INTRODUCTION TO LOGIC

Lecture 8

Identity and Definite Descriptions

Dr. James Studd

> The analysis of the beginning would thus yield the notion of the unity of being and not-being—or, in a more reflected form, the unity of differentiatedness and non-differentiatedness, or the identity of identity and non-identity.
>
> Hegel
> The Science of Logic

Outline

(1) The language of predicate logic with identity: $\mathcal{L}_=$

- Syntax
- Semantics
- Proof theory

(2) Formalisation in $\mathcal{L}_=$

- Numerical quantifiers
- Definite descriptions

The logicians’ sense of ‘identical’

In English, we use the words ‘identity’/‘identical’ in a number of different ways.

The logicians' sense of 'identical'

In English, we use the words 'identity'/'identical' in a number of different ways.

Wider uses of 'identity'/'identical'

(1) Mancunians have a strong sense of cultural identity.

The logicians' sense of 'identical'

In English, we use the words 'identity'/'identical' in a number of different ways.

Wider uses of 'identity'/'identical'

(1) Mancunians have a strong sense of cultural identity.

(2) Dr. Jekyll has multiple identities.

The logicians' sense of 'identical'

In English, we use the words 'identity'/'identical' in a number of different ways.

Wider uses of 'identity'/'identical'

(1) Mancunians have a strong sense of cultural identity.

(2) Dr. Jekyll has multiple identities.

(3) Jedward are almost completely identical.

The logicians' sense of 'identical'

In English, we use the words 'identity'/'identical' in a number of different ways.

Wider uses of 'identity'/'identical'

(1) Mancunians have a strong sense of cultural identity.

(2) Dr. Jekyll has multiple identities.

(3) Jedward are almost completely identical.

The sense of 'identity' used in (3) is sometimes called 'qualitative identity'.

The logicians' sense of 'identical'

In English, we use the words 'identity'/'identical' in a number of different ways.

Wider uses of 'identity'/'identical'

(1) Mancunians have a strong sense of cultural identity.

(2) Dr. Jekyll has multiple identities.

(3) Jedward are almost completely identical.

The sense of 'identity' used in (3) is sometimes called 'qualitative identity'.

- (3) says that John and Edward are almost exactly similar in every respect.

The logicians' sense of 'identical'

In English, we use the words 'identity'/'identical' in a number of different ways.

Wider uses of 'identity'/'identical'

(1) Mancunians have a strong sense of cultural identity.

(2) Dr. Jekyll has multiple identities.

(3) Jedward are almost completely identical.

The sense of 'identity' used in (3) is sometimes called 'qualitative identity'.

- (3) says that John and Edward are almost exactly similar in every respect.

None of these uses of 'identical' is the logicians' use.

In logic, we always use 'identical' in the following strict sense

A is identical to B iff A is the very same thing as B
i.e. A and B are one and the same thing.

In logic, we always use 'identical' in the following strict sense

> A is identical to B iff A is the very same thing as B
> i.e. A and B are one and the same thing.

This is sometimes called 'numerical identity'

In logic, we always use 'identical' in the following strict sense

A is identical to B iff A is the very same thing as B
i.e. A and B are one and the same thing.

This is sometimes called 'numerical identity'

(Unless otherwise stated 'identity'/'identical' henceforth mean numerical identity/numerically identical.)

In logic, we always use ‘identical’ in the following strict sense

A is identical to B iff A is the very same thing as B
i.e. A and B are one and the same thing.

This is sometimes called ‘numerical identity’

(Unless otherwise stated ‘identity’/‘identical’ henceforth mean numerical identity/numerically identical.)

Examples

- George Orwell is identical to Eric Arthur Blair

In logic, we always use 'identical' in the following strict sense

A is identical to B iff A is the very same thing as B
i.e. A and B are one and the same thing.

This is sometimes called 'numerical identity'

(Unless otherwise stated 'identity'/'identical' henceforth mean numerical identity/numerically identical.)

Examples

- George Orwell is identical to Eric Arthur Blair
- Dr. Jekyll is identical to Mr. Hyde

In logic, we always use ‘identical’ in the following strict sense

A is identical to B iff A is the very same thing as B
i.e. A and B are one and the same thing.

This is sometimes called ‘numerical identity’

(Unless otherwise stated ‘identity’/‘identical’ henceforth mean numerical identity/numerically identical.)

Examples

- George Orwell is identical to Eric Arthur Blair
- Dr. Jekyll is identical to Mr. Hyde
- John is not identical to Edward

A third formal language

The new language makes a single addition to $\mathcal{L}_2$.

A third formal language

The new language makes a single addition to $\mathcal{L}_2$.

The language $\mathcal{L}_=$

The language $\mathcal{L}_=$ of predicate logic with identity adds a single binary predicate letter to the language of predicate logic $\mathcal{L}_2$.

- $\mathcal{L}_=$ adds the identity predicate $=$ to $\mathcal{L}_2$

A third formal language

The new language makes a single addition to $\mathcal{L}_2$.

The language $\mathcal{L}_=$

The language $\mathcal{L}_=$ of predicate logic with identity adds a single binary predicate letter to the language of predicate logic $\mathcal{L}_2$.

- $\mathcal{L}_=$ adds the identity predicate $=$ to $\mathcal{L}_2$

$=$ differs from the other predicate letters in several way.

A third formal language

The new language makes a single addition to $\mathcal{L}_2$.

The language $\mathcal{L}_=$

The language $\mathcal{L}_=$ of predicate logic with identity adds a single binary predicate letter to the language of predicate logic $\mathcal{L}_2$.

- $\mathcal{L}_=$ adds the identity predicate $=$ to $\mathcal{L}_2$

$=$ differs from the other predicate letters in several way.

- P, R^2, etc., are non-logical expressions.
  Different $\mathcal{L}_2$-structures interpret them differently.

A third formal language

The new language makes a single addition to $\mathcal{L}_2$.

The language $\mathcal{L}_=$

The language $\mathcal{L}_=$ of predicate logic with identity adds a single binary predicate letter to the language of predicate logic $\mathcal{L}_2$.

- $\mathcal{L}_=$ adds the identity predicate $=$ to $\mathcal{L}_2$

$=$ differs from the other predicate letters in several way.

- P, R^2, etc., are non-logical expressions.
  Different $\mathcal{L}_2$-structures interpret them differently.
- $=$ is treated as a logical expression.
  It always has the same interpretation in any structure.

A third formal language

The new language makes a single addition to $\mathcal{L}_2$.

The language $\mathcal{L}_=$

The language $\mathcal{L}_=$ of predicate logic with identity adds a single binary predicate letter to the language of predicate logic $\mathcal{L}_2$.

- $\mathcal{L}_=$ adds the identity predicate $=$ to $\mathcal{L}_2$

$=$ differs from the other predicate letters in several way.

- P, R^2, etc., are non-logical expressions.
  Different $\mathcal{L}_2$-structures interpret them differently.
- $=$ is treated as a logical expression.
  It always has the same interpretation in any structure.
- Minor difference: we write $a = b$ (rather than $=ab$).

Syntax

We make a slight change to the definition of atomic formula.

Definition (atomic formulae of $\mathcal{L}_=$)

All atomic formulae of $\mathcal{L}_2$ are atomic formulae of $\mathcal{L}_=$. Furthermore, if s and t are variables or constants, then $s = t$ is an atomic formula of $\mathcal{L}_=$.

Syntax

We make a slight change to the definition of atomic formula.

Definition (atomic formulae of $\mathcal{L}_=$)

All atomic formulae of $\mathcal{L}_2$ are atomic formulae of $\mathcal{L}_=$. Furthermore, if s and t are variables or constants, then $s=t$ is an atomic formula of $\mathcal{L}_=$.

The definition of formula and sentence is otherwise just like the definition for $\mathcal{L}_2$.

Examples

- Atomic $\mathcal{L}_=$-formulae:

Syntax

We make a slight change to the definition of atomic formula.

