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.
<table>
<thead>
<tr>
<th>Argument</th>
<th>Valid</th>
</tr>
</thead>
</table>
| (1) Zeno is a tortoise.  
(2) All tortoises are toothless.  
Therefore, (C) Zeno is toothless. |
Argument

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

Formalisation

(1) \( Pa \)
(2) \( \forall x (Px \rightarrow Qx) \)
(C) \( Qa \)

Dictionary: \( a \): Zeno. \( P \): ...is a tortoise. \( Q \): ...is toothless
**Argument**

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

**Formalisation**

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

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

What is it for this $L_2$-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 $\Gamma$ be a set of sentences of $\mathcal{L}_1$ and $\phi$ a sentence of $\mathcal{L}_1$. 
Validity

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

Let $\Gamma$ be a set of sentences of $\mathcal{L}_1$ and $\phi$ a sentence of $\mathcal{L}_1$.

**Definition**

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

(i) all sentences in $\Gamma$ are true; and

(ii) $\phi$ is false.
Recall the definition of validity for $\mathcal{L}_1$. Let $\Gamma$ be a set of sentences of $\mathcal{L}_1$ and $\phi$ a sentence of $\mathcal{L}_1$.

**Definition**

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

(i) all sentences in $\Gamma$ are true; and

(ii) $\phi$ 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 $\Gamma$ be a set of sentences of $\mathcal{L}_2$ and $\phi$ a sentence of $\mathcal{L}_2$.

**Definition**

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

(i) all sentences in $\Gamma$ are true; and

(ii) $\phi$ 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 $\Gamma$ be a set of sentences of $\mathcal{L}_2$ and $\phi$ a sentence of $\mathcal{L}_2$.

**Definition**

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

(i) all sentences in $\Gamma$ are *true*; and
(ii) $\phi$ 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.

<table>
<thead>
<tr>
<th>$\mathcal{L}_1$-structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-logical expressions in $\mathcal{L}_1$: $P, Q, R, \ldots$</td>
</tr>
</tbody>
</table>
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.

\[ \mathcal{L}_1 \text{-structures} \]

- Non-logical expressions in \( \mathcal{L}_1 \): \( P, Q, R, \ldots \)
- An \( \mathcal{L}_1 \)-structure \( A \) 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.

<table>
<thead>
<tr>
<th>(\mathcal{L}_1)-structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Non-logical expressions in (\mathcal{L}_1): (P, Q, R, \ldots)</td>
</tr>
<tr>
<td>• An (\mathcal{L}_1)-structure (A) assigns each sentence letter a semantic value (specifically, a truth-value: (T) or (F)).</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>(\mathcal{L}_2)-structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Non-logical expressions: (P^1, Q^1, R^1, \ldots)</td>
</tr>
</tbody>
</table>
**Structures**

Structures interpret non-logical expressions.

<table>
<thead>
<tr>
<th>$\mathcal{L}_1$-structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Non-logical expressions in $\mathcal{L}_1$: $P, Q, R, \ldots$</td>
</tr>
<tr>
<td>• An $\mathcal{L}_1$-structure $\mathcal{A}$ assigns each sentence letter a semantic value (specifically, a truth-value: T or F).</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$\mathcal{L}_2$-structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Non-logical expressions: $P^1, Q^1, R^1, \ldots$</td>
</tr>
<tr>
<td>$P^2, Q^2, R^2, \ldots$</td>
</tr>
<tr>
<td>$\vdots$</td>
</tr>
</tbody>
</table>
**Structures**

Structures interpret non-logical expressions.

<table>
<thead>
<tr>
<th>$\mathcal{L}_1$-structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Non-logical expressions in $\mathcal{L}_1$: $P, Q, R, \ldots$</td>
</tr>
<tr>
<td>• An $\mathcal{L}_1$-structure $\mathcal{A}$ assigns each sentence letter a semantic value (specifically, a truth-value: T or F).</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$\mathcal{L}_2$-structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Non-logical expressions: $P^1, Q^1, R^1, \ldots$</td>
</tr>
<tr>
<td>$P^2, Q^2, R^2, \ldots$</td>
</tr>
<tr>
<td>$\vdots$</td>
</tr>
<tr>
<td>$a, b, c, \ldots$</td>
</tr>
</tbody>
</table>
# Structures

Structures interpret non-logical expressions.

<table>
<thead>
<tr>
<th>$\mathcal{L}_1$-structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-logical expressions in $\mathcal{L}_1$: $P, Q, R, \ldots$</td>
</tr>
<tr>
<td>An $\mathcal{L}_1$-structure $\mathcal{A}$ assigns each sentence letter a semantic value (specifically, a truth-value: T or F).</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$\mathcal{L}_2$-structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-logical expressions: $P^1, Q^1, R^1, \ldots$ $P^2, Q^2, R^2, \ldots$ $\vdots$ $a, b, c, \ldots$</td>
</tr>
<tr>
<td>An $\mathcal{L}_2$-structure $\mathcal{A}$ assigns each predicate and constant a semantic value (specifically, what?).</td>
</tr>
</tbody>
</table>
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.

‘Benedict Cumberbatch is an actor.’
Semantics in English

Start with a semantics for simple English sentences.

‘Benedict Cumberbatch is an actor.’

The sentence is true
Semantics in English

Start with a semantics for simple English sentences.

‘Benedict Cumberbatch 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.

‘Benedict Cumberbatch 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.

‘Benedict Cumberbatch 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.

<table>
<thead>
<tr>
<th>expression</th>
<th>semantic value</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘Benedict Cumberbatch’</td>
<td>Benedict Cumberbatch</td>
</tr>
<tr>
<td>‘is an actor’</td>
<td>the property of being an actor</td>
</tr>
</tbody>
</table>
Semantics in English

Start with a semantics for simple English sentences.

‘Benedict Cumberbatch 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.

<table>
<thead>
<tr>
<th>expression</th>
<th>semantic value</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘Benedict Cumberbatch’</td>
<td>Benedict Cumberbatch</td>
</tr>
<tr>
<td>‘is an actor’</td>
<td>the property of being an actor</td>
</tr>
</tbody>
</table>

…because Cumberbatch has the property of being an actor.
Semantics in English

Start with a semantics for simple English sentences.

‘Benedict Cumberbatch 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.

<table>
<thead>
<tr>
<th>expression</th>
<th>semantic value</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘Benedict Cumberbatch ’</td>
<td>Benedict Cumberbatch</td>
</tr>
<tr>
<td>‘is an actor’</td>
<td>the property of being an actor</td>
</tr>
</tbody>
</table>

…because Cumberbatch has the property of being an actor.
…because |‘Benedict Cumberbatch ’| has |‘is an actor’|.

