First Order Language

Ming Hsiung

School of Philosophy and Social Development
South China Normal University

Contents

Contents

Example

- Socrates is a philosopher.
- Socrates influences Plato.
- Socrates influences all philosophers.
- Some philosopher influences all philosophers.

Symbols	Interpretations
$P(x)$	x is a philosopher
$Q(x_1, x_2)$	x_1 influences x_2
a	Socrates
b	Plato

- Socrates is a philosopher.
- $P(a)$

Symbols	Interpretations
$P(x)$	x is a philosopher
$Q(x_1, x_2)$	x_1 influences x_2
a	Socrates
b	Plato

- Socrates is a philosopher.
- $P(a)$

Symbols	Interpretations
$P(x)$	x is a philosopher
$Q(x_1, x_2)$	x_1 influences x_2
a	Socrates
b	Plato

- Socrates is a philosopher.
- $P(a)$

Symbols	Interpretations
$P(x)$	x is a philosopher
$Q(x_1, x_2)$	x_1 influences x_2
a	Socrates
b	Plato

- Socrates influences Plato.
- $Q(a, b)$

Symbols	Interpretations
$P(x)$	x is a philosopher
$Q(x_1, x_2)$	x_1 influences x_2
a	Socrates
b	Plato

- Socrates influences Plato.
- $Q(a, b)$

Symbols	Interpretations
$P(x)$	x is a philosopher
$Q(x_1, x_2)$	x_1 influences x_2
a	Socrates
b	Plato

- Socrates influences Plato.
- $Q(a, b)$

Symbols	Interpretations
$P(x)$	x is a philosopher
$Q(x_1, x_2)$	x_1 influences x_2
a	Socrates
b	Plato

- Socrates influences all philosophers.
- $\forall x(P(x) \to Q(a, x))$

Symbols	Interpretations
$P(x)$	x is a philosopher
$Q(x_1, x_2)$	x_1 influences x_2
a	Socrates
b	Plato

- Socrates influences all philosophers.
- $\forall x(P(x) \rightarrow Q(a, x))$

Symbols	Interpretations
$P(x)$	x is a philosopher
$Q(x_1, x_2)$	x_1 influences x_2
a	Socrates
b	Plato

- Socrates influences all philosophers.
- $\forall x(P(x) \to Q(a, x))$

Symbols	Interpretations
$P(x)$	x is a philosopher
$Q(x_1, x_2)$	x_1 influences x_2
a	Socrates
b	Plato

- Some philosopher influences all philosophers.
- $\exists x_1 (P(x_1) \wedge \forall x_2 (P(x_2) \to Q(x_1, x_2)))$

Symbols	Interpretations
$P(x)$	x is a philosopher
$Q(x_1, x_2)$	x_1 influences x_2
a	Socrates
b	Plato

- Some philosopher influences all philosophers.
- $\exists x_1(P(x_1) \wedge \forall x_2(P(x_2) \rightarrow Q(x_1, x_2))$

Symbols	Interpretations
$P(x)$	x is a philosopher
$Q(x_1, x_2)$	x_1 influences x_2
a	Socrates
b	Plato

- Some philosopher influences all philosophers.
- $\exists x_1 (P(x_1) \wedge \forall x_2 (P(x_2) \rightarrow Q(x_1, x_2))$

A toy structure

- A (non-empty) domain A (the set of human beings, or any other set including all philosophers).
- the property R_1 (being a philosophy), as the interpretation of P
- the relation R_2 (influencing), as the interpretation of Q
- the philosopher Socrates, as the interpretation of a
- the philosopher Plato, as the interpretation of b

$$\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$$

The similarity type of $\mathfrak{A}$ is

$$\langle 1, 2; -; 2 \rangle$$

A toy structure

- A (non-empty) domain A (the set of human beings, or any other set including all philosophers).
- the property R_1 (being a philosophy), as the interpretation of P
- the relation R_2 (influencing), as the interpretation of Q
- the philosopher Socrates, as the interpretation of a
- the philosopher Plato, as the interpretation of b

$$\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$$

The similarity type of $\mathfrak{A}$ is

$$\langle 1, 2; -; 2 \rangle$$

A toy structure

- A (non-empty) domain A (the set of human beings, or any other set including all philosophers).
- the property R_1 (being a philosophy), as the interpretation of P
- the relation R_2 (influencing), as the interpretation of Q
- the philosopher Socrates, as the interpretation of a
- the philosopher Plato, as the interpretation of b

