INTRODUCTION TO LOGIC

5 The Semantics of Predicate Logic

Volker Halbach

We could forget about philosophy. Settle down and maybe get into semantics.

Woody Allen, *Mr. Big*

Outline

1. Validity.
2. Semantics for simple English sentences.
3. Semantics for $\mathcal{L}_2$-formulae.
4. $\mathcal{L}_2$-structures.

Argument	Valid
(1) Zeno is a tortoise.	
(2) All tortoises are toothless.	
Therefore, (C) Zeno is toothless.	

Argument **Valid**

(1) Zeno is a tortoise.
(2) All tortoises are toothless.
Therefore, (C) Zeno is toothless.

Formalisation

(1) Pa
(2) $\forall x\,(Px \to Qx)$
(C) Qa

Dictionary: a: Zeno. P:…is a tortoise. Q: …is toothless

Argument **Valid**

(1) Zeno is a tortoise.
(2) All tortoises are toothless.
Therefore, (C) Zeno is toothless.

Formalisation

(1) Pa
(2) $\forall x\,(Px \to Qx)$
(C) Qa

Dictionary: a: Zeno. P:…is a tortoise. Q: …is toothless

What is it for this $\mathcal{L}_2$-argument to be valid?

Validity

Recall the definition of validity for $\mathcal{L}_1$.

Validity

Recall the definition of validity for $\mathcal{L}_1$.
Let Γ be a set of sentences of $\mathcal{L}_1$ and ϕ a sentence of $\mathcal{L}_1$.

Validity

Recall the definition of validity for $\mathcal{L}_1$.
Let Γ be a set of sentences of $\mathcal{L}_1$ and ϕ a sentence of $\mathcal{L}_1$.

Definition

The argument with all sentences in Γ as premisses and ϕ as conclusion is *valid* iff there is no $\mathcal{L}_1$-structure under which:

(i) all sentences in Γ are true; and

(ii) ϕ is false.

Validity

Recall the definition of validity for $\mathcal{L}_1$.
Let Γ be a set of sentences of $\mathcal{L}_1$ and ϕ a sentence of $\mathcal{L}_1$.

Definition

The argument with all sentences in Γ as premisses and ϕ as conclusion is *valid* iff there is no $\mathcal{L}_1$-structure under which:

(i) all sentences in Γ are true; and

(ii) ϕ is false.

We use an exactly analogous definition for $\mathcal{L}_2$, replacing '$\mathcal{L}_1$' everywhere above with '$\mathcal{L}_2$'.

Validity

Recall the definition of validity for $\mathcal{L}_1$.
Let Γ be a set of sentences of $\mathcal{L}_2$ and ϕ a sentence of $\mathcal{L}_2$.

Definition

The argument with all sentences in Γ as premisses and ϕ as conclusion is *valid* iff there is no $\mathcal{L}_2$-structure under which:

(i) all sentences in Γ are true; and

(ii) ϕ is false.

We use an exactly analogous definition for $\mathcal{L}_2$, replacing '$\mathcal{L}_1$' everywhere above with '$\mathcal{L}_2$'.

Validity

Recall the definition of validity for $\mathcal{L}_1$.
Let Γ be a set of sentences of $\mathcal{L}_2$ and ϕ a sentence of $\mathcal{L}_2$.

Definition

The argument with all sentences in Γ as premisses and ϕ as conclusion is *valid* iff there is no $\mathcal{L}_2$**-structure** under which:

(i) all sentences in Γ are **true**; and

(ii) ϕ is **false**.

We use an exactly analogous definition for $\mathcal{L}_2$, replacing '$\mathcal{L}_1$' everywhere above with '$\mathcal{L}_2$'.
It remains to define: *$\mathcal{L}_2$-structure*, *truth in an $\mathcal{L}_2$-structure*

Structures

Structures interpret non-logical expressions.

Structures

Structures interpret non-logical expressions.

$\mathcal{L}_1$-structures

- Non-logical expressions in $\mathcal{L}_1$: $P, Q, R, \ldots$.

Structures

Structures interpret non-logical expressions.

$\mathcal{L}_1$-structures

- Non-logical expressions in $\mathcal{L}_1$: $P, Q, R, \ldots$.
- An $\mathcal{L}_1$-structure $\mathcal{A}$ assigns each sentence letter a semantic value (specifically, a truth-value: T or F).

Structures

Structures interpret non-logical expressions.

$\mathcal{L}_1$-structures

- Non-logical expressions in $\mathcal{L}_1$: $P, Q, R, \ldots$.
- An $\mathcal{L}_1$-structure $\mathcal{A}$ assigns each sentence letter a semantic value (specifically, a truth-value: T or F).

$\mathcal{L}_2$ is a richer language. This calls for richer structures.

Structures

Structures interpret non-logical expressions.

$\mathcal{L}_1$-structures

- Non-logical expressions in $\mathcal{L}_1$: $P, Q, R, \ldots$.
- An $\mathcal{L}_1$-structure $\mathcal{A}$ assigns each sentence letter a semantic value (specifically, a truth-value: T or F).

$\mathcal{L}_2$ is a richer language. This calls for richer structures.

$\mathcal{L}_2$-structures

- Non-logical expressions: $P^1, Q^1, R^1, \ldots$

Structures

Structures interpret non-logical expressions.

$\mathcal{L}_1$-structures

- Non-logical expressions in $\mathcal{L}_1$: $P, Q, R, \ldots$.
- An $\mathcal{L}_1$-structure $\mathcal{A}$ assigns each sentence letter a semantic value (specifically, a truth-value: T or F).

$\mathcal{L}_2$ is a richer language. This calls for richer structures.

$\mathcal{L}_2$-structures

- Non-logical expressions: $$\begin{array}{l} P^1, Q^1, R^1, \ldots \\ P^2, Q^2, R^2, \ldots \\ \quad\vdots \end{array}$$

Structures

Structures interpret non-logical expressions.

$\mathcal{L}_1$-structures

- Non-logical expressions in $\mathcal{L}_1$: $P, Q, R, \ldots$.
- An $\mathcal{L}_1$-structure $\mathcal{A}$ assigns each sentence letter a semantic value (specifically, a truth-value: T or F).

$\mathcal{L}_2$ is a richer language. This calls for richer structures.

$\mathcal{L}_2$-structures

- Non-logical expressions: $\begin{array}{l} P^1, Q^1, R^1, \ldots \\ P^2, Q^2, R^2, \ldots \\ \quad\vdots \\ a, b, c, \ldots \end{array}$

Structures

Structures interpret non-logical expressions.

$\mathcal{L}_1$-structures

- Non-logical expressions in $\mathcal{L}_1$: $P, Q, R, \ldots$.
- An $\mathcal{L}_1$-structure $\mathcal{A}$ assigns each sentence letter a semantic value (specifically, a truth-value: T or F).

$\mathcal{L}_2$ is a richer language. This calls for richer structures.

$\mathcal{L}_2$-structures

- Non-logical expressions: $$\begin{array}{l} P^1, Q^1, R^1, \ldots \\ P^2, Q^2, R^2, \ldots \\ \quad\vdots \\ a, b, c, \ldots \end{array}$$
- An $\mathcal{L}_2$-structure $\mathcal{A}$ assigns each predicate and constant a semantic value (specifically, what?).

I could present all definitions on 4 slides. Most slides just help to motivate these definitions.

Semantics in English

Start with a semantics for simple English sentences.

Semantics in English

Start with a semantics for simple English sentences.

'Maggie Smith is an actor.'

Semantics in English

Start with a semantics for simple English sentences.

'Maggie Smith is an actor.'

The sentence is true

Semantics in English

Start with a semantics for simple English sentences.

'Maggie Smith is an actor.'

The sentence is true (i.e.: its semantic value is: T).

Semantics in English

Start with a semantics for simple English sentences.

'Maggie Smith is an actor.'

The sentence is true (i.e.: its semantic value is: T).
…because of the relationship between the semantic values of its constituents.

Semantics in English

Start with a semantics for simple English sentences.

‘Maggie Smith is an actor.’

The sentence is true (i.e.: its semantic value is: T).
…because of the relationship between the semantic values of its constituents.

expression	*semantic value*
‘Maggie Smith ’	Maggie Smith
‘is an actor’	the property of *being an actor*

Semantics in English

Start with a semantics for simple English sentences.

'Maggie Smith is an actor.'

The sentence is true (i.e.: its semantic value is: T).
…because of the relationship between the semantic values of its constituents.

expression	*semantic value*
'Maggie Smith '	Maggie Smith
'is an actor'	the property of *being an actor*

…because Maggie Smith has the property of *being an actor*.

Semantics in English

Start with a semantics for simple English sentences.

'Maggie Smith is an actor.'

The sentence is true (i.e.: its semantic value is: T).
...because of the relationship between the semantic values of its constituents.

expression	*semantic value*
'Maggie Smith '	Maggie Smith
'is an actor'	the property of *being an actor*

...because Maggie Smith has the property of *being an actor*.
...because |'Maggie Smith '| has |'is an actor'|.

Notation

When e is an expression, we write $|e|$ for its semantic value.

Similarly:

'Mary likes Maggie Smith' is true iff
Mary stands in the relation of *liking* to Maggie Smith

Similarly:

'Mary likes Maggie Smith' is true iff
Mary stands in the relation of *liking* to Maggie Smith

In other words:

|'Mary likes Maggie Smith'| = T iff
|'Mary'| stands in |'likes'| to |'Maggie Smith'|

Semantic values for English expressions

expression	*semantic value*
designator	object
unary predicate	property (alias: unary relation)
binary predicate	binary relation

Semantic values for English expressions

expression	*semantic value*
designator	object
unary predicate	property (alias: unary relation)
binary predicate	binary relation

Examples

- |‘Maggie Smith’| = Maggie Smith
- |‘is an actor’| = the property of *being an actor*
- |‘likes’| = the relation of *liking*

Semantic values for English expressions

expression	*semantic value*
designator	object
unary predicate	property (alias: unary relation)
binary predicate	binary relation

Examples

- |'Maggie Smith'| = Maggie Smith
- |'is an actor'| = the property of *being an actor*
- |'likes'| = the relation of *liking*

We'll take this one step further, by saying more about properties and relations.

Properties

For the purposes here, we identify properties with sets.

Properties

For the purposes here, we identify properties with sets.

Property (alias: unary relation)

A *unary relation* P is a set of zero or more objects.

Properties

For the purposes here, we identify properties with sets.

Property (alias: unary relation)

A *unary relation* P is a set of zero or more objects.

Specifically, P is the set of objects that have the property.

Properties

For the purposes here, we identify properties with sets.

Property (alias: unary relation)

A *unary relation* P is a set of zero or more objects.

Specifically, P is the set of objects that have the property.

Informally: $d \in P$ indicates that d has property P.

Properties

For the purposes here, we identify properties with sets.

Property (alias: unary relation)

A *unary relation* P is a set of zero or more objects.

Specifically, P is the set of objects that have the property.

Informally: $d \in P$ indicates that d has property P.

Example

The property of *being an actor*

Properties

For the purposes here, we identify properties with sets.

Property (alias: unary relation)

A *unary relation* P is a set of zero or more objects.

Specifically, P is the set of objects that have the property.