Notation

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

‘Mary reveres Benedict Cumberbatch’ is true iff Mary stands in the relation of *revering* to Mr Cumberbatch.
Similarly:

‘Mary reveres Benedict Cumberbatch’ is true iff
Mary stands in the relation of *revering* to Mr Cumberbatch

In other words:

\[ |\text{‘Mary reveres Benedict Cumberbatch’}| = \top \text{ iff } |\text{‘Mary’| stands in |‘reveres’| to |‘Benedict Cumberbatch’}| \]
### Semantic values for English expressions

<table>
<thead>
<tr>
<th>expression</th>
<th>semantic value</th>
</tr>
</thead>
<tbody>
<tr>
<td>designator</td>
<td>object</td>
</tr>
<tr>
<td>unary predicate</td>
<td>property (alias: unary relation)</td>
</tr>
<tr>
<td>binary predicate</td>
<td>binary relation</td>
</tr>
</tbody>
</table>

We'll take this one step further, by saying more about properties and relations.
Semantic values for English expressions

<table>
<thead>
<tr>
<th>expression</th>
<th>semantic value</th>
</tr>
</thead>
<tbody>
<tr>
<td>designator</td>
<td>object</td>
</tr>
<tr>
<td>unary predicate</td>
<td>property (alias: unary relation)</td>
</tr>
<tr>
<td>binary predicate</td>
<td>binary relation</td>
</tr>
</tbody>
</table>

Examples

- |‘Benedict Cumberbatch’| = Mr Cumberbatch
- |‘is an actor’| = the property of being an actor
- |‘reveres’| = the relation of revering
### Semantic values for English expressions

<table>
<thead>
<tr>
<th>expression</th>
<th>semantic value</th>
</tr>
</thead>
<tbody>
<tr>
<td>designator</td>
<td>object</td>
</tr>
<tr>
<td>unary predicate</td>
<td>property (alias: unary relation)</td>
</tr>
<tr>
<td>binary predicate</td>
<td>binary relation</td>
</tr>
</tbody>
</table>

#### Examples

- ‘Benedict Cumberbatch’ = Mr Cumberbatch
- ‘is an actor’ = the property of *being an actor*
- ‘reveres’ = the relation of *revering*

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 $P$. 

Example: /The property of being an actor = the set of actors = $\{d: d$ is an actor$\}$ = $\{Daniel Craig, B. Cumberbatch, ...\}$.
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

$= \{d : d \text{ is an actor}\}$

$= \{\text{Daniel Craig, B. Cumberbatch, …}\}$
Relations

Recall that we identify binary relations with sets of pairs.
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.
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 revering = \( \{ \langle d, e \rangle : d \text{ reveres } e \} \)
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 *revering* $= \{ \langle d, e \rangle : d \text{ reveres } e \}$

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 revering $= \{ \langle d, e \rangle : d$ reveres $e \}$

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 *revering* = $\{ \langle d, e \rangle : d$ reveres $e \}$

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:

‘Benedict Cumberbatch is an actor’ is true
Putting this all together:

‘Benedict Cumberbatch is an actor’ is true
iff ‘Benedict Cumberbatch’ has ‘is an actor’
Putting this all together:

‘Benedict Cumberbatch is an actor’ is true
iff ‘Benedict Cumberbatch’ has ‘is an actor’
iff Cumberbatch ∈ the set of actors
Putting this all together:

‘Benedict Cumberbatch is an actor’ is true
iff ‘Benedict Cumberbatch’| has ‘is an actor’|
iff Cumberbatch ∈ the set of actors

Similarly:

‘Mary reveres Benedict Cumberbatch’ is true
Putting this all together:

‘Benedict Cumberbatch is an actor’ is true
\[\text{iff } |\text{Benedict Cumberbatch}| \text{ has } |\text{is an actor}|\]
\[\text{iff Cumberbatch } \in \text{ the set of actors}\]

Similarly:

‘Mary reveres Benedict Cumberbatch’ is true
\[\text{iff } |\text{Mary}| \text{ stands in } |\text{reveres}| \text{ to } |\text{Benedict Cumberbatch}|\]
Putting this all together:

‘Benedict Cumberbatch is an actor’ is true
iff ‘Benedict Cumberbatch’ has ‘is an actor’
iff Cumberbatch ∈ the set of actors

Similarly:

‘Mary reveres Benedict Cumberbatch’ is true
iff ‘Mary’ stands in ‘reveres’ to ‘Benedict Cumberbatch’
iff ⟨Mary, B. Cumberbatch⟩ ∈ {⟨d, e⟩ : d reveres 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:

<table>
<thead>
<tr>
<th>$\mathcal{L}_2$-expression</th>
<th>semantic value</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant: $a$</td>
<td>object: $</td>
</tr>
<tr>
<td>sentence letter: $P$</td>
<td>truth-value: $</td>
</tr>
<tr>
<td>unary predicate letter: $P^1$</td>
<td>unary relation: $</td>
</tr>
<tr>
<td>binary predicate letter: $P^2$</td>
<td>binary relation: $</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>$\mathcal{L}_2$-expression</th>
<th>semantic value</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant: $a$</td>
<td>object: $</td>
</tr>
<tr>
<td>sentence letter: $P$</td>
<td>truth-value: $</td>
</tr>
<tr>
<td>unary predicate letter: $P^1$</td>
<td>unary relation: $</td>
</tr>
<tr>
<td>binary predicate letter: $P^2$</td>
<td>binary relation: $</td>
</tr>
</tbody>
</table>

- $|P^1b| = T$ 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:

<table>
<thead>
<tr>
<th>$\mathcal{L}_2$-expression</th>
<th>semantic value</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant: $a$</td>
<td>object: $</td>
</tr>
<tr>
<td>sentence letter: $P$</td>
<td>truth-value: $</td>
</tr>
<tr>
<td>unary predicate letter: $P^1$</td>
<td>unary relation: $</td>
</tr>
<tr>
<td>binary predicate letter: $P^2$</td>
<td>binary relation: $</td>
</tr>
</tbody>
</table>

- $|P^1b| = T$ iff $|b| \in |P^1|$
- $|R^2ab| = T$ iff $\langle |a|, |b| \rangle \in |R^2|$
Atomic Sentences

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:

<table>
<thead>
<tr>
<th>$\mathcal{L}_2$-expression</th>
<th>semantic value</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant: $a$</td>
<td>object: $</td>
</tr>
<tr>
<td>sentence letter: $P$</td>
<td>truth-value: $</td>
</tr>
<tr>
<td>unary predicate letter: $P^1$</td>
<td>unary relation: $</td>
</tr>
<tr>
<td>binary predicate letter: $P^2$</td>
<td>binary relation: $</td>
</tr>
</tbody>
</table>

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

Notation: $|e|_A$ is the semantic value of $e$ in $\mathcal{L}_2$-structure $A$. 
Semantics for atomic $L_2$-formulae

We have the semantics for $L_2$-sentences like $Pa$. 

