have $m, \nu \models F$

If F is a sentence, ν has no influence in the satisfaction relation.(why?)

For sentence F, we say

- F is *true* in m if $m \models F$
- Otherwise, F is *false* in m.

Example: satisfiability

Example 2.3

Consider $\mathbf{S} = (\{s/1, +/2\}, \{\})$ *and formula* $\exists z. s(x) + y \approx s(z)$

Consider model $m = (\mathbb{N}; succ, +^{\mathbb{N}})$ *and assignment* $\nu = \{x \mapsto 3, y \mapsto 2\}$

$m^{\nu}(s(x) + y) = m^{\nu}(s(x)) +^{\mathbb{N}} m^{\nu}(y) = succ(m^{\nu}(x)) +^{\mathbb{N}} 2 = succ(3) + 2 = 6$

$m^{\nu[z \mapsto 5]}(s(x) + y) = m^{\nu}(s(x) + y) = 6$ *//Since z does not occur in the term*

$m^{\nu[z \mapsto 5]}(s(z)) = 6$

Therefore, $m, \nu[z \mapsto 5] \models s(x) + y \approx s(z)$.

$m, \nu \models \exists z. s(x) + y \approx s(z)$.

Extended satisfiability

We extend the usage of $\models$.

Definition 2.6

Let Σ be a (possibly infinite) set of formulas.
$m, \nu \models \Sigma$ if $m, \nu \models F$ for each $F \in \Sigma$.

Definition 2.7

Let M be a (possibly infinite) set of models.
$M \models F$ if for each $m \in M$, $m \models F$.

Implication and equivalence

Definition 2.8
Let Σ be a (possibly infinite) set of formulas.
$\Sigma \models F$ *if for each model m and assignment ν if $m, \nu \models \Sigma$ then $m, \nu \models F$.*

$\Sigma \models F$ is read Σ implies F. If $\{G\} \models F$ then we may write $G \models F$.

Definition 2.9
Let $F \equiv G$ if $G \models F$ and $F \models G$.

The above are semantic definitions.
Later, we will see the connection between logical connective $\Rightarrow$ and semantic implication $\models$.
We also need to prove that $\equiv$ are closely related to $\Leftrightarrow$.

The above definitions may appear to be abuse of notation.

Topic 2.2

FOL examples

Example: non-standard models

Example 2.4

Consider $\mathbf{S} = (\{\mathbf{0}/0, s/1, +/2\}, \{\})$ *and formula* $\exists z. s(x) + y \approx s(z)$

Unexpected model: *Let* $m = (\{a, b\}^*; \epsilon, append\_a, concat)$.

- *The domain of* m *is the set of all strings over alphabet* $\{a, b\}$.
- $append\_a$: *appends a in the input and*
- $concat$: *joins two strings.*

Let $\nu = \{x \mapsto ab, y \mapsto ba\}$.
Since $m, \nu[z \mapsto abab] \models s(x) + y \approx s(z)$,
$m, \nu \models \exists z. s(x) + y \approx s(z)$.

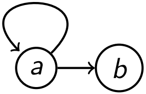
Exercise 2.2

- *Show* $m, \nu[y \mapsto bb] \not\models \exists z. s(x) + y \approx s(z)$
- *Give an assignment* ν *s.t.* $m, \nu \models x \not\approx 0 \Rightarrow \exists y.\ x \approx s(y)$.
  Show $m \not\models \forall x.\ x \not\approx 0 \Rightarrow \exists y.\ x \approx s(y)$.

Example: graph models

Example 2.5

Consider $\mathbf{S} = (\{\}, \{E/2\})$ *and* $m = (\{a, b\}; \{(a, a), (a, b)\})$.
m may be viewed as the following graph.



$$m, \{x \to a\} \models E(x, x) \wedge \exists y.(E(x, y) \wedge \neg E(y, y))$$

Exercise 2.3

Give another model and assignment that satisfies the above formula

Example : counting

Example 2.6

Consider $\mathbf{S} = (\{\}, \{E/2\})$
The following sentence is false in all the models with one element domain

$$\forall x. \neg E(x, x) \wedge \exists x \exists y. E(x, y)$$

Exercise 2.4

a. Give a sentence that is true only in a model that has more than two elements in its domains
b. Give a sentence that is true only in infinite models
c. Does the negation of the sentence in b satisfies only finite models.