$$\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$$

The similarity type of $\mathfrak{A}$ is

$$\langle 1, 2; -; 2 \rangle$$

A toy structure

- A (non-empty) domain A (the set of human beings, or any other set including all philosophers).
- the property R_1 (being a philosophy), as the interpretation of P
- the relation R_2 (influencing), as the interpretation of Q
- the philosopher Socrates, as the interpretation of a
- the philosopher Plato, as the interpretation of b

$$\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$$

The similarity type of $\mathfrak{A}$ is

$$\langle 1, 2; -; 2 \rangle$$

A toy structure

- A (non-empty) domain A (the set of human beings, or any other set including all philosophers).
- the property R_1 (being a philosophy), as the interpretation of P
- the relation R_2 (influencing), as the interpretation of Q
- the philosopher Socrates, as the interpretation of a
- the philosopher Plato, as the interpretation of b

$$\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$$

The similarity type of $\mathfrak{A}$ is

$$\langle 1, 2; -; 2 \rangle$$

A toy structure

- A (non-empty) domain A (the set of human beings, or any other set including all philosophers).
- the property R_1 (being a philosophy), as the interpretation of P
- the relation R_2 (influencing), as the interpretation of Q
- the philosopher Socrates, as the interpretation of a
- the philosopher Plato, as the interpretation of b

$$\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$$

The similarity type of $\mathfrak{A}$ is

$$\langle 1, 2; -; 2 \rangle$$

The alphabet for a toy language

- (individual) variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$
- quantifiers: $\forall$, $\exists$
- constant symbols: a, b
- predicate symbols: P, Q
- function symbols: none
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P, Q, a, b \rangle$$

The alphabet for a toy language

- (individual) variables: $x_0, x_1, \ldots$.
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: a, b
- predicate symbols: P, Q
- function symbols: none
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P, Q, a, b \rangle$$

The alphabet for a toy language

- (individual) variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$
- quantifiers: $\forall$, $\exists$
- constant symbols: a, b
- predicate symbols: P, Q
- function symbols: none
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P, Q, a, b \rangle$$

The alphabet for a toy language

- (individual) variables: $x_0, x_1, \ldots$.
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: a, b
- predicate symbols: P, Q
- function symbols: none
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P, Q, a, b \rangle$$

The alphabet for a toy language

- (individual) variables: $x_0, x_1, \ldots$.
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: a, b
- predicate symbols: P, Q
- function symbols: none
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P, Q, a, b \rangle$$

The alphabet for a toy language

- (individual) variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: a, b
- predicate symbols: P, Q
- function symbols: none
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P, Q, a, b \rangle$$

The alphabet for a toy language

- (individual) variables: $x_0, x_1, \ldots$.
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$
- quantifiers: $\forall$, $\exists$
- constant symbols: a, b
- predicate symbols: P, Q
- function symbols: none
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P, Q, a, b \rangle$$

The alphabet for a toy language

- (individual) variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: a, b
- predicate symbols: P, Q
- function symbols: none
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P, Q, a, b \rangle$$

Terms and formulas for the toy language

- Terms are variables or constant symbols.

- $\perp$ is a formula.
- If t, t_1, and t_2 are terms, then expressions of form $P(t)$ or $Q(t_1, t_2)$ are formulas.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Terms and formulas for the toy language

- Terms are variables or constant symbols.

- $\bot$ is a formula.
- If t, t_1, and t_2 are terms, then expressions of form $P(t)$ or $Q(t_1, t_2)$ are formulas.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Terms and formulas for the toy language

- Terms are variables or constant symbols.

- $\perp$ is a formula.
- If t, t_1, and t_2 are terms, then expressions of form $P(t)$ or $Q(t_1, t_2)$ are formulas.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Terms and formulas for the toy language

- Terms are variables or constant symbols.

- $\bot$ is a formula.
- If t, t_1, and t_2 are terms, then expressions of form $P(t)$ or $Q(t_1, t_2)$ are formulas.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Terms and formulas for the toy language

- Terms are variables or constant symbols.

- $\perp$ is a formula.
- If t, t_1, and t_2 are terms, then expressions of form $P(t)$ or $Q(t_1, t_2)$ are formulas.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Terms and formulas for the toy language

- Terms are variables or constant symbols.

- $\perp$ is a formula.
- If t, t_1, and t_2 are terms, then expressions of form $P(t)$ or $Q(t_1, t_2)$ are formulas.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Terms and formulas for the toy language

- Terms are variables or constant symbols.