In English: $\text{th}_\text{edesignator} \text{Benedict Cumberbatch}$ 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 $L_2$-structure $A$: $a, b, c, \ldots$ are assigned a constant semantic value in $A$. Variables: $x, y, z, \ldots$ are not. What object each variable denotes is specified with a variable assignment.
Semantics for atomic $L_2$-formulae

We have the semantics for $L_2$-sentences like $Pa$.
What about $L_2$-formulae like $Px$?
Semantics for atomic $L_2$-formulae

We have the semantics for $L_2$-sentences like $Pa$. What about $L_2$-formulae like $Px$?

In English:

- The designator ‘Benedict Cumberbatch’ 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 ‘Benedict Cumberbatch’ 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 ‘Benedict Cumberbatch’ 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 ‘Benedict Cumberbatch’ 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 \( L_2 \)-formulae**

We have the semantics for \( L_2 \)-sentences like \( Pa \).

What about \( L_2 \)-formulae like \( Px \)?

In English:

- The designator ‘Benedict Cumberbatch’ 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 \( L_2 \)-structure \( A \):

- \( a, b, c, \ldots \) are assigned a constant semantic value in \( A \).
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 ‘Benedict Cumberbatch’ 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 $L_2$-formulae

We have the semantics for $L_2$-sentences like $Pa$. What about $L_2$-formulae like $Px$?

In English:

- The designator ‘Benedict Cumberbatch’ 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 $L_2$-structure $A$:

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

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 $\alpha$.

\[
\begin{array}{cccccccc}
  x & y & z & x_1 & y_1 & z_1 & x_2 & \ldots \\
  \text{Mercury} & \text{Venus} & \text{Venus} & \text{Neptune} & \text{Mars} & \text{Venus} & \text{Mars} & \ldots \\
\end{array}
\]
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 $\alpha$.

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

Notation

We write $|x|^\alpha$ for the object $\alpha$ 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 $\alpha$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>$x_1$</th>
<th>$y_1$</th>
<th>$z_1$</th>
<th>$x_2$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>Venus</td>
<td>Venus</td>
<td>Neptune</td>
<td>Mars</td>
<td>Venus</td>
<td>Mars</td>
<td>...</td>
</tr>
</tbody>
</table>

Notation

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

We use lower case Greek letters: $\alpha, \beta, \gamma$ 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 $\alpha$.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>Mercury</td>
</tr>
<tr>
<td>$y$</td>
<td>Venus</td>
</tr>
<tr>
<td>$z$</td>
<td>Venus</td>
</tr>
<tr>
<td>$x_1$</td>
<td>Neptune</td>
</tr>
<tr>
<td>$y_1$</td>
<td>Mars</td>
</tr>
<tr>
<td>$z_1$</td>
<td>Venus</td>
</tr>
<tr>
<td>$x_2$</td>
<td>Mars</td>
</tr>
</tbody>
</table>

Notation

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

We use lower case Greek letters: $\alpha$, $\beta$, $\gamma$ 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 $\alpha$.

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>$x_1$</th>
<th>$y_1$</th>
<th>$z_1$</th>
<th>$x_2$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mercury</td>
<td>Venus</td>
<td>Venus</td>
<td>Neptune</td>
<td>Mars</td>
<td>Venus</td>
<td>Mars</td>
<td>...</td>
</tr>
</tbody>
</table>

Notation

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

We use lower case Greek letters: $\alpha$, $\beta$, $\gamma$ 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 $\alpha$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>$x_1$</th>
<th>$y_1$</th>
<th>$z_1$</th>
<th>$x_2$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>Venus</td>
<td>Venus</td>
<td>Neptune</td>
<td>Mars</td>
<td>Venus</td>
<td>Mars</td>
<td>...</td>
</tr>
</tbody>
</table>

Notation

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

We use lower case Greek letters: $\alpha$, $\beta$, $\gamma$ for assignments.

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

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

Notation

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

We use lower case Greek letters: $\alpha, \beta, \gamma$ for assignments.

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

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>$x_1$</th>
<th>$y_1$</th>
<th>$z_1$</th>
<th>$x_2$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>Venus</td>
<td>Venus</td>
<td>Neptune</td>
<td>Mars</td>
<td>Venus</td>
<td>Mars</td>
<td>...</td>
</tr>
</tbody>
</table>

Notation
We write $|x|^\alpha$ for the object $\alpha$ assigns to $x$.
We use lower case Greek letters: $\alpha, \beta, \gamma$ for assignments.

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

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>$y$</td>
<td>$z$</td>
<td>$x_1$</td>
<td>$y_1$</td>
<td>$z_1$</td>
<td>$x_2$</td>
</tr>
<tr>
<td>Mercury</td>
<td>Venus</td>
<td>Venus</td>
<td>Neptune</td>
<td>Mars</td>
<td>Venus</td>
<td>Mars</td>
</tr>
</tbody>
</table>

Notation
We write $|x|^\alpha$ for the object $\alpha$ assigns to $x$.
We use lower case Greek letters: $\alpha$, $\beta$, $\gamma$ 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 \( |e|_A^\alpha \) for the semantic value of expression \( e \) in the structure \( A \) under the variable assignment \( \alpha \).
Once $x$ has been assigned an object, the semantics for $Px$ are much like the semantics for $Pa$.

We write $|e|^\alpha_A$ for the semantic value of expression $e$ in the structure $A$ under the variable assignment $\alpha$.

- $|Px|^\alpha_A = T$ iff $|x|^\alpha \in |P^1|^A$  
  (NB: $|x|^\alpha_A = |x|^\alpha$)
Once $x$ has been assigned an object, the semantics for $Px$ are much like the semantics for $Pa$.

We write $|e|^\alpha_A$ for the semantic value of expression $e$ in the structure $A$ under the variable assignment $\alpha$.

- $|Px|^\alpha_A = T \text{ iff } |x|^\alpha \in |P^1|_A$  
  (NB: $|x|^\alpha_A = |x|^\alpha$)
- $|Rx\,y|^\alpha_A = T \text{ iff } \langle |x|^\alpha, |y|^\alpha \rangle \in |R^2|_A$
Once \( x \) has been assigned an object, the semantics for \( Px \) are much like the semantics for \( Pa \).

We write \(|e|^{\alpha}_A\) for the semantic value of expression \( e \) in the structure \( A \) under the variable assignment \( \alpha \).

- \(|Px|^{\alpha}_A = \text{true} \iff |x|^{\alpha} \in |P^1|_A\)  
  \hspace{1cm} \text{(NB: } |x|^{\alpha}_A = |x|^{\alpha})

- \(|Rx\, y|^{\alpha}_A = \text{true} \iff \langle |x|^{\alpha}, |y|^{\alpha} \rangle \in |R^2|_A\)

Note: semantic values of constants and predicates are unaffected by the assignment (e.g. \(|P|^{\alpha}_A = |P|_A\), \(|a|^{\alpha}_A = |a|_A\)).
Once $x$ has been assigned an object, the semantics for $Px$ are much like the semantics for $Pa$.

