Symbolic Artificial Intelligence (COMP3008)
Lecture 2: First-Order Logic
6 October 2025

Daniel Karapetyan
daniel.karapetyan@nottingham.ac.uk

First-Order Logic

First-Order Logic

- To formalise knowledge and reasoning, we need a formal language to describe knowledge
- There exist many formal languages to describe knowledge
- The first one we will study is the First-Order Logic (FOL)

Main characteristics of FOL

- FOL is a declarative language
- Designed to express knowledge
- Precise: each fact can only be true or false (unlike in fuzzy logic)
- Defines which strings are valid sentences (syntax) and what it means for them to be true (semantics)
- Unlike natural languages (e.g. English), FOL is completely abstract; it does not have the tools to describe real-world phenomena
  - The language uses abstract symbols, e.g. 'x', 'Dog2', . . .
  - The mapping of those symbols to real-world phenomena is up to the user

Examples

1. $\text{Dog}(\text{fido}) \lor \text{Cat}(\text{fido})$
2. $\text{Dog}(x) \rightarrow \text{Animal}(x)$
3. $\exists x.\text{Dog}(x) \land \text{Lovely}(x)$
4. $\forall x.\big(\text{Dog}(x) \lor \text{Cat}(x)\big) \rightarrow \text{Animal}(x)$

Types

As it concerns FOL, there are two ‘data types’:

- Boolean
- Domain of discourse (domain)

Operations such as $\wedge$, $\vee$ and $\neg$ work with Booleans

Domain elements are used in quantifiers $\forall$ and $\exists$ and equality $=$, and they can be passed as parameters

Alphabet

Logical symbols:

- Quantifiers $\forall$ and $\exists$ for domain elements
- Logical connectives $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$
- Equality $=$ to compare domain elements
- Punctuation '(', ')' and other brackets for readability: '[', ']', '{', '}'
- Variables for domain elements – usually lower case letters, optionally provided with indices, e.g. x, y_1, $z_{3,8}$, ...

Non-logical symbols:

- Predicate symbols: 'functions' that return Booleans; usually begin with capital letters, e.g. $P(x)$, Q, $IsTasty(y)$
- Function symbols: functions that return domain elements; usually begin with lower-case letters, e.g. $f(x)$, g, $madeOf(y)$

Logical connectives

$\neg A$ is negation (logical 'not'): it is true iff A is false.

$A \wedge B$ is conjunction (logical 'and'): both A and B are true.

$A \vee B$ is disjunction (logical 'or'): at least one of A and B is true.

$A \rightarrow B$ can be read as 'if A then B': if A is true then B must also be true. $A \rightarrow B$ is short for $\neg A \vee B$. Sometimes denoted as $\supset$.

$A \leftrightarrow B$ is 'if and only if': both A and B are either false or true. $A \leftrightarrow B$ is short for $(A \rightarrow B) \wedge (B \rightarrow A)$. Sometimes denoted as $\equiv$.

A	$\neg A$
False	True
True	False

A	B	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
False	False	False	False	True	True
False	True	False	True	True	False
True	False	False	True	False	False
True	True	True	True	True	True

Precedence

We will use these rules (different sources may give different rules)

- Apply all the functions, predicates, and '='
- Apply $\neg$ to as little as possible

$$\neg A \land B \qquad \text{means} \qquad (\neg A) \land B$$

- Then apply the quantifiers to as little as possible

$$\forall x.A(x) \land B(x) \land C \qquad \text{means} \qquad \big(\forall x.A(x) \land B(x)\big) \land C$$

- Then apply $\land$ to as little as possible

$$A \to B \land \neg C \land D \qquad \text{means} \qquad A \to \Big(\big(B \land (\neg C)\big) \land D\Big)$$

- Then apply $\lor$ to as little as possible
- Then apply $\to$ to as little as possible
- Then apply $\leftrightarrow$ to as little as possible

Note about quantifiers and precedence

Consider this expression:

$$\forall x. A(x) \wedge B(x)$$

One could interpreted it as

$$\big(\forall x. A(x)\big) \wedge B(x)$$

but it clearly means

$$\forall x. \big(A(x) \wedge B(x)\big)$$

as otherwise x in $B(x)$ would not be defined

De Morgan's laws and their generalisation

De Morgan's laws:

$$\neg(A_1 \wedge A_2 \wedge \cdots \wedge A_n) = \neg A_1 \vee \neg A_2 \vee \cdots \vee \neg A_n$$

$$\neg(A_1 \vee A_2 \vee \cdots \vee A_n) = \neg A_1 \wedge \neg A_2 \wedge \cdots \wedge \neg A_n$$