- $\perp$ is a formula.
- If t, t_1, and t_2 are terms, then expressions of form $P(t)$ or $Q(t_1, t_2)$ are formulas.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Contents

Example

- 0 is not the successor of any number.
- 0 is less than 1.
- 0 is an even number.
- Any number multiplied by 0 is still equal to 0.

Symbols	Interpretations
$x_1 \doteq x_2$	x_1 is equal to x_2
$\boldsymbol{S}x$	the successor of x
$x_1 + x_2$	the sum of x_1 and x_2
$x_1 \times x_2$	the product of x_1 and x_2
$\mathbf{0}$	zero

- 0 is not (equal to) the successor of any number.
- $\forall x \neg(0 \doteq \boldsymbol{S}x)$

Symbols	Interpretations
$x_1 \doteq x_2$	x_1 is equal to x_2
$\boldsymbol{S}x$	the successor of x
$x_1 + x_2$	the sum of x_1 and x_2
$x_1 \times x_2$	the product of x_1 and x_2
$\mathbf{0}$	zero

- 0 is not (equal to) the successor of any number.
- $\forall x \neg (0 \doteq \boldsymbol{S}x)$

Symbols	Interpretations
$x_1 \doteq x_2$	x_1 is equal to x_2
$\boldsymbol{S}x$	the successor of x
$x_1 + x_2$	the sum of x_1 and x_2
$x_1 \times x_2$	the product of x_1 and x_2
$\mathbf{0}$	zero

- 0 is not (equal to) the successor of any number.
- $\forall x \neg (0 \doteq \boldsymbol{S}x)$

Symbols	Interpretations
$x_1 \doteq x_2$	x_1 is equal to x_2
$\boldsymbol{S}x$	the successor of x
$x_1 + x_2$	the sum of x_1 and x_2
$x_1 \times x_2$	the product of x_1 and x_2
$\mathbf{0}$	zero

- 0 is an even number.
- $\exists x\, (\mathbf{0} \doteq \boldsymbol{S}\boldsymbol{S}\mathbf{0} \times x)$

Symbols	Interpretations
$x_1 \doteq x_2$	x_1 is equal to x_2
$\boldsymbol{S}x$	the successor of x
$x_1 + x_2$	the sum of x_1 and x_2
$x_1 \times x_2$	the product of x_1 and x_2
$\mathbf{0}$	zero

- 0 is an even number.
- $\exists x\,(\mathbf{0} \doteq \boldsymbol{S}\boldsymbol{S}\mathbf{0} \times x)$

Symbols	Interpretations
$x_1 \doteq x_2$	x_1 is equal to x_2
$\boldsymbol{S}x$	the successor of x
$x_1 + x_2$	the sum of x_1 and x_2
$x_1 \times x_2$	the product of x_1 and x_2
$\mathbf{0}$	zero

- 0 is an even number.
- $\exists x\,(\mathbf{0} \doteq \boldsymbol{S}\boldsymbol{S}\mathbf{0} \times x)$

Symbols	Interpretations
$x_1 \doteq x_2$	x_1 is equal to x_2
$\boldsymbol{S}x$	the successor of x
$x_1 + x_2$	the sum of x_1 and x_2
$x_1 \times x_2$	the product of x_1 and x_2
$\mathbf{0}$	zero

- Any number multiplied by 0 is still equal to 0.
- $\forall x(x \times \mathbf{0} \doteq \mathbf{0})$

Symbols	Interpretations
$x_1 \doteq x_2$	x_1 is equal to x_2
$\boldsymbol{S}x$	the successor of x
$x_1 + x_2$	the sum of x_1 and x_2
$x_1 \times x_2$	the product of x_1 and x_2
$\mathbf{0}$	zero

- Any number multiplied by 0 is still equal to 0.
- $\forall x(x \times \mathbf{0} \doteq \mathbf{0})$

Symbols	Interpretations
$x_1 \doteq x_2$	x_1 is equal to x_2
$\boldsymbol{S}x$	the successor of x
$x_1 + x_2$	the sum of x_1 and x_2
$x_1 \times x_2$	the product of x_1 and x_2
$\mathbf{0}$	zero

- Any number multiplied by 0 is still equal to 0.
- $\forall x(x \times \mathbf{0} \doteq \mathbf{0})$