Definition (atomic formulae of $\mathcal{L}_=$)

All atomic formulae of $\mathcal{L}_2$ are atomic formulae of $\mathcal{L}_=$. Furthermore, if s and t are variables or constants, then $s = t$ is an atomic formula of $\mathcal{L}_=$.

The definition of formula and sentence is otherwise just like the definition for $\mathcal{L}_2$.

Examples

- Atomic $\mathcal{L}_=$-formulae: $c = a$,

Syntax

We make a slight change to the definition of atomic formula.

Definition (atomic formulae of $\mathcal{L}_=$)

All atomic formulae of $\mathcal{L}_2$ are atomic formulae of $\mathcal{L}_=$. Furthermore, if s and t are variables or constants, then $s = t$ is an atomic formula of $\mathcal{L}_=$.

The definition of formula and sentence is otherwise just like the definition for $\mathcal{L}_2$.

Examples

- Atomic $\mathcal{L}_=$-formulae: $c = a$, $x = y_3$,

Syntax

We make a slight change to the definition of atomic formula.

Definition (atomic formulae of $\mathcal{L}_=$)

All atomic formulae of $\mathcal{L}_2$ are atomic formulae of $\mathcal{L}_=$. Furthermore, if s and t are variables or constants, then $s = t$ is an atomic formula of $\mathcal{L}_=$.

The definition of formula and sentence is otherwise just like the definition for $\mathcal{L}_2$.

Examples

- Atomic $\mathcal{L}_=$-formulae: $c = a$, $x = y_3$, $x = a$,

Syntax

We make a slight change to the definition of atomic formula.

Definition (atomic formulae of $\mathcal{L}_=$)

All atomic formulae of $\mathcal{L}_2$ are atomic formulae of $\mathcal{L}_=$. Furthermore, if s and t are variables or constants, then $s = t$ is an atomic formula of $\mathcal{L}_=$.

The definition of formula and sentence is otherwise just like the definition for $\mathcal{L}_2$.

Examples

- Atomic $\mathcal{L}_=$-formulae: $c = a$, $x = y_3$, $x = a$, R^2ax.

Syntax

We make a slight change to the definition of atomic formula.

Definition (atomic formulae of $\mathcal{L}_=$)

All atomic formulae of $\mathcal{L}_2$ are atomic formulae of $\mathcal{L}_=$. Furthermore, if s and t are variables or constants, then $s = t$ is an atomic formula of $\mathcal{L}_=$.

The definition of formula and sentence is otherwise just like the definition for $\mathcal{L}_2$.

Examples

- Atomic $\mathcal{L}_=$-formulae: $c = a$, $x = y_3$, $x = a$, R^2ax.
- Complex $\mathcal{L}_=$-formulae:

Syntax

We make a slight change to the definition of atomic formula.

Definition (atomic formulae of $\mathcal{L}_=$)

All atomic formulae of $\mathcal{L}_2$ are atomic formulae of $\mathcal{L}_=$. Furthermore, if s and t are variables or constants, then $s = t$ is an atomic formula of $\mathcal{L}_=$.

The definition of formula and sentence is otherwise just like the definition for $\mathcal{L}_2$.

Examples

- Atomic $\mathcal{L}_=$-formulae: $c = a$, $x = y_3$, $x = a$, R^2ax.
- Complex $\mathcal{L}_=$-formulae: $\neg\, x = y$,

Syntax

We make a slight change to the definition of atomic formula.

Definition (atomic formulae of $\mathcal{L}_=$)

All atomic formulae of $\mathcal{L}_2$ are atomic formulae of $\mathcal{L}_=$. Furthermore, if s and t are variables or constants, then $s=t$ is an atomic formula of $\mathcal{L}_=$.

The definition of formula and sentence is otherwise just like the definition for $\mathcal{L}_2$.

Examples

- Atomic $\mathcal{L}_=$-formulae: $c=a$, $x=y_3$, $x=a$, R^2ax.
- Complex $\mathcal{L}_=$-formulae: $\neg\, x=y$, $\forall x(Rxy_2 \to y_2=x)$.

Semantics

Semantics

The definition of structure is just the same as before.

Definition: $\mathcal{L}_=$-structure

An $\mathcal{L}_=$-structure is simply an $\mathcal{L}_2$-structure.

Semantics

The definition of structure is just the same as before.

Definition: $\mathcal{L}_=$-structure

An $\mathcal{L}_=$-structure is simply an $\mathcal{L}_2$-structure.

Why no change?

Semantics

The definition of structure is just the same as before.

Definition: $\mathcal{L}_=$-structure

An $\mathcal{L}_=$-structure is simply an $\mathcal{L}_2$-structure.

Why no change?

- Structures interpret non-logical expressions like P and a.

Semantics

The definition of structure is just the same as before.

Definition: $\mathcal{L}_=$-structure

An $\mathcal{L}_=$-structure is simply an $\mathcal{L}_2$-structure.

Why no change?

- Structures interpret non-logical expressions like P and a.
- Structures do not interpret logical expressions like $\neg$ and $\forall x$.

Semantics

The definition of structure is just the same as before.

Definition: $\mathcal{L}_=$-structure

An $\mathcal{L}_=$-structure is simply an $\mathcal{L}_2$-structure.

Why no change?

- Structures interpret non-logical expressions like P and a.
- Structures do not interpret logical expressions like $\neg$ and $\forall x$.
- The fixed interpretation of logical expressions is specified in the definition of satisfaction.

Semantics

The definition of structure is just the same as before.

Definition: $\mathcal{L}_=$-structure

An $\mathcal{L}_=$-structure is simply an $\mathcal{L}_2$-structure.

Why no change?

- Structures interpret non-logical expressions like P and a.
- Structures do not interpret logical expressions like $\neg$ and $\forall x$.
- The fixed interpretation of logical expressions is specified in the definition of satisfaction.
  e.g. $|\neg\phi|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ iff $|\phi|^{\alpha}_{\mathcal{A}} = \mathrm{F}$

Semantics

The definition of structure is just the same as before.

Definition: $\mathcal{L}_=$-structure

An $\mathcal{L}_=$-structure is simply an $\mathcal{L}_2$-structure.

Why no change?

- Structures interpret non-logical expressions like P and a.
- Structures do not interpret logical expressions like $\neg$ and $\forall x$.
- The fixed interpretation of logical expressions is specified in the definition of satisfaction.
  e.g. $|\neg\phi|^{\alpha}_{\mathcal{A}} = \text{T}$ iff $|\phi|^{\alpha}_{\mathcal{A}} = \text{F}$
- Similarly $=$ is treated as a logical expression, which is not assigned a semantic value by the structure.

Semantics

The definition of structure is just the same as before.

Definition: $\mathcal{L}_=$-structure

An $\mathcal{L}_=$-structure is simply an $\mathcal{L}_2$-structure.

Why no change?

- Structures interpret non-logical expressions like P and a.
- Structures do not interpret logical expressions like $\neg$ and $\forall x$.
- The fixed interpretation of logical expressions is specified in the definition of satisfaction.
  e.g. $|\neg\phi|^\alpha_\mathcal{A} = \mathrm{T}$ iff $|\phi|^\alpha_\mathcal{A} = \mathrm{F}$
- Similarly $=$ is treated as a logical expression, which is not assigned a semantic value by the structure.
- The fixed interpretation of $=$ is specified in the definition of satisfaction.

Let $\mathcal{A}$ be an $\mathcal{L}_=$-structure (i.e. an $\mathcal{L}_2$-structure).
Truth in $\mathcal{A}$ is defined just as before with one addition:

Let $\mathcal{A}$ be an $\mathcal{L}_=$-structure (i.e. an $\mathcal{L}_2$-structure).
Truth in $\mathcal{A}$ is defined just as before with one addition:

Definition: satisfaction of identity statements

(ix) $|s = t|^{\alpha}_{\mathcal{A}} = \text{T}$ if and only if $|s|^{\alpha}_{\mathcal{A}} = |t|^{\alpha}_{\mathcal{A}}$.