Informally: $d \in P$ indicates that d has property P.

Example

The property of *being an actor*

$$\begin{aligned} &= \text{the set of actors} \\ &= \{d : d \text{ is an actor}\} \\ &= \{\text{Emma Stone, B. Cumberbatch}, \ldots\} \end{aligned}$$

Relations

Recall that we identify binary relations with sets of pairs.

Relations

Recall that we identify binary relations with sets of pairs.

Binary relation

A *binary relation* R is a set of zero or more pairs of objects.

Relations

Recall that we identify binary relations with sets of pairs.

Binary relation

A *binary relation* R is a set of zero or more pairs of objects.

R is the set of pairs $\langle d, e\rangle$ such that d stands in R to e.

Relations

Recall that we identify binary relations with sets of pairs.

Binary relation

A *binary relation* R is a set of zero or more pairs of objects.

R is the set of pairs $\langle d, e \rangle$ such that d stands in R to e.

Informally: $\langle d, e \rangle \in R$ indicates that d bears R to e.

Relations

Recall that we identify binary relations with sets of pairs.

Binary relation

A *binary relation* R is a set of zero or more pairs of objects.

R is the set of pairs $\langle d, e\rangle$ such that d stands in R to e.

Informally: $\langle d, e\rangle \in R$ indicates that d bears R to e.

Example

The relation of *liking* = $\{\langle d, e\rangle : d \text{ likes } e\}$

Relations

Recall that we identify binary relations with sets of pairs.

Binary relation

A *binary relation* R is a set of zero or more pairs of objects.

R is the set of pairs $\langle d, e \rangle$ such that d stands in R to e.

Informally: $\langle d, e \rangle \in R$ indicates that d bears R to e.

Example

The relation of *liking* $= \{\langle d, e \rangle : d \text{ likes } e\}$

Similarly:

A ternary (3-ary) relation is a set of triples (3-tuples).

Relations

Recall that we identify binary relations with sets of pairs.

Binary relation

A *binary relation* R is a set of zero or more pairs of objects.

R is the set of pairs $\langle d, e\rangle$ such that d stands in R to e.

Informally: $\langle d, e\rangle \in R$ indicates that d bears R to e.

Example

The relation of *liking* $= \{\langle d, e\rangle : d \text{ likes } e\}$

Similarly:

A ternary (3-ary) relation is a set of triples (3-tuples).
A quaternary (4-ary) relation is a set of quadruples (4-tuples).

Relations

Recall that we identify binary relations with sets of pairs.

Binary relation

A *binary relation* R is a set of zero or more pairs of objects.

R is the set of pairs $\langle d, e\rangle$ such that d stands in R to e.

Informally: $\langle d, e\rangle \in R$ indicates that d bears R to e.

Example

The relation of *liking* $= \{\langle d, e\rangle : d \text{ likes } e\}$

Similarly:

A ternary (3-ary) relation is a set of triples (3-tuples).
A quaternary (4-ary) relation is a set of quadruples (4-tuples).
etc.

Putting this all together:

Putting this all together:

‘Maggie Smith is an actor’ is true

Putting this all together:

‘Maggie Smith is an actor’ is true
iff |‘Maggie Smith’| has |‘is an actor’|

Putting this all together:

'Maggie Smith is an actor' is true
  iff |'Maggie Smith'| has |'is an actor'|
  iff Maggie Smith $\in$ the set of actors

Putting this all together:

‘Maggie Smith is an actor’ is true
  iff |‘Maggie Smith’| has |‘is an actor’|
  iff Maggie Smith $\in$ the set of actors

Similarly:

‘Mary likes Maggie Smith’ is true

Putting this all together:

'Maggie Smith is an actor' is true
    iff |'Maggie Smith'| has |'is an actor'|
    iff Maggie Smith ∈ the set of actors

Similarly:

'Mary likes Maggie Smith' is true
    iff |'Mary'| stands in |'likes'| to |'Maggie Smith'|

Putting this all together:

‘Maggie Smith is an actor’ is true
iff |‘Maggie Smith’| has |‘is an actor’|
iff Maggie Smith $\in$ the set of actors

Similarly:

‘Mary likes Maggie Smith’ is true
iff |‘Mary’| stands in |‘likes’| to |‘Maggie Smith’|
iff $\langle$Mary, M. Smith$\rangle \in \{\langle d, e\rangle : d \text{ likes } e\}$

Semantics for atomic $\mathcal{L}_2$-sentences

The semantics for atomic $\mathcal{L}_2$-sentences is similar.

Semantics for atomic $\mathcal{L}_2$-sentences

The semantics for atomic $\mathcal{L}_2$-sentences is similar.

An $\mathcal{L}_2$-structure specifies semantic values for $\mathcal{L}_2$-expressions:

$\mathcal{L}_2$-expression	*semantic value*
constant: a	object: $\lvert a \rvert$
sentence letter: P	truth-value: $\lvert P \rvert$ (i.e. T or F)
unary predicate letter: P^1	unary relation: $\lvert P^1 \rvert$ (i.e. a set)
binary predicate letter: P^2	binary relation: $\lvert P^2 \rvert$ (a set of pairs)

Semantics for atomic $\mathcal{L}_2$-sentences

The semantics for atomic $\mathcal{L}_2$-sentences is similar.

An $\mathcal{L}_2$-structure specifies semantic values for $\mathcal{L}_2$-expressions:

$\mathcal{L}_2$-expression	*semantic value*
constant: a	object: $\lvert a \rvert$
sentence letter: P	truth-value: $\lvert P \rvert$ (i.e. T or F)
unary predicate letter: P^1	unary relation: $\lvert P^1 \rvert$ (i.e. a set)
binary predicate letter: P^2	binary relation: $\lvert P^2 \rvert$ (a set of pairs)

- $|P^1 b| = \mathrm{T}$ iff $|b| \in |P^1|$

Semantics for atomic $\mathcal{L}_2$-sentences

The semantics for atomic $\mathcal{L}_2$-sentences is similar.

An $\mathcal{L}_2$-structure specifies semantic values for $\mathcal{L}_2$-expressions:

$\mathcal{L}_2$-expression	*semantic value*
constant: a	object: $|a|$
sentence letter: P	truth-value: $|P|$ (i.e. T or F)
unary predicate letter: P^1	unary relation: $|P^1|$ (i.e. a set)
binary predicate letter: P^2	binary relation: $|P^2|$ (a set of pairs)

- $|P^1 b| = \mathrm{T}$ iff $|b| \in |P^1|$
- $|R^2 ab| = \mathrm{T}$ iff $\langle |a|, |b| \rangle \in |R^2|$

Semantics for atomic $\mathcal{L}_2$-sentences

The semantics for atomic $\mathcal{L}_2$-sentences is similar.

An $\mathcal{L}_2$-structure specifies semantic values for $\mathcal{L}_2$-expressions:

$\mathcal{L}_2$-expression	*semantic value*
constant: a	object: $\lvert a\rvert$
sentence letter: P	truth-value: $\lvert P\rvert$ (i.e. T or F)
unary predicate letter: P^1	unary relation: $\lvert P^1\rvert$ (i.e. a set)
binary predicate letter: P^2	binary relation: $\lvert P^2\rvert$ (a set of pairs)

- $|P^1 b| = \mathrm{T}$ iff $|b| \in |P^1|$
- $|R^2 ab| = \mathrm{T}$ iff $\langle |a|, |b| \rangle \in |R^2|$

Notation: $|e|_{\mathcal{A}}$ is the semantic value of e in $\mathcal{L}_2$-structure $\mathcal{A}$.

Semantics for atomic $\mathcal{L}_2$-sentences

The semantics for atomic $\mathcal{L}_2$-sentences is similar.

An $\mathcal{L}_2$-structure specifies semantic values for $\mathcal{L}_2$-expressions:

$\mathcal{L}_2$-expression	*semantic value*
constant: a	object: $\lvert a \rvert_{\mathcal{A}}$
sentence letter: P	truth-value: $\lvert P \rvert_{\mathcal{A}}$ (i.e. T or F)
unary predicate letter: P^1	unary relation: $\lvert P^1 \rvert_{\mathcal{A}}$ (i.e. a set)
binary predicate letter: P^2	binary relation: $\lvert P^2 \rvert_{\mathcal{A}}$ (a set of pairs)

- $|P^1 b|_{\mathcal{A}} = \mathrm{T}$ iff $|b|_{\mathcal{A}} \in |P^1|_{\mathcal{A}}$
- $|R^2 ab|_{\mathcal{A}} = \mathrm{T}$ iff $\langle |a|_{\mathcal{A}}, |b|_{\mathcal{A}} \rangle \in |R^2|_{\mathcal{A}}$

Notation: $|e|_{\mathcal{A}}$ is the semantic value of e in $\mathcal{L}_2$-structure $\mathcal{A}$.

Semantics for atomic $\mathcal{L}_2$-formulae

We have the semantics for $\mathcal{L}_2$-*sentences* like Pa.

Semantics for atomic $\mathcal{L}_2$-formulae

We have the semantics for $\mathcal{L}_2$-*sentences* like Pa.
What about $\mathcal{L}_2$-*formulae* like Px?

Semantics for atomic $\mathcal{L}_2$-formulae

We have the semantics for $\mathcal{L}_2$-*sentences* like Pa.
What about $\mathcal{L}_2$-*formulae* like Px?

In English:

- The designator 'Maggie Smith' has a constant semantic value.

Semantics for atomic $\mathcal{L}_2$-formulae

We have the semantics for $\mathcal{L}_2$-*sentences* like Pa.
What about $\mathcal{L}_2$-*formulae* like Px?

In English:

- The designator 'Maggie Smith' has a constant semantic value.
- Pronouns, such as 'it', do not.

Semantics for atomic $\mathcal{L}_2$-formulae

We have the semantics for $\mathcal{L}_2$-*sentences* like Pa.
What about $\mathcal{L}_2$-*formulae* like Px?

In English:

- The designator 'Maggie Smith' has a constant semantic value.
- Pronouns, such as 'it', do not.
  'it' refers to different objects depending on the context.

Semantics for atomic $\mathcal{L}_2$-formulae

We have the semantics for $\mathcal{L}_2$-*sentences* like Pa.
What about $\mathcal{L}_2$-*formulae* like Px?

In English:

- The designator 'Maggie Smith' has a constant semantic value.
- Pronouns, such as 'it', do not.
  'it' refers to different objects depending on the context.

Something similar happens in an $\mathcal{L}_2$-structure $\mathcal{A}$:

Semantics for atomic $\mathcal{L}_2$-formulae

We have the semantics for $\mathcal{L}_2$-*sentences* like Pa.
What about $\mathcal{L}_2$-*formulae* like Px?

In English:

- The designator 'Maggie Smith' has a constant semantic value.
- Pronouns, such as 'it', do not.
  'it' refers to different objects depending on the context.

Something similar happens in an $\mathcal{L}_2$-structure $\mathcal{A}$:

- $a, b, c, \ldots$ are assigned a constant semantic value in $\mathcal{A}$.

Semantics for atomic $\mathcal{L}_2$-formulae

We have the semantics for $\mathcal{L}_2$-*sentences* like Pa.
What about $\mathcal{L}_2$-*formulae* like Px?