The structure of natural numbers

- The domain $\mathbb{N}$ of (all) natural numbers.
- the equality relation $=$, as the interpretation of $\doteq$
- the successor operation $'$, as the interpretation of S
- the addition operation $+$, as the interpretation of $+$
- the multiplication operation $\times$, as the interpretation of $\times$
- the number 0, as the interpretation of $\mathbf{0}$

$$\mathfrak{N} = \langle \mathbb{N}, ', +, \times, 0 \rangle$$

The similarity type of $\mathfrak{N}$ is

$$\langle -; 1, 2, 2; 1 \rangle$$

The structure of natural numbers

- The domain $\mathbb{N}$ of (all) natural numbers.
- the equality relation $=$, as the interpretation of $\doteq$
- the successor operation $'$, as the interpretation of S
- the addition operation $+$, as the interpretation of $+$
- the multiplication operation $\times$, as the interpretation of $\times$
- the number 0, as the interpretation of $\mathbf{0}$

$$\mathfrak{N} = \langle \mathbb{N}, ', +, \times, 0 \rangle$$

The similarity type of $\mathfrak{N}$ is

$$\langle -; 1, 2, 2; 1 \rangle$$

The structure of natural numbers

- The domain $\mathbb{N}$ of (all) natural numbers.
- the equality relation $=$, as the interpretation of $\doteq$
- the successor operation $'$, as the interpretation of S
- the addition operation $+$, as the interpretation of $\mathbf{+}$
- the multiplication operation $\times$, as the interpretation of $\mathbf{\times}$
- the number 0, as the interpretation of $\mathbf{0}$

$$\mathfrak{N} = \langle \mathbb{N}, ', +, \times, 0 \rangle$$

The similarity type of $\mathfrak{N}$ is

$$\langle -; 1, 2, 2; 1 \rangle$$

The structure of natural numbers

- The domain $\mathbb{N}$ of (all) natural numbers.
- the equality relation $=$, as the interpretation of $\doteq$
- the successor operation $'$, as the interpretation of S
- the addition operation $+$, as the interpretation of $+$
- the multiplication operation $\times$, as the interpretation of $\times$
- the number 0, as the interpretation of $\mathbf{0}$

$$\mathfrak{N} = \langle \mathbb{N}, ', +, \times, 0 \rangle$$

The similarity type of $\mathfrak{N}$ is

$$\langle -; 1, 2, 2; 1 \rangle$$

The structure of natural numbers

- The domain $\mathbb{N}$ of (all) natural numbers.
- the equality relation $=$, as the interpretation of $\doteq$
- the successor operation $'$, as the interpretation of $\boldsymbol{S}$
- the addition operation $+$, as the interpretation of $\boldsymbol{+}$
- the multiplication operation $\times$, as the interpretation of $\boldsymbol{\times}$
- the number 0, as the interpretation of $\boldsymbol{0}$

$$\mathfrak{N} = \langle \mathbb{N}, ', +, \times, 0 \rangle$$

The similarity type of $\mathfrak{N}$ is

$$\langle -; 1, 2, 2; 1 \rangle$$

The structure of natural numbers

- The domain $\mathbb{N}$ of (all) natural numbers.
- the equality relation $=$, as the interpretation of $\doteq$
- the successor operation $'$, as the interpretation of $\boldsymbol{S}$
- the addition operation $+$, as the interpretation of $\boldsymbol{+}$
- the multiplication operation $\times$, as the interpretation of $\boldsymbol{\times}$
- the number 0, as the interpretation of $\mathbf{0}$

$$\mathfrak{N} = \langle \mathbb{N}, ', +, \times, 0 \rangle$$

The similarity type of $\mathfrak{N}$ is

$$\langle -; 1, 2, 2; 1 \rangle$$

The structure of natural numbers

- The domain $\mathbb{N}$ of (all) natural numbers.
- the equality relation $=$, as the interpretation of $\doteq$
- the successor operation $'$, as the interpretation of S
- the addition operation $+$, as the interpretation of $+$
- the multiplication operation $\times$, as the interpretation of $\times$
- the number 0, as the interpretation of $\mathbf{0}$

$$\mathfrak{N} = \langle \mathbb{N}, ', +, \times, 0 \rangle$$

The similarity type of $\mathfrak{N}$ is

$$\langle -; 1, 2, 2; 1 \rangle$$

The arithmetic language

- variables: $x_0, x_1, \ldots$.
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\mathbf{0}$
- predicate symbols: $\doteq$ (which, again, is always uploaded, and so is deliberately not mentioned)
- function symbols: S, $+$, $\times$
- auxiliary symbols:) , (.