Generalisations:

$$\neg \forall x. A(x) = \exists x. \neg A(x)$$

$$\neg \exists x. A(x) = \forall x. \neg A(x)$$

Arity of function and predicate symbols

- Each predicate and function symbol has *arity*, i.e. the number of parameters
- Arity can be 0, 1, 2, etc.
- Predicates with arity 0 are called *propositional variables*; they take value ‘true’ or ‘false’
- Functions with arity 0 are called *constant symbols*; they stand for domain elements such as ‘COMP3008 module’, ‘Number 73’, ‘Cat’, etc.
- For predicates and functions of arity 0, we skip the parentheses: P, f, etc. instead of $P()$, $f()$, etc.

Example of an FOL formula

$$\exists x. x = f(x) \wedge x = g(x)$$

The order of operations is as follows:

$$\exists x. \big((x = f(x)) \wedge (x = g(x))\big)$$

It reads as

'There exists x such that x is equal to $f(x)$ and x is equal to $g(x)$, where f and g are functions of arity 1'

For example, if our domain of discourse is the set of all integers, $f(x) = 2x$ and $g(x) = x^2$, then the above statement is true (why?)

Well-Formed Formulas

- Not every sequence of alphabet symbols is a valid expression, e.g.

$$\neg \wedge x \forall$$

  has no meaning
  - Similarly, not every sequence of words in English has a meaning, e.g. ‘Run 15 laptop or’ is grammatically incorrect
- Valid formulas are called Well-Formed Formulas (WFF)
- WFFs are defined inductively via terms and formulas:
  - *Term* is an expression that ‘returns’ a domain element
  - *Formula* is an expression that ‘returns’ Boolean

Formation rules (grammar)

Formation rules inductively define terms and formulas

A term is either

- A variable
- Any expression in the form of $f(t_1, t_2, \ldots, t_n)$, where t_i is a term and f is a function symbol of arity n

A formula is any expression in the following forms:

- $P(t_1, t_2, \ldots, t_n)$, where t_i is a term and P is a predicate symbol of arity n
- $\neg\phi$, where ϕ is a formula
- $t_1 = t_2$, where t_1 and t_2 are terms
- $\phi \land \psi$, $\phi \lor \psi$, $\phi \to \psi$ and $\phi \leftrightarrow \psi$, where ϕ and ψ are formulas
- $\forall x.\phi$ and $\exists x.\phi$, where x is a variable name and ϕ is a formula

Variable scope

- Variables in FOL can be *free* or *bound*
  - Bound variables are defined by quantifiers, e.g. $\forall x.P(x)$
  - Free variables are defined outside the formula (their values are given), e.g. $P(x) \lor Q(x)$
- The *scope* of a variable is the part of the formula where it can be used
- Variables defined by a quantifier are only accessible within the scope of that quantifier
- In this example, x is free and y is bound:

$$\overbrace{P(x) \lor \underbrace{\exists y.Q(y) \land R(x,y)}_{\text{Scope of } y}}^{\text{Scope of } x}$$

- If the name of a variable is reused, it refers to the innermost definition, e.g. $P(x) \lor \exists x.Q(x)$ uses two variables x, and $Q(x)$ refers to the bound x

Sentences

Sentences are special cases of formulas:

A WFF that does not have free variables is called *sentence*

To convert a formula with free variables into a sentence, we need to substitute values for every free variable in that formula

Semantics

Semantics

- Consider the following sentence:

$$\forall x. PM(x) \rightarrow MP(x) \wedge Popular(x)$$

  What does it mean?
  Is it true?

Semantics

- Consider the following sentence:

$$\forall x.PM(x) \rightarrow MP(x) \wedge Popular(x)$$

  What does it mean?
  Is it true?
- Cannot answer these questions: we don't know what the non-logical symbols mean and how they behave
- Without precise interpretation of each non-logical symbol, cannot represent knowledge
- Semantics is concerned with the meaning of sentences (but not the mapping to real world)
- *Logical interpretation* defines the semantics of FOL
  - It specifies how non-logical symbols behave

Interpretation

Interpretation $\mathfrak{I}$ is a pair $\langle \mathcal{D}, \mathcal{I} \rangle$, where