Let $\mathcal{A}$ be an $\mathcal{L}_=$-structure (i.e. an $\mathcal{L}_2$-structure).
Truth in $\mathcal{A}$ is defined just as before with one addition:

Definition: satisfaction of identity statements

(ix) $|s{=}t|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|s|^{\alpha}_{\mathcal{A}} = |t|^{\alpha}_{\mathcal{A}}$.

Note: $=$ is used in both $\mathcal{L}_=$ and the metalanguage.

Let $\mathcal{A}$ be an $\mathcal{L}_=$-structure (i.e. an $\mathcal{L}_2$-structure).
Truth in $\mathcal{A}$ is defined just as before with one addition:

Definition: satisfaction of identity statements

(ix) $|s = t|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|s|^{\alpha}_{\mathcal{A}} = |t|^{\alpha}_{\mathcal{A}}$.

Note: $=$ is used in both $\mathcal{L}_=$ and the metalanguage.

Let $\mathcal{A}$ be an $\mathcal{L}_=$-structure (i.e. an $\mathcal{L}_2$-structure).
Truth in $\mathcal{A}$ is defined just as before with one addition:

Definition: satisfaction of identity statements

(ix) $|s=t|^\alpha_\mathcal{A} = \text{T}$ if and only if $|s|^\alpha_\mathcal{A} = |t|^\alpha_\mathcal{A}$.

Note: $=$ is used in both $\mathcal{L}_=$ and the metalanguage.

Let $\mathcal{A}$ be an $\mathcal{L}_=$-structure (i.e. an $\mathcal{L}_2$-structure).
Truth in $\mathcal{A}$ is defined just as before with one addition:

Definition: satisfaction of identity statements

(ix) $|s=t|^{\alpha}_{\mathcal{A}} = \mathrm{T}$ if and only if $|s|^{\alpha}_{\mathcal{A}} = |t|^{\alpha}_{\mathcal{A}}$.

Note: $=$ is used in both $\mathcal{L}_=$ and the metalanguage.

The other definitions from Chapter 5 carry over directly to $\mathcal{L}_=$.

- Valid
- Logical truth
- Contradiction
- Logically equivalent
- Semantically consistent

These are defined just as before replacing '$\mathcal{L}_2$' with '$\mathcal{L}_=$'.

Worked example

$\forall x \, \forall y \, x = y$ isn't logically true.

Worked example

$\forall x \, \forall y \, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Worked example

$\forall x\, \forall y\, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof.

Worked example

$\forall x\, \forall y\, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.

Worked example

$\forall x \, \forall y \, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.

Sufficient to prove (STP:) $\forall x \, \forall y \, x = y$ is false in $\mathcal{A}$ under α.

Worked example

$\forall x \, \forall y \, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.

Sufficient to prove (STP:) $\forall x \, \forall y \, x = y$ is false in $\mathcal{A}$ under α.

Now: $|\forall x \forall y \, x = y|_{\mathcal{A}}^{\alpha} = \mathrm{T}$ iff $|\forall y \, x = y|_{\mathcal{A}}^{\beta} = \mathrm{T}$ for every β differing from α at most in x.

Worked example

$\forall x \, \forall y \, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.

Sufficient to prove (STP:) $\forall x \, \forall y \, x = y$ is false in $\mathcal{A}$ under α.

Now: $|\forall x \forall y \, x = y|^{\alpha}_{\mathcal{A}} = \mathrm{F}$ iff $|\forall y \, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ for some β differing from α at most in x.

Worked example

$\forall x\, \forall y\, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.

Sufficient to prove (STP:) $\forall x\, \forall y\, x = y$ is false in $\mathcal{A}$ under α.

Now: $|\forall x \forall y\, x = y|^{\alpha}_{\mathcal{A}} = \mathrm{F}$ iff $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ for some β differing from α at most in x.

STP: $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ for some assignment β differing from α at most in x.

Worked example

$\forall x\, \forall y\, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.

Sufficient to prove (STP:) $\forall x\, \forall y\, x = y$ is false in $\mathcal{A}$ under α.

Now: $|\forall x \forall y\, x = y|^{\alpha}_{\mathcal{A}} = \text{F}$ iff $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \text{F}$ for some β differing from α at most in x.

STP: $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \text{F}$ for some assignment β differing from α at most in x.

But: $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \text{T}$ iff $|x = y|^{\gamma}_{\mathcal{A}} = \text{T}$ for every γ differing from β at most in y.

Worked example

$\forall x \, \forall y \, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_{=}$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.

Sufficient to prove (STP:) $\forall x \, \forall y \, x = y$ is false in $\mathcal{A}$ under α.

Now: $|\forall x \forall y \, x = y|^{\alpha}_{\mathcal{A}} = \mathrm{F}$ iff $|\forall y \, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ for some β differing from α at most in x.

STP: $|\forall y \, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ for some assignment β differing from α at most in x.

But: $|\forall y \, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ iff $|x = y|^{\gamma}_{\mathcal{A}} = \mathrm{F}$ for some γ differing from β at most in y.

Worked example

$\forall x \, \forall y \, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.
Sufficient to prove (STP:) $\forall x \, \forall y \, x = y$ is false in $\mathcal{A}$ under α.

Now: $|\forall x \forall y \, x = y|^{\alpha}_{\mathcal{A}} = \mathrm{F}$ iff $|\forall y \, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ for some β differing from α at most in x.

STP: $|\forall y \, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ for some assignment β differing from α at most in x.

But: $|\forall y \, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ iff $|x = y|^{\gamma}_{\mathcal{A}} = \mathrm{F}$ for some γ differing from β at most in y.

STP: $|x = y|^{\gamma}_{\mathcal{A}} = \mathrm{F}$ for some γ differing from β in at most y.

Worked example

$\forall x\,\forall y\,x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.

Sufficient to prove (STP:) $\forall x\,\forall y\,x = y$ is false in $\mathcal{A}$ under α.

Now: $|\forall x \forall y\, x = y|^{\alpha}_{\mathcal{A}} = \mathrm{F}$ iff $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ for some β differing from α at most in x.

STP: $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ for some assignment β differing from α at most in x.

But: $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ iff $|x = y|^{\gamma}_{\mathcal{A}} = \mathrm{F}$ for some γ differing from β at most in y.

STP: $|x = y|^{\gamma}_{\mathcal{A}} = \mathrm{F}$ for some γ differing from α in at most x and y.

Worked example

$\forall x\,\forall y\, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.

Sufficient to prove (STP:) $\forall x\,\forall y\, x = y$ is false in $\mathcal{A}$ under α.

Now: $|\forall x \forall y\, x = y|^{\alpha}_{\mathcal{A}} = \text{F}$ iff $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \text{F}$ for some β differing from α at most in x.

STP: $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \text{F}$ for some assignment β differing from α at most in x.

But: $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \text{F}$ iff $|x = y|^{\gamma}_{\mathcal{A}} = \text{F}$ for some γ differing from β at most in y.

STP: $|x = y|^{\gamma}_{\mathcal{A}} = \text{F}$ for some γ differing from α in at most x and y.

So: Let γ assign x to 1 and y to 2 (otherwise agreeing with α)

Worked example

$\forall x \, \forall y \, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.

Sufficient to prove (STP:) $\forall x \, \forall y \, x = y$ is false in $\mathcal{A}$ under α.

Now: $|\forall x \forall y \, x = y|^{\alpha}_{\mathcal{A}} = \text{F}$ iff $|\forall y \, x = y|^{\beta}_{\mathcal{A}} = \text{F}$ for some β differing from α at most in x.

STP: $|\forall y \, x = y|^{\beta}_{\mathcal{A}} = \text{F}$ for some assignment β differing from α at most in x.