Topic 2.3

What is theory?

Theory

"Theory is a contemplative and rational type abstract ... thinking, or .... ."

- Wikipedia

Example 2.7

Scientific and economic theories

- *Newton's theory of Gravity*
- *Theory of evolution*
- *Theory of marginal utility*

Conspiracy theories

Example 2.8

- *9/11 is an inside job*
- *Trump is a Russian mole*
- $R + L = J$

They may sound silly.

However, they are still theories.

FOL has no knowledge

First-order logic(FOL) provides a grammar for rational abstract thinking.

However, FOL carries no knowledge of any subject matter.

It was not obvious. 16th century philosopher René Descartes tried to prove

$$\text{Inherent structure of logic} \Rightarrow \text{God exists.}$$

Theory crafting needs something more than logic

$$\text{Theory} = \text{Subject knowledge} + \text{FOL}$$

Now we will formally define theories in logic.

Decidablity and Complexity

- FOL validity is undecidable
- We restrict the problem in two ways
  - Theories : limits on the space of models
  - Logics/Fragments : limits on the structure of formulas

Topic 2.4

Theories

Defining theories

The subject knowledge can be expressed in the following two ways

- the set of acceptable models
- the set of valid sentences in the subject

Example 2.9

Model m with $D_m = \mathbb{N}$ is the only model we consider for the theory of natural numbers.

We can also define the theory using the set of valid sentences over natural numbers. e.g. $\forall x.\ x + 1 \not\approx 0$.

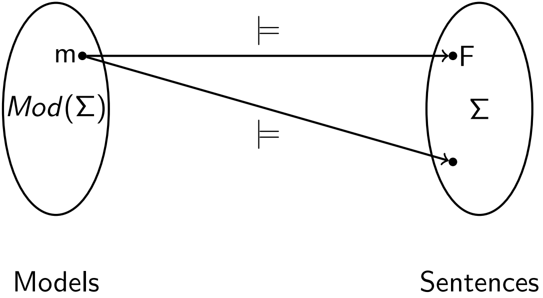
Now let us define this formally.

Definability of a class of models

Definition 2.10

For a set Σ of sentences in signature $\mathbf{S}$*, let* $Mod(\Sigma)$ *be a class of models such that*

$$Mod(\Sigma) = \{m \mid \textit{for all } F \in \Sigma.\ m \models F\}.$$



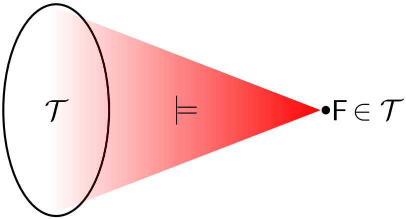
Theories

Definition 2.11

A theory $\mathcal{T}$ *is a set of sentences closed under implication, i.e.,*

$$\text{if } \mathcal{T} \models F \text{ then } F \in \mathcal{T}.$$

Abuse of notation, $\models$ is also used for implication



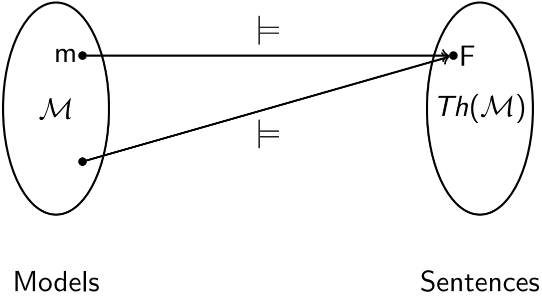
Sentences

Theory of Models

Definition 2.12

For a set $\mathcal{M}$ of models for signature **S**, *let* $Th(\mathcal{M})$ *be the set of* **S**-*sentences that are true in every model in* Σ,*i.e.,*

$$Th(\mathcal{M}) = \{F|\ \textit{for all } m \in \mathcal{M}.\ m \models F\}$$



Theory and models

Theorem 2.1
$Th(\mathcal{M})$ is a theory

Proof.
Consider F such that $Th(\mathcal{M}) \models F$.

Therefore, F is true in every model in $\mathcal{M}$.

Therefore,$F \in Th(\mathcal{M})$.