In English:

- The designator 'Maggie Smith' has a constant semantic value.
- Pronouns, such as 'it', do not.
  'it' refers to different objects depending on the context.

Something similar happens in an $\mathcal{L}_2$-structure $\mathcal{A}$:

- $a, b, c, \ldots$ are assigned a constant semantic value in $\mathcal{A}$.
- Variables: $x, y, z, \ldots$ are not.

Semantics for atomic $\mathcal{L}_2$-formulae

We have the semantics for $\mathcal{L}_2$-*sentences* like Pa.
What about $\mathcal{L}_2$-*formulae* like Px?

In English:

- The designator 'Maggie Smith' has a constant semantic value.
- Pronouns, such as 'it', do not.
  'it' refers to different objects depending on the context.

Something similar happens in an $\mathcal{L}_2$-structure $\mathcal{A}$:

- $a, b, c, \ldots$ are assigned a constant semantic value in $\mathcal{A}$.
- Variables: $x, y, z, \ldots$ are not.

What object each variable denotes is specified with a *variable assignment*.

Variable assignments

Variable assignment

A variable assignment assigns an object to each variable.

Variable assignments

Variable assignment

A variable assignment assigns an object to each variable.

One can think of a variable assignment as an infinite list.

Variable assignments

Variable assignment

A variable assignment assigns an object to each variable.

One can think of a variable assignment as an infinite list.

Example: the assignment α.

x	y	z	x_1	y_1	z_1	x_2	...
Mercury	Venus	Venus	Neptune	Mars	Venus	Mars	

Variable assignments

Variable assignment

A variable assignment assigns an object to each variable.

One can think of a variable assignment as an infinite list.

Example: the assignment α.

x	y	z	x_1	y_1	z_1	x_2	...
Mercury	Venus	Venus	Neptune	Mars	Venus	Mars	

Notation

We write $|x|^{\alpha}$ for the object α assigns to x.

Variable assignments

Variable assignment

A variable assignment assigns an object to each variable.

One can think of a variable assignment as an infinite list.

Example: the assignment α.

x	y	z	x_1	y_1	z_1	x_2	…
Mercury	Venus	Venus	Neptune	Mars	Venus	Mars	

Notation

We write $|x|^\alpha$ for the object α assigns to x.
We use lower case Greek letters: α, β, γ for assignments.

Variable assignments

Variable assignment

A variable assignment assigns an object to each variable.

One can think of a variable assignment as an infinite list.

Example: the assignment α.

x	y	z	x_1	y_1	z_1	x_2	…
Mercury	Venus	Venus	Neptune	Mars	Venus	Mars	

Notation

We write $|x|^\alpha$ for the object α assigns to x.
We use lower case Greek letters: α, β, γ for assignments.

e.g. $|x|^\alpha =$

Variable assignments

Variable assignment

A variable assignment assigns an object to each variable.

One can think of a variable assignment as an infinite list.

Example: the assignment α.

x	y	z	x_1	y_1	z_1	x_2	…
Mercury	Venus	Venus	Neptune	Mars	Venus	Mars	

Notation

We write $|x|^{\alpha}$ for the object α assigns to x.
We use lower case Greek letters: α, β, γ for assignments.

e.g. $|x|^{\alpha} = \text{Mercury}$

Variable assignments

Variable assignment

A variable assignment assigns an object to each variable.

One can think of a variable assignment as an infinite list.

Example: the assignment α.

x	y	z	x_1	y_1	z_1	x_2	...
Mercury	Venus	Venus	Neptune	Mars	Venus	Mars	

Notation

We write $|x|^\alpha$ for the object α assigns to x.
We use lower case Greek letters: α, β, γ for assignments.

e.g. $|x|^\alpha$ = Mercury; $|y|^\alpha$ =

Variable assignments

Variable assignment

A variable assignment assigns an object to each variable.

One can think of a variable assignment as an infinite list.

Example: the assignment α.

x	y	z	x_1	y_1	z_1	x_2	…
Mercury	**Venus**	Venus	Neptune	Mars	Venus	Mars	

Notation

We write $|x|^\alpha$ for the object α assigns to x.
We use lower case Greek letters: α, β, γ for assignments.

e.g. $|x|^\alpha$ = Mercury; $|y|^\alpha$ =Venus

Variable assignments

Variable assignment

A variable assignment assigns an object to each variable.

One can think of a variable assignment as an infinite list.

Example: the assignment α.

x	y	z	x_1	y_1	z_1	x_2	...
Mercury	Venus	Venus	Neptune	Mars	Venus	Mars	

Notation

We write $|x|^\alpha$ for the object α assigns to x.
We use lower case Greek letters: α, β, γ for assignments.

e.g. $|x|^\alpha$ = Mercury; $|y|^\alpha$ =Venus; $|x_2|^\alpha$ =

Variable assignments

Variable assignment

A variable assignment assigns an object to each variable.

One can think of a variable assignment as an infinite list.

Example: the assignment α.

x	y	z	x_1	y_1	z_1	x_2	…
Mercury	Venus	Venus	Neptune	Mars	Venus	**Mars**	

Notation

We write $|x|^\alpha$ for the object α assigns to x.
We use lower case Greek letters: α, β, γ for assignments.

e.g. $|x|^\alpha$ = Mercury; $|y|^\alpha$ =Venus; $|x_2|^\alpha$ = Mars.

Once x has been assigned an object, the semantics for Px are much like the semantics for Pa.

Once x has been assigned an object, the semantics for Px are much like the semantics for Pa.

We write $|e|^{\alpha}_{\mathcal{A}}$ for the semantic value of expression e in the structure $\mathcal{A}$ *under the variable assignment* α.

Once x has been assigned an object, the semantics for Px are much like the semantics for Pa.

We write $|e|^{\alpha}_{\mathcal{A}}$ for the semantic value of expression e in the structure $\mathcal{A}$ *under the variable assignment* α.

- $|Px|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $|x|^{\alpha} \in |P^1|_{\mathcal{A}}$ (NB: $|x|^{\alpha}_{\mathcal{A}} = |x|^{\alpha}$)

Once x has been assigned an object, the semantics for Px are much like the semantics for Pa.

We write $|e|^{\alpha}_{\mathcal{A}}$ for the semantic value of expression e in the structure $\mathcal{A}$ *under the variable assignment* α.

- $|Px|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $|x|^{\alpha} \in |P^1|_{\mathcal{A}}$ (NB: $|x|^{\alpha}_{\mathcal{A}} = |x|^{\alpha}$)
- $|Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $\langle |x|^{\alpha}, |y|^{\alpha} \rangle \in |R^2|_{\mathcal{A}}$

Once x has been assigned an object, the semantics for Px are much like the semantics for Pa.

We write $|e|^{\alpha}_{\mathcal{A}}$ for the semantic value of expression e in the structure $\mathcal{A}$ *under the variable assignment* α.

- $|Px|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $|x|^{\alpha} \in |P^1|_{\mathcal{A}}$ (NB: $|x|^{\alpha}_{\mathcal{A}} = |x|^{\alpha}$)
- $|Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $\langle |x|^{\alpha}, |y|^{\alpha} \rangle \in |R^2|_{\mathcal{A}}$

Note: semantic values of constants and predicates are unaffected by the assignment (e.g. $|P|^{\alpha}_{\mathcal{A}} = |P|_{\mathcal{A}}$, $|a|^{\alpha}_{\mathcal{A}} = |a|_{\mathcal{A}}$).

Once x has been assigned an object, the semantics for Px are much like the semantics for Pa.

We write $|e|^{\alpha}_{\mathcal{A}}$ for the semantic value of expression e in the structure $\mathcal{A}$ *under the variable assignment* α.

- $|Px|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $|x|^{\alpha} \in |P^1|_{\mathcal{A}}$ (NB: $|x|^{\alpha}_{\mathcal{A}} = |x|^{\alpha}$)
- $|Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $\langle |x|^{\alpha}, |y|^{\alpha} \rangle \in |R^2|_{\mathcal{A}}$

Note: semantic values of constants and predicates are unaffected by the assignment (e.g. $|P|^{\alpha}_{\mathcal{A}} = |P|_{\mathcal{A}}$, $|a|^{\alpha}_{\mathcal{A}} = |a|_{\mathcal{A}}$).

- $|Rab|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $\langle |a|_{\mathcal{A}}, |b|_{\mathcal{A}} \rangle \in |R|_{\mathcal{A}}$

Once x has been assigned an object, the semantics for Px are much like the semantics for Pa.

We write $|e|^{\alpha}_{\mathcal{A}}$ for the semantic value of expression e in the structure $\mathcal{A}$ *under the variable assignment* α.

- $|Px|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $|x|^{\alpha} \in |P^1|_{\mathcal{A}}$ (NB: $|x|^{\alpha}_{\mathcal{A}} = |x|^{\alpha}$)
- $|Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $\langle |x|^{\alpha}, |y|^{\alpha} \rangle \in |R^2|_{\mathcal{A}}$

Note: semantic values of constants and predicates are unaffected by the assignment (e.g. $|P|^{\alpha}_{\mathcal{A}} = |P|_{\mathcal{A}}$, $|a|^{\alpha}_{\mathcal{A}} = |a|_{\mathcal{A}}$).

- $|Rab|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $\langle |a|_{\mathcal{A}}, |b|_{\mathcal{A}} \rangle \in |R|_{\mathcal{A}}$
- $|Rxb|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $\langle |x|^{\alpha}, |b|_{\mathcal{A}} \rangle \in |R|_{\mathcal{A}}$

Once x has been assigned an object, the semantics for Px are much like the semantics for Pa.

We write $|e|^{\alpha}_{\mathcal{A}}$ for the semantic value of expression e in the structure $\mathcal{A}$ *under the variable assignment* α.

- $|Px|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $|x|^{\alpha} \in |P^1|_{\mathcal{A}}$ (NB: $|x|^{\alpha}_{\mathcal{A}} = |x|^{\alpha}$)
- $|Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $\langle |x|^{\alpha}, |y|^{\alpha} \rangle \in |R^2|_{\mathcal{A}}$

Note: semantic values of constants and predicates are unaffected by the assignment (e.g. $|P|^{\alpha}_{\mathcal{A}} = |P|_{\mathcal{A}}$, $|a|^{\alpha}_{\mathcal{A}} = |a|_{\mathcal{A}}$).

- $|Rab|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $\langle |a|_{\mathcal{A}}, |b|_{\mathcal{A}} \rangle \in |R|_{\mathcal{A}}$
- $|Rxb|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $\langle |x|^{\alpha}, |b|_{\mathcal{A}} \rangle \in |R|_{\mathcal{A}}$

Similarly for other atomic formulae.