$$\mathcal{L}_A = \langle S, +, \times, \mathbf{0} \rangle$$

The arithmetic language

- variables: $x_0, x_1, \ldots$.
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\mathbf{0}$
- predicate symbols: $\doteq$ (which, again, is always uploaded, and so is deliberately not mentioned)
- function symbols: S, $+$, $\times$
- auxiliary symbols:) , (.

$$\mathcal{L}_A = \langle \boldsymbol{S}, +, \times, \mathbf{0} \rangle$$

The arithmetic language

- variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\mathbf{0}$
- predicate symbols: $\doteq$ (which, again, is always uploaded, and so is deliberately not mentioned)
- function symbols: S, $+$, $\times$
- auxiliary symbols:) , (.

$$\mathcal{L}_A = \langle \mathbf{S}, +, \times, \mathbf{0} \rangle$$

The arithmetic language

- variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\mathbf{0}$
- predicate symbols: $\doteq$ (which, again, is always uploaded, and so is deliberately not mentioned)
- function symbols: S, $+$, $\times$
- auxiliary symbols:) , (.

$$\mathcal{L}_A = \langle S, +, \times, \mathbf{0} \rangle$$

The arithmetic language

- variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\mathbf{0}$
- predicate symbols: $\doteq$ (which, again, is always uploaded, and so is deliberately not mentioned)
- function symbols: S, $+$, $\times$
- auxiliary symbols:) , (.

$$\mathcal{L}_A = \langle S, +, \times, \mathbf{0} \rangle$$

The arithmetic language

- variables: x_0, x_1, ….
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\mathbf{0}$
- predicate symbols: $\doteq$ (which, again, is always uploaded, and so is deliberately not mentioned)
- function symbols: S, $+$, $\times$
- auxiliary symbols:) , (.

$$\mathcal{L}_A = \langle S, +, \times, \mathbf{0} \rangle$$

The arithmetic language

- variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\mathbf{0}$
- predicate symbols: $\doteq$ (which, again, is always uploaded, and so is deliberately not mentioned)
- function symbols: S, $+$, $\times$
- auxiliary symbols:) , (.

$$\mathcal{L}_A = \langle S, +, \times, \mathbf{0} \rangle$$

The arithmetic language

- variables: x_0, x_1, ….
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\mathbf{0}$
- predicate symbols: $\doteq$ (which, again, is always uploaded, and so is deliberately not mentioned)
- function symbols: $\boldsymbol{S}$, $+$, $\times$
- auxiliary symbols:) , (.

$$\mathcal{L}_A = \langle \boldsymbol{S}, +, \times, \mathbf{0} \rangle$$

Terms of the arithmetic language

- Variables are terms (of $\mathcal{L}_A$).
- 0 is a term.
- If t is a term, so is St.
- If t_1 and t_2 are terms, so are $t_1 + t_2$ and $t_1 \times t_2$.
- Only expressions obtained by the above rules are terms.

Formulas of the arithmetic language

- $\bot$ is a formula (of $\mathcal{L}_A$).
- If t_1 and t_2 are terms, then $(t_1 \doteq t_2)$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Formulas of the arithmetic language

- $\bot$ is a formula (of $\mathcal{L}_A$).
- If t_1 and t_2 are terms, then $(t_1 \doteq t_2)$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Formulas of the arithmetic language

- $\bot$ is a formula (of $\mathcal{L}_A$).
- If t_1 and t_2 are terms, then $(t_1 \doteq t_2)$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Formulas of the arithmetic language

- $\perp$ is a formula (of $\mathcal{L}_A$).
- If t_1 and t_2 are terms, then $(t_1 \doteq t_2)$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Formulas of the arithmetic language

- $\bot$ is a formula (of $\mathcal{L}_A$).
- If t_1 and t_2 are terms, then $(t_1 \doteq t_2)$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Formulas of the arithmetic language

- $\perp$ is a formula (of $\mathcal{L}_A$).
- If t_1 and t_2 are terms, then $(t_1 \doteq t_2)$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Formulas are exactly those expressions obtained by the above rules.

Contents

Structure

A structure $\mathfrak{A}$ is an ordered sequence

$$\langle A, R_1, \dots, R_n, F_1, \dots, F_m, \{c_k | i \in K\} \rangle,$$

where

- A is a non-empty set.
- R_1, …, R_n are relations on A.
- F_1, …, F_m are functions on A.
- for all $k \in K$, c_k is an element of A, and K is an index set.

E.g.

