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.
What of argument 2?

<table>
<thead>
<tr>
<th>Argument 2</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Zeno is a tortoise.</td>
<td></td>
</tr>
<tr>
<td>(2) All tortoises are toothless.</td>
<td></td>
</tr>
<tr>
<td>Therefore, (C) Zeno is toothless.</td>
<td></td>
</tr>
</tbody>
</table>
What of argument 2?

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

Formalisation

| (1) $Ta$ |
| (2) $\forall x (Tx \rightarrow Lx)$ |
| (C) $La$ |

Dictionary: $a$: Zeno. $T$: ...is a tortoise. $L$: ...is toothless.
What of argument 2?

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

Formalisation

(1) \(Ta\)
(2) \(\forall x(Tx \rightarrow Lx)\)
(C) \(La\)

Dictionary: \(a\): Zeno. \(T\): ...is a tortoise. \(L\): ...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 $\mathcal{L}_1$. The argument with all sentences in $\Gamma$ as premisses and $\phi$ as conclusion is valid if and only if 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 ' $\mathcal{L}_2$'.
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** if and only if there is no $\mathcal{L}_1$-structure under which:

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

(ii) $\phi$ is false.
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 if and only if 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** if and only if 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 if and only if 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.

$\mathcal{L}_1$-structures

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

$L_1$-structures

- Non-logical expressions in $L_1$: $P, Q, R, \ldots$
- An $L_1$ structure $\mathcal{A}$ assigns each sentence letter a semantic value
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 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.
Structures

Structures interpret non-logical expressions.

$L_1$-structures

- Non-logical expressions in $L_1$: $P, Q, R, \ldots$
- An $L_1$ structure $A$ assigns each sentence letter a semantic value (specifically, a truth-value: T or F)

$L_2$ is a richer language. This calls for richer structures.

$L_2$-structures

- Non-logical expressions: $P^1, Q^1, R^1, \ldots$
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: $P^1, Q^1, R^1, \ldots$
- $P^2, Q^2, R^2, \ldots$
- $\vdots$
Structures

Structures interpret non-logical expressions.

**$L_1$-structures**

- Non-logical expressions in $L_1$: $P, Q, R, \ldots$
- An $L_1$ structure $A$ assigns each sentence letter a semantic value (specifically, a truth-value: T or F)

$L_2$ is a richer language. This calls for richer structures.

**$L_2$-structures**

- Non-logical expressions: $P^1, Q^1, R^1, \ldots$
  $P^2, Q^2, R^2, \ldots$
  $\vdots$
  $a, b, c, \ldots$
Structures

Structures interpret non-logical expressions.

<table>
<thead>
<tr>
<th>Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>$\mathcal{L}_1$-structures</strong></td>
</tr>
<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>Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>$\mathcal{L}_2$-structures</strong></td>
</tr>
<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>
<tr>
<td>• An $\mathcal{L}_2$-structure $\mathcal{A}$ assigns each predicate and constant a semantic value</td>
</tr>
</tbody>
</table>
Structures

Structures interpret non-logical expressions.

$L_1$-structures

- Non-logical expressions in $L_1$: $P, Q, R, \ldots$
- An $L_1$ structure $A$ assigns each sentence letter a semantic value (specifically, a truth-value: $T$ or $F$)

$L_2$ is a richer language. This calls for richer structures.

$L_2$-structures

- Non-logical expressions: $P^1, Q^1, R^1, \ldots$
  $P^2, Q^2, R^2, \ldots$
  \vdots
  $a, b, c, \ldots$
- An $L_2$-structure $A$ assigns each predicate and constant a semantic value (specifically, what?)
Semantics in English

Start with a semantics for simple English sentences.
Semantics in English

Start with a semantics for simple English sentences.

‘Bertrand Russell is a philosopher’
Semantics in English

Start with a semantics for simple English sentences.

‘Bertrand Russell is a philosopher’

The sentence is true
Semantics in English

Start with a semantics for simple English sentences.

‘Bertrand Russell is a philosopher’

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

Start with a semantics for simple English sentences.

‘Bertrand Russell is a philosopher’

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.

‘Bertrand Russell is a philosopher’

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>‘Bertrand Russell’</td>
<td>Russell</td>
</tr>
<tr>
<td>‘is a philosopher’</td>
<td>the property of being a philosopher</td>
</tr>
</tbody>
</table>
Semantics in English

Start with a semantics for simple English sentences.

‘Bertrand Russell is a philosopher’

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>‘Bertrand Russell’</td>
<td>Russell</td>
</tr>
<tr>
<td>‘is a philosopher’</td>
<td>the property of being a philosopher</td>
</tr>
</tbody>
</table>

...because Russell has the property of being a philosopher.
Semantics in English

Start with a semantics for simple English sentences.