But: $|\forall y \, x = y|^{\beta}_{\mathcal{A}} = \text{F}$ iff $|x = y|^{\gamma}_{\mathcal{A}} = \text{F}$ for some γ differing from β at most in y.

STP: $|x = y|^{\gamma}_{\mathcal{A}} = \text{F}$ for some γ differing from α in at most x and y.

So: Let γ assign x to 1 and y to 2 (otherwise agreeing with α)
Then $|x|^{\gamma} \neq |y|^{\gamma}$;

Worked example

$\forall x\,\forall y\, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.

Sufficient to prove (STP:) $\forall x\,\forall y\, x = y$ is false in $\mathcal{A}$ under α.

Now: $|\forall x \forall y\, x = y|^{\alpha}_{\mathcal{A}} = \text{F}$ iff $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \text{F}$ for some β differing from α at most in x.

STP: $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \text{F}$ for some assignment β differing from α at most in x.

But: $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \text{F}$ iff $|x = y|^{\gamma}_{\mathcal{A}} = \text{F}$ for some γ differing from β at most in y.

STP: $|x = y|^{\gamma}_{\mathcal{A}} = \text{F}$ for some γ differing from α in at most x and y.

So: Let γ assign x to 1 and y to 2 (otherwise agreeing with α) Then $|x|^{\gamma} \neq |y|^{\gamma}$; so $|x = y|^{\gamma}_{\mathcal{A}} = \text{F}$.

Worked example

$\forall x\, \forall y\, x = y$ isn't logically true.

Counterexample: let $\mathcal{A}$ be an $\mathcal{L}_=$-structure with domain $\{1, 2\}$.

Proof. Let α be an assignment over $\mathcal{A}$.

Sufficient to prove (STP:) $\forall x\, \forall y\, x = y$ is false in $\mathcal{A}$ under α.

Now: $|\forall x \forall y\, x = y|^{\alpha}_{\mathcal{A}} = \mathrm{F}$ iff $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ for some β differing from α at most in x.

STP: $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ for some assignment β differing from α at most in x.

But: $|\forall y\, x = y|^{\beta}_{\mathcal{A}} = \mathrm{F}$ iff $|x = y|^{\gamma}_{\mathcal{A}} = \mathrm{F}$ for some γ differing from β at most in y.

STP: $|x = y|^{\gamma}_{\mathcal{A}} = \mathrm{F}$ for some γ differing from α in at most x and y.

So: Let γ assign x to 1 and y to 2 (otherwise agreeing with α)
Then $|x|^{\gamma} \neq |y|^{\gamma}$; so $|x = y|^{\gamma}_{\mathcal{A}} = \mathrm{F}$. QED

Proof theory

Natural Deduction for $\mathcal{L}_=$ has the same rules as Natural Deduction for $\mathcal{L}_2$ with the addition of rules for $=$.

Proof theory

Natural Deduction for $\mathcal{L}_=$ has the same rules as Natural Deduction for $\mathcal{L}_2$ with the addition of rules for $=$.

=Intro

Any assumption of the form $t = t$ where t is a constant can and must be discharged.

Proof theory

Natural Deduction for $\mathcal{L}_=$ has the same rules as Natural Deduction for $\mathcal{L}_2$ with the addition of rules for $=$.

=Intro

Any assumption of the form $t = t$ where t is a constant can and must be discharged.

A proof with an application of =Intro looks like this:

$$\frac{[t = t]}{\vdots}$$

Proof theory

Natural Deduction for $\mathcal{L}_=$ has the same rules as Natural Deduction for $\mathcal{L}_2$ with the addition of rules for $=$.

=Intro

Any assumption of the form $t = t$ where t is a constant can and must be discharged.

A proof with an application of =Intro looks like this:

$$\frac{[t = t]}{\vdots}$$

Example: prove $\vdash \forall z(z = z)$

Proof theory

Natural Deduction for $\mathcal{L}_=$ has the same rules as Natural Deduction for $\mathcal{L}_2$ with the addition of rules for $=$.

=Intro

Any assumption of the form $t = t$ where t is a constant can and must be discharged.

A proof with an application of =Intro looks like this:

$$\frac{[t = t]}{\vdots}$$

Example: prove $\vdash \forall z(z = z)$

$$a = a$$

Proof theory

Natural Deduction for $\mathcal{L}_=$ has the same rules as Natural Deduction for $\mathcal{L}_2$ with the addition of rules for $=$.

=Intro

Any assumption of the form $t = t$ where t is a constant can and must be discharged.

A proof with an application of =Intro looks like this:

$$\frac{[t = t]}{\vdots}$$

Example: prove $\vdash \forall z(z = z)$

$$[a = a]$$

Proof theory

Natural Deduction for $\mathcal{L}_=$ has the same rules as Natural Deduction for $\mathcal{L}_2$ with the addition of rules for $=$.

=Intro

Any assumption of the form $t = t$ where t is a constant can and must be discharged.

A proof with an application of =Intro looks like this:

$$\frac{[t = t]}{\vdots}$$

Example: prove $\vdash \forall z(z = z)$

$$\frac{[a = a]}{\forall z(z = z)}$$

=Elim

If s and t are constants, the result of appending $\phi[t/v]$ to a proof of $\phi[s/v]$ and a proof of $s=t$ or $t=s$ is a proof of $\phi[t/v]$.

$$\dfrac{\begin{matrix}\vdots \\ \phi[s/v]\end{matrix} \qquad \begin{matrix}\vdots \\ s=t\end{matrix}}{\phi[t/v]}\ {=}\mathrm{Elim} \qquad\qquad \dfrac{\begin{matrix}\vdots \\ \phi[s/v]\end{matrix} \qquad \begin{matrix}\vdots \\ t=s\end{matrix}}{\phi[t/v]}\ {=}\mathrm{Elim}$$

Worked example: prove the following.

$\vdash \forall x\, \forall y\, (Rxy \to (x = y \to Ryx))$

Worked example: prove the following.

$\vdash \forall x \forall y\,(Rxy \rightarrow (x = y \rightarrow Ryx))$

Rab

Worked example: prove the following.

$\vdash \forall x\, \forall y\, (Rxy \rightarrow (x = y \rightarrow Ryx))$

$$Rab \qquad a = b$$

$$\frac{\begin{array}{c}\vdots \\ \phi[s/v]\end{array} \qquad \begin{array}{c}\vdots \\ s = t\end{array}}{\phi[t/v]} = \text{Elim} \qquad \frac{\begin{array}{c}\vdots \\ \phi[s/v]\end{array} \qquad \begin{array}{c}\vdots \\ t = s\end{array}}{\phi[t/v]} = \text{Elim}$$

Worked example: prove the following.

$\vdash \forall x\, \forall y\, (Rxy \to (x = y \to Ryx))$

$$\frac{Rab \qquad a = b}{Raa}$$

$$\frac{\begin{matrix}\vdots \\ \phi[s/v]\end{matrix} \qquad \begin{matrix}\vdots \\ s = t\end{matrix}}{\phi[t/v]} = \text{Elim} \qquad\qquad \frac{\begin{matrix}\vdots \\ \phi[s/v]\end{matrix} \qquad \begin{matrix}\vdots \\ t = s\end{matrix}}{\phi[t/v]} = \text{Elim}$$

Worked example: prove the following.

$\vdash \forall x \, \forall y \, (Rxy \to (x = y \to Ryx))$

$$\frac{Rab \qquad a = b}{Raa} \qquad a = b$$

$$\frac{\begin{matrix} \vdots & & \vdots \\ \phi[s/v] & & s = t \end{matrix}}{\phi[t/v]} \text{=Elim} \qquad \frac{\begin{matrix} \vdots & & \vdots \\ \phi[s/v] & & t = s \end{matrix}}{\phi[t/v]} \text{=Elim}$$

Worked example: prove the following.