- $\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$
- $\mathfrak{A} = \langle \mathbb{N}, ', +, \times, 0 \rangle$

Structure

A structure $\mathfrak{A}$ is an ordered sequence

$$\langle A, R_1, \ldots, R_n, F_1, \ldots, F_m, \{c_k | i \in K\} \rangle,$$

where

- A is a non-empty set.
- R_1, ..., R_n are relations on A.
- F_1, ..., F_m are functions on A.
- for all $k \in K$, c_k is an element of A, and K is an index set.

E.g.

- $\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$
- $\mathfrak{A} = \langle \mathbb{N}, ', +, \times, 0 \rangle$

Structure

A structure $\mathfrak{A}$ is an ordered sequence

$$\langle A, R_1, \ldots, R_n, F_1, \ldots, F_m, \{c_k | i \in K\} \rangle,$$

where

- A is a non-empty set.
- R_1, ..., R_n are relations on A.
- F_1, ..., F_m are functions on A.
- for all $k \in K$, c_k is an element of A, and K is an index set.

E.g.

- $\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$
- $\mathfrak{A} = \langle \mathbb{N}, ', +, \times, 0 \rangle$

Structure

A structure $\mathfrak{A}$ is an ordered sequence

$$\langle A, R_1, \ldots, R_n, F_1, \ldots, F_m, \{c_k | i \in K\} \rangle,$$

where

- A is a non-empty set.
- R_1, ..., R_n are relations on A.
- F_1, ..., F_m are functions on A.
- for all $k \in K$, c_k is an element of A, and K is an index set.

E.g.

- $\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$
- $\mathfrak{A} = \langle \mathbb{N}, ', +, \times, 0 \rangle$

Structure

A structure $\mathfrak{A}$ is an ordered sequence

$$\langle A, R_1, \ldots, R_n, F_1, \ldots, F_m, \{c_k | i \in K\} \rangle,$$

where

- A is a non-empty set.
- R_1, …, R_n are relations on A.
- F_1, …, F_m are functions on A.
- for all $k \in K$, c_k is an element of A, and K is an index set.

E.g.

- $\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$
- $\mathfrak{A} = \langle \mathbb{N}, ', +, \times, 0 \rangle$

Structure

A structure $\mathfrak{A}$ is an ordered sequence

$$\langle A, R_1, \ldots, R_n, F_1, \ldots, F_m, \{c_k | i \in K\} \rangle,$$

where

- A is a non-empty set.
- R_1, …, R_n are relations on A.
- F_1, …, F_m are functions on A.
- for all $k \in K$, c_k is an element of A, and K is an index set.

E.g.

- $\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$
- $\mathfrak{A} = \langle \mathbb{N}, ', +, \times, 0 \rangle$

Structure

A structure $\mathfrak{A}$ is an ordered sequence

$$\langle A, R_1, \ldots, R_n, F_1, \ldots, F_m, \{c_k | i \in K\} \rangle,$$

where

- A is a non-empty set.
- R_1, ..., R_n are relations on A.
- F_1, ..., F_m are functions on A.
- for all $k \in K$, c_k is an element of A, and K is an index set.

E.g.

- $\mathfrak{A} = \langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$
- $\mathfrak{A} = \langle \mathbb{N}, ', +, \times, 0 \rangle$

Similarity type

The similarity type of a structure $\mathfrak{A}$

$$\langle A, R_1, \ldots, R_n, F_1, \ldots, F_m, \{c_i | i \in I\} \rangle$$

is a sequence

$$\langle r_1, \ldots, r_n; a_1, \ldots, a_m; \kappa \rangle$$

where $R_i \subseteq A^{r_i}$, $F_j : A^{a_j} \to A$, the size of $\{c_k | k \in K\}$ is κ.

E.g.

Structure	Similarity type
$\langle A, R_1, R_2, \text{Socrates}, \text{Plato} \rangle$	$\langle 1, 2; -; 2 \rangle$
$\langle \mathbb{N}, ', +, \times, 0 \rangle$	$\langle -; 1, 2, 2; 1 \rangle$
$\langle G, R \rangle$	$\langle 1; -; - \rangle$

Similarity type

The similarity type of a structure $\mathfrak{A}$

$$\langle A, R_1, \ldots, R_n, F_1, \ldots, F_m, \{c_i | i \in I\}\rangle$$

is a sequence

$$\langle r_1, \ldots, r_n; a_1, \ldots, a_m; \kappa\rangle$$

where $R_i \subseteq A^{r_i}$, $F_j : A^{a_j} \to A$, the size of $\{c_k | k \in K\}$ is κ.

E.g.