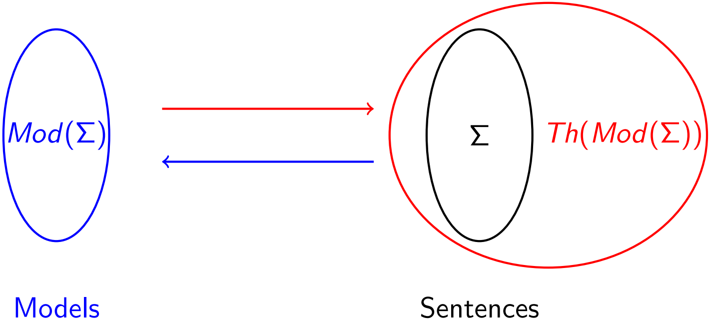
$Th(\mathcal{M})$ is closed under implication. $\square$

Consequences

Definition 2.13

For a set Σ of sentences, let $Cn(\Sigma)$ be the set of consequences of Σ, i.e.,

$$Cn(\Sigma) = Th(Mod(\Sigma)).$$



Exercise 2.5

Show for a theory $\mathcal{T}$, $\mathcal{T} = Cn(\mathcal{T})$.

Example: theory of lists

Example 2.10

Let us suppose our subject of interest is lists.

First we need to fix our signature.

We should be interested in the following functions and predicates

- :: - *constructor for extending a list*
- *head - function to pick head of a list*
- *tail - function to pick tail of a list*
- *atom - predicate that checks if something is constructed using* :: *or not*

The signature is

$$\mathbf{S} = (\{:: /2, head/1, tail/1\}, \{atom/1\})$$

Example: theory of lists

$\approx$ is equality within the logical syntax.

Let Σ consists of

1. $\forall x, y.\ head(x :: y) \approx x$
2. $\forall x, y.\ tail(x :: y) \approx y$
3. $\forall x.\ atom(x) \vee head(x) :: tail(x) \approx x$
4. $\forall x, y.\ \neg atom(x :: y)$

$\mathcal{T}_{list} = Th(Mod(\Sigma))$ is the set of valid sentences over lists.

The sentences in $\mathcal{T}_{list}$ may not be true on the non-list models.

Exercise 2.6

Why empty list is not explicitly encoded?

Axiomatizable

A set is decidable if there is an algorithm to check membership

Definition 2.14

A theory $\mathcal{T}$ is axiomatizable if there is a decidable set Σ s.t. $\mathcal{T} = Cn(\Sigma)$.

Definition 2.15

A theory $\mathcal{T}$ is finitely axiomatizable if there is a finite set Σ s.t. $\mathcal{T} = Cn(\Sigma)$.

$\mathcal{T}$-satisfiability,validity

Definition 2.16

A formula F *$\mathcal{T}$-satisfiable* *if there is model m s.t.* $m \models \mathcal{T} \cup \{F\}$.
$\mathcal{T}$-satisfiability is usually written as $m \models_{\mathcal{T}} F$.

Definition 2.17

A formula F is *$\mathcal{T}$-valid* *if* $\mathcal{T} \models F$.
$\mathcal{T}$-validity is usually written as $\models_{\mathcal{T}} F$.

Topic 2.5

Decidability

Decidable theories

Definition 2.18

Let $\mathcal{T} = Th(Mod(\Sigma))$. $\mathcal{T}$ is decidable if there is an algorithm that, for each sentence F, can decide (in finite time) whether $F \in \mathcal{T}$ or not.

Definition 2.19 (Equivalent to 2.18)

There is an algorithm that, for each sentence F, can decide (in finite time) whether $\Sigma \Rightarrow F$ or not.

Complexity of decidability

A theory may have axioms with some structure

and we can exploit the structure

to avoid the mindless enumeration of proofs and search them efficiently.

The dedicated search procedures are called decision procedures.

We often show decidability of a theory by providing a decision procedure.