$\vdash \forall x\, \forall y\, (Rxy \to (x = y \to Ryx))$

$$\dfrac{\dfrac{Rab \qquad a=b}{Raa} \qquad a=b}{Rba}$$

$$\dfrac{\begin{matrix}\vdots \\ \phi[s/v]\end{matrix} \qquad \begin{matrix}\vdots \\ s=t\end{matrix}}{\phi[t/v]}\ {=}\text{Elim} \qquad\qquad \dfrac{\begin{matrix}\vdots \\ \phi[s/v]\end{matrix} \qquad \begin{matrix}\vdots \\ t=s\end{matrix}}{\phi[t/v]}\ {=}\text{Elim}$$

Worked example: prove the following.

$\vdash \forall x\, \forall y\, (Rxy \to (x = y \to Ryx))$

$$\dfrac{\dfrac{\dfrac{Rab \qquad a=b}{Raa} \qquad a=b}{Rba}}{a=b \to Rba}$$

Worked example: prove the following.

$\vdash \forall x\, \forall y\, (Rxy \rightarrow (x = y \rightarrow Ryx))$

$$\dfrac{\dfrac{\dfrac{Rab \qquad [a=b]}{Raa} \qquad a=b}{Rba}}{a=b \rightarrow Rba}$$

Worked example: prove the following.

$\vdash \forall x \forall y\, (Rxy \to (x = y \to Ryx))$

$$
\dfrac{\dfrac{\dfrac{Rab \qquad [a=b]}{Raa} \qquad [a=b]}{Rba}}{a=b \to Rba}
$$

Worked example: prove the following.

$\vdash \forall x\, \forall y\, (Rxy \to (x = y \to Ryx))$

$$\dfrac{\dfrac{\dfrac{\dfrac{Rab \qquad [a=b]}{Raa} \qquad [a=b]}{Rba}}{a=b \to Rba}}{Rab \to (a=b \to Rba)}$$

Worked example: prove the following.

$\vdash \forall x\, \forall y\, (Rxy \to (x = y \to Ryx))$

$$\dfrac{\dfrac{\dfrac{\dfrac{[Rab] \qquad [a=b]}{Raa} \qquad [a=b]}{Rba}}{a=b \to Rba}}{Rab \to (a=b \to Rba)}$$

Worked example: prove the following.

$\vdash \forall x\, \forall y\, (Rxy \to (x = y \to Ryx))$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[Rab] \qquad [a=b]}{Raa} \qquad [a=b]}{Rba}}{a=b \to Rba}}{Rab \to (a=b \to Rba)}}{\forall y\, (Ray \to (a=y \to Rya))}$$

Worked example: prove the following.

$\vdash \forall x\, \forall y\, (Rxy \to (x = y \to Ryx))$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[Rab] \qquad [a=b]}{Raa} \qquad [a=b]}{Rba}}{a=b \to Rba}}{Rab \to (a=b \to Rba)}}{\forall y\, (Ray \to (a=y \to Rya))}}{\forall x\, \forall y\, (Rxy \to (x=y \to Ryx))}
$$

Adequacy

Soundness and Completeness still hold.

Adequacy

Soundness and Completeness still hold.

Let Γ be a set of $\mathcal{L}_=$-sentences and ϕ an $\mathcal{L}_=$-sentence.

Theorem (adequacy)

$\Gamma \vdash \phi$ if and only if $\Gamma \models \phi$.

Formalisation with identity

Using $=$ one can formalise 'is [identical to]' in English.

Formalise:

William II is Wilhelm II.

Formalisation: $a = b$.
Dictionary: a: William II. b: Wilhelm II.

Formalisation with identity

Using $=$ one can formalise 'is [identical to]' in English.

Formalise:

William II is Wilhelm II.

Formalisation: $a = b$.
Dictionary: a: William II. b: Wilhelm II.

Note: don't confuse the 'is' of identity with the 'is' of predication.

Formalisation with identity

Using $=$ one can formalise 'is [identical to]' in English.

Formalise:

William II is Wilhelm II.

Formalisation: $a = b$.
Dictionary: a: William II. b: Wilhelm II.

Note: don't confuse the 'is' of identity with the 'is' of predication.

Formalise:

Wilhelm II is an emperor.

Formalisation with identity

Using $=$ one can formalise 'is [identical to]' in English.

Formalise:

William II is Wilhelm II.

Formalisation: $a = b$.
Dictionary: a: William II. b: Wilhelm II.

Note: don't confuse the 'is' of identity with the 'is' of predication.

Formalise:

Wilhelm II is an emperor.

Formalisation: Ea.
Dictionary: a: Wilhelm. E: ... is an emperor.

Formalisation with identity

Using = one can formalise ‘is [identical to]’ in English.

Formalise:

William II is Wilhelm II.

Formalisation: $a = b$.
Dictionary: a: William II. b: Wilhelm II.

Note: don’t confuse the ‘is’ of identity with the ‘is’ of predication.

Formalise:

Wilhelm II is an emperor.

Formalisation: Ea.
Dictionary: a: Wilhelm. E: … is an emperor.

Here ‘is’ forms part of the predicate ‘is an emperor.’

Identity can also be used to formalise numerical quantifiers.

Identity can also be used to formalise numerical quantifiers.

Dictionary: P: … is a perfect being.

Formalise

(1) There are at least two perfect beings.

Identity can also be used to formalise numerical quantifiers.

Dictionary: P: … is a perfect being.

Formalise

(1) There are at least two perfect beings.
Incorrect formalisation: $\exists x \exists y (Px \land Py)$.

Identity can also be used to formalise numerical quantifiers.

Dictionary: P: … is a perfect being.

Formalise

(1) There are at least two perfect beings.
Incorrect formalisation: $\exists x \exists y(Px \land Py)$.
Correct formalisation: $\exists x \exists y(Px \land Py \land \neg x = y)$.

Identity can also be used to formalise numerical quantifiers.

Dictionary: P: … is a perfect being.

Formalise

(1) There are at least two perfect beings.
Incorrect formalisation: $\exists x \exists y(Px \wedge Py)$.
Correct formalisation: $\exists x \exists y(Px \wedge Py \wedge \neg x = y)$.

(2) There is at most one perfect being.

Identity can also be used to formalise numerical quantifiers.

Dictionary: P: … is a perfect being.

Formalise

(1) There are at least two perfect beings.
Incorrect formalisation: $\exists x \exists y(Px \land Py)$.
Correct formalisation: $\exists x \exists y(Px \land Py \land \neg x = y)$.

(2) There is at most one perfect being.
Formalisation: $\neg \exists x \exists y(Px \land Py \land \neg x = y)$.

Identity can also be used to formalise numerical quantifiers.

Dictionary: P: … is a perfect being.

Formalise

(1) There are at least two perfect beings.
Incorrect formalisation: $\exists x \exists y (Px \land Py)$.
Correct formalisation: $\exists x \exists y (Px \land Py \land \neg x = y)$.

(2) There is at most one perfect being.
Formalisation: $\neg \exists x \exists y (Px \land Py \land \neg x = y)$.
Alternative formalisation: $\forall x \forall y ((Px \land Py) \rightarrow x = y)$.

Identity can also be used to formalise numerical quantifiers.

Dictionary: P: … is a perfect being.

Formalise

(1) There are at least two perfect beings.
Incorrect formalisation: $\exists x \exists y(Px \wedge Py)$.
Correct formalisation: $\exists x \exists y(Px \wedge Py \wedge \neg x = y)$.

(2) There is at most one perfect being.
Formalisation: $\neg \exists x \exists y(Px \wedge Py \wedge \neg x = y)$.
Alternative formalisation: $\forall x \forall y((Px \wedge Py) \rightarrow x = y)$.

(3) There is exactly one perfect being.

Identity can also be used to formalise numerical quantifiers.

Dictionary: P: … is a perfect being.

Formalise

(1) There are at least two perfect beings.
Incorrect formalisation: $\exists x \exists y(Px \wedge Py)$.
Correct formalisation: $\exists x \exists y(Px \wedge Py \wedge \neg x = y)$.