$\mathcal{D}$ is a non-empty set called *domain of discourse* (or *domain*), and

$\mathcal{I}$ gives mapping of every non-logical symbol

$\mathcal{I}$ is defined as follows:

- For a functional symbol *f* of arity *n*
  $\mathcal{I}[f] : \underbrace{\mathcal{D} \times \mathcal{D} \times \mathcal{D} \times \cdots \times \mathcal{D}}_{n \text{ times}} \rightarrow \mathcal{D}$
  - Recall that constant symbol is a functional symbol of arity 0; then interpretation of a constant symbol *a* is defined as $\mathcal{I}[a] \in \mathcal{D}$
- For a predicate symbol *P* of arity *n*
  $\mathcal{I}[P] : \underbrace{\mathcal{D} \times \mathcal{D} \times \mathcal{D} \times \cdots \times \mathcal{D}}_{n \text{ times}} \rightarrow \{\text{True}, \text{False}\}$
- $\mathcal{I}[P]$ can also be seen as a relation: $\mathcal{I}[P] \subseteq \underbrace{\mathcal{D} \times \mathcal{D} \times \mathcal{D} \times \cdots \times \mathcal{D}}_{n \text{ times}}$

Sufficiency of logical interpretation

> Interpretation $\mathfrak{I} = \langle \mathcal{D}, \mathcal{I} \rangle$ is all you need to know about non-logical symbols to evaluate an FOL sentence as true or false

In other words, to say if a sentence is true, all you need is to know is

- The set of elements in the domain
- Which inputs satisfy predicate symbols
- Which element does each function return for every possible input

Question

Evaluate sentence

$$\exists x.A(x) \wedge \forall y.B(x, y)$$

given the following interpretation $\mathfrak{I}$:

$$\mathcal{D} = \{1, 2, 3\}$$

x	$A(x)$
1	True
2	False
3	True

x	y	$B(x, y)$
1	1	False
1	2	True
1	3	False
2	1	True
2	2	True
2	3	True
3	1	True
3	2	False
3	3	True

Answer

We need to find at least one value of $x \in \mathcal{D}$ such that $A(x) = \text{True}$ and $B(x, y) = \text{True}$ for every $y \in \mathcal{D}$.

Since $A(2) = \text{False}$, we can exclude $x = 2$.

For $x = 1$, predicate $B(x, y)$ evaluates to False for $y = 1$, hence $x = 1$ does not satisfy the sentence.

For $x = 3$, predicate $B(x, y)$ evaluates to False for $y = 2$, hence $x = 1$ does not satisfy the sentence.

We conclude that interpretation $\mathfrak{I}$ does not satisfy this sentence; the sentence evaluates to False.

Satisfiability

Satisfaction

Having interpretation $\mathfrak{I}$, we can determine if a sentence α is *satisfied* by $\mathfrak{I}$:

$$\mathfrak{I} \models \alpha$$

If α is a formula with free variables, satisfiability then depends on the variable assignment μ:

$$\mathfrak{I}, \mu \models \alpha$$

(we say that α is satisfied by $\mathfrak{I}$ and μ)

Same notation can be used for a set S of sentences/formulas:

$$\mathfrak{I} \models S \qquad \mathfrak{I}, \mu \models S$$

We can also use symbol $\not\models$ to say that a formula is not satisfied, i.e. it is false under interpretation $\mathfrak{I}$ (and variable assignment μ)

Rules of interpretation

Let P be a predicate symbol of arity n, t_i be a term, $d_i = \mathfrak{I}[t_i]$ (or $d_i = \mu[t_i]$ for free variables), α and β be formulas, and x be a variable

Then the rules of interpretation are as follows

- $\mathfrak{I}, \mu \models P(t_1, t_2, \dots, t_n)$ iff $\langle d_1, d_2, \dots, d_n \rangle \in \mathcal{I}[P]$
- $\mathfrak{I}, \mu \models (t_1 = t_2)$ iff d_1 is the same object as d_2
- $\mathfrak{I}, \mu \models \neg\alpha$ iff $\mathfrak{I}, \mu \not\models \alpha$
- $\mathfrak{I}, \mu \models (\alpha \wedge \beta)$ iff $\mathfrak{I}, \mu \models \alpha$ and $\mathfrak{I}, \mu \models \beta$
- $\mathfrak{I}, \mu \models (\alpha \vee \beta)$ iff $\mathfrak{I}, \mu \models \alpha$ or $\mathfrak{I}, \mu \models \beta$
- $\mathfrak{I}, \mu \models \exists x.\alpha$ iff for some $d \in \mathcal{D}$ holds $\mathfrak{I}, \mu \models \alpha$, where $\mu[x] = d$
- $\mathfrak{I}, \mu \models \forall x.\alpha$ iff for every $d \in \mathcal{D}$ holds $\mathfrak{I}, \mu \models \alpha$, where $\mu[x] = d$