Worked example

Let $\mathcal{L}_2$-structure $\mathcal{A}$ be such that:

- $|a|_{\mathcal{A}} = \text{Venus}$
- $|b|_{\mathcal{A}} = \text{Mars}$
- $|P^1|_{\mathcal{A}} = \{\text{Saturn, Mars}\}$
- $|R^2|_{\mathcal{A}} = \{\langle \text{Venus, Mars} \rangle\}$

Let assignments α and β be such that:

	x	y	z
α:	Saturn	Mars	Jupiter
β:	Venus	Venus	Venus

Worked example

Let $\mathcal{L}_2$-structure $\mathcal{A}$ be such that:

- $|a|_{\mathcal{A}} = \text{Venus}$
- $|b|_{\mathcal{A}} = \text{Mars}$
- $|P^1|_{\mathcal{A}} = \{\text{Saturn, Mars}\}$
- $|R^2|_{\mathcal{A}} = \{\langle\text{Venus, Mars}\rangle\}$

Let assignments α and β be such that:

	x	y	z
α:	Saturn	Mars	Jupiter
β:	Venus	Venus	Venus

Compute the following:

$\lvert x\rvert_{\mathcal{A}}^{\alpha} =$	$\lvert x\rvert_{\mathcal{A}}^{\beta} =$	$\lvert a\rvert_{\mathcal{A}}^{\alpha} =$
$\lvert Py\rvert_{\mathcal{A}}^{\alpha} =$	$\lvert Py\rvert_{\mathcal{A}}^{\beta} =$	$\lvert Pb\rvert_{\mathcal{A}}^{\alpha} =$
$\lvert Rxy\rvert_{\mathcal{A}}^{\alpha} =$	$\lvert Rxy\rvert_{\mathcal{A}}^{\beta} =$	$\lvert Rab\rvert_{\mathcal{A}}^{\alpha} =$

Worked example

Let $\mathcal{L}_2$-structure $\mathcal{A}$ be such that:

- $|a|_{\mathcal{A}} = \text{Venus}$
- $|b|_{\mathcal{A}} = \text{Mars}$
- $|P^1|_{\mathcal{A}} = \{\text{Saturn, Mars}\}$
- $|R^2|_{\mathcal{A}} = \{\langle\text{Venus, Mars}\rangle\}$

Let assignments α and β be such that:

	x	y	z
α:	**Saturn**	Mars	Jupiter
β:	Venus	Venus	Venus

Compute the following:

$\lvert x\rvert_{\mathcal{A}}^{\alpha} = \text{Saturn}$	$\lvert x\rvert_{\mathcal{A}}^{\beta} =$	$\lvert a\rvert_{\mathcal{A}}^{\alpha} =$
$\lvert Py\rvert_{\mathcal{A}}^{\alpha} =$	$\lvert Py\rvert_{\mathcal{A}}^{\beta} =$	$\lvert Pb\rvert_{\mathcal{A}}^{\alpha} =$
$\lvert Rxy\rvert_{\mathcal{A}}^{\alpha} =$	$\lvert Rxy\rvert_{\mathcal{A}}^{\beta} =$	$\lvert Rab\rvert_{\mathcal{A}}^{\alpha} =$

Worked example

Let $\mathcal{L}_2$-structure $\mathcal{A}$ be such that:

- $|a|_{\mathcal{A}} = \text{Venus}$
- $|b|_{\mathcal{A}} = \text{Mars}$
- $|P^1|_{\mathcal{A}} = \{\text{Saturn, Mars}\}$
- $|R^2|_{\mathcal{A}} = \{\langle\text{Venus, Mars}\rangle\}$

Let assignments α and β be such that:

	x	y	z
α:	Saturn	Mars	Jupiter
β:	**Venus**	Venus	Venus

Compute the following:

$\lvert x\rvert^{\alpha}_{\mathcal{A}} = \text{Saturn}$	$\lvert x\rvert^{\beta}_{\mathcal{A}} = \text{Venus}$	$\lvert a\rvert^{\alpha}_{\mathcal{A}} =$
$\lvert Py\rvert^{\alpha}_{\mathcal{A}} =$	$\lvert Py\rvert^{\beta}_{\mathcal{A}} =$	$\lvert Pb\rvert^{\alpha}_{\mathcal{A}} =$
$\lvert Rxy\rvert^{\alpha}_{\mathcal{A}} =$	$\lvert Rxy\rvert^{\beta}_{\mathcal{A}} =$	$\lvert Rab\rvert^{\alpha}_{\mathcal{A}} =$

Worked example

Let $\mathcal{L}_2$-structure $\mathcal{A}$ be such that:

- $|a|_{\mathcal{A}} = \textbf{Venus}$
- $|b|_{\mathcal{A}} = \text{Mars}$
- $|P^1|_{\mathcal{A}} = \{\text{Saturn, Mars}\}$
- $|R^2|_{\mathcal{A}} = \{\langle\text{Venus, Mars}\rangle\}$

Let assignments α and β be such that:

	x	y	z
α:	Saturn	Mars	Jupiter
β:	Venus	Venus	Venus

Compute the following:

$|x|^{\alpha}_{\mathcal{A}} = \text{Saturn}$ $\quad$ $|x|^{\beta}_{\mathcal{A}} = \text{Venus}$ $\quad$ $|a|^{\alpha}_{\mathcal{A}} = \text{Venus}$

$|Py|^{\alpha}_{\mathcal{A}} =$ $\quad$ $|Py|^{\beta}_{\mathcal{A}} =$ $\quad$ $|Pb|^{\alpha}_{\mathcal{A}} =$

$|Rxy|^{\alpha}_{\mathcal{A}} =$ $\quad$ $|Rxy|^{\beta}_{\mathcal{A}} =$ $\quad$ $|Rab|^{\alpha}_{\mathcal{A}} =$

Worked example

Let $\mathcal{L}_2$-structure $\mathcal{A}$ be such that:

- $|a|_{\mathcal{A}}$ = Venus
- $|b|_{\mathcal{A}}$ = Mars
- **$|P^1|_{\mathcal{A}} = \{$Saturn, Mars$\}$**
- $|R^2|_{\mathcal{A}} = \{\langle$Venus, Mars$\rangle\}$

Let assignments α and β be such that:

	x	y	z
α:	Saturn	**Mars**	Jupiter
β:	Venus	Venus	Venus

Compute the following:

$|x|^{\alpha}_{\mathcal{A}}$ = Saturn $\quad$ $|x|^{\beta}_{\mathcal{A}}$ = Venus $\quad$ $|a|^{\alpha}_{\mathcal{A}}$ = Venus

$|Py|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ $\quad$ $|Py|^{\beta}_{\mathcal{A}} =$ $\quad$ $|Pb|^{\alpha}_{\mathcal{A}} =$

$|Rxy|^{\alpha}_{\mathcal{A}} =$ $\quad$ $|Rxy|^{\beta}_{\mathcal{A}} =$ $\quad$ $|Rab|^{\alpha}_{\mathcal{A}} =$

Worked example

Let $\mathcal{L}_2$-structure $\mathcal{A}$ be such that:

- $|a|_{\mathcal{A}} = \text{Venus}$
- $|b|_{\mathcal{A}} = \text{Mars}$
- $|P^1|_{\mathcal{A}} = \{\textbf{Saturn, Mars}\}$
- $|R^2|_{\mathcal{A}} = \{\langle\text{Venus, Mars}\rangle\}$

Let assignments α and β be such that:

	x	y	z
α:	Saturn	Mars	Jupiter
β:	Venus	**Venus**	Venus

Compute the following:

$$|x|^{\alpha}_{\mathcal{A}} = \text{Saturn} \qquad |x|^{\beta}_{\mathcal{A}} = \text{Venus} \qquad |a|^{\alpha}_{\mathcal{A}} = \text{Venus}$$

$$|Py|^{\alpha}_{\mathcal{A}} = \text{T} \qquad |Py|^{\beta}_{\mathcal{A}} = \text{F} \qquad |Pb|^{\alpha}_{\mathcal{A}} =$$

$$|Rxy|^{\alpha}_{\mathcal{A}} = \qquad |Rxy|^{\beta}_{\mathcal{A}} = \qquad |Rab|^{\alpha}_{\mathcal{A}} =$$

Worked example

Let $\mathcal{L}_2$-structure $\mathcal{A}$ be such that:

- $|a|_{\mathcal{A}} = \text{Venus}$
- $|b|_{\mathcal{A}} = \textbf{Mars}$
- $|P^1|_{\mathcal{A}} = \{\textbf{Saturn, Mars}\}$
- $|R^2|_{\mathcal{A}} = \{\langle \text{Venus, Mars} \rangle\}$

Let assignments α and β be such that:

	x	y	z
α:	Saturn	Mars	Jupiter
β:	Venus	Venus	Venus

Compute the following:

$\lvert x\rvert^{\alpha}_{\mathcal{A}} = \text{Saturn}$	$\lvert x\rvert^{\beta}_{\mathcal{A}} = \text{Venus}$	$\lvert a\rvert^{\alpha}_{\mathcal{A}} = \text{Venus}$
$\lvert Py\rvert^{\alpha}_{\mathcal{A}} = \text{T}$	$\lvert Py\rvert^{\beta}_{\mathcal{A}} = \text{F}$	$\lvert Pb\rvert^{\alpha}_{\mathcal{A}} = \text{T}$
$\lvert Rxy\rvert^{\alpha}_{\mathcal{A}} =$	$\lvert Rxy\rvert^{\beta}_{\mathcal{A}} =$	$\lvert Rab\rvert^{\alpha}_{\mathcal{A}} =$

Worked example

Let $\mathcal{L}_2$-structure $\mathcal{A}$ be such that:

- $|a|_{\mathcal{A}}$ = Venus
- $|b|_{\mathcal{A}}$ = Mars
- $|P^1|_{\mathcal{A}} = \{$Saturn, Mars$\}$
- $|R^2|_{\mathcal{A}} = \{\langle$**Venus, Mars**$\rangle\}$

Let assignments α and β be such that:

	x	y	z
α:	**Saturn**	**Mars**	Jupiter
β:	Venus	Venus	Venus

Compute the following:

$|x|^{\alpha}_{\mathcal{A}}$ = Saturn $\quad$ $|x|^{\beta}_{\mathcal{A}}$ = Venus $\quad$ $|a|^{\alpha}_{\mathcal{A}}$ = Venus

$|Py|^{\alpha}_{\mathcal{A}}$ = T $\quad$ $|Py|^{\beta}_{\mathcal{A}}$ = F $\quad$ $|Pb|^{\alpha}_{\mathcal{A}}$ = T

$|Rxy|^{\alpha}_{\mathcal{A}}$ = F $\quad$ $|Rxy|^{\beta}_{\mathcal{A}}$ = $\quad$ $|Rab|^{\alpha}_{\mathcal{A}}$ =

Worked example

Let $\mathcal{L}_2$-structure $\mathcal{A}$ be such that:

- $|a|_{\mathcal{A}} = \text{Venus}$
- $|b|_{\mathcal{A}} = \text{Mars}$
- $|P^1|_{\mathcal{A}} = \{\text{Saturn, Mars}\}$
- $|R^2|_{\mathcal{A}} = \{\langle \textbf{Venus, Mars} \rangle\}$

Let assignments α and β be such that:

	x	y	z
α:	Saturn	Mars	Jupiter
β:	**Venus**	**Venus**	Venus

Compute the following:

$|x|^{\alpha}_{\mathcal{A}} = \text{Saturn}$ $\quad$ $|x|^{\beta}_{\mathcal{A}} = \text{Venus}$ $\quad$ $|a|^{\alpha}_{\mathcal{A}} = \text{Venus}$

$|Py|^{\alpha}_{\mathcal{A}} = \text{T}$ $\quad$ $|Py|^{\beta}_{\mathcal{A}} = \text{F}$ $\quad$ $|Pb|^{\alpha}_{\mathcal{A}} = \text{T}$