(2) There is at most one perfect being.
Formalisation: $\neg\exists x \exists y(Px \wedge Py \wedge \neg x = y)$.
Alternative formalisation: $\forall x \forall y((Px \wedge Py) \rightarrow x = y)$.

(3) There is exactly one perfect being.
Formalisation: $\exists x Px \wedge \forall x \forall y((Px \wedge Py) \rightarrow x = y)$.

Identity can also be used to formalise numerical quantifiers.

Dictionary: P: ... is a perfect being.

Formalise

(1) There are at least two perfect beings.
Incorrect formalisation: $\exists x \exists y(Px \land Py)$.
Correct formalisation: $\exists x \exists y(Px \land Py \land \neg x = y)$.

(2) There is at most one perfect being.
Formalisation: $\neg \exists x \exists y(Px \land Py \land \neg x = y)$.
Alternative formalisation: $\forall x \forall y((Px \land Py) \to x = y)$.

(3) There is exactly one perfect being.
Formalisation: $\exists x Px \land \forall x \forall y((Px \land Py) \to x = y)$.
Alternative formalisation: $\exists x(Px \land \forall y(Py \to y = x))$.

Definite descriptions

Examples of definite descriptions:

- ‘the Queen’
- ‘Bellerophon’s winged horse’
- ‘the author of Ulysses’

Definite descriptions

Examples of definite descriptions:

- ‘the Queen’
- ‘Bellerophon’s winged horse’
- ‘the author of Ulysses’

In $\mathcal{L}_2$: the best we can do is to formalise definite descriptions as constants.

Definite descriptions

Examples of definite descriptions:

- ‘the Queen’
- ‘Bellerophon’s winged horse’
- ‘the author of Ulysses’

In $\mathcal{L}_2$: the best we can do is to formalise definite descriptions as constants.

But this isn’t perfect…

Example

Belleraphon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

The obvious formalisation with constants is valid.

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

The obvious formalisation with constants is valid.

Formalisation: premiss: $\neg Rb$. Conclusion: $\exists x(x = b)$.
Dictionary: R: $\ldots$is real. b: Bellerophon's winged horse.

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

The obvious formalisation with constants is valid.

Formalisation: premiss: $\neg Rb$. Conclusion: $\exists x(x = b)$.
Dictionary: R: ...is real. b: Bellerophon's winged horse.

$$b = b$$

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

The obvious formalisation with constants is valid.

Formalisation: premiss: $\neg Rb$. Conclusion: $\exists x(x = b)$.
Dictionary: R: ...is real. b: Bellerophon's winged horse.

$$[b = b]$$

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

The obvious formalisation with constants is valid.

Formalisation: premiss: $\neg Rb$. Conclusion: $\exists x(x = b)$.
Dictionary: R: …is real. b: Bellerophon's winged horse.

$$\frac{[b = b]}{\exists x(x = b)}$$

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

The obvious formalisation with constants is valid.

Formalisation: premiss: $\neg Rb$. Conclusion: $\exists x(x = b)$.
Dictionary: R: ...is real. b: Bellerophon's winged horse.

$$\frac{[b = b]}{\exists x(x = b)}$$

(In fact: the conclusion is a logical truth.)

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

The obvious formalisation with constants is valid.

Formalisation: premiss: $\neg Rb$. Conclusion: $\exists x(x = b)$.
Dictionary: R: ...is real. b: Bellerophon's winged horse.

$$\frac{[b = b]}{\exists x(x = b)}$$

(In fact: the conclusion is a logical truth.)

Source of the trouble:

- $\mathcal{L}_=$-constants always refer to an object in a $\mathcal{L}_=$-structure.
- definite descriptions may fail to pick out a unique object.

Russell's theory of descriptions.

There's a better way to formalise definite descriptions in $\mathcal{L}_=$.

Russell's theory of descriptions.

There's a better way to formalise definite descriptions in $\mathcal{L}_=$.

Formalise:

The author of Ulysses wrote Dubliners.

Russell's theory of descriptions.

There's a better way to formalise definite descriptions in $\mathcal{L}_=$.

Formalise:

The author of Ulysses wrote Dubliners.

Russell analyses this as the conjunction of two claims.

(i) There is exactly one author of Ulysses

(ii) and it wrote Dubliners.

Russell's theory of descriptions.

There's a better way to formalise definite descriptions in $\mathcal{L}_=$.

Formalise:

The author of Ulysses wrote Dubliners.

Russell analyses this as the conjunction of two claims.

(i) There is exactly one author of Ulysses

(ii) and it wrote Dubliners.

Dictionary: A: ...is an author of Ulysses.
W: ...wrote Dubliners.

Russell's theory of descriptions.

There's a better way to formalise definite descriptions in $\mathcal{L}_=$.

Formalise:

The author of Ulysses wrote Dubliners.

Russell analyses this as the conjunction of two claims.

(i) There is exactly one author of Ulysses

(ii) and it wrote Dubliners.

Dictionary: A: ... is an author of Ulysses.
W: ... wrote Dubliners.

Formalisation: $\exists x\big(Ax \land \forall y(Ay \to y = x)$

Russell's theory of descriptions.

There's a better way to formalise definite descriptions in $\mathcal{L}_=$.

Formalise:

The author of Ulysses wrote Dubliners.

Russell analyses this as the conjunction of two claims.

(i) There is exactly one author of Ulysses

(ii) and it wrote Dubliners.

Dictionary: A: …is an author of Ulysses.
W: …wrote Dubliners.

Formalisation: $\exists x\big(Ax \land \forall y(Ay \to y = x) \land Wx\big)$

Formalise:

Bellerophon's winged horse isn't real.

R: ...is real. B: ...is a winged horse belonging to Bellerophon.

Formalise:

Bellerophon's winged horse isn't real.

R: …is real. B: …is a winged horse belonging to Bellerophon.

On Russell's view this can have two readings.

Formalise:

Bellerophon's winged horse isn't real.

R: ...is real. B: ...is a winged horse belonging to Bellerophon.

On Russell's view this can have two readings.

Paraphrase 1: (i) there is exactly one winged horse belonging to Bellerophon and (ii) it is not real.

Formalise:

Bellerophon's winged horse isn't real.

R: ...is real. B: ...is a winged horse belonging to Bellerophon.

On Russell's view this can have two readings.

Paraphrase 1: (i) there is exactly one winged horse belonging to Bellerophon and (ii) it is not real.

Formalisation 1: $\exists x\big(Bx \land \forall y(By \to y = x) \land \neg Rx\big)$.

Formalise:

Bellerophon's winged horse isn't real.

R: ... is real. B: ... is a winged horse belonging to Bellerophon.

On Russell's view this can have two readings.

Paraphrase 1: (i) there is exactly one winged horse belonging to Bellerophon and (ii) it is not real.

Formalisation 1: $\exists x\big(Bx \land \forall y(By \to y = x) \land \neg Rx\big)$.

Dubious: this is true only if there are non-real things .

Formalise:

Bellerophon's winged horse isn't real.

R: ...is real. B: ...is a winged horse belonging to Bellerophon.

On Russell's view this can have two readings.

Paraphrase 1: (i) there is exactly one winged horse belonging to Bellerophon and (ii) it is not real.

Formalisation 1: $\exists x\big(Bx \wedge \forall y(By \to y = x) \wedge \neg Rx\big)$.

Dubious: this is true only if there are non-real things .

Paraphrase 2: It's not the case that $\big($(i) there is exactly one winged horse belonging to Bellerophon and (ii) it is real$\big)$.

Formalise:

Bellerophon's winged horse isn't real.

R: . . . is real. B: . . . is a winged horse belonging to Bellerophon.

On Russell's view this can have two readings.

Paraphrase 1: (i) there is exactly one winged horse belonging to Bellerophon and (ii) it is not real.

Formalisation 1: $\exists x\big(Bx \land \forall y(By \to y = x) \land \neg Rx\big)$.

Dubious: this is true only if there are non-real things .