Logical entailment

Satisfaction of a sentence generally depends on interpretation

Instead of defining an interpretation, we can put ‘constraints’ on the interpretation, e.g.

$$\text{if } \mathfrak{I} \models \alpha \text{ then } \mathfrak{I} \models \neg(\beta \wedge \neg\alpha) \text{ for any } \mathfrak{I}$$

Let *S* be a set of sentences and α be a sentence

If every interpretation $\mathfrak{I}$ that satisfies *S* also satisfies α then we say that

- α is a *logical consequence* of *S*
- *S logically entails* α
- $S \models \alpha$

Classification of sentences

Sentence α is

Unsatisfiable	if no interpretation satisfies α
Satisfiable	if at least one interpretation satisfies α
Not valid	if at least one interpretation does not satisfy α
Valid	if every interpretation satisfies α

These definitions can be applied to a set of sentences $S = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$; just let $\alpha = \alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n$

We can also allow free variables; e.g., formula α is satisfiable if there exists an interpretation $\mathfrak{I}$ and a variable assignment μ such that $\mathfrak{I}, \mu \models \alpha$

Logical validity

A sentence α is *logically valid* (*valid*) if it is satisfied by every interpretation

An equivalent definition: a formula is valid if it is a logical consequence of an empty set:

$$S \models \alpha, \text{ where } S = \emptyset$$

We write '$\models \alpha$' if α is valid and '$\not\models \alpha$' for not valid

For example

$$\models \alpha \vee \neg\alpha$$

Observe that entailment can be reduced to validity:

$$\text{if} \quad \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \models \beta \quad \text{then} \quad \models (\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \rightarrow \beta$$

Unsatisfiability

Sentence α is *unsatisfiable* if there is no $\mathfrak{I}$ such that $\mathfrak{I} \models \alpha$

E.g. $\alpha = \alpha' \wedge \neg\alpha'$ is unsatisfiable

Unsatisfiability can be expressed using entailment:

- Let us define FALSE as $\exists x.\neg(x = x)$
  (you can use any other unsatisfiable sentence)
- Recall that S entails a sentence β if every interpretation $\mathfrak{I}$ that satisfies S also satisfies β
- If S is unsatisfiable, there is no interpretation that satisfies S, hence any formula including FALSE is a consequence of S
- $\boxed{\text{A common way of showing that } S \text{ is unsatisfiable is } S \models \text{FALSE}}$

Summary

Concept	Notation	Definition
Satisfaction	$\mathfrak{I}, \mu \models \alpha$	α is satisfied by the interpretation
Entailment	$S \models \alpha$	α is a logical consequence of S (it is satisfied by every interpretation that satisfies S)
Satisfiability	(no notation)	there exists an interpretation that satisfies α
Validity	$\models \alpha$	α is satisfied by every interpretation
Unsatisfiability	$S \models \text{FALSE}$	there exists no interpretation that would satisfy S

In-class exercises

Z3 program (SMT2)

```
(declare-const A Bool)
(declare-const B Bool)
(declare-const C Bool)
(assert (or A B))
(assert (or B C))
(check-sat)
(get-model)
```

Z3 program (SMT2)

```
(declare-const A Bool)
(declare-const B Bool)
(declare-const C Bool)
(assert (or A B))
(assert (or B C))
(check-sat)
(get-model)
```

1. Declare propositional variables (predicates of arity 0) A, B and C
2. Express our knowledge using FOL: $A \vee B$ and $B \vee C$
3. Ask Z3 to verify whether our program is satisfiable
4. Ask to print out an interpretation that satisfies our program

Z3 output

```
sat
(model
   (define-fun A () Bool
     false)
   (define-fun B () Bool
     true)
   (define-fun C () Bool
     false)
)
```

This means that $A = \text{False}$, $B = \text{True}$ and $C = \text{False}$

Are there any other models that satisfy our program?