$|Rxy|^{\alpha}_{\mathcal{A}} = \text{F}$ $\quad$ $|Rxy|^{\beta}_{\mathcal{A}} = \text{F}$ $\quad$ $|Rab|^{\alpha}_{\mathcal{A}} =$

Worked example

Let $\mathcal{L}_2$-structure $\mathcal{A}$ be such that:

- $|a|_{\mathcal{A}} = \textbf{Venus}$
- $|b|_{\mathcal{A}} = \textbf{Mars}$
- $|P^1|_{\mathcal{A}} = \{\text{Saturn, Mars}\}$
- $|R^2|_{\mathcal{A}} = \{\langle \textbf{Venus, Mars} \rangle\}$

Let assignments α and β be such that:

	x	y	z
α:	Saturn	Mars	Jupiter
β:	Venus	Venus	Venus

Compute the following:

$$|x|^{\alpha}_{\mathcal{A}} = \text{Saturn} \qquad |x|^{\beta}_{\mathcal{A}} = \text{Venus} \qquad |a|^{\alpha}_{\mathcal{A}} = \text{Venus}$$

$$|Py|^{\alpha}_{\mathcal{A}} = \text{T} \qquad |Py|^{\beta}_{\mathcal{A}} = \text{F} \qquad |Pb|^{\alpha}_{\mathcal{A}} = \text{T}$$

$$|Rxy|^{\alpha}_{\mathcal{A}} = \text{F} \qquad |Rxy|^{\beta}_{\mathcal{A}} = \text{F} \qquad |Rab|^{\alpha}_{\mathcal{A}} = \text{T}$$

Semantics for quantifiers

Whether the following sentence is true depends on which things there are:

Everything is material.

Thus the truth of sentences depends on which objects there are and this needs to be taken into account in determining truth values.

In English, the truth-value of a quantified sentence depends on how widely the quantifiers range.

In English, the truth-value of a quantified sentence depends on how widely the quantifiers range.

Everyone can hear the lecturer.

In English, the truth-value of a quantified sentence depends on how widely the quantifiers range.

Everyone can hear the lecturer.

The context supplies a 'domain' telling us what 'everyone' ranges over.

In English, the truth-value of a quantified sentence depends on how widely the quantifiers range.

Everyone can hear the lecturer.

The context supplies a 'domain' telling us what 'everyone' ranges over. 20

Domain: the set of people in South Schools

Everyone can hear the lecturer.

In English, the truth-value of a quantified sentence depends on how widely the quantifiers range.

Everyone can hear the lecturer.

The context supplies a 'domain' telling us what 'everyone' ranges over. 20

Domain: the set of people in South Schools

Everyone can hear the lecturer. T

In English, the truth-value of a quantified sentence depends on how widely the quantifiers range.

Everyone can hear the lecturer.

The context supplies a 'domain' telling us what 'everyone' ranges over. 20

Domain: the set of people in South Schools

Everyone can hear the lecturer. T

Domain: the set of everyone in the world

Everyone can hear the lecturer.

In English, the truth-value of a quantified sentence depends on how widely the quantifiers range.

Everyone can hear the lecturer.

The context supplies a 'domain' telling us what 'everyone' ranges over.

Domain: the set of people in South Schools	
Everyone can hear the lecturer.	T

Domain: the set of everyone in the world	
Everyone can hear the lecturer.	F

An $\mathcal{L}_2$-structure $\mathcal{A}$ specifies a non-empty set $D_{\mathcal{A}}$ as the domain.

An $\mathcal{L}_2$-structure $\mathcal{A}$ specifies a non-empty set $D_{\mathcal{A}}$ as the domain.
A *variable assignment over* $\mathcal{A}$ assigns a member of $D_{\mathcal{A}}$ to each variable.

An $\mathcal{L}_2$-structure $\mathcal{A}$ specifies a non-empty set $D_{\mathcal{A}}$ as the domain.
A *variable assignment over* $\mathcal{A}$ assigns a member of $D_{\mathcal{A}}$ to each variable.

Semantics for $\forall / \exists$ (first approximation):

$|\forall x Px|_{\mathcal{A}} = \mathrm{T}$
iff every member of $D_{\mathcal{A}}$ is in $|P|_{\mathcal{A}}$

An $\mathcal{L}_2$-structure $\mathcal{A}$ specifies a non-empty set $D_{\mathcal{A}}$ as the domain.
A *variable assignment over* $\mathcal{A}$ assigns a member of $D_{\mathcal{A}}$ to each variable.

Semantics for ∀/∃ (first approximation):

$|\forall x Px|_{\mathcal{A}} = \mathrm{T}$
iff every member of $D_{\mathcal{A}}$ is in $|P|_{\mathcal{A}}$
iff every assignment α of x to a member of $D_{\mathcal{A}}$ is such that $|x|^{\alpha} \in |P|_{\mathcal{A}}$

An $\mathcal{L}_2$-structure $\mathcal{A}$ specifies a non-empty set $D_{\mathcal{A}}$ as the domain.
A *variable assignment over* $\mathcal{A}$ assigns a member of $D_{\mathcal{A}}$ to each variable.

Semantics for ∀/∃ (first approximation):

$|\forall x Px|_{\mathcal{A}} = \mathrm{T}$
iff every member of $D_{\mathcal{A}}$ is in $|P|_{\mathcal{A}}$
iff every assignment α of x to a member of $D_{\mathcal{A}}$ is such that $|x|^{\alpha} \in |P|_{\mathcal{A}}$
iff every assignment α over $\mathcal{A}$ is such that $|Px|^{\alpha}_{\mathcal{A}} = T$

An $\mathcal{L}_2$-structure $\mathcal{A}$ specifies a non-empty set $D_{\mathcal{A}}$ as the domain.
A *variable assignment over* $\mathcal{A}$ assigns a member of $D_{\mathcal{A}}$ to each variable.

Semantics for $\forall/\exists$ (first approximation):

$|\forall x Px|_{\mathcal{A}} = \mathrm{T}$
iff every member of $D_{\mathcal{A}}$ is in $|P|_{\mathcal{A}}$
iff every assignment α of x to a member of $D_{\mathcal{A}}$ is such that $|x|^{\alpha} \in |P|_{\mathcal{A}}$
iff every assignment α over $\mathcal{A}$ is such that $|Px|^{\alpha}_{\mathcal{A}} = T$

Similarly:

$|\exists x Px|_{\mathcal{A}} = \mathrm{T}$
iff some member of $D_{\mathcal{A}}$ is in $|P|_{\mathcal{A}}$

An $\mathcal{L}_2$-structure $\mathcal{A}$ specifies a non-empty set $D_{\mathcal{A}}$ as the domain.
A *variable assignment over* $\mathcal{A}$ assigns a member of $D_{\mathcal{A}}$ to each variable.

Semantics for ∀/∃ (first approximation):

$|\forall x Px|_{\mathcal{A}} = \mathrm{T}$
iff every member of $D_{\mathcal{A}}$ is in $|P|_{\mathcal{A}}$
iff every assignment α of x to a member of $D_{\mathcal{A}}$ is such that $|x|^{\alpha} \in |P|_{\mathcal{A}}$
iff every assignment α over $\mathcal{A}$ is such that $|Px|^{\alpha}_{\mathcal{A}} = T$

Similarly:

$|\exists x Px|_{\mathcal{A}} = \mathrm{T}$
iff some member of $D_{\mathcal{A}}$ is in $|P|_{\mathcal{A}}$
iff some assignment α of x to a member of $D_{\mathcal{A}}$ is such that $|x|^{\alpha} \in |P|_{\mathcal{A}}$

An $\mathcal{L}_2$-structure $\mathcal{A}$ specifies a non-empty set $D_{\mathcal{A}}$ as the domain.
A *variable assignment over* $\mathcal{A}$ assigns a member of $D_{\mathcal{A}}$ to each variable.

Semantics for $\forall/\exists$ (first approximation):

$|\forall x Px|_{\mathcal{A}} = \mathrm{T}$
iff every member of $D_{\mathcal{A}}$ is in $|P|_{\mathcal{A}}$
iff every assignment α of x to a member of $D_{\mathcal{A}}$ is such that $|x|^{\alpha} \in |P|_{\mathcal{A}}$
iff every assignment α over $\mathcal{A}$ is such that $|Px|^{\alpha}_{\mathcal{A}} = T$

Similarly:

$|\exists x Px|_{\mathcal{A}} = \mathrm{T}$
iff some member of $D_{\mathcal{A}}$ is in $|P|_{\mathcal{A}}$
iff some assignment α of x to a member of $D_{\mathcal{A}}$ is such that $|x|^{\alpha} \in |P|_{\mathcal{A}}$
iff some assignment α over $\mathcal{A}$ is such that $|Px|^{\alpha}_{\mathcal{A}} = T$

An $\mathcal{L}_2$-structure $\mathcal{A}$ specifies a non-empty set $D_\mathcal{A}$ as the domain.
A *variable assignment over* $\mathcal{A}$ assigns a member of $D_\mathcal{A}$ to each variable.

Semantics for $\forall/\exists$ (first approximation):

$|\forall x Px|_\mathcal{A} = \mathrm{T}$
iff every member of $D_\mathcal{A}$ is in $|P|_\mathcal{A}$
iff every assignment α of x to a member of $D_\mathcal{A}$ is such that $|x|^\alpha \in |P|_\mathcal{A}$
iff every assignment α over $\mathcal{A}$ is such that $|Px|^\alpha_\mathcal{A} = T$

Similarly:

$|\exists x Px|_\mathcal{A} = \mathrm{T}$
iff some member of $D_\mathcal{A}$ is in $|P|_\mathcal{A}$
iff some assignment α of x to a member of $D_\mathcal{A}$ is such that $|x|^\alpha \in |P|_\mathcal{A}$
iff some assignment α over $\mathcal{A}$ is such that $|Px|^\alpha_\mathcal{A} = T$

This is correct but the general case is more complex.

The semantics of quantifiers is complicated by the need to deal with multiple quantifiers in sentences such as $\forall x \exists y Rxy$.

The semantics of quantifiers is complicated by the need to deal with multiple quantifiers in sentences such as $\forall x \exists y Rxy$.

Suppose we try to evaluate this as before in $\mathcal{A}$ with domain $D_{\mathcal{A}}$.

The semantics of quantifiers is complicated by the need to deal with multiple quantifiers in sentences such as $\forall x \exists y Rxy$.

Suppose we try to evaluate this as before in $\mathcal{A}$ with domain $D_{\mathcal{A}}$.

$|\forall x \exists y Rxy|_{\mathcal{A}} = \mathrm{T}$
  iff every assignment α over $\mathcal{A}$ is such that $|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$

The semantics of quantifiers is complicated by the need to deal with multiple quantifiers in sentences such as $\forall x \exists y Rxy$.

Suppose we try to evaluate this as before in $\mathcal{A}$ with domain $D_{\mathcal{A}}$.

$|\forall x \exists y Rxy|_{\mathcal{A}} = \mathrm{T}$
iff every assignment α over $\mathcal{A}$ is such that $|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathbf{T}$

To progress any further we need to be able evaluate $\exists y Rxy$ *under an assignment* α *of an object to* x.