‘Bertrand Russell is a philosopher’

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>‘Bertrand Russell’</td>
<td>Russell</td>
</tr>
<tr>
<td>‘is a philosopher’</td>
<td>the property of being a philosopher</td>
</tr>
</tbody>
</table>

...because Russell has the property of being a philosopher.

...because |‘Bertrand Russell’| has |‘is a philosopher’|.

Notation

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

‘Alonzo Church reveres Bertrand Russell’ is true iff Church stands in the relation of *revering* to Russell
Similarly:

‘Alonzo Church reveres Bertrand Russell’ is true iff Church stands in the relation of \textit{revering} to Russell

In other words:

\[
\langle \text{‘Alonzo Church reveres Bertrand Russell’} \rangle = T \iff \\
\langle \text{‘Alonzo Church’} \rangle \text{ stands in } \langle \text{‘reveres’} \rangle \text{ to } \langle \text{‘Bertrand Russell’} \rangle
\]
### 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>
# 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**

- |‘Bertrand Russell’| = Russell
- |‘is a philosopher’| = the property of *being a philosopher*
- |‘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

- ‘Bertrand Russell’ = Russell
- ‘is a philosopher’ = the property of *being a philosopher*
- ‘reveres’ = the relation of *revering*

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

In logic, we identify properties with sets.
Properties

In logic, we identify properties with sets.

Property (alias: unary relation)

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

In logic, 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

In logic, 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

In logic, 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 a philosopher*
Properties

In logic, 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 a philosopher

$= \text{the set of philosophers}$

$= \{d : d \text{ is a philosopher}\}$

$= \{\text{Descartes, Kant, Russell, ...} \}$
Recall that we identify binary relations with sets of pairs.

Example: The relation of revering is \{⟨d,e⟩ : d 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.
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 $e$. 

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.
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* $= \{ \langle d, e \rangle : d$ 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* $= \{ \langle d, e \rangle : d$ 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).
e tc.
Putting this all together:
Putting this all together:

‘Bertrand Russell is a philosopher’ is true
Putting this all together:

‘Bertrand Russell is a philosopher’ is true iff |‘Bertrand Russell’| has |‘is a philosopher’|
Putting this all together:

‘Bertrand Russell is a philosopher’ is true
  iff  ‘Bertrand Russell’ has ‘is a philosopher’
  iff  Russell ∈ the set of philosophers
Putting this all together:

‘Bertrand Russell is a philosopher’ is true
iff |‘Bertrand Russell’| has |‘is a philosopher’|
iff Russell ∈ the set of philosophers

Similarly:

‘Alonzo Church reveres Russell’ is true
Putting this all together:

‘Bertrand Russell is a philosopher’ is true
iff |‘Bertrand Russell’| has |‘is a philosopher’|
iff Russell ∈ the set of philosophers

Similarly:

‘Alonzo Church reveres Russell’ is true
iff |‘Alonzo Church’| stands in |‘reveres’| to |‘Russell’|
Putting this all together:

‘Bertrand Russell is a philosopher’ is true
iff |‘Bertrand Russell’| has |‘is a philosopher’|
iff Russell ∈ the set of philosophers

Similarly:

‘Alonzo Church reveres Russell’ is true
iff |‘Alonzo Church’| stands in |‘reveres’| to |‘Russell’|
iff ⟨Church, Russell⟩ ∈ {⟨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: $P^1$</td>
<td>unary relation: $</td>
</tr>
<tr>
<td>binary predicate: $P^2$</td>
<td>binary relation: $</td>
</tr>
</tbody>
</table>

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

**Example:**

- If $a$ is an object in the structure, then $|a|$ is the object.
- If $P$ is a truth-value, then $|P|$ is either T or F.
- If $P^1$ is a unary relation, then $|P^1|$ is a set.
- If $P^2$ is a binary relation, then $|P^2|$ is 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:

<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: $P^1$</td>
<td>unary relation: $</td>
</tr>
<tr>
<td>binary predicate: $P^2$</td>
<td>binary relation: $</td>
</tr>
</tbody>
</table>

- $|Pb| = \text{T} \text{ iff } |b| \text{ has } |P|$
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: $P^1$</td>
<td>unary relation: $</td>
</tr>
<tr>
<td>binary predicate: $P^2$</td>
<td>binary relation: $</td>
</tr>
</tbody>
</table>

- $|Pb| = T$ iff $|b|$ has $|P|$ iff $|b| \in |P|$
Semantics for atomic $L_2$-sentences

The semantics for atomic $L_2$-sentences is similar.