Z3 solver

- In terms of FOL, Z3 program is a sentence:

$$S = S_1 \wedge S_2 \wedge \dots$$

  where S_1, S_2, ... are assertions

- Z3 can only test satisfiability
  - If the program is satisfiable, Z3 finds a 'model' (i.e. interpretation):

$$\text{find interpretation } \mathfrak{I} \text{ such that } \mathfrak{I} \models S$$

  - If the program is unsatisfiable, Z3 can only return 'unsat'

Question

What does this mean?

$$S \not\models \alpha$$

Question

What does this mean?

$$S \not\models \alpha$$

It means 'does not entail':

Entails:	Every interpretation $\mathfrak{I}$ that satisfies S also satisfies α
Does not entail:	There exists at least one interpretation $\mathfrak{I}$ that satisfies S but does not satisfy α

Question

Consider the following FOL sentence:

$$A \lor \exists x . B(x)$$

1. Is it satisfiable?
2. Is it valid?
3. Is it not valid?
4. Is it unsatisfiable?

Approach

Let α be our sentence

Answer two questions:

1. Is there an interpretation $\mathfrak{I}_{\text{sat}}$ that satisfies α?
2. Is there an interpretation $\mathfrak{I}_{\text{not-sat}}$ that does not satisfy α?

α is satisfiable if $\mathfrak{I}_{\text{sat}}$ exists
α is unsatisfiable if $\mathfrak{I}_{\text{sat}}$ does not exist
α is not valid if $\mathfrak{I}_{\text{non-sat}}$ exists
α is valid if $\mathfrak{I}_{\text{non-sat}}$ does not exist

Answer

$$\alpha : A \vee \exists x.B(x)$$

1. Is there an interpretation $\mathfrak{I}_{\mathrm{sat}}$ that satisfies α?

Answer

$$\alpha : A \vee \exists x.B(x)$$

1. Is there an interpretation $\mathfrak{I}_{\text{sat}}$ that satisfies α?
   Let $A = $ TRUE and $B(x) = $ TRUE

Answer

$$\alpha : A \lor \exists x.B(x)$$

1. Is there an interpretation $\mathfrak{I}_{\text{sat}}$ that satisfies α?
   Let $A =$ TRUE and $B(x) =$ TRUE
2. Is there an interpretation $\mathfrak{I}_{\text{not-sat}}$ that does not satisfy α?

Answer

$$\alpha : A \vee \exists x.B(x)$$

1. Is there an interpretation $\mathfrak{I}_{\text{sat}}$ that satisfies α?
   Let $A = \text{TRUE}$ and $B(x) = \text{TRUE}$
2. Is there an interpretation $\mathfrak{I}_{\text{not-sat}}$ that does not satisfy α?
   Let $A = \text{FALSE}$ and $B(x) = \text{FALSE}$

Answer

$$\alpha : A \lor \exists x.B(x)$$

1. Is there an interpretation $\mathfrak{I}_{\text{sat}}$ that satisfies α?
   Let $A = \text{TRUE}$ and $B(x) = \text{TRUE}$
2. Is there an interpretation $\mathfrak{I}_{\text{not-sat}}$ that does not satisfy α?
   Let $A = \text{FALSE}$ and $B(x) = \text{FALSE}$

α is satisfiable if $\mathfrak{I}_{\text{sat}}$ exists
α is unsatisfiable if $\mathfrak{I}_{\text{sat}}$ does not exist
α is not valid if $\mathfrak{I}_{\text{non-sat}}$ exists
α is valid if $\mathfrak{I}_{\text{non-sat}}$ does not exist

Conclusion: α is satisfiable and not valid

Self-study exercises

Exercise 1

Consider the following FOL formula:

$$\forall x \forall y.(P(f(x), y) \rightarrow f(x) = x \lor \neg P(x, y))$$

1. Place parenthesis to show the order of the operations
2. Translate it into English
3. Suggest a real-world example which fits the above formula

Exercise 2

Let x_1, x_2 and x_3 be variables, f and g be function symbols and A, B and C be predicate symbols. Assume that the arity of function and predicate symbols is appropriate for the context.

Which of the following are WFFs?