We write $|e|_A^\alpha$ for the semantic value of expression $e$ in the structure $A$ under the variable assignment $\alpha$.

- $|Px|_A^\alpha = T \iff |x|_A^\alpha \in |P|_A$  
  (NB: $|x|_A^\alpha = |x|_A$)
- $|Rx y|_A^\alpha = T \iff \langle |x|_A^\alpha, |y|_A^\alpha \rangle \in |R^2|_A$

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

- $|Rab|_A^\alpha = T \iff \langle |a|_A, |b|_A \rangle \in |R|_A$
Once $x$ has been assigned an object, the semantics for $Px$ are much like the semantics for $Pa$.

We write $|e|^\alpha_A$ for the semantic value of expression $e$ in the structure $A$ under the variable assignment $\alpha$.

- $|Px|^\alpha_A = \text{T iff } |x|^\alpha \in |P|^1_A$  
  (NB: $|x|_A^\alpha = |x|_A^\alpha$)

- $|Rx\,y|^\alpha_A = \text{T iff } \langle |x|_A^\alpha, |y|_A^\alpha \rangle \in |R|^2_A$

Note: semantic values of constants and predicates are unaffected by the assignment (e.g. $|P|^\alpha_A = |P|_A$, $|a|^\alpha_A = |a|_A$).

- $|Rab|^\alpha_A = \text{T iff } \langle |a|_A, |b|_A \rangle \in |R|_A$

- $|Rx\,b|^\alpha_A = \text{T iff } \langle |x|_A^\alpha, |b|_A \rangle \in |R|_A$
Once $x$ has been assigned an object, the semantics for $Px$ are much like the semantics for $Pa$.

We write $|e|_A^\alpha$ for the semantic value of expression $e$ in the structure $A$ under the variable assignment $\alpha$.

- $|Px|_A^\alpha = \text{T iff } |x|^\alpha \in |P^1|_A$  
  (NB: $|x|_A^\alpha = |x|^\alpha$)
- $|Rx y|_A^\alpha = \text{T iff } \langle |x|^\alpha, |y|^\alpha \rangle \in |R^2|_A$

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

- $|Rab|_A^\alpha = \text{T iff } \langle |a|_A, |b|_A \rangle \in |R|_A$
- $|Rx b|_A^\alpha = \text{T iff } \langle |x|^\alpha, |b|_A \rangle \in |R|_A$

Similarly for other atomic formulae.
Worked example

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

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

Let assignments $\alpha$ and $\beta$ be such that:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$:</td>
<td>Saturn</td>
<td>Mars</td>
<td>Jupiter</td>
</tr>
<tr>
<td>$\beta$:</td>
<td>Venus</td>
<td>Venus</td>
<td>Venus</td>
</tr>
</tbody>
</table>
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 $\alpha$ and $\beta$ be such that:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>Saturn</td>
<td>Mars</td>
<td>Jupiter</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Venus</td>
<td>Venus</td>
<td>Venus</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x|_\mathcal{A}^\alpha = \text{Saturn}$
- $|x|_\mathcal{A}^\beta = \text{Venus}$
- $|a|_\mathcal{A}^\alpha = \text{Venus}$
- $|P|_\mathcal{A}^\alpha = \text{T}$
- $|P|_\mathcal{A}^\beta = \text{F}$
- $|Rx|_\mathcal{A}^\alpha = \text{F}$
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 $\alpha$ and $\beta$ be such that:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$:</td>
<td>Saturn</td>
<td>Mars</td>
<td>Jupiter</td>
</tr>
<tr>
<td>$\beta$:</td>
<td>Venus</td>
<td>Venus</td>
<td>Venus</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x\alpha|_\mathcal{A} = \text{Saturn}$
- $|x\beta|_\mathcal{A} = \text{Venus}$
- $|a\alpha|_\mathcal{A} = \text{Venus}$
- $|P^1y\alpha|_\mathcal{A} = T$
- $|P^1y\beta|_\mathcal{A} = F$
- $|Pb\alpha|_\mathcal{A} = T$
- $|Rx y\alpha|_\mathcal{A} = F$
- $|Rx y\beta|_\mathcal{A} = F$
- $|Rx b\alpha|_\mathcal{A} = F$
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 $\alpha$ and $\beta$ be such that:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$: Saturn</td>
<td>Mars</td>
<td>Jupiter</td>
</tr>
<tr>
<td>$\beta$: Venus</td>
<td>Venus</td>
<td>Venus</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x\alpha|_\mathcal{A} = \text{Saturn}$
- $|x\beta|_\mathcal{A} = \text{Venus}$
- $|a\alpha|_\mathcal{A} = \text{Venus}$
- $|P y\alpha|_\mathcal{A} = \text{Saturn}$
- $|P y\beta|_\mathcal{A} = \text{Venus}$
- $|P b\alpha|_\mathcal{A} = \text{Venus}$
- $|R x y\alpha|_\mathcal{A} = \text{Saturn}$
- $|R x y\beta|_\mathcal{A} = \text{Venus}$
- $|R x b\alpha|_\mathcal{A} = \text{Venus}$
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 $\alpha$ and $\beta$ be such that:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>Saturn</td>
<td>Mars</td>
<td>Jupiter</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Venus</td>
<td>Venus</td>
<td>Venus</td>
</tr>
</tbody>
</table>

Compute the following:

|   | $|x|_\mathcal{A}^\alpha$ = Saturn | $|x|_\mathcal{A}^\beta$ = Venus | $|a|_\mathcal{A}^\alpha$ = Venus |
|---|----------------------------------|-------------------------------|-------------------------------|
| $|P y|_\mathcal{A}^\alpha$ | $|P y|_\mathcal{A}^\beta$ | $|P b|_\mathcal{A}^\alpha$ | $|R x y|_\mathcal{A}^\alpha$ | $|R x y|_\mathcal{A}^\beta$ | $|R x b|_\mathcal{A}^\alpha$ |
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 $\alpha$ and $\beta$ be such that:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$:</td>
<td>Saturn</td>
<td>Mars</td>
<td>Jupiter</td>
</tr>
<tr>
<td>$\beta$:</td>
<td>Venus</td>
<td>Venus</td>
<td>Venus</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x^\alpha|_\mathcal{A} = \text{Saturn}$
- $|x^\beta|_\mathcal{A} = \text{Venus}$
- $|a^\alpha|_\mathcal{A} = \text{Venus}$
- $|P^1y^\alpha|_\mathcal{A} = \text{T}$
- $|P^1y^\beta|_\mathcal{A} = \text{F}$
- $|P^1b^\alpha|_\mathcal{A} = \text{F}$
- $|Rx^\alpha y|_\mathcal{A} = \text{F}$
- $|Rx^\beta y|_\mathcal{A} = \text{F}$
- $|Rx^\alpha b|_\mathcal{A} = \text{F}$
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 $\alpha$ and $\beta$ be such that:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$:</td>
<td>Saturn</td>
<td>Mars</td>
</tr>
<tr>
<td>$\beta$:</td>
<td>Venus</td>
<td>Venus</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x|_\mathcal{A}^\alpha = \text{Saturn}$
- $|x|_\mathcal{A}^\beta = \text{Venus}$
- $|a|_\mathcal{A}^\alpha = \text{Venus}$
- $|P y|_\mathcal{A}^\alpha = T$
- $|P y|_\mathcal{A}^\beta = F$
- $|P b|_\mathcal{A}^\alpha = $
- $|R x y|_\mathcal{A}^\alpha = $
- $|R x y|_\mathcal{A}^\beta = $
- $|R x b|_\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 $\alpha$ and $\beta$ be such that:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$:</td>
<td>Saturn</td>
<td>Mars</td>
<td>Jupiter</td>
</tr>
<tr>
<td>$\beta$:</td>
<td>Venus</td>
<td>Venus</td>
<td>Venus</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x|_\mathcal{A}^\alpha = \text{Saturn}$
- $|x|_\mathcal{A}^\beta = \text{Venus}$
- $|a|_\mathcal{A}^\alpha = \text{Venus}$
- $|P y|_\mathcal{A}^\alpha = T$
- $|P y|_\mathcal{A}^\beta = F$
- $|P b|_\mathcal{A}^\alpha = T$
- $|R x y|_\mathcal{A}^\alpha = $
- $|R x y|_\mathcal{A}^\beta = $
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 $\alpha$ and $\beta$ be such that:

\[
\begin{array}{ccc}
x & y & z \\
\alpha: & \text{Saturn} & \text{Mars} & \text{Jupiter} \\
\beta: & \text{Venus} & \text{Venus} & \text{Venus}
\end{array}
\]

Compute the following:

- $|x|_\mathcal{A}^\alpha = \text{Saturn}$
- $|x|_\mathcal{A}^\beta = \text{Venus}$
- $|a|_\mathcal{A}^\alpha = \text{Venus}$
- $|P y|_\mathcal{A}^\alpha = \text{T}$
- $|P y|_\mathcal{A}^\beta = \text{F}$
- $|P b|_\mathcal{A}^\alpha = \text{T}$
- $|R x y|_\mathcal{A}^\alpha = \text{F}$
- $|R x y|_\mathcal{A}^\beta =$
- $|R x b|_\mathcal{A}^\alpha =$
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 $\alpha$ and $\beta$ be such that:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$:</td>
<td>Saturn</td>
<td>Mars</td>
<td>Jupiter</td>
</tr>
<tr>
<td>$\beta$:</td>
<td>Venus</td>
<td>Venus</td>
<td>Venus</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x|_\mathcal{A}^\alpha = \text{Saturn}$
- $|x|_\mathcal{A}^\beta = \text{Venus}$
- $|a|_\mathcal{A}^\alpha = \text{Venus}$
- $|P^1y|_\mathcal{A}^\alpha = T$
- $|P^1y|_\mathcal{A}^\beta = F$
- $|Pb|_\mathcal{A}^\alpha = T$
- $|Rx^2y|_\mathcal{A}^\alpha = F$
- $|Rx^2y|_\mathcal{A}^\beta = F$
- $|Rx^2b|_\mathcal{A}^\alpha = \text{blank}$
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 $\alpha$ and $\beta$ be such that:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$:</td>
<td>Saturn</td>
<td>Mars</td>
<td>Jupiter</td>
</tr>
<tr>
<td>$\beta$:</td>
<td>Venus</td>
<td>Venus</td>
<td>Venus</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x|_\mathcal{A}^\alpha = \text{Saturn}$
- $|x|_\mathcal{A}^\beta = \text{Venus}$
- $|a|_\mathcal{A}^\alpha = \text{Venus}$
- $|P y|_\mathcal{A}^\alpha = \text{T}$
- $|P y|_\mathcal{A}^\beta = \text{F}$
- $|P b|_\mathcal{A}^\alpha = \text{T}$
- $|R x y|_\mathcal{A}^\alpha = \text{F}$
- $|R x y|_\mathcal{A}^\beta = \text{F}$
- $|Rx b|_\mathcal{A}^\alpha = \text{F}$
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 who ‘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 who ‘everyone’ ranges over.

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 who ‘everyone’ ranges over.

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 who ‘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.
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 who ‘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. An assignment over $\mathcal{A}$ assigns a member of $D_\mathcal{A}$ to each variable.
An $L_2$-structure $\mathcal{A}$ specifies a non-empty set $D_\mathcal{A}$ as the domain. An assignment over $\mathcal{A}$ assigns a member of $D_\mathcal{A}$ to each variable.

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

$|\forall x P x|_\mathcal{A} = T$

iff every member of $D_\mathcal{A}$ is in $|P|_\mathcal{A}$
An $L_2$-structure $\mathcal{A}$ specifies a non-empty set $D_\mathcal{A}$ as the domain. An *assignment over* $\mathcal{A}$ assigns a member of $D_\mathcal{A}$ to each variable.

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

\[
\left| \forall x P x \right|_{\mathcal{A}} = T
\]
iff every member of $D_\mathcal{A}$ is in $|P|_{\mathcal{A}}$
iff every assignment $\alpha$ 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. An assignment over $\mathcal{A}$ assigns a member of $D_\mathcal{A}$ to each variable.

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

$$|\forall x P x|_\mathcal{A} = T$$

iff every member of $D_\mathcal{A}$ is in $|P|_\mathcal{A}$

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

iff every assignment $\alpha$ over $\mathcal{A}$ is such that $|P x|^\alpha = T$
An $\mathcal{L}_2$-structure $\mathcal{A}$ specifies a non-empty set $D_\mathcal{A}$ as the domain. An assignment over $\mathcal{A}$ assigns a member of $D_\mathcal{A}$ to each variable.

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

$|\forall x P x|_\mathcal{A} = T$

iff every member of $D_\mathcal{A}$ is in $|P|_\mathcal{A}$

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

iff every assignment $\alpha$ over $\mathcal{A}$ is such that $|P x|^\alpha = T$

Similarly:

$|\exists x P x|_\mathcal{A} = T$

iff some member of $D_\mathcal{A}$ is in $|P|_\mathcal{A}$
An $L_2$-structure $\mathcal{A}$ specifies a non-empty set $D_\mathcal{A}$ as the domain. An assignment over $\mathcal{A}$ assigns a member of $D_\mathcal{A}$ to each variable.

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

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

Similarly:

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

$T_{\text{h}}$ is correct but the general case is more complex.
An $\mathcal{L}_2$-structure $\mathcal{A}$ specifies a non-empty set $D_\mathcal{A}$ as the domain. An assignment over $\mathcal{A}$ assigns a member of $D_\mathcal{A}$ to each variable.