Example decidable and undecidable theories

Example 2.11

Two arithmetics over natural numbers.

$$\text{Decidable}\quad \text{Presburger [3EXPTIME]}\left\{\begin{array}{l}\forall x \neg(x+1 \approx 0)\\ \forall x \forall y(x+1 \approx y+1 \Rightarrow x \approx y)\\ F(0) \wedge (\forall x(F(x) \Rightarrow F(x+1)) \Rightarrow \forall x F(x))\\ \forall x(x+0 \approx x)\\ \forall x \forall y(x+(y+1) \approx (x+y)+1)\end{array}\right.$$

$$\left.\begin{array}{l}\forall x \neg(x+1 \approx 0)\\ \forall x \forall y(x+1 \approx y+1 \Rightarrow x \approx y)\\ F(0) \wedge (\forall x(F(x) \Rightarrow F(x+1)) \Rightarrow \forall x F(x))\\ \forall x(x+0 \approx x)\\ \forall x \forall y(x+(y+1) \approx (x+y)+1)\\ \forall x, y(x \cdot 0 \approx 0)\\ \forall(x \cdot (y+1) \approx x \cdot y + x)\end{array}\right\}\text{Peano}\quad\text{Undecidable}$$

The third axiom is a schema.(It will be explained shortly!)

Exercise 2.7

Prove commutativity of $+$ in Presburger arithmetic.

Example: theory of equality $\mathcal{T}_E$

We have treated equality as part of FOL syntax and added special proof rules for it.

We can also treat equality as yet another predicate.

We can encode the behavior of equality as the set of following axioms.

1. $\forall x . x \approx x$
2. $\forall x, y . x \approx y \Rightarrow y \approx x$
3. $\forall x, y, z.\ x \approx y \wedge y \approx z \Rightarrow x \approx z$
4. for each $f/n \in \mathbf{F}$
   $\forall x_1, .., x_n, y_1, .., y_n.\ x_1 \approx y_1 \wedge .. \wedge x_n \approx y_n \Rightarrow f(x_1, .., x_n) \approx f(y_1, .., y_n)$
5. for each $P/n \in \mathbf{R}$
   $\forall x_1, .., x_n, y_1, .., y_n.\ x_1 \approx y_1 \wedge .. \wedge x_n \approx y_n \wedge P(x_1, .., x_n) \Rightarrow P(y_1, .., y_n)$

The last two axioms are called schema, because they define a set of axioms using a pattern.

Topic 2.6

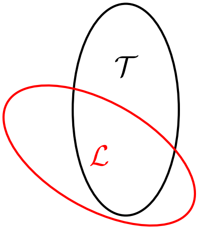
Fragments/Logics

Fragments

We may restrict $\mathcal{T}$ syntactically to achieve decidablity or low complexity.

Definition 2.20

Let $\mathcal{T}$ be a theory and $\mathcal{L}$ be a set of $\mathbf{S}$-sentences. $\mathcal{L}$ wrt $\mathcal{T}$ is decidable if there is an algorithm that takes $F \in \mathcal{L}$ as input and returns if $F \in \mathcal{T}$ or not.



Sentences

Example : fragments

Example 2.12 (Horn clauses)

$\mathcal{L} = \{\forall x.\, A_1(x) \wedge \cdots \wedge A_n(x) \Rightarrow B(x) | A_i \textit{ and } B \textit{ are atomic}\}$

Example 2.13 (Integer difference logic)

$\mathcal{L} =$ *linear arithmetic formulas that contain atoms with only two variables and with opposite signs [quadratic complexity].*

Quantifier-free fragments

Quantifier-free(QF) fragment has free variables that are assumed to be existentially quantified.(unlike FOL clauses!!)

Often, the quantifier-free fragments of theories have efficient decision procedures.

Example 2.14

The following is a QF formula in the theory of equality

$$f(x) \approx y \wedge (x \approx g(a, z) \vee h(x) \approx g(b))$$

QF of $\mathcal{T}$ of equality has an efficient decision procedure.
Otherwise, the theory is undecidable.

Example of logics

Some times the fragments are also referred as logics.

- quantifier-free theory of equality and uninterpreted function symbols (QF_EUF)
- quantifier-free theory of linear rational arithmetic (QF_LRA)
- quantifier-free theory of uninterpreted function and linear integer arithmetic (QF_UFLIA)

Topic 2.7

SMTLIB

Visit SMTLIB

http://smtlib.cs.uiowa.edu/

Topic 2.8

Problems

Axioms for predicates

Exercise 2.8

a. Write axioms for odd numbers in first order logic

b. Write axioms for even numbers in first order logic

c. Write axioms for divisibility in first order logic

d. Using the axioms write a resolution proof for the following statement

Between any two prime numbers greater than 2, there is an even number.

End of Lecture 2