Structure	Similarity type
$\langle A, R_1, R_2,$ Socrates, Plato$\rangle$	$\langle 1, 2; -; 2\rangle$
$\langle \mathbb{N}, ', +, \times, 0\rangle$	$\langle -; 1, 2, 2; 1\rangle$
$\langle G, R\rangle$	$\langle 1; -; -\rangle$

About identity relation

Stipulate:

- All structures possess an identity (equality) relation, unless otherwise claimed.
- We always omit the identity relation in a structure.
- Accordingly, we do not mention the arity of the identity relation in the similarity type.

The language of a similarity type

We can use a first-order language to describe **all** of the structures with a same similarity type $\langle r_1, \ldots, r_n; a_1, \ldots, a_m; \kappa \rangle$. We give this language by showing its alphabet:

- variables: x_0, x_1, ….
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\overline{c_k}$ for all $k \in K$
- predicate symbols: P_1, …, P_n, ($\doteq$)
- function symbols: f_1, …, f_m
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P_1, \ldots, P_n, f_1, \ldots, f_m, \{\overline{c_k} | k \in K\} \rangle$$

The language of a similarity type

We can use a first-order language to describe **all** of the structures with a same similarity type $\langle r_1, \ldots, r_n; a_1, \ldots, a_m; \kappa \rangle$. We give this language by showing its alphabet:

- variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\overline{c_k}$ for all $k \in K$
- predicate symbols: P_1, ..., P_n, ($\doteq$)
- function symbols: f_1, ..., f_m
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P_1, \ldots, P_n, f_1, \ldots, f_m, \{\overline{c_k} | k \in K\} \rangle$$

The language of a similarity type

We can use a first-order language to describe **all** of the structures with a same similarity type $\langle r_1, \ldots, r_n; a_1, \ldots, a_m; \kappa \rangle$. We give this language by showing its alphabet:

- variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\overline{c_k}$ for all $k \in K$
- predicate symbols: P_1, ..., P_n, ($\doteq$)
- function symbols: f_1, ..., f_m
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P_1, \ldots, P_n, f_1, \ldots, f_m, \{\overline{c_k} | k \in K\} \rangle$$

The language of a similarity type

We can use a first-order language to describe **all** of the structures with a same similarity type $\langle r_1, \ldots, r_n; a_1, \ldots, a_m; \kappa \rangle$. We give this language by showing its alphabet:

- variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\overline{c_k}$ for all $k \in K$
- predicate symbols: P_1, ..., P_n, ($\doteq$)
- function symbols: f_1, ..., f_m
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P_1, \ldots, P_n, f_1, \ldots, f_m, \{\overline{c_k} | k \in K\} \rangle$$

The language of a similarity type

We can use a first-order language to describe **all** of the structures with a same similarity type $\langle r_1, \ldots, r_n; a_1, \ldots, a_m; \kappa\rangle$. We give this language by showing its alphabet:

- variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\overline{c_k}$ for all $k \in K$
- predicate symbols: P_1, ..., P_n, ($\doteq$)
- function symbols: f_1, ..., f_m
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P_1, \ldots, P_n, f_1, \ldots, f_m, \{\overline{c_k} | k \in K\}\rangle$$

The language of a similarity type

We can use a first-order language to describe **all** of the structures with a same similarity type $\langle r_1, \ldots, r_n; a_1, \ldots, a_m; \kappa \rangle$. We give this language by showing its alphabet:

- variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\overline{c_k}$ for all $k \in K$
- predicate symbols: P_1, ..., P_n, ($\doteq$)
- function symbols: f_1, ..., f_m
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P_1, \ldots, P_n, f_1, \ldots, f_m, \{\overline{c_k} | k \in K\} \rangle$$

The language of a similarity type

We can use a first-order language to describe **all** of the structures with a same similarity type $\langle r_1, \ldots, r_n; a_1, \ldots, a_m; \kappa \rangle$. We give this language by showing its alphabet:

- variables: x_0, x_1, ….
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\overline{c_k}$ for all $k \in K$
- predicate symbols: P_1, …, P_n, ($\doteq$)
- function symbols: f_1, …, f_m
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P_1, \ldots, P_n, f_1, \ldots, f_m, \{\overline{c_k} | k \in K\} \rangle$$

The language of a similarity type

We can use a first-order language to describe **all** of the structures with a same similarity type $\langle r_1, \ldots, r_n; a_1, \ldots, a_m; \kappa \rangle$. We give this language by showing its alphabet:

- variables: x_0, x_1, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$
- quantifiers: $\forall$, $\exists$
- constant symbols: $\overline{c_k}$ for all $k \in K$
- predicate symbols: P_1, ..., P_n, ($\doteq$)
- function symbols: f_1, ..., f_m
- auxiliary symbols:) , (.

$$\mathcal{L} = \langle P_1, \ldots, P_n, f_1, \ldots, f_m, \{\overline{c_k} | k \in K\} \rangle$$

Terms of $\mathcal{L}$

- Variables and constant symbols are terms (of $\mathcal{L}$).
- For any $1 \leq i \leq m$, if t_1, ...t_{a_i} are terms, so is $f_i(t_1, \ldots, t_{a_i})$.
- Only the expressions obtained by the above rules are the terms.