An $L_2$-structure specifies semantic values for $L_2$-expressions:

<table>
<thead>
<tr>
<th>$L_2$-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: $P^1$</td>
<td>unary relation: $</td>
</tr>
<tr>
<td>binary predicate: $P^2$</td>
<td>binary relation: $</td>
</tr>
</tbody>
</table>

- $|Pb| = T$ iff $|b|$ has $|P|$ 
  iff $|b| \in |P|$
- $|Rab| = T$ iff $|a|$ stands in $|R|$ to $|b|$
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: $P^1$</td>
<td>unary relation: $</td>
</tr>
<tr>
<td>binary predicate: $P^2$</td>
<td>binary relation: $</td>
</tr>
</tbody>
</table>

- $|Pb| = T$ iff $|b|$ has $|P|$  
  iff $|b| \in |P|$  
- $|Rab| = T$ iff $|a|$ stands in $|R|$ to $|b|$  
  iff $\langle |a|, |b| \rangle \in |R|$
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: $P^1$</td>
<td>unary relation: $</td>
</tr>
<tr>
<td>binary predicate: $P^2$</td>
<td>binary relation: $</td>
</tr>
</tbody>
</table>

- $|Pb| = T$ iff $|b|$ has $|P|$ iff $|b| \in |P|$
- $|Rab| = T$ iff $|a|$ stands in $|R|$ to $|b|$ iff $\langle |a|, |b| \rangle \in |R|$

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

We have the semantics for $\mathcal{L}_2$-sentences like $Pa$. 
We have the semantics for $L_2$-sentences like $Pa$. What about $L_2$-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 ‘Russell’ 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 ‘Russell’ 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 ‘Russell’ 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 ‘Russell’ 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 ‘Russell’ 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 ‘Russell’ 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 $A$:

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

<table>
<thead>
<tr>
<th>Variable assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>A variable assignment assigns an object to each variable.</td>
</tr>
</tbody>
</table>

Example: the assignment $\alpha$.  
\[
\begin{align*}
x & \mapsto 1 \\
y & \mapsto 1 \\
z & \mapsto 1 \\
x & \mapsto 2
\end{align*}
\]

Mercury  Venus  Venus  Neptune  Mars  Venus  Mars  \ldots

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. $|x|\alpha = \text{Mercury}$; $|y|\alpha = \text{Venus}$; $|x_2|\alpha = \text{Mars}$. 
Variable assignments

A variable assignment assigns an object to each variable.

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

\[
\begin{align*}
\alpha & : x \rightarrow 1, y \rightarrow 1, z \rightarrow 1, x \rightarrow 2, \ldots \\
& = \text{Mercury, Venus, Venus, Neptune, Mars, Venus, Mars, ...}
\end{align*}
\]
Variable assignments

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>$\ldots$</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>$\ldots$</td>
</tr>
</tbody>
</table>
Variable assignments

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 $x$. 
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>$x$</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. $|x|^\alpha =$
Variable assignments

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>x₁</th>
<th>y₁</th>
<th>z₁</th>
<th>x₂</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>
</tr>
</tbody>
</table>

Example: the assignment $\alpha$.

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

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<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>x₁</td>
<td>y₁</td>
<td>z₁</td>
</tr>
<tr>
<td>Mercury</td>
<td>Venus</td>
<td>Venus</td>
<td>Neptune</td>
<td>Mars</td>
<td>Venus</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 $\alpha$.

<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. $|x|^\alpha = \text{Mercury}; |y|^\alpha = \text{Venus}$
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>
</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>
</tr>
<tr>
<td>Mercury</td>
<td>Venus</td>
<td>Venus</td>
<td>Neptune</td>
<td>Mars</td>
<td>Venus</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 = \text{...} |
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. $|x|^\alpha = \text{Mercury}$; $|y|^\alpha = \text{Venus}$; $|x_2|^\alpha = \text{Mars}$. 
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 $P\!x$ are much like the semantics for $P\!a$

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

- $|Px|^\alpha_{\mathcal{A}} = T$ iff $|x|^\alpha$ has $|P|_{\mathcal{A}}$

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

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

- $|Rxb|^\alpha_{\mathcal{A}} = T$ iff $\langle |x|_{\mathcal{A}}, |b|_{\mathcal{A}} \rangle \in |R|_{\mathcal{A}}$
Once \( x \) has been assigned an object, the semantics for \( Px \) are much like the semantics for \( Pa \).

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

- \( |Px|_{\mathcal{A}}^\alpha = T \) iff \( |x|_{\mathcal{A}}^\alpha \) has \( |P|_{\mathcal{A}} \)  
  \( \text{(NB: } |x|_{\mathcal{A}}^\alpha = |x|_{\alpha}) \)
Once $x$ has been assigned an object, the semantics for $P \cdot x$ are much like the semantics for $P \cdot a$

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

- $|P \cdot x|^\alpha_A = T$ iff $|x|^\alpha$ has $|P|^A$
  
  iff $|x|^\alpha \in |P|^A$

(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 $|e|^\alpha_\mathcal{A}$ for the semantic value of expression $e$ in the structure $\mathcal{A}$ under the variable assignment $\alpha$.