1. $f(x_1) \wedge g(x_2)$
2. $A \vee (B \wedge \exists x_1 . C(x_1))$
3. $B(A(x_1))$
4. $\exists A. (f(x_1) = x_2) \wedge A$
5. $[\forall x_1 \exists x_2 . A(x_1, x_2) \wedge \neg B(x_2)] \vee [\exists x_1 \exists x_2 \exists x_3 . (f(x_1) = g(x_1, x_2, x_3)) \wedge (A(x_1) = B(x_2))]$

Exercise 3: Background

It is easy to come up with simple examples of entailment, e.g.

$$\{\forall x.\neg(Dog(x) \wedge Cat(x)),\ Dog(oscar)\} \models \neg Cat(oscar)$$

We have not yet studied how to prove such an entailment, yet the proof is intuitive:

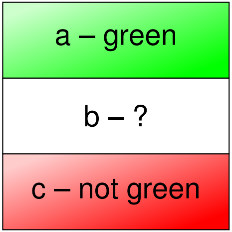
- Let $x = oscar$
- From the first sentence: $\neg(Dog(oscar) \wedge Cat(oscar))$
- Substitute the second sentence: $\neg(\text{TRUE} \wedge Cat(oscar))$
- Simplify: $\neg Cat(oscar)$

If such an example is scaled up, with hundreds of sentences, an automated reasoning system may be very handy

However, examples do not have to be large to be non-trivial

Exercise 3: Example of a less trivial entailment

There are three blocks: *a*, *b* and *c*, as shown in the figure. We know that *a* is green and *c* is not green. We do not know the colour of *b*. Question: is there a green block on top of a non-green block?



Our knowledge can be expressed using the following FOL sentences:

1. $OnTop(a, b)$
2. $OnTop(b, c)$
3. $Green(a)$
4. $\neg Green(c)$

Hypothesis: $\exists x \exists y . OnTop(x, y) \wedge Green(x) \wedge \neg Green(y)$

Exercise 3: Exercise

Using the FOL language, prove or disprove the hypothesis in the previous slide

You need to demonstrate that the sentences that express our knowledge entail the hypothesis

Hint: consider two complementary families of interpretations

Self-study exercises – answers

Exercise 1: Answers

The original formula is as follows

$$\forall x \forall y.(P(f(x), y) \supset f(x) = x \vee \neg P(x, y))$$

1. After placing parenthesis:

$$\forall x \forall y.\Big(P(f(x), y) \supset \big((f(x) = x) \vee (\neg P(x, y))\big)\Big)$$

2. Translating into English: "For all x and y, if $P(f(x), y)$ then $f(x)$ is equal to x or $P(x, y)$ is false."
3. An example of an interpretation (you may have a completely different example): Let $f(x)$ be the closest friend of x and $P(x, y)$ be true if x and y are married. Then the sentence reads as follows: for any x and y, if the closest friend of x is married to y then either x is their own closest friend, or x is not married to y.

Exercise 2: Answers

1. $f(x_1) \wedge g(x_2)$
   This is not a WFF; f and g are functions; logical ‘and’ cannot be applied to terms
2. $A \vee (B \wedge \exists x_1 . C(x_1))$
   This is a WFF
3. $B(A(x_1))$
   This is not a WFF; $A(x_1)$ is a formula whereas the parameter of B as to be a term
4. $\exists A. (f(x_1) = x_2) \wedge A$
   This is not a WFF; A is a predicate whereas quantifiers require variables
5. $[\forall x_1 \exists x_2 . A(x_1, x_2) \wedge \neg B(x_2)] \vee [\exists x_1 \exists x_2 \exists x_3 . (f(x_1) = g(x_1, x_2, x_3)) \wedge (A(x_1) = B(x_2))]$
   This is not a WFF; equality connective in $A(x_1) = B(x_2)$ cannot be used between predicates; instead, we had to use ‘$\equiv$’ for ‘if and only if’

Exercise 3: Answer

There are two cases: (1) $\mathfrak{I} \models \mathit{Green}(b)$ and (2) $\mathfrak{I} \models \neg \mathit{Green}(b)$

1. If $\mathfrak{I} \models \mathit{Green}(b)$ then for $x = b$ and $y = c$, the following holds: $\mathfrak{I} \models \mathit{OnTop}(x, y) \wedge \mathit{Green}(x) \wedge \neg \mathit{Green}(y)$, hence the hypothesis is correct
2. If $\mathfrak{I} \models \neg \mathit{Green}(b)$ then for $x = a$ and $y = b$, the following holds: $\mathfrak{I} \models \mathit{OnTop}(x, y) \wedge \mathit{Green}(x) \wedge \neg \mathit{Green}(y)$, hence the hypothesis is correct

Conclusion: for any interpretation $\mathfrak{I}$, the hypothesis

$$\mathrm{KB} \models \exists x \exists y . \mathit{OnTop}(x, y) \wedge \mathit{Green}(x) \wedge \neg \mathit{Green}(y)$$

is correct; there is a green block right on top of a non-green block