Paraphrase 2: It's not the case that $\big($(i) there is exactly one winged horse belonging to Bellerophon and (ii) it is real$\big)$.

Formalisation 2: $\neg\exists x\big(Bx \land \forall y(By \to y = x) \land Rx\big)$.

Example **Not valid**

Belleropon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

We can capture its non-validity by using the second formalisation of the premiss.

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

We can capture its non-validity by using the second formalisation of the premiss.

Dictionary: R: . . . is real.
B: . . . is a winged horse belonging to Bellerophon.

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

We can capture its non-validity by using the second formalisation of the premiss.

Dictionary: R: …is real.
B: …is a winged horse belonging to Bellerophon.

Formalisation

Premiss: $\neg\exists x\big(Bx \wedge \forall y(By \to y = x) \wedge Rx\big)$.

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

We can capture its non-validity by using the second formalisation of the premiss.

Dictionary: R: …is real.
B: …is a winged horse belonging to Bellerophon.

Formalisation

Premiss: $\neg\exists x\big(Bx \land \forall y(By \to y = x) \land Rx\big)$.
Conclusion: $\exists x Bx$.

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

We can capture its non-validity by using the second formalisation of the premiss.

Dictionary: R: . . . is real.
B: . . . is a winged horse belonging to Bellerophon.

Formalisation

Premiss: $\neg\exists x\big(Bx \wedge \forall y(By \to y = x) \wedge Rx\big)$.
Conclusion: $\exists x Bx$.

The structure $\mathcal{A}$ is a counterexample to this argument.

$$D_{\mathcal{A}} = \{x : x \text{ is a horse}\};\ |B|_{\mathcal{A}} = \emptyset.$$

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

We can capture its non-validity by using the second formalisation of the premiss.

Dictionary: R: …is real.
B: …is a winged horse belonging to Bellerophon.

Formalisation **Not valid**

Premiss: $\neg\exists x\big(Bx \wedge \forall y(By \rightarrow y = x) \wedge Rx\big)$.
Conclusion: $\exists x Bx$.

The structure $\mathcal{A}$ is a counterexample to this argument.

$$D_{\mathcal{A}} = \{x : x \text{ is a horse}\};\ |B|_{\mathcal{A}} = \emptyset.$$

Example **Not valid**

Bellerophon's winged horse isn't real; so there is something that is Bellerophon's winged horse.

We can capture its non-validity by using the second formalisation of the premiss.

Dictionary: R: …is real.
B: …is a winged horse belonging to Bellerophon.

Formalisation **Not valid**

Premiss: $\neg\exists x\big(Bx \land \forall y(By \to y = x) \land Rx\big)$.
Conclusion: $\exists x Bx$.

The structure $\mathcal{A}$ is a counterexample to this argument.

$$D_{\mathcal{A}} = \{x : x \text{ is a horse}\};\ |B|_{\mathcal{A}} = \emptyset.$$

(It doesn't matter what the extension of R is here.)

Multiple descriptions

We deal with these much like multiple quantifiers.

Formalise

The author of Ulysses likes the author of the Odyssey

Dictionary: U: …is an author of Ulysses
O: …is an author of the Odyssey. L: …likes …

Multiple descriptions

We deal with these much like multiple quantifiers.

Formalise

The author of Ulysses likes the author of the Odyssey

Dictionary: U: …is an author of Ulysses
O: …is an author of the Odyssey. L: …likes …

It's helpful to break this into two steps.

Partial formalisation:

$$\exists x_1\big(Ux_1 \wedge \forall y_1(Uy_1 \to y_1 = x_1)$$
$$\wedge\ x_1 \text{ likes the author of the Odyssey}\big)$$

Multiple descriptions

We deal with these much like multiple quantifiers.

Formalise

The author of Ulysses likes the author of the Odyssey

Dictionary: U: …is an author of Ulysses
O: …is an author of the Odyssey. L: …likes …

It's helpful to break this into two steps.

Partial formalisation:

$$\exists x_1 \big(Ux_1 \land \forall y_1 (Uy_1 \to y_1 = x_1) \\ \land\ x_1 \text{ likes the author of the Odyssey} \big)$$

It remains to formalise 'x_1 likes the author of the Odyssey'.

x_1 likes the author of the Odyssey

Paraphrase: the author of the Odyssey is liked by x_1.

x_1 likes the author of the Odyssey

Paraphrase: the author of the Odyssey is liked by x_1.

Formalisation: $\exists x_2\big(Ox_2 \wedge \forall y_2(Oy_2 \to y_2 = x_2) \wedge Lx_1x_2\big)$.

x_1 likes the author of the Odyssey

Paraphrase: the author of the Odyssey is liked by x_1.

Formalisation: $\exists x_2\big(Ox_2 \land \forall y_2(Oy_2 \to y_2 = x_2) \land Lx_1x_2\big)$.

Finally, we put this together with what we had before.

The author of Ulysses likes the author of the Odyssey

$$\exists x_1\big(Ux_1 \land \forall y_1(Uy_1 \to y_1 = x_1) \\ \land x_1 \text{ likes the author of the Odyssey}\big).$$

x_1 likes the author of the Odyssey

Paraphrase: the author of the Odyssey is liked by x_1.

Formalisation: $\exists x_2 \big(Ox_2 \land \forall y_2 (Oy_2 \to y_2 = x_2) \land Lx_1x_2\big)$.

Finally, we put this together with what we had before.

The author of Ulysses likes the author of the Odyssey

$$\exists x_1 \big(Ux_1 \land \forall y_1 (Uy_1 \to y_1 = x_1) \land x_1 \text{ likes the author of the Odyssey}\big).$$

x_1 likes the author of the Odyssey

Paraphrase: the author of the Odyssey is liked by x_1.

Formalisation: $\exists x_2\big(Ox_2 \land \forall y_2(Oy_2 \to y_2 = x_2) \land Lx_1x_2\big)$.

Finally, we put this together with what we had before.

The author of Ulysses likes the author of the Odyssey

$$\exists x_1\big(Ux_1 \land \forall y_1(Uy_1 \to y_1 = x_1) \\ \land\, x_1 \text{ likes the author of the Odyssey}\big).$$

$$\exists x_1\big(Ux_1 \land \forall y_1(Uy_1 \to y_1 = x_1) \\ \land\, \exists x_2\big(Ox_2 \land \forall y_2(Oy_2 \to y_2 = x_2) \land Lx_1x_2\big)\big).$$

x_1 likes the author of the Odyssey

Paraphrase: the author of the Odyssey is liked by x_1.

Formalisation: $\exists x_2\big(Ox_2 \land \forall y_2(Oy_2 \to y_2 = x_2) \land Lx_1x_2\big)$.

Finally, we put this together with what we had before.

The author of Ulysses likes the author of the Odyssey

$$\exists x_1\big(Ux_1 \land \forall y_1(Uy_1 \to y_1 = x_1) \\ \land x_1 \text{ likes the author of the Odyssey}\big).$$

$$\exists x_1\big(Ux_1 \land \forall y_1(Uy_1 \to y_1 = x_1) \\ \land \exists x_2\big(Ox_2 \land \forall y_2(Oy_2 \to y_2 = x_2) \land Lx_1x_2\big)\big).$$

x_1 likes the author of the Odyssey

Paraphrase: the author of the Odyssey is liked by x_1.

Formalisation: $\exists x_2\big(Ox_2 \land \forall y_2(Oy_2 \to y_2 = x_2) \land Lx_1x_2\big)$.

Finally, we put this together with what we had before.

The author of Ulysses likes the author of the Odyssey

$$\exists x_1\big(Ux_1 \land \forall y_1(Uy_1 \to y_1 = x_1) \\ \land x_1 \text{ likes the author of the Odyssey}\big).$$

$$\exists x_1\big(Ux_1 \land \forall y_1(Uy_1 \to y_1 = x_1) \\ \land \exists x_2\big(Ox_2 \land \forall y_2(Oy_2 \to y_2 = x_2) \land Lx_1x_2\big)\big).$$

x_1 likes the author of the Odyssey

Paraphrase: the author of the Odyssey is liked by x_1.