- $|Px|^\alpha_\mathcal{A} = T$ iff $|x|^\alpha$ has $|P|^\mathcal{A}$ iff $|x|^\alpha \in |P|^\mathcal{A}$ (NB: $|x|^\alpha_\mathcal{A} = |x|^\alpha$)
- $|Rxy|^\alpha_\mathcal{A} = T$ iff $|x|^\alpha$ stands in $|R|^\mathcal{A}$ to $|y|^\alpha$
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 \( |x|_A^\alpha \) has \( |P|_A \) iff \( |x|_A^\alpha \in |P|_A \) (NB: \( |x|_A^\alpha = |x|_A^\alpha \))
- \( |Rxy|_A^\alpha = T \) iff \( |x|_A^\alpha \) stands in \( |R|_A \) to \( |y|_A^\alpha \) iff \( \langle |x|_A^\alpha, |y|_A^\alpha \rangle \in |R|_A \)
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 $\mathcal{A}$ under the variable assignment $\alpha$.

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

- $|Rxy|_A^\alpha = T$ iff $|x|_A^\alpha$ stands in $|R|_A$ to $|y|_A^\alpha$ iff $\langle |x|_A^\alpha, |y|_A^\alpha \rangle \in |R|_A$

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

- $|Rxy|_A^\alpha = T$ iff $|x|_A^\alpha$ stands in $|R|_A$ to $|y|_A$
  
  
  iff $\langle |x|_A^\alpha, |y|_A^\alpha \rangle \in |R|_A$

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

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

- $|Rxy|_{\mathcal{A}}^\alpha = T$ iff $|x|_{\mathcal{A}}^\alpha$ stands in $|R|_{\mathcal{A}}$ to $|y|_{\mathcal{A}}^\alpha$ iff $\langle |x|_{\mathcal{A}}^\alpha, |y|_{\mathcal{A}}^\alpha \rangle \in |R|_{\mathcal{A}}$

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

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

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

- $|Px|^\alpha_A = T$ iff $|x|^\alpha$ has $|P|_A$
  - iff $|x|^\alpha \in |P|_A$
- $|Rxy|^\alpha_A = T$ iff $|x|^\alpha$ stands in $|R|_A$ to $|y|^\alpha$
  - iff $\langle |x|^\alpha, |y|^\alpha \rangle \in |R|_A$

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

- $|Rab|^\alpha_A = T$ iff $\langle |a|_A, |b|_A \rangle \in |R|_A$
- $|Rx|^\alpha_A = T$ 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:

- $|a|_\mathcal{A} = \text{Alonzo Church}$
- $|b|_\mathcal{A} = \text{Bertrand Russell}$
- $|P|_\mathcal{A} = \{\text{Frege, Russell}\}$
- $|R|_\mathcal{A} = \{\langle\text{Church, Russell}\rangle\}$

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>Frege</td>
<td>Russell</td>
<td>Wittgenstein</td>
</tr>
<tr>
<td>$\beta$:</td>
<td>Church</td>
<td>Church</td>
<td>Church</td>
</tr>
</tbody>
</table>
Worked example

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

- $|a|_\mathcal{A} = \text{Alonzo Church}$
- $|b|_\mathcal{A} = \text{Bertrand Russell}$
- $|P|_\mathcal{A} = \{\text{Frege, Russell}\}$
- $|R|_\mathcal{A} = \{\langle \text{Church, Russell} \rangle\}$

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>Frege</td>
<td>Russell</td>
<td>Wittgenstein</td>
</tr>
<tr>
<td>$\beta$:</td>
<td>Church</td>
<td>Church</td>
<td>Church</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x|_\mathcal{A}^\alpha = \text{Frege}$
- $|x|_\mathcal{A}^\beta = \text{Church}$
- $|a|_\mathcal{A}^\alpha = \text{Church}$
- $|Py|_\mathcal{A}^\alpha = T$
- $|Py|_\mathcal{A}^\beta = F$
- $|Pb|_\mathcal{A}^\alpha = T$
- $|Rx|_\mathcal{A}^\alpha = F$
- $|Rx|_\mathcal{A}^\beta = F$
- $|Rx|_\mathcal{A}^\alpha = F$
Worked example

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

- $|a|_\mathcal{A} = \text{Alonzo Church}$
- $|b|_\mathcal{A} = \text{Bertrand Russell}$
- $|P|_\mathcal{A} = \{\text{Frege, Russell}\}$
- $|R|_\mathcal{A} = \{\langle\text{Church, Russell}\rangle\}$

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

\[
\begin{array}{ccc}
  x & y & z \\
  \alpha: & \text{Frege} & \text{Russell} & \text{Wittgenstein} \\
  \beta: & \text{Church} & \text{Church} & \text{Church} \\
\end{array}
\]

Compute the following:

- $|x|_\mathcal{A}^\alpha = \text{Frege}$
- $|x|_\mathcal{A}^\beta =$
- $|a|_\mathcal{A}^\alpha =$
- $|P|_\mathcal{A}^\alpha =$
- $|P|_\mathcal{A}^\beta =$
- $|Pb|_\mathcal{A}^\alpha =$
- $|Rxy|_\mathcal{A}^\alpha =$
- $|Rxy|_\mathcal{A}^\beta =$
- $|Rx|_\mathcal{A}^\alpha =$
Worked example

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

- $|a|_{\mathcal{A}} = \text{Alonzo Church}$
- $|b|_{\mathcal{A}} = \text{Bertrand Russell}$
- $|P|_{\mathcal{A}} = \{\text{Frege, Russell}\}$
- $|R|_{\mathcal{A}} = \{\langle\text{Church, Russell}\rangle\}$

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

<table>
<thead>
<tr>
<th></th>
<th>$\alpha$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>Frege</td>
<td>Church</td>
</tr>
<tr>
<td>$y$</td>
<td>Russell</td>
<td>Church</td>
</tr>
<tr>
<td>$z$</td>
<td>Wittgenstein</td>
<td>Church</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x|^\alpha_{\mathcal{A}} = \text{Frege}$
- $|x|^\beta_{\mathcal{A}} = \text{Church}$
- $|a|^\alpha_{\mathcal{A}} =$
- $|Py|^\alpha_{\mathcal{A}} =$
- $|Py|^\beta_{\mathcal{A}} =$
- $|Pb|^\alpha_{\mathcal{A}} =$
- $|Rxy|^\alpha_{\mathcal{A}} =$
- $|Rxy|^\beta_{\mathcal{A}} =$
- $|Rxb|^\alpha_{\mathcal{A}} =$
Worked example

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

- $|a|_\mathcal{A} = \text{Alonzo Church}$
- $|b|_\mathcal{A} = \text{Bertrand Russell}$
- $|P|_\mathcal{A} = \{\text{Frege, Russell}\}$
- $|R|_\mathcal{A} = \{\langle\text{Church, Russell}\rangle\}$

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>Frege</td>
<td>Russell</td>
<td>Wittgenstein</td>
</tr>
<tr>
<td>$\beta$:</td>
<td>Church</td>
<td>Church</td>
<td>Church</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x|_\mathcal{A}^\alpha = \text{Frege}$
- $|x|_\mathcal{A}^\beta = \text{Church}$
- $|a|_\mathcal{A}^\alpha = \text{Church}$
- $|P|_\mathcal{A}^\alpha = \text{Frege}$
- $|P|_\mathcal{A}^\beta = \text{Church}$
- $|Pb|_\mathcal{A}^\alpha = \text{Church}$
- $|Rxy|_\mathcal{A}^\alpha = \text{Frege}$
- $|Rxy|_\mathcal{A}^\beta = \text{Church}$
- $|Rx b|_\mathcal{A}^\alpha = \text{Church}$
Worked example

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

- $|a|_\mathcal{A} = \text{Alonzo Church}$
- $|b|_\mathcal{A} = \text{Bertrand Russell}$
- $|P|_\mathcal{A} = \{\text{Frege, Russell}\}$
- $|R|_\mathcal{A} = \{\langle \text{Church, Russell} \rangle\}$

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>Frege</td>
<td>Russell</td>
<td>Wittgenstein</td>
</tr>
<tr>
<td>$\beta$:</td>
<td>Church</td>
<td>Church</td>
<td>Church</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x|_\mathcal{A}^\alpha = \text{Frege}$
- $|x|_\mathcal{A}^\beta = \text{Church}$
- $|a|_\mathcal{A}^\alpha = \text{Church}$
- $|Py|_\mathcal{A}^\alpha = \text{T}$
- $|Py|_\mathcal{A}^\beta = $ (Blank)
- $|Pb|_\mathcal{A}^\alpha = $ (Blank)
- $|Rxy|_\mathcal{A}^\alpha = $ (Blank)
- $|Rxy|_\mathcal{A}^\beta = $ (Blank)
- $|Rx b|_\mathcal{A}^\alpha = $ (Blank)
Worked example

Let $\mathcal{L}_2$-structure $\mathcal{A}$ be such that:
- $|a|_\mathcal{A} = \text{Alonzo Church}$
- $|b|_\mathcal{A} = \text{Bertrand Russell}$
- $|P|_\mathcal{A} = \{\text{Frege, Russell}\}$
- $|R|_\mathcal{A} = \{\langle\text{Church, Russell}\rangle\}$

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>Frege</td>
<td>Russell</td>
<td>Wittgenstein</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Church</td>
<td>Church</td>
<td>Church</td>
</tr>
</tbody>
</table>

Compute the following:

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

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

$|Rxy|_\mathcal{A}^\alpha = \quad |Rxy|_\mathcal{A}^\beta = \quad |Rx b|_\mathcal{A}^\alpha =$
Worked example