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

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

Similarly:

\[ |\exists x P_x|_\mathcal{A} = T \]
iff some member of $D_\mathcal{A}$ is in $|P|_\mathcal{A}$
iff some assignment $\alpha$ of $x$ to a member of $D_\mathcal{A}$ is such that $|x|^\alpha \in |P|_\mathcal{A}$
iff some assignment $\alpha$ over $\mathcal{A}$ is such that $|P_x|^\alpha_\mathcal{A} = T$
An $L_2$-structure $\mathcal{A}$ specifies a non-empty set $D_\mathcal{A}$ as the domain. An *assignment over* $\mathcal{A}$ assigns a member of $D_\mathcal{A}$ to each variable.

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

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

Similarly:

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

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 $A$ with domain $D_A$. 
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 $D_{\mathcal{A}}$.

$$|\forall x \exists y Rxy|_{\mathcal{A}} = T$$

iff every assignment $\alpha$ over $\mathcal{A}$ is such that $|\exists y Rxy|_{\mathcal{A}}^{\alpha} = T$
The semantics of quantifiers is complicated by the need to deal with multiple quantifiers in sentences such as $\forall x \exists y Rx y$.

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

$$|\forall x \exists y Rx y|_{\mathcal{A}} = T$$

iff every assignment $\alpha$ over $\mathcal{A}$ is such that $|\exists y Rx y|_{\mathcal{A}}^{\alpha} = T$

To progress any further we need to be able evaluate $\exists y Rx y$ under an assignment $\alpha$ of an object to $x$. 
How to determine $|\exists y Rx y|_A^\alpha$?
How to determine $|\exists y Rx y|^\alpha_A$?

$|\exists y Rx y|^\alpha_A = T$

iff some $d$ in $D_A$ is such that $\langle |x|^\alpha, d \rangle \in |R|^A$
How to determine $|\exists yRx y|^\alpha_A$?

$|\exists yRx y|^\alpha_A = T$

iff some $d$ in $D_A$ is such that $\langle |x|^\alpha, d \rangle \in |R|_A$

iff some assignment $\beta$ over $A$ is such that $\langle |x|^\alpha, |y|^\beta \rangle \in |R|_A$
How to determine $|\exists y Rx y|_A^\alpha$?

$|\exists y Rx y|_A^\alpha = T$

iff some $d$ in $D_A$ is such that $\langle |x|^\alpha, d \rangle \in |R|_A$

iff some assignment $\beta$ over $A$ is such that $\langle |x|^\alpha, |y|^\beta \rangle \in |R|_A$

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

Say that $\beta$ differs from $\alpha$ in $y$ at most if $|\nu|^\alpha = |\nu|^\beta$ for all variables $\nu$ with the possible exception of $y$. 
How to determine $|\exists yRx y|^\alpha_A$?

$|\exists yRx y|^\alpha_A = T$

iff some $d$ in $D_A$ is such that $\langle |x|^\alpha, d \rangle \in |R|_A$

iff some assignment $\beta$ over $A$ is such that $\langle |x|^\alpha, |y|^\beta \rangle \in |R|_A$

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

Say that $\beta$ differs from $\alpha$ in $y$ at most if $|\nu|^\alpha = |\nu|^\beta$ for all variables $\nu$ with the possible exception of $y$.

$|\exists yRx y|^\alpha_A = T$

iff some assignment $\beta$ over $A$ which differs from $\alpha$ in $y$ at most
is such that $\langle |x|^\alpha, |y|^\beta \rangle \in |R|_A$
How to determine $|\exists yRx y|_A^\alpha$?

$|\exists yRx y|_A^\alpha = T$
iff some $d$ in $D_A$ is such that $\langle |x|^\alpha, d \rangle \in |R|_A$
iff some assignment $\beta$ over $A$ is such that $\langle |x|^\alpha, |y|^\beta \rangle \in |R|_A$

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

Say that $\beta$ differs from $\alpha$ in $y$ at most if $|\nu|^\alpha = |\nu|^\beta$ for all variables $\nu$ with the possible exception of $y$.

$|\exists yRx y|_A^\alpha = T$
iff some assignment $\beta$ over $A$ which differs from $\alpha$ in $y$ at most
is such that $\langle |x|^\alpha, |y|^\beta \rangle \in |R|_A$
How to determine $|∃yRx y|^α_A$?

$|∃yRx y|^α_A = T$

iff some $d$ in $D_A$ is such that $⟨|x|^α, d⟩ ∈ |R|^A$

iff some assignment $β$ over $A$ is such that $⟨|x|^α, |y|^β⟩ ∈ |R|^A$

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

Say that $β$ differs from $α$ in $y$ at most if $|ν|^α = |ν|^β$ for all variables $ν$ with the possible exception of $y$.

$|∃yRx y|^α_A = T$

iff some assignment $β$ over $A$ which differs from $α$ in $y$ at most
is such that $⟨|x|^β, |y|^β⟩ ∈ |R|^A$
How to determine $|\exists y Rx y|^\alpha_A$?

$|\exists y Rx y|^\alpha_A = T$

iff some $d$ in $D_A$ is such that $\langle |x|^\alpha, d \rangle \in |R|^A$

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

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

Say that $\beta$ differs from $\alpha$ in $y$ at most if $|\nu|^\alpha = |\nu|^\beta$ for all variables $\nu$ with the possible exception of $y$.

$|\exists y Rx y|^\alpha_A = T$

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

iff some assignment $\beta$ over $A$ which differs from $\alpha$ in $y$ at most is such that $|Rx y|^\beta_A = T$
Here's the full specification of an $\mathcal{L}_2$-structure.
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 \( D_A \)
Here’s the full specification of an $\mathcal{L}_2$-structure.

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

1. a domain: a non-empty set $D_A$
2. a semantic value for each predicate 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 and constant.

<table>
<thead>
<tr>
<th>$\mathcal{L}_2$-expression</th>
<th>semantic value in $\mathcal{A}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant: $a$</td>
<td>object: $</td>
</tr>
<tr>
<td>sentence letter: $P$</td>
<td>truth-value: $</td>
</tr>
<tr>
<td>unary predicate letter: $P^1$</td>
<td>unary relation: $</td>
</tr>
<tr>
<td>binary predicate letter: $P^2$</td>
<td>binary relation: $</td>
</tr>
<tr>
<td>ternary predicate letter: $P^3$</td>
<td>ternary relation: $</td>
</tr>
<tr>
<td>etc.</td>
<td></td>
</tr>
</tbody>
</table>
Summary of semantics of $\mathcal{L}_2$

Let $\mathcal{A}$ be an $\mathcal{L}_2$-structure and $\alpha$ an assignment over $\mathcal{A}$. 
Summary of semantics of $\mathcal{L}_2$

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

### Atomic formulae

Let $\Phi^n$ 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 $\alpha$ an assignment over $\mathcal{A}$.