Formalisation: $\exists x_2\big(Ox_2 \land \forall y_2(Oy_2 \to y_2 = x_2) \land Lx_1x_2\big)$.

Finally, we put this together with what we had before.

The author of Ulysses likes the author of the Odyssey

$$\exists x_1\big(Ux_1 \land \forall y_1(Uy_1 \to y_1 = x_1) \land x_1 \text{ likes the author of the Odyssey}\big).$$

$$\exists x_1\big(Ux_1 \land \forall y_1(Uy_1 \to y_1 = x_1) \land \exists x_2\big(Ox_2 \land \forall y_2(Oy_2 \to y_2 = x_2) \land Lx_1x_2\big)\big).$$

Logical constants

$\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$ and $=$ are our only logical expressions.

Logical constants

$\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$ and $=$ are our only logical expressions. 45

This raises two questions:

Logical constants

$\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$ and $=$ are our only logical expressions. 45

This raises two questions:

Q1 What's special about these expressions?

Logical constants

$\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$ and $=$ are our only logical expressions. 45

This raises two questions:

Q1 What's special about these expressions?

A1 Alfred Tarski proposes to analyse topic neutrality in terms of 'permutation invariance'

Logical constants

$\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$ and $=$ are our only logical expressions. 45

This raises two questions:

Q1 What's special about these expressions?

A1 Alfred Tarski proposes to analyse topic neutrality in terms of 'permutation invariance'

- Roughly: logical expressions are the ones whose meaning is insensitive to which object is which.

Logical constants

$\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$ and $=$ are our only logical expressions. 45

This raises two questions:

Q1 What's special about these expressions?

A1 Alfred Tarski proposes to analyse topic neutrality in terms of 'permutation invariance'

- Roughly: logical expressions are the ones whose meaning is insensitive to which object is which.
- See Tarski 'What are Logical Notions?' *History and Philosophy of Logic 7*, 143–154.

Q2 What happens if we add more logical constants?

Q2 What happens if we add more logical constants?

A2 This is the business of philosophical logic.

Q2 What happens if we add more logical constants?

A2 This is the business of philosophical logic.

Extension of $\mathcal{L}_2$	New logical expressions
Generalised quantifiers	more than half infinitely many, etc.
Modal logic	It is necessarily the case that It is possibly the case that
Deontic logic	It is obligatory that It is permissible that

Q2 What happens if we add more logical constants?

A2 This is the business of philosophical logic.

Extension of $\mathcal{L}_2$	New logical expressions
Generalised quantifiers	more than half infinitely many, etc.
Modal logic	It is necessarily the case that It is possibly the case that
Deontic logic	It is obligatory that It is permissible that

See the finals paper 127: Philosophical Logic.

Decidability

There's an important difference between $\mathcal{L}_1$ and $\mathcal{L}_=$.

Decidability

There's an important difference between $\mathcal{L}_1$ and $\mathcal{L}_=$.

Let Γ be a finite set of sentences and ϕ a sentence.

Propositional Case

When these are all $\mathcal{L}_1$-sentences, we have a single effective procedure to determine whether or not $\Gamma \vDash \phi$.

Decidability

There's an important difference between $\mathcal{L}_1$ and $\mathcal{L}_=$.
Let Γ be a finite set of sentences and ϕ a sentence.

Propositional Case

When these are all $\mathcal{L}_1$-sentences, we have a single effective procedure to determine whether or not $\Gamma \vDash \phi$.

- Construct a full truth-table

Decidability

There's an important difference between $\mathcal{L}_1$ and $\mathcal{L}_=$.
Let Γ be a finite set of sentences and ϕ a sentence.

Propositional Case

When these are all $\mathcal{L}_1$-sentences, we have a single effective procedure to determine whether or not $\Gamma \vDash \phi$.

- Construct a full truth-table

This method can easily be automated.

Decidability

There's an important difference between $\mathcal{L}_1$ and $\mathcal{L}_=$.
Let Γ be a finite set of sentences and ϕ a sentence.

Propositional Case

When these are all $\mathcal{L}_1$-sentences, we have a single effective procedure to determine whether or not $\Gamma \vDash \phi$.

- Construct a full truth-table

This method can easily be automated.

Predicate Case

When these are $\mathcal{L}_=$-sentences, we have two methods.

Decidability

There's an important difference between $\mathcal{L}_1$ and $\mathcal{L}_=$.
Let Γ be a finite set of sentences and ϕ a sentence.

Propositional Case

When these are all $\mathcal{L}_1$-sentences, we have a single effective procedure to determine whether or not $\Gamma \vDash \phi$.

- Construct a full truth-table

This method can easily be automated.

Predicate Case

When these are $\mathcal{L}_=$-sentences, we have two methods.

- To establish $\Gamma \vDash \phi$ we construct a Natural Deduction proof.

Decidability

There's an important difference between $\mathcal{L}_1$ and $\mathcal{L}_=$.
Let Γ be a finite set of sentences and ϕ a sentence.

Propositional Case

When these are all $\mathcal{L}_1$-sentences, we have a single effective procedure to determine whether or not $\Gamma \vDash \phi$.

- Construct a full truth-table

This method can easily be automated.

Predicate Case

When these are $\mathcal{L}_=$-sentences, we have two methods.

- To establish $\Gamma \vDash \phi$ we construct a Natural Deduction proof.
- To establish $\Gamma \nvDash \phi$ we construct a counterexample.

Decidability

There's an important difference between $\mathcal{L}_1$ and $\mathcal{L}_=$.
Let Γ be a finite set of sentences and ϕ a sentence.

Propositional Case

When these are all $\mathcal{L}_1$-sentences, we have a single effective procedure to determine whether or not $\Gamma \vDash \phi$.

- Construct a full truth-table

This method can easily be automated.

Predicate Case

When these are $\mathcal{L}_=$-sentences, we have two methods.

- To establish $\Gamma \vDash \phi$ we construct a Natural Deduction proof.
- To establish $\Gamma \nvDash \phi$ we construct a counterexample.

But: we need to know whether or not the argument is valid before we know which method to apply.

Is there a single effective procedure for determining whether or not an $\mathcal{L}_=$-argument is valid?

Is there a single effective procedure for determining whether or not an $\mathcal{L}_=$-argument is valid?

- On two natural regimentations of ‘effective procedure’ the answer is negative.

Is there a single effective procedure for determining whether or not an $\mathcal{L}_=$-argument is valid?

- On two natural regimentations of 'effective procedure' the answer is negative.

Theorem (Church-Turing 1936/7)

There is no 'recursive' or 'Turing computable' method for deciding whether an $\mathcal{L}_=$-argument with finitely many premisses is valid.

Is there a single effective procedure for determining whether or not an $\mathcal{L}_=$-argument is valid?

- On two natural regimentations of ‘effective procedure’ the answer is negative.

Theorem (Church-Turing 1936/7)

There is no ‘recursive’ or ‘Turing computable’ method for deciding whether an $\mathcal{L}_=$-argument with finitely many premisses is valid.

- We cannot write a computer programme that, when applied to an $\mathcal{L}_=$-argument, delivers a ‘yes’/‘no’ output according to whether the argument is valid or not.

Is there a single effective procedure for determining whether or not an $\mathcal{L}_=$-argument is valid?

- On two natural regimentations of 'effective procedure' the answer is negative.

Theorem (Church-Turing 1936/7)

There is no 'recursive' or 'Turing computable' method for deciding whether an $\mathcal{L}_=$-argument with finitely many premisses is valid.

- We cannot write a computer programme that, when applied to an $\mathcal{L}_=$-argument, delivers a 'yes'/'no' output according to whether the argument is valid or not.
- This holds even if no restrictions are imposed on the memory, disk space, computation time, etc.

fin