Terms of $\mathcal{L}$

- Variables and constant symbols are terms (of $\mathcal{L}$).
- For any $1 \leq i \leq m$, if t_1, ...t_{a_i} are terms, so is $f_i(t_1, \ldots, t_{a_i})$.
- Only the expressions obtained by the above rules are the terms.

Terms of $\mathcal{L}$

- Variables and constant symbols are terms (of $\mathcal{L}$).
- For any $1 \leq i \leq m$, if t_1, ...t_{a_i} are terms, so is $f_i(t_1, \ldots, t_{a_i})$.
- Only the expressions obtained by the above rules are the terms.

Formulas of $\mathcal{L}$

- $\bot$ is a formula (of $\mathcal{L}$).
- For any terms t_1 and t_2, $t_1 \doteq t_2$ is a formula.
- For any $1 \leq i \leq n$, if t_1, ..., t_{r_i} are terms, then $P_i(t_1, \dots, t_{r_i})$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Only expressions obtained by the above rules are formulas.

Formulas of $\mathcal{L}$

- $\bot$ is a formula (of $\mathcal{L}$).
- For any terms t_1 and t_2, $t_1 \doteq t_2$ is a formula.
- For any $1 \leq i \leq n$, if t_1, ..., t_{r_i} are terms, then $P_i(t_1, \ldots, t_{r_i})$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Only expressions obtained by the above rules are formulas.

Formulas of $\mathcal{L}$

- $\bot$ is a formula (of $\mathcal{L}$).
- For any terms t_1 and t_2, $t_1 \doteq t_2$ is a formula.
- For any $1 \leq i \leq n$, if t_1, ..., t_{r_i} are terms, then $P_i(t_1, \ldots, t_{r_i})$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Only expressions obtained by the above rules are formulas.

Formulas of $\mathcal{L}$

- $\bot$ is a formula (of $\mathcal{L}$).
- For any terms t_1 and t_2, $t_1 \doteq t_2$ is a formula.
- For any $1 \leq i \leq n$, if t_1, ..., t_{r_i} are terms, then $P_i(t_1, \ldots, t_{r_i})$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Only expressions obtained by the above rules are formulas.

Formulas of $\mathcal{L}$

- $\perp$ is a formula (of $\mathcal{L}$).
- For any terms t_1 and t_2, $t_1 \doteq t_2$ is a formula.
- For any $1 \leq i \leq n$, if t_1, ..., t_{r_i} are terms, then $P_i(t_1, \ldots, t_{r_i})$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Only expressions obtained by the above rules are formulas.

Formulas of $\mathcal{L}$

- $\bot$ is a formula (of $\mathcal{L}$).
- For any terms t_1 and t_2, $t_1 \doteq t_2$ is a formula.
- For any $1 \leq i \leq n$, if t_1, ..., t_{r_i} are terms, then $P_i(t_1, \ldots, t_{r_i})$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Only expressions obtained by the above rules are formulas.

Formulas of $\mathcal{L}$

- $\bot$ is a formula (of $\mathcal{L}$).
- For any terms t_1 and t_2, $t_1 \doteq t_2$ is a formula.
- For any $1 \leq i \leq n$, if t_1, ..., t_{r_i} are terms, then $P_i(t_1, \ldots, t_{r_i})$ is a formula.
- If φ is a formula, so is $(\neg\varphi)$.
- If φ and ψ are formulas, so is $(\varphi \star \psi)$, where $\star$ is $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$.
- If φ is a formula, then for any variable x, $((\forall x)\varphi)$ and $((\exists x)\varphi)$ are also formulas.
- Only expressions obtained by the above rules are formulas.

Other notions

- free occurrence of a variable
- free variable in a formula
- sentence
- substitution $s[t/x]$
- substitution $\varphi[t/x]$
- t is free for x in φ

Thanks for your attention!

Q & A