**Atomic formulae**

Let $\Phi^n$ 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 $\Phi^n$ by $\mathcal{A}$. 
Summary of semantics of $\mathcal{L}_2$

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

Atomic formulae

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

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

(i) $|\Phi t_1|_\mathcal{A}^\alpha = T$ if and only if $|t_1|_\mathcal{A}^\alpha \in |\Phi|_\mathcal{A}$
Summary of semantics of $\mathcal{L}_2$

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

**Atomic formulae**

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

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

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

$|\Phi^2 t_1 t_2|_{\mathcal{A}}^\alpha = T$ if and only if $\langle |t_1|_{\mathcal{A}}^\alpha, |t_2|_{\mathcal{A}}^\alpha \rangle \in |\Phi^2|_{\mathcal{A}}$
Summary of semantics of $\mathcal{L}_2$

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

**Atomic formulae**

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

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

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

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

$|\Phi^3 t_1 t_2 t_3|_\mathcal{A}^\alpha = T$ if and only if $\langle |t_1|_\mathcal{A}^\alpha, |t_2|_\mathcal{A}^\alpha, |t_3|_\mathcal{A}^\alpha \rangle \in |\Phi^3|_\mathcal{A}$
Summary of semantics of $\mathcal{L}_2$

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

**Atomic formulae**

Let $\Phi^n$ 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 $\Phi^n$ 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 $\alpha$ if $t$ is a variable.

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

<table>
<thead>
<tr>
<th>Rule</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ii)</td>
<td>$</td>
</tr>
<tr>
<td>(iii)</td>
<td>$</td>
</tr>
<tr>
<td>(iv)</td>
<td>$</td>
</tr>
<tr>
<td>(v)</td>
<td>$</td>
</tr>
<tr>
<td>(vi)</td>
<td>$</td>
</tr>
</tbody>
</table>
These are the semantic clauses for $\forall \nu$ and $\exists \nu$. 
These are the semantic clauses for $\forall \nu$ and $\exists \nu$.

<table>
<thead>
<tr>
<th>Quantifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>(vii) $</td>
</tr>
</tbody>
</table>
These are the semantic clauses for $\forall \nu$ and $\exists \nu$.

<table>
<thead>
<tr>
<th>Quantifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>(vii) $</td>
</tr>
<tr>
<td>(viii) $</td>
</tr>
</tbody>
</table>
These are the semantic clauses for $\forall \nu$ and $\exists \nu$.

### Quantifiers

<table>
<thead>
<tr>
<th>Clause</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(vii)</td>
<td>$</td>
</tr>
<tr>
<td>(viii)</td>
<td>$</td>
</tr>
</tbody>
</table>

These clauses determine the truth value of any formula in a structure $A$ under some variable assignment $\alpha$ over $A$ inductively.
These are the semantic clauses for $\forall \nu$ and $\exists \nu$.

<table>
<thead>
<tr>
<th>Quantifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>(vii) $</td>
</tr>
<tr>
<td>(viii) $</td>
</tr>
</tbody>
</table>

These clauses determine the truth value of any formula in a structure $A$ under some variable assignment $\alpha$ over $A$ 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 $L_2$-structure $A$. 
Truth

We haven’t yet said what it is for a sentence to be true in an $L_2$-structure $A$.

We’ve said what it is for a formula to be true in an $L_2$-structure $A$ under an assignment over $A$. 

Truth

We haven’t yet said what it is for a sentence to be true in an $L_2$-structure $A$.

We’ve said what it is for a formula to be true in an $L_2$-structure $A$ under an assignment over $A$.

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

We haven’t yet said what it is for a *sentence* to be *true* in an \( L_2 \)-structure \( A \).

We’ve said what it is for a *formula* to be true in an \( L_2 \)-structure \( A \) *under an assignment* over \( A \).

(We’ve defined \( |\phi|^\alpha_A \); we want now to define \( |\phi|_A \).)

**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 \( A \).

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

(We’ve defined \( \models_A^\alpha \); we want now to define \( \models_A \).)

**Fact about sentences**

The truth-value of a sentence does *not* depend on the assignment.
For \( \alpha \) and \( \beta \) over \( A \): \( \models_A^\alpha = \models_A^\beta \) (when \( \phi \) is a sentence).
Truth

We haven’t yet said what it is for a sentence to be true in an \(L_2\)-structure \(A\).

We’ve said what it is for a formula to be true in an \(L_2\)-structure \(A\) under an assignment over \(A\).

(We’ve defined \(\models^\alpha_A\); we want now to define \(\models_A\).)

Fact about sentences

The truth-value of a sentence does not depend on the assignment.
For \(\alpha\) and \(\beta\) over \(A\): \(\models^\alpha_A = \models^\beta_A\) (when \(\phi\) is a sentence).

A sentence \(\phi\) is true in an \(L_2\)-structure \(A\) (in symbols: \(\models_A = T\))
iff \(\models^\alpha_A = T\) for all variable assignments \(\alpha\) over \(A\).
**Truth**

We haven’t yet said what it is for a sentence to be *true* in an $L_2$-structure $\mathcal{A}$.

We’ve said what it is for a *formula* to be true in an $L_2$-structure $\mathcal{A}$ under an assignment over $\mathcal{A}$.

(We’ve defined $|\phi|^\alpha_A$; we want now to define $|\phi|_A$.)

**Fact about sentences**

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

For $\alpha$ and $\beta$ over $\mathcal{A}$: $|\phi|^\alpha_A = |\phi|^\beta_A$ (when $\phi$ is a sentence).

A sentence $\phi$ is *true in an $L_2$-structure $\mathcal{A}$* (in symbols: $|\phi|_A = T$) iff $|\phi|^\alpha_A = T$ for all variable assignments $\alpha$ over $\mathcal{A}$.

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

We haven’t yet said what it is for a *sentence* to be *true* in an $L_2$-structure $A$.

We’ve said what it is for a *formula* to be true in an $L_2$-structure $A$ *under an assignment over* $A$.

(We’ve defined $|\phi|^\alpha_A$; we want now to define $|\phi|_A$.)

**Fact about sentences**

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

For $\alpha$ and $\beta$ over $A$: $|\phi|^\alpha_A = |\phi|^\beta_A$ (when $\phi$ is a sentence).

A sentence $\phi$ is *true in an* $L_2$-structure $A$ (in symbols: $|\phi|_A = T$) iff $|\phi|^\alpha_A = T$ for all variable assignments $\alpha$ over $A$.

equivalently: $|\phi|^\alpha_A = T$ for some variable assignment $\alpha$ over $A$.