How to determine $|\exists y Rxy|^{\alpha}_{\mathcal{A}}$?

How to determine $|\exists y Rxy|^{\alpha}_{\mathcal{A}}$?

$|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$

iff some d in $D_{\mathcal{A}}$ is such that $\langle |x|^{\alpha}, d\rangle \in |R|_{\mathcal{A}}$

How to determine $|\exists y Rxy|^{\alpha}_{\mathcal{A}}$?

$|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$

iff some d in $D_{\mathcal{A}}$ is such that $\langle |x|^{\alpha}, d\rangle \in |R|_{\mathcal{A}}$

iff some assignment β over $\mathcal{A}$ is such that $\langle |x|^{\alpha}, |y|^{\beta}\rangle \in |R|_{\mathcal{A}}$

How to determine $|\exists y Rxy|^{\alpha}_{\mathcal{A}}$?

$|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$

iff some d in $D_{\mathcal{A}}$ is such that $\langle |x|^{\alpha}, d\rangle \in |R|_{\mathcal{A}}$

iff some assignment β over $\mathcal{A}$ is such that $\langle |x|^{\alpha}, |y|^{\beta}\rangle \in |R|_{\mathcal{A}}$

We don't have to keep track of multiple assignments:

Say that β *differs from* α *in* y *at most* if $|v|^{\alpha} = |v|^{\beta}$ for all variables v with the possible exception of y.

How to determine $|\exists y Rxy|^{\alpha}_{\mathcal{A}}$?

$|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$

iff some d in $D_{\mathcal{A}}$ is such that $\langle |x|^{\alpha}, d \rangle \in |R|_{\mathcal{A}}$

iff some assignment β over $\mathcal{A}$ is such that $\langle |x|^{\alpha}, |y|^{\beta} \rangle \in |R|_{\mathcal{A}}$

We don't have to keep track of multiple assignments:

Say that β *differs from α in y at most* if $|v|^{\alpha} = |v|^{\beta}$ for all variables v with the possible exception of y.

$|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$

iff some assignment β over $\mathcal{A}$ which differs from α in y at most
is such that $\langle |x|^{\alpha}, |y|^{\beta} \rangle \in |R|_{\mathcal{A}}$

How to determine $|\exists y Rxy|^{\alpha}_{\mathcal{A}}$?

$|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$

iff some d in $D_{\mathcal{A}}$ is such that $\langle |x|^{\alpha}, d\rangle \in |R|_{\mathcal{A}}$

iff some assignment β over $\mathcal{A}$ is such that $\langle |x|^{\alpha}, |y|^{\beta}\rangle \in |R|_{\mathcal{A}}$

We don't have to keep track of multiple assignments:

Say that β *differs from* α *in* y *at most* if $|v|^{\alpha} = |v|^{\beta}$ for all variables v with the possible exception of y.

$|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$

iff some assignment β over $\mathcal{A}$ which differs from α in y at most
is such that $\langle |x|^{\alpha}, |y|^{\beta}\rangle \in |R|_{\mathcal{A}}$

How to determine $|\exists y Rxy|^{\alpha}_{\mathcal{A}}$?

$|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$

iff some d in $D_{\mathcal{A}}$ is such that $\langle |x|^{\alpha}, d\rangle \in |R|_{\mathcal{A}}$

iff some assignment β over $\mathcal{A}$ is such that $\langle |x|^{\alpha}, |y|^{\beta}\rangle \in |R|_{\mathcal{A}}$

We don't have to keep track of multiple assignments:

Say that β *differs from* α *in* y *at most* if $|v|^{\alpha} = |v|^{\beta}$ for all variables v with the possible exception of y.

$|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$

iff some assignment β over $\mathcal{A}$ which differs from α in y at most
is such that $\langle |x|^{\beta}, |y|^{\beta}\rangle \in |R|_{\mathcal{A}}$

How to determine $|\exists y Rxy|^{\alpha}_{\mathcal{A}}$?

$|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$

iff some d in $D_{\mathcal{A}}$ is such that $\langle |x|^{\alpha}, d\rangle \in |R|_{\mathcal{A}}$

iff some assignment β over $\mathcal{A}$ is such that $\langle |x|^{\alpha}, |y|^{\beta}\rangle \in |R|_{\mathcal{A}}$

We don't have to keep track of multiple assignments:

Say that β *differs from α in y at most* if $|v|^{\alpha} = |v|^{\beta}$ for all variables v with the possible exception of y.

$|\exists y Rxy|^{\alpha}_{\mathcal{A}} = \mathrm{T}$

iff some assignment β over $\mathcal{A}$ which differs from α in y at most
is such that $\langle |x|^{\beta}, |y|^{\beta}\rangle \in |R|_{\mathcal{A}}$

iff some assignment β over $\mathcal{A}$ which differs from α in y at most is such that $|Rxy|^{\beta}_{\mathcal{A}} = \mathrm{T}$

$\mathcal{L}_2$-structures

Here's the full specification of an $\mathcal{L}_2$-structure.

$\mathcal{L}_2$-structures

Here's the full specification of an $\mathcal{L}_2$-structure.

An $\mathcal{L}_2$-structure $\mathcal{A}$ supplies two things

$\mathcal{L}_2$-structures

Here's the full specification of an $\mathcal{L}_2$-structure.

An $\mathcal{L}_2$-structure $\mathcal{A}$ supplies two things

(1) a domain: a non-empty set $D_{\mathcal{A}}$

$\mathcal{L}_2$-structures

Here's the full specification of an $\mathcal{L}_2$-structure.

An $\mathcal{L}_2$-structure $\mathcal{A}$ supplies two things

1. a domain: a non-empty set $D_{\mathcal{A}}$
2. a semantic value for each predicate letter and constant.

$\mathcal{L}_2$-structures

Here's the full specification of an $\mathcal{L}_2$-structure.

An $\mathcal{L}_2$-structure $\mathcal{A}$ supplies two things

1. a domain: a non-empty set $D_{\mathcal{A}}$
2. a semantic value for each predicate letter and constant.

$\mathcal{L}_2$-expression	semantic value in $\mathcal{A}$
constant: a	object: $\lvert a \rvert_{\mathcal{A}}$ in $D_{\mathcal{A}}$
sentence letter: P	truth-value: $\lvert P \rvert_{\mathcal{A}}$ (= T or F)
unary predicate letter: P^1	unary relation: $\lvert P^1 \rvert_{\mathcal{A}}$ (i.e. a set)
binary predicate letter: P^2	binary relation: $\lvert P^2 \rvert_{\mathcal{A}}$ (a set of pairs)
ternary predicate letter: P^3	ternary relation: $\lvert P^3 \rvert_{\mathcal{A}}$ (a set of triples)
etc.	

Summary of semantics of $\mathcal{L}_2$

Let $\mathcal{A}$ be an $\mathcal{L}_2$-structure and α an assignment over $\mathcal{A}$.

Summary of semantics of $\mathcal{L}_2$

Let $\mathcal{A}$ be an $\mathcal{L}_2$-structure and α an assignment over $\mathcal{A}$.

Atomic formulae

Let Φ^n be a n-ary predicate letter ($n > 0$) and let $t_1, t_2, \ldots$ be variables or constants.

Summary of semantics of $\mathcal{L}_2$

Let $\mathcal{A}$ be an $\mathcal{L}_2$-structure and α an assignment over $\mathcal{A}$.

Atomic formulae

Let Φ^n be a n-ary predicate letter ($n > 0$) and let $t_1, t_2, \ldots$ be variables or constants.

- $|\Phi^n|^\alpha_\mathcal{A}$ is the n-ary relation assigned to Φ^n by $\mathcal{A}$.

Summary of semantics of $\mathcal{L}_2$

Let $\mathcal{A}$ be an $\mathcal{L}_2$-structure and α an assignment over $\mathcal{A}$.

Atomic formulae

Let Φ^n be a n-ary predicate letter ($n > 0$) and let $t_1, t_2, \ldots$ be variables or constants.

- $|\Phi^n|^\alpha_\mathcal{A}$ is the n-ary relation assigned to Φ^n by $\mathcal{A}$.
- $|t|^\alpha_\mathcal{A}$ is the object t denotes in $\mathcal{A}$ if t is a constant.
- $|t|^\alpha_\mathcal{A}$ is the object assigned to t by α if t is a variable.

(i) $|\Phi^1 t_1|^\alpha_\mathcal{A} = \text{T}$ if and only if $|t_1|^\alpha_\mathcal{A} \in |\Phi^1|_\mathcal{A}$

Summary of semantics of $\mathcal{L}_2$

Let $\mathcal{A}$ be an $\mathcal{L}_2$-structure and α an assignment over $\mathcal{A}$.

Atomic formulae

Let Φ^n be a n-ary predicate letter ($n > 0$) and let $t_1, t_2, \ldots$ be variables or constants.

- $|\Phi^n|^\alpha_\mathcal{A}$ is the n-ary relation assigned to Φ^n by $\mathcal{A}$.
- $|t|^\alpha_\mathcal{A}$ is the object t denotes in $\mathcal{A}$ if t is a constant.
- $|t|^\alpha_\mathcal{A}$ is the object assigned to t by α if t is a variable.

(i) $|\Phi^1 t_1|^\alpha_\mathcal{A} = \mathrm{T}$ if and only if $|t_1|^\alpha_\mathcal{A} \in |\Phi^1|_\mathcal{A}$

$|\Phi^2 t_1 t_2|^\alpha_\mathcal{A} = \mathrm{T}$ if and only if $\langle |t_1|^\alpha_\mathcal{A}, |t_2|^\alpha_\mathcal{A} \rangle \in |\Phi^2|_\mathcal{A}$

Summary of semantics of $\mathcal{L}_2$

Let $\mathcal{A}$ be an $\mathcal{L}_2$-structure and α an assignment over $\mathcal{A}$.

Atomic formulae

Let Φ^n be a n-ary predicate letter ($n > 0$) and let $t_1, t_2, \ldots$ be variables or constants.

- $|\Phi^n|^\alpha_\mathcal{A}$ is the n-ary relation assigned to Φ^n by $\mathcal{A}$.
- $|t|^\alpha_\mathcal{A}$ is the object t denotes in $\mathcal{A}$ if t is a constant.
- $|t|^\alpha_\mathcal{A}$ is the object assigned to t by α if t is a variable.

(i) $|\Phi^1 t_1|^\alpha_\mathcal{A} = \mathrm{T}$ if and only if $|t_1|^\alpha_\mathcal{A} \in |\Phi^1|_\mathcal{A}$
$|\Phi^2 t_1 t_2|^\alpha_\mathcal{A} = \mathrm{T}$ if and only if $\langle |t_1|^\alpha_\mathcal{A}, |t_2|^\alpha_\mathcal{A} \rangle \in |\Phi^2|_\mathcal{A}$
$|\Phi^3 t_1 t_2 t_3|^\alpha_\mathcal{A} = \mathrm{T}$ if and only if $\langle |t_1|^\alpha_\mathcal{A}, |t_2|^\alpha_\mathcal{A}, |t_3|^\alpha_\mathcal{A} \rangle \in |\Phi^3|_\mathcal{A}$

Summary of semantics of $\mathcal{L}_2$

Let $\mathcal{A}$ be an $\mathcal{L}_2$-structure and α an assignment over $\mathcal{A}$.