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

- $|a|_A = \text{Alonzo Church}$
- $|b|_A = \text{Bertrand Russell}$
- $|P|_A = \{\text{Frege, Russell}\}$
- $|R|_A = \{\langle \text{Church, Russell} \rangle\}$

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>$\alpha$: Frege</td>
<td>Russell</td>
<td>Wittgenstein</td>
</tr>
<tr>
<td>$\beta$: Church</td>
<td>Church</td>
<td>Church</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x|_A^\alpha = \text{Frege}$
- $|x|_A^\beta = \text{Church}$
- $|a|_A^\alpha = \text{Church}$
- $|Py|_A^\alpha = \text{T}$
- $|Py|_A^\beta = \text{F}$
- $|Pb|_A^\alpha = \text{T}$
- $|Rxy|_A^\alpha = \text{ }$
- $|Rxy|_A^\beta = \text{ }$
- $|Rx'b|_A^\alpha = \text{ }$
Worked example

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

- $|a|_\mathcal{A} = \text{Alonzo Church}$
- $|b|_\mathcal{A} = \text{Bertrand Russell}$
- $|P|_\mathcal{A} = \{\text{Frege, Russell}\}$
- $|R|_\mathcal{A} = \{\langle\text{Church, Russell}\rangle\}$

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>Frege</td>
<td>Russell</td>
<td>Wittgenstein</td>
</tr>
<tr>
<td>$\beta$:</td>
<td>Church</td>
<td>Church</td>
<td>Church</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x|_\mathcal{A}^\alpha = \text{Frege}$  $|x|_\mathcal{A}^\beta = \text{Church}$  $|a|_\mathcal{A}^\alpha = \text{Church}$
- $|Py|_\mathcal{A}^\alpha = T$  $|Py|_\mathcal{A}^\beta = F$  $|Pb|_\mathcal{A}^\alpha = T$
- $|Rxy|_\mathcal{A}^\alpha = F$  $|Rxy|_\mathcal{A}^\beta = $  $|Rx b|_\mathcal{A}^\alpha = $
Worked example

Let $L_2$-structure $A$ be such that:

- $|a|_A = \text{Alonzo Church}$
- $|b|_A = \text{Bertrand Russell}$
- $|P|_A = \{\text{Frege, Russell}\}$
- $|R|_A = \{\langle\text{Church, Russell}\rangle\}$

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>Frege</td>
<td>Russell</td>
<td>Wittgenstein</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Church</td>
<td>Church</td>
<td>Church</td>
</tr>
</tbody>
</table>

Compute the following:

- $|x|_{\alpha} = \text{Frege}$
- $|x|_{\beta} = \text{Church}$
- $|a|_{\alpha} = \text{Church}$
- $|Py|_{\alpha} = T$
- $|Py|_{\beta} = F$
- $|Pb|_{\alpha} = T$
- $|Rxy|_{\alpha} = F$
- $|Rxy|_{\beta} = F$
- $|Rx|_{\alpha} = $
**Worked example**

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

- $|a|_\mathcal{A} = \text{Alonzo Church}$
- $|b|_\mathcal{A} = \text{Bertrand Russell}$
- $|P|_\mathcal{A} = \{\text{Frege, Russell}\}$
- $|R|_\mathcal{A} = \{\langle\text{Church, Russell}\rangle\}$

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>Frege</td>
<td>Russell</td>
<td>Wittgenstein</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Church</td>
<td>Church</td>
<td>Church</td>
</tr>
</tbody>
</table>

**Compute the following:**

- $|x|_\mathcal{A}^\alpha = \text{Frege}$  
  $|x|_\mathcal{A}^\beta = \text{Church}$  
  $|a|_\mathcal{A}^\alpha = \text{Church}$
- $|Py|_\mathcal{A}^\alpha = \text{T}$  
  $|Py|_\mathcal{A}^\beta = \text{F}$  
  $|Pb|_\mathcal{A}^\alpha = \text{T}$
- $|Rxy|_\mathcal{A}^\alpha = \text{F}$  
  $|Rxy|_\mathcal{A}^\beta = \text{F}$  
  $|Rx b|_\mathcal{A}^\alpha = \text{F}$
Semantics for quantifiers

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

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

Almost everyone attended the first lecture.
Semantics for quantifiers

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

Almost everyone attended the first lecture.

The context supplies a ‘domain’ telling us who ‘everyone’ ranges over
Semantics for quantifiers

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

Almost everyone attended the first lecture.

The context supplies a ‘domain’ telling us who ‘everyone’ ranges over

**Domain:** the set of first-year Oxford philosophy students

Almost every first-year Oxford philosophy student attended the first lecture.
Semantics for quantifiers

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

Almost everyone attended the first lecture.