Now you know what truth is.
Definition

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

Let $\Gamma$ be a set of sentences of $\mathcal{L}_2$ and $\phi$ a sentence of $\mathcal{L}_2$. The argument with all sentences in $\Gamma$ as premises and $\phi$ as conclusion is valid if and only if there is no $\mathcal{L}_2$-structure in which all sentences in $\Gamma$ are true and $\phi$ 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 $\Gamma$ be a set of sentences of $\mathcal{L}_2$ and $\phi$ a sentence of $\mathcal{L}_2$. The argument with all sentences in $\Gamma$ as premisses and $\phi$ as conclusion is valid if and only if there is no $\mathcal{L}_2$-structure in which all sentences in $\Gamma$ are true and $\phi$ 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 $\Gamma$ as premisses and $\phi$ as conclusion is valid, is abbreviated as $\Gamma \models \phi$. 
Definition

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

Thus, $\Gamma \models \phi$ iff there is no $\mathcal{L}_2$-structure such that $|\phi|_A = F$ and for all sentences $\gamma$ in $\Gamma$, $|\gamma|_A = T$. 
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

\[ \forall x (P^1 x \to Q^1 x) \not\equiv \forall x (\neg P^1 x \to \neg Q^1 x) \]

The symbol \( \not\equiv \) is used to claim that the argument is *not* valid.
Example

\[ \forall x (P^1 x \rightarrow Q^1 x) \not\equiv \forall x (\neg P^1 x \rightarrow \neg Q^1 x) \]

The symbol \( \not\equiv \) is used to claim that the argument is \textit{not} valid.

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

\[
|P^1|_A = \emptyset \\
|Q^1|_A = \{\text{Oxford}\}
\]

What \( B \) assigns to other constants and predicate letters doesn’t matter.

Claim

\( B \) is a counterexample to the argument.
At first I show that the premiss is true in $\mathcal{B}$. Let $\alpha$ be any variable assignment over $\mathcal{B}$.

\[ x^\alpha_\mathcal{B} \notin \emptyset \]
\[ x^\alpha_\mathcal{B} \notin P^1|_\mathcal{B} \]
\[ P^1x^\alpha_\mathcal{B} = F \]
\[ P^1x \rightarrow Q^1x|^\alpha_\mathcal{B} = T \]
At first I show that the premiss is true in $\mathcal{B}$. Let $\alpha$ be any variable assignment over $\mathcal{B}$.

\[
\begin{align*}
|x|^\alpha_\mathcal{B} & \notin \emptyset \\
|x|^\alpha_\mathcal{B} & \notin |P^1|_\mathcal{B} \\
|P^1x|^\alpha_\mathcal{B} & = T \\
|P^1x \rightarrow Q^1x|^\alpha_\mathcal{B} & = T
\end{align*}
\]

So $|P^1x \rightarrow Q^1x|^\alpha_\mathcal{B} = T$ for all variable assignments $\alpha$ over $\mathcal{B}$.
At first I show that the premiss is true in $\mathcal{B}$. Let $\alpha$ be any variable assignment over $\mathcal{B}$.

\[
\begin{align*}
|x|^\alpha_\mathcal{B} & \notin \emptyset \\
|x|^\alpha_\mathcal{B} & \notin |P^1|_\mathcal{B} \\
|P^1x|^\alpha_\mathcal{B} & = F \\
|P^1x \rightarrow Q^1x|^\alpha_\mathcal{B} & = T
\end{align*}
\]

So $|P^1x \rightarrow Q^1x|^\alpha_\mathcal{B} = T$ for all variable assignments $\alpha$ over $\mathcal{B}$ and therefore

\[|\forall x (P^1x \rightarrow Q^1x)|_\mathcal{B} = T\]
At first I show that the premiss is true in \( \mathcal{B} \). Let \( \alpha \) be any variable assignment over \( \mathcal{B} \).

\[
\begin{align*}
|x|^\alpha_{\mathcal{B}} & \notin \emptyset \\
|x|^\alpha_{\mathcal{B}} & \notin |P^1|_{\mathcal{B}} \\
|P^1x|^\alpha_{\mathcal{B}} & = F \\
|P^1x \rightarrow Q^1x|^\alpha_{\mathcal{B}} & = T
\end{align*}
\]

So \( |P^1x \rightarrow Q^1x|_\mathcal{B}^\alpha = T \) for all variable assignments \( \alpha \) over \( \mathcal{B} \) and therefore

\[
|\forall x \ (P^1x \rightarrow Q^1x)|_\mathcal{B} = T
\]

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

\[
\begin{align*}
|x|_{B}^{\beta} & \notin \emptyset \\
|x|_{B}^{\beta} & \notin |P^1|_{B} \\
|P^1 x|_{B}^{\beta} & = \text{F} \\
|\neg P^1 x|_{B}^{\beta} & = \text{T}
\end{align*}
\]
I still need to show that $\forall x \left( \neg P^1 x \rightarrow \neg Q^1 x \right)$ is false in $\mathcal{B}$. Let $\beta$ be a variable assignment over $\mathcal{B}$. Then $|x|_\mathcal{B}^\beta = \text{Oxford}$.

\[
\begin{align*}
|x|_\mathcal{B}^\beta & \notin \emptyset \\
|x|_\mathcal{B}^\beta & \notin |P^1|_\mathcal{B} \\
|P^1 x|_\mathcal{B}^\beta & = F \\
|\neg P^1 x|_\mathcal{B}^\beta & = T
\end{align*}
\]

and similarly:

\[
\begin{align*}
|x|_\mathcal{B}^\beta & \in \{ \text{Oxford} \} \\
|x|_\mathcal{B}^\beta & \in |Q^1|_\mathcal{B} \\
|Q^1 x|_\mathcal{B}^\beta & = T \\
|\neg Q^1 x|_\mathcal{B}^\beta & = F
\end{align*}
\]
I still need to show that $\forall x \left( \neg P^1 x \rightarrow \neg Q^1 x \right)$ is false in $\mathcal{B}$. Let $\beta$ be a variable assignment over $\mathcal{B}$. Then $|x|^\beta_\mathcal{B} = \text{Oxford}$.

$$|x|^\beta_\mathcal{B} \notin \emptyset$$
$$|x|^\beta_\mathcal{B} \notin |P^1|^\mathcal{B}$$
$$|P^1 x|^\beta_\mathcal{B} = F$$
$$|\neg P^1 x|^\beta_\mathcal{B} = T$$

and similarly:
$$|x|^\beta_\mathcal{B} \in \{\text{Oxford}\}$$
$$|x|^\beta_\mathcal{B} \in |Q^1|^\mathcal{B}$$
$$|Q^1 x|^\beta_\mathcal{B} = T$$
$$|\neg Q^1 x|^\beta_\mathcal{B} = F$$

So I have $|\left( \neg P^1 x \rightarrow \neg Q^1 x \right)|^\beta_\mathcal{B} = F$ and therefore
$$|\forall x \left( \neg P^1 x \rightarrow \neg Q^1 x \right)|^\mathcal{B} = F$$

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