Atomic formulae

Let Φ^n be a n-ary predicate letter ($n > 0$) and let $t_1, t_2, \ldots$ be variables or constants.

- $|\Phi^n|^\alpha_\mathcal{A}$ is the n-ary relation assigned to Φ^n by $\mathcal{A}$.
- $|t|^\alpha_\mathcal{A}$ is the object t denotes in $\mathcal{A}$ if t is a constant.
- $|t|^\alpha_\mathcal{A}$ is the object assigned to t by α if t is a variable.

(i) $|\Phi^1 t_1|^\alpha_\mathcal{A} = \mathrm{T}$ if and only if $|t_1|^\alpha_\mathcal{A} \in |\Phi^1|_\mathcal{A}$
$|\Phi^2 t_1 t_2|^\alpha_\mathcal{A} = \mathrm{T}$ if and only if $\langle |t_1|^\alpha_\mathcal{A}, |t_2|^\alpha_\mathcal{A} \rangle \in |\Phi^2|_\mathcal{A}$
$|\Phi^3 t_1 t_2 t_3|^\alpha_\mathcal{A} = \mathrm{T}$ if and only if $\langle |t_1|^\alpha_\mathcal{A}, |t_2|^\alpha_\mathcal{A}, |t_3|^\alpha_\mathcal{A} \rangle \in |\Phi^3|_\mathcal{A}$
etc.

The semantics for connectives are just like those for $\mathcal{L}_1$.

Semantics for connectives

(ii) $|\neg\phi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|\phi|^{\alpha}_{\mathcal{A}} = \mathrm{F}$.

(iii) $|\phi \wedge \psi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|\phi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ and $|\psi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$.

(iv) $|\phi \vee \psi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|\phi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ or $|\psi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$.

(v) $|\phi \rightarrow \psi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|\phi|^{\alpha}_{\mathcal{A}} = \mathrm{F}$ or $|\psi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$.

(vi) $|\phi \leftrightarrow \psi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|\phi|^{\alpha}_{\mathcal{A}} = |\psi|^{\alpha}_{\mathcal{A}}$.

These are the semantic clauses for $\forall v$ and $\exists v$.

These are the semantic clauses for $\forall v$ and $\exists v$.

Quantifiers

(vii) $|\forall v\, \phi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|\phi|^{\beta}_{\mathcal{A}} = \mathrm{T}$ for all variable assignments β over $\mathcal{A}$ differing from α in v at most.

These are the semantic clauses for $\forall v$ and $\exists v$.

Quantifiers

(vii) $|\forall v\, \phi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|\phi|^{\beta}_{\mathcal{A}} = \mathrm{T}$ for all variable assignments β over $\mathcal{A}$ differing from α in v at most.

(viii) $|\exists v\, \phi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|\phi|^{\beta}_{\mathcal{A}} = \mathrm{T}$ for at least one variable assignment β over $\mathcal{A}$ differing from α in v at most.

These are the semantic clauses for $\forall v$ and $\exists v$.

Quantifiers

(vii) $|\forall v\, \phi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|\phi|^{\beta}_{\mathcal{A}} = \mathrm{T}$ for all variable assignments β over $\mathcal{A}$ differing from α in v at most.

(viii) $|\exists v\, \phi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|\phi|^{\beta}_{\mathcal{A}} = \mathrm{T}$ for at least one variable assignment β over $\mathcal{A}$ differing from α in v at most.

These clauses determine the truth value of any formula in a structure $\mathcal{A}$ under some variable assignment α over $\mathcal{A}$ inductively.

These are the semantic clauses for $\forall v$ and $\exists v$.

Quantifiers

(vii) $|\forall v\, \phi|^{\alpha}_{\mathcal{A}} = T$ if and only if $|\phi|^{\beta}_{\mathcal{A}} = T$ for all variable assignments β over $\mathcal{A}$ differing from α in v at most.

(viii) $|\exists v\, \phi|^{\alpha}_{\mathcal{A}} = T$ if and only if $|\phi|^{\beta}_{\mathcal{A}} = T$ for at least one variable assignment β over $\mathcal{A}$ differing from α in v at most.

These clauses determine the truth value of any formula in a structure $\mathcal{A}$ under some variable assignment α over $\mathcal{A}$ inductively.

However, we lack a simple decision procedure (in contrast to $\mathcal{L}_1$ and the truth table method).

Truth

We haven't yet said what it is for a *sentence* to be *true* in an $\mathcal{L}_2$-structure $\mathcal{A}$.

Truth

We haven't yet said what it is for a *sentence* to be *true* in an $\mathcal{L}_2$-structure $\mathcal{A}$.

We've said what it is for a *formula* to be true in an $\mathcal{L}_2$-structure $\mathcal{A}$ *under an assignment over* $\mathcal{A}$.

Truth

We haven't yet said what it is for a *sentence* to be *true* in an $\mathcal{L}_2$-structure $\mathcal{A}$.

We've said what it is for a *formula* to be true in an $\mathcal{L}_2$-structure $\mathcal{A}$ *under an assignment over* $\mathcal{A}$.

(We've defined $|\phi|_{\mathcal{A}}^{\alpha}$; we want now to define $|\phi|_{\mathcal{A}}$.)

Truth

We haven't yet said what it is for a *sentence* to be *true* in an $\mathcal{L}_2$-structure $\mathcal{A}$.

We've said what it is for a *formula* to be true in an $\mathcal{L}_2$-structure $\mathcal{A}$ *under an assignment over* $\mathcal{A}$.

(We've defined $|\phi|^{\alpha}_{\mathcal{A}}$; we want now to define $|\phi|_{\mathcal{A}}$.)

Fact about sentences

The truth-value of a sentence does *not* depend on the assignment.

Truth

We haven't yet said what it is for a *sentence* to be *true* in an $\mathcal{L}_2$-structure $\mathcal{A}$.

We've said what it is for a *formula* to be true in an $\mathcal{L}_2$-structure $\mathcal{A}$ *under an assignment over* $\mathcal{A}$.

(We've defined $|\phi|^{\alpha}_{\mathcal{A}}$; we want now to define $|\phi|_{\mathcal{A}}$.)

Fact about sentences

The truth-value of a sentence does *not* depend on the assignment.

For α and β over $\mathcal{A}$: $|\phi|^{\alpha}_{\mathcal{A}} = |\phi|^{\beta}_{\mathcal{A}}$ (when ϕ is a sentence).

Truth

We haven't yet said what it is for a *sentence* to be *true* in an $\mathcal{L}_2$-structure $\mathcal{A}$.

We've said what it is for a *formula* to be true in an $\mathcal{L}_2$-structure $\mathcal{A}$ *under an assignment over* $\mathcal{A}$.

(We've defined $|\phi|^{\alpha}_{\mathcal{A}}$; we want now to define $|\phi|_{\mathcal{A}}$.)

Fact about sentences

The truth-value of a sentence does *not* depend on the assignment.
For α and β over $\mathcal{A}$: $|\phi|^{\alpha}_{\mathcal{A}} = |\phi|^{\beta}_{\mathcal{A}}$ (when ϕ is a sentence).

A sentence ϕ is *true in an $\mathcal{L}_2$-structure $\mathcal{A}$* (in symbols: $|\phi|_{\mathcal{A}} = T$) iff $|\phi|^{\alpha}_{\mathcal{A}} = T$ for all variable assignments α over $\mathcal{A}$.

Truth

We haven't yet said what it is for a *sentence* to be *true* in an $\mathcal{L}_2$-structure $\mathcal{A}$.

We've said what it is for a *formula* to be true in an $\mathcal{L}_2$-structure $\mathcal{A}$ *under an assignment over* $\mathcal{A}$.

(We've defined $|\phi|^{\alpha}_{\mathcal{A}}$; we want now to define $|\phi|_{\mathcal{A}}$.)

Fact about sentences

The truth-value of a sentence does *not* depend on the assignment.

For α and β over $\mathcal{A}$: $|\phi|^{\alpha}_{\mathcal{A}} = |\phi|^{\beta}_{\mathcal{A}}$ (when ϕ is a sentence).

A sentence ϕ is *true in an* $\mathcal{L}_2$*-structure* $\mathcal{A}$ (in symbols: $|\phi|_{\mathcal{A}} = \mathrm{T}$) iff $|\phi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ for all variable assignments α over $\mathcal{A}$.

equivalently: $|\phi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ for some variable assignment α over $\mathcal{A}$.

Truth

We haven't yet said what it is for a *sentence* to be *true* in an $\mathcal{L}_2$-structure $\mathcal{A}$.

We've said what it is for a *formula* to be true in an $\mathcal{L}_2$-structure $\mathcal{A}$ *under an assignment over* $\mathcal{A}$.

(We've defined $|\phi|^{\alpha}_{\mathcal{A}}$; we want now to define $|\phi|_{\mathcal{A}}$.)

Fact about sentences

The truth-value of a sentence does *not* depend on the assignment.
For α and β over $\mathcal{A}$: $|\phi|^{\alpha}_{\mathcal{A}} = |\phi|^{\beta}_{\mathcal{A}}$ (when ϕ is a sentence).

A sentence ϕ is *true in an $\mathcal{L}_2$-structure* $\mathcal{A}$ (in symbols: $|\phi|_{\mathcal{A}} = T$) iff $|\phi|^{\alpha}_{\mathcal{A}} = T$ for all variable assignments α over $\mathcal{A}$.

equivalently: $|\phi|^{\alpha}_{\mathcal{A}} = T$ for some variable assignment α over $\mathcal{A}$.

Now you know what truth is.

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$\neg(((P \land Q) \to (P \lor \neg R_{45})) \leftrightarrow \neg((P_3 \lor R) \lor R))$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$(((P \wedge Q) \rightarrow (P \vee \neg R_{45})) \leftrightarrow \neg((P_3 \vee R) \vee R))$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$((P \wedge Q) \to (P \vee \neg R_{45})) \qquad \neg((P_3 \vee R) \vee R)$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$((P \land Q) \to (P \lor \neg R_{45})) \qquad ((P_3 \lor R) \lor R)$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$((P \wedge Q) \to (P \vee \neg R_{45})) \qquad (P_3 \vee R) \quad R$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$((P \wedge Q) \to (P \vee \neg R_{45})) \qquad (P_3 \vee R)$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$((P \wedge Q) \to (P \vee \neg R_{45})) \qquad P_3 \quad R$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$((P \wedge Q) \rightarrow (P \vee \neg R_{45})) \qquad\qquad P_3$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$((P \land Q) \to (P \lor \neg R_{45}))$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$(P \wedge Q) \qquad (P \vee \neg R_{45})$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$(P \wedge Q) \qquad P \quad \neg R_{45}$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$(P \wedge Q) \qquad\qquad \neg R_{45}$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$(P \wedge Q) \qquad\qquad R_{45}$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$(P \wedge Q)$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$P \quad Q$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

$$P$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

Sentences of $\mathcal{L}_2$ are built up from sentences and/or formulae (possibly with free occurrences of variables):

$$\neg\forall x\,(Px \to \neg\exists y\,Rxy)$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

Sentences of $\mathcal{L}_2$ are built up from sentences and/or formulae (possibly with free occurrences of variables):

$$\forall x\,(Px \to \neg\exists y\,Rxy)$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

Sentences of $\mathcal{L}_2$ are built up from sentences and/or formulae (possibly with free occurrences of variables):

$$(Px \to \neg \exists y \, Rxy)$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

Sentences of $\mathcal{L}_2$ are built up from sentences and/or formulae (possibly with free occurrences of variables):

$$Px \qquad \neg\exists y\, Rxy$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

Sentences of $\mathcal{L}_2$ are built up from sentences and/or formulae (possibly with free occurrences of variables):

$$Px \qquad \exists y\, Rxy$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

Sentences of $\mathcal{L}_2$ are built up from sentences and/or formulae (possibly with free occurrences of variables):

$$Px \qquad\qquad Rxy$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

Sentences of $\mathcal{L}_2$ are built up from sentences and/or formulae (possibly with free occurrences of variables):

$$Px$$

Why do we need variable assignments? Why can't we just define truth first for atomic sentences and then for longer and longer sentences as in $\mathcal{L}_1$?