The context supplies a ‘domain’ telling us who ‘everyone’ ranges over

**Domain: the set of first-year Oxford philosophy students**

Almost every first-year Oxford philosophy student attended the first lecture.  

T
Semantics for quantifiers

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

Almost everyone attended the first lecture.

The context supplies a ‘domain’ telling us who ‘everyone’ ranges over

**Domain: the set of first-year Oxford philosophy students**
Almost every first-year Oxford philosophy student attended the first lecture. $T$

**Domain: the set of everyone in the world**
Almost everyone in the world attended the first lecture.
Semantics for quantifiers

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

Almost everyone attended the first lecture.

The context supplies a ‘domain’ telling us who ‘everyone’ ranges over

Domain: the set of first-year Oxford philosophy students
Almost every first-year Oxford philosophy student attended the first lecture. T

Domain: the set of everyone in the world
Almost everyone in the world attended the first lecture. F
An $L_2$-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 $\mathcal{L}_2$-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):**

\[
|\forall x P x|_\mathcal{A} = T \quad \text{iff every member of } D_\mathcal{A} \text{ has } |P|_\mathcal{A}
\]
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 $D_\mathcal{A}$ has $|P|_\mathcal{A}$
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 $D_\mathcal{A}$ has $|P|_\mathcal{A}$

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 $|Px|^\alpha = T$
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):**

$|\forall xP x|_{\mathcal{A}} = T$  
iff every member of $D_{\mathcal{A}}$ has $|P|_{\mathcal{A}}$  
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 $|Px|_{\alpha}^{\alpha} = T$

Similarly:

$|\exists xP x|_{\mathcal{A}} = T$  
iff some member of $D_{\mathcal{A}}$ has $|P|_{\mathcal{A}}$
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):**

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

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

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 $D_\mathcal{A}$ has $|P|_\mathcal{A}$

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

This 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):**

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

Similarly:

\[
|\exists xPx|_\mathcal{A} = T \\
\text{iff some member of } D_\mathcal{A} \text{ has } |P|_\mathcal{A} \\
\text{iff some assignment } \alpha \text{ of } x \text{ to a member of } D_\mathcal{A} \text{ is such that } |x|^{\alpha} \in |P|_\mathcal{A} \\
\text{iff some assignment } \alpha \text{ over } \mathcal{A} \text{ is such that } |Px|^{\alpha}_\mathcal{A} = T
\]
An $\mathcal{L}_2$-structure $\mathcal{A}$ specifies a non-empty set $D_\mathcal{A}$ as the domain. An assignment over $\mathcal{A}$ assigns a member of $D_\mathcal{A}$ to each variable.

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

$|\forall xPx|_\mathcal{A} = T$
  iff every member of $D_\mathcal{A}$ has $|P|_\mathcal{A}$
  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 $|Px|_\alpha = T$

Similarly:

$|\exists xPx|_\mathcal{A} = T$
  iff some member of $D_\mathcal{A}$ has $|P|_\mathcal{A}$
  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 $|Px|_\alpha = T$

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 under $\mathcal{A}$ 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 under $\mathcal{A}$ with domain $D_\mathcal{A}$

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

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

Suppose we try to evaluate this as before under $\mathcal{A}$ with domain $D_A$

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

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

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

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

iff some $d$ in $D_A$ is such that $|x|^\alpha$ stands in $|R|^A$ to $d$
How to determine $|\exists y R xy|_A^\alpha$?

$|\exists y R xy|_A^\alpha = T$

iff some $d$ in $D_A$ is such that $|x|_A^\alpha$ stands in $|R|_A$ to $d$

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

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

iff some $d$ in $D_A$ is such that $|x|_A^{\alpha}$ stands in $|R|_A$ to $d$

iff some assignment $\beta$ over $A$ is such that $|x|_A^{\alpha}$ stands in $|R|_A$ to $|y|_A^{\beta}$

So we don’t have to keep track of multiple assignments:

Say that $\beta$ differs from $\alpha$ in $y$ at most if $|v|_A^{\alpha} = |v|_A^{\beta}$ for all variables $v$ with the possible exception of $y$. 
How to determine $\exists y Rxy|_A^\alpha$?

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

iff some $d$ in $D_A$ is such that $|x|^\alpha$ stands in $|R|_A$ to $d$

iff some assignment $\beta$ over $A$ is such that $|x|^\alpha$ stands in $|R|_A$ to $|y|^\beta$

So we don’t have to keep track of multiple assignments:

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

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

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

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

iff some $d$ in $D_A$ is such that $|x|^\alpha$ stands in $|R|^A$ to $d$

iff some assignment $\beta$ over $A$ is such that $|x|^\alpha$ stands in $|R|^A$ to $|y|^\beta$

So we don’t have to keep track of multiple assignments:

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

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

iff some assignment $\beta$ over $A$ which differs from $\alpha$ in $y$ at most

is such that $|x|^\alpha$ stands in $|R|^A$ to $|y|^\beta$
How to determine $|\exists y R xy|^\alpha_A$?