Sentences of $\mathcal{L}_1$ are built up from other sentences:

Sentences of $\mathcal{L}_2$ are built up from sentences and/or formulae (possibly with free occurrences of variables):

Definition

Let Γ be a set of sentences of $\mathcal{L}_2$ and ϕ a sentence of $\mathcal{L}_2$. The argument with all sentences in Γ as premisses and ϕ as conclusion is valid if and only if there is no $\mathcal{L}_2$-structure in which all sentences in Γ are true and ϕ is false.

Definition

Let Γ be a set of sentences of $\mathcal{L}_2$ and ϕ a sentence of $\mathcal{L}_2$. The argument with all sentences in Γ as premisses and ϕ as conclusion is valid if and only if there is no $\mathcal{L}_2$-structure in which all sentences in Γ are true and ϕ is false.

This makes precise the informal characterisation of valid arguments: in a valid argument the premisses can't be true while the conclusion is false – independently of what exists (arbitrary domain), what proper names designate and what predicate expressions mean.

Definition

Let Γ be a set of sentences of $\mathcal{L}_2$ and ϕ a sentence of $\mathcal{L}_2$. The argument with all sentences in Γ as premisses and ϕ as conclusion is valid if and only if there is no $\mathcal{L}_2$-structure in which all sentences in Γ are true and ϕ is false.

This makes precise the informal characterisation of valid arguments: in a valid argument the premisses can't be true while the conclusion is false – independently of what exists (arbitrary domain), what proper names designate and what predicate expressions mean.

That the argument with all sentences in Γ as premisses and ϕ as conclusion is valid, is abbreviated as $\Gamma \vDash \phi$.

Definition

Let Γ be a set of sentences of $\mathcal{L}_2$ and ϕ a sentence of $\mathcal{L}_2$. The argument with all sentences in Γ as premisses and ϕ as conclusion is valid if and only if there is no $\mathcal{L}_2$-structure in which all sentences in Γ are true and ϕ is false.

This makes precise the informal characterisation of valid arguments: in a valid argument the premisses can't be true while the conclusion is false – independently of what exists (arbitrary domain), what proper names designate and what predicate expressions mean.

That the argument with all sentences in Γ as premisses and ϕ as conclusion is valid, is abbreviated as $\Gamma \vDash \phi$.

Thus, $\Gamma \vDash \phi$ iff there is no $\mathcal{L}_2$-structure such that $|\phi|_{\mathcal{A}} = \mathrm{F}$ and for all sentences γ in Γ, $|\gamma|_{\mathcal{A}} = \mathrm{T}$.

In general, it's difficult to prove that an argument in $\mathcal{L}_2$ is valid by proving a claim about all $\mathcal{L}_2$-structures as there is no method to go through *all* $\mathcal{L}_2$-structures.

This is in contrast to $\mathcal{L}_1$ where one can systematically check out all $\mathcal{L}_1$-structures using truth tables.

In general, it's difficult to prove that an argument in $\mathcal{L}_2$ is valid by proving a claim about all $\mathcal{L}_2$-structures as there is no method to go through *all* $\mathcal{L}_2$-structures.

This is in contrast to $\mathcal{L}_1$ where one can systematically check out all $\mathcal{L}_1$-structures using truth tables.

In order to show that an argument in $\mathcal{L}_2$ is *not* valid, one can specify an $\mathcal{L}_2$-structure in which all premisses are true and the conclusion is false. Such an $\mathcal{L}_2$-structure is called a counterexample to the argument.

Example

$\forall x\,(P^1x \to Q^1x) \nvDash \forall x\,(\neg P^1x \to \neg Q^1x)$

The symbol $\nvDash$ is used to claim that the argument is *not* valid.

Example

$\forall x\,(P^1x \to Q^1x) \nvDash \forall x\,(\neg P^1x \to \neg Q^1x)$

The symbol $\nvDash$ is used to claim that the argument is *not* valid.

Let $\mathcal{B}$ be an $\mathcal{L}_2$-structure with $\{\text{Oxford}\}$ as its domain and

$$|P^1|_{\mathcal{A}} = \varnothing$$
$$|Q^1|_{\mathcal{A}} = \{\text{Oxford}\}$$

What $\mathcal{B}$ assigns to other constants and predicate letters doesn't matter.

Claim

$\mathcal{B}$ is a counterexample to the argument.

At first I show that the premiss is true in $\mathcal{B}$. Let α be any variable assignment over $\mathcal{B}$.

$$|x|^{\alpha}_{\mathcal{B}} \notin \varnothing$$
$$|x|^{\alpha}_{\mathcal{B}} \notin |P^1|_{\mathcal{B}}$$
$$|P^1x|^{\alpha}_{\mathcal{B}} = \mathrm{F}$$
$$|P^1x \to Q^1x|^{\alpha}_{\mathcal{B}} = \mathrm{T}$$

At first I show that the premiss is true in $\mathcal{B}$. Let α be any variable assignment over $\mathcal{B}$.

$$
\begin{aligned}
&|x|^{\alpha}_{\mathcal{B}} \notin \varnothing \\
&|x|^{\alpha}_{\mathcal{B}} \notin |P^1|_{\mathcal{B}} \\
&|P^1x|^{\alpha}_{\mathcal{B}} = \mathrm{F} \\
&|P^1x \to Q^1x|^{\alpha}_{\mathcal{B}} = \mathrm{T}
\end{aligned}
$$

So $|P^1x \to Q^1x|^{\alpha}_{\mathcal{B}} = \mathrm{T}$ for all variable assignments α over $\mathcal{B}$

At first I show that the premiss is true in $\mathcal{B}$. Let α be any variable assignment over $\mathcal{B}$.

$$|x|^{\alpha}_{\mathcal{B}} \notin \varnothing$$
$$|x|^{\alpha}_{\mathcal{B}} \notin |P^1|_{\mathcal{B}}$$
$$|P^1 x|^{\alpha}_{\mathcal{B}} = \mathrm{F}$$
$$|P^1 x \to Q^1 x|^{\alpha}_{\mathcal{B}} = \mathrm{T}$$

So $|P^1 x \to Q^1 x|^{\alpha}_{\mathcal{B}} = \mathrm{T}$ for all variable assignments α over $\mathcal{B}$ and therefore

$$|\forall x\,(P^1 x \to Q^1 x)|_{\mathcal{B}} = \mathrm{T}$$

At first I show that the premiss is true in $\mathcal{B}$. Let α be any variable assignment over $\mathcal{B}$.

$$
\begin{aligned}
&|x|^{\alpha}_{\mathcal{B}} \notin \varnothing \\
&|x|^{\alpha}_{\mathcal{B}} \notin |P^1|_{\mathcal{B}} \\
&|P^1x|^{\alpha}_{\mathcal{B}} = \mathrm{F} \\
&|P^1x \to Q^1x|^{\alpha}_{\mathcal{B}} = \mathrm{T}
\end{aligned}
$$

So $|P^1x \to Q^1x|^{\alpha}_{\mathcal{B}} = \mathrm{T}$ for all variable assignments α over $\mathcal{B}$ and therefore

$$|\forall x\,(P^1x \to Q^1x)|_{\mathcal{B}} = \mathrm{T}$$

So the premiss is true in $\mathcal{B}$.

I still need to show that $\forall x\,(\neg P^1x \to \neg Q^1x)$ is false in $\mathcal{B}$. Let β be a variable assignment over $\mathcal{B}$. Then $|x|^{\beta}_{\mathcal{B}} = \text{Oxford}$.

$$|x|^{\beta}_{\mathcal{B}} \notin \varnothing$$

$$|x|^{\beta}_{\mathcal{B}} \notin |P^1|_{\mathcal{B}}$$

$$|P^1x|^{\beta}_{\mathcal{B}} = \mathrm{F}$$

$$|\neg P^1x|^{\beta}_{\mathcal{B}} = \mathrm{T}$$

I still need to show that $\forall x\,(\neg P^1x \to \neg Q^1x)$ is false in $\mathcal{B}$. Let β be a variable assignment over $\mathcal{B}$. Then $|x|^{\beta}_{\mathcal{B}} = \text{Oxford}$.

$$\begin{aligned}
|x|^{\beta}_{\mathcal{B}} &\notin \varnothing \\
|x|^{\beta}_{\mathcal{B}} &\notin |P^1|_{\mathcal{B}} \\
|P^1x|^{\beta}_{\mathcal{B}} &= \text{F} \\
|\neg P^1x|^{\beta}_{\mathcal{B}} &= \text{T}
\end{aligned}$$

and similarly:

$$\begin{aligned}
|x|^{\beta}_{\mathcal{B}} &\in \{\text{Oxford}\} \\
|x|^{\beta}_{\mathcal{B}} &\in |Q^1|_{\mathcal{B}} \\
|Q^1x|^{\beta}_{\mathcal{B}} &= \text{T} \\
|\neg Q^1x|^{\beta}_{\mathcal{B}} &= \text{F}
\end{aligned}$$

I still need to show that $\forall x\,(\neg P^1x \to \neg Q^1x)$ is false in $\mathcal{B}$. Let β be a variable assignment over $\mathcal{B}$. Then $|x|^\beta_\mathcal{B} = \text{Oxford}$.

$$|x|^\beta_\mathcal{B} \notin \varnothing$$

$$|x|^\beta_\mathcal{B} \notin |P^1|_\mathcal{B}$$

$$|P^1x|^\beta_\mathcal{B} = \text{F}$$

$$|\neg P^1x|^\beta_\mathcal{B} = \text{T}$$

and similarly:

$$|x|^\beta_\mathcal{B} \in \{\text{Oxford}\}$$

$$|x|^\beta_\mathcal{B} \in |Q^1|_\mathcal{B}$$

$$|Q^1x|^\beta_\mathcal{B} = \text{T}$$

$$|\neg Q^1x|^\beta_\mathcal{B} = \text{F}$$

So I have $|(\neg P^1x \to \neg Q^1x)|^\beta_\mathcal{B} = \text{F}$ and therefore

$$|\forall x\,(\neg P^1x \to \neg Q^1x)|_\mathcal{B} = \text{F}$$

So the conclusion is false in $\mathcal{B}$.