$|\exists y R xy|^\alpha_A = T$

iff some $d$ in $D_A$ is such that $|x|^\alpha$ stands in $|R|^A$ to $d$

iff some assignment $\beta$ over $A$ is such that $|x|^\alpha$ stands in $|R|^A$ to $|y|^\beta$

So we don’t have to keep track of multiple assignments:

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

$|\exists y R xy|^\alpha_A = T$

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

$|\exists yRx_\alpha|_A = T$

iff some $d$ in $D_A$ is such that $|x^\alpha$ stands in $|R|_A$ to $d$

iff some assignment $\beta$ over $A$ is such that $|x^\alpha$ stands in $|R|_A$ to $|y^\beta$

So we don’t have to keep track of multiple assignments:

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

$|\exists yRx_\alpha|_A = T$

iff some assignment $\beta$ over $A$ which differs from $\alpha$ in $y$ at most is such that $|x^\beta$ stands in $|R|_A$ to $|y^\beta$

iff some assignment $\beta$ over $A$ which differs from $\alpha$ in $y$ at most is such that $|Rxy^\beta|_A = T$
\(\mathcal{L}_2\)-structures

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 $D_\mathcal{A}$
$\mathcal{L}_2$-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 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: $P^1$</td>
<td>unary relation: $</td>
</tr>
<tr>
<td>binary predicate: $P^2$</td>
<td>binary relation: $</td>
</tr>
<tr>
<td>ternary predicate: $P^3$</td>
<td>ternary relation: $</td>
</tr>
<tr>
<td>etc. etc.</td>
<td></td>
</tr>
</tbody>
</table>
Summary of semantics of $\mathcal{L}_2$

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

**Atomic formulae**

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

- $|\Phi^n|_A^\alpha$ is the $n$-ary relation assigned to $\Phi^n$ by $A$. 
Summary of semantics of $L_2$

Let $A$ be an $L_2$-structure and $\alpha$ an assignment over $A$.

**Atomic formulae**

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

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

(i) $|\Phi^1t_1|^{\alpha}_A = T$ if and only if $|t_1|^{\alpha}_A \in |\Phi^1|_A$
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) $|\Phi^1 t_1_\mathcal{A}^{\alpha} = T$ 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 $t_1, t_2, \ldots$ 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) $|\Phi^1t_1|_\mathcal{A}^\alpha = T$ if and only if $|t_1|_\mathcal{A}^\alpha \in |\Phi^1|_\mathcal{A}$

$|\Phi^2t_1t_2|_\mathcal{A}^\alpha = T$ if and only if $\langle |t_1|_\mathcal{A}^\alpha, |t_2|_\mathcal{A}^\alpha \rangle \in |\Phi^2|_\mathcal{A}$

$|\Phi^3t_1t_2t_3|_\mathcal{A}^\alpha = T$ 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.

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

(i) $|\Phi^1 t_1|_A^\alpha = T$ if and only if $|t_1|_A^\alpha \in |\Phi^1|_A$

$|\Phi^2 t_1 t_2|_A^\alpha = T$ if and only if $\langle |t_1|_A^\alpha, |t_2|_A^\alpha \rangle \in |\Phi^2|_A$

$|\Phi^3 t_1 t_2 t_3|_A^\alpha = T$ if and only if $\langle |t_1|_A^\alpha, |t_2|_A^\alpha, |t_3|_A^\alpha \rangle \in |\Phi^3|_A$

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

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

**Quantifiers**

**(vii)** $|\forall v \phi|^\alpha_A = T$ if and only if $|\phi|^\beta_A = T$ for all variable assignments $\beta$ over $A$ differing from $\alpha$ in $v$ at most.
These are the semantic clauses for $\forall v$ and $\exists v$.

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

Just one detail remains.
Truth

Just one detail remains.

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

Just one detail remains.

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 $\mathcal{A}$.
Truth

Just one detail remains.

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 $|\phi|_A^\alpha$; we want now to define $|\phi|_A$.)
Truth

Just one detail remains.

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 $|\phi|_A^\alpha$; we want now to define $|\phi|_A$.)

Fact about sentences

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

Just one detail remains.

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 $|\phi|_A^\alpha$; 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 $A$: $|\phi|_A^\alpha = |\phi|_A^\beta$ (when $\phi$ is a sentence).

50
Truth

Just one detail remains.

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 $|\phi|_\mathcal{A}^\alpha$; we want now to define $|\phi|_\mathcal{A}$.)

**Fact about sentences**

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

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

Just one detail remains.

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 $|\phi|_A^\alpha$; 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 $A$: $|\phi|_A^\alpha = |\phi|_A^\beta$ (when $\phi$ is a sentence).

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

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