Formale Semantik
05. Prädikatenlogik

Roland Schäfer

Institut für Germanistische Sprachwissenschaft
Friedrich-Schiller-Universität Jena

Achtung: Folien in Überarbeitung. Englische Teile sind noch von 2007!
Stets aktuelle Fassungen: `https://github.com/rsling/VL-Semantik`

Inhalt

Why predicate calculus?

Weak compositionality in SL

- properties/relations vs. individuals

Weak compositionality in SL

- properties/relations vs. individuals
- *Martin is an expert on inversion and Martin is a good climber.*

Weak compositionality in SL

- properties/relations vs. individuals
- *Martin is an <u>e</u>xpert on inversion and Martin is a good <u>c</u>limber.*
- ...becomes $E \wedge C$

Weak compositionality in SL

- properties/relations vs. individuals
- *Martin is an expert on inversion and Martin is a good climber.*
- ...becomes $E \wedge C$
- compositionality resticted to level of connected propositional atoms

Some desirable deductions

- important generalizations about all and some individuals (which have property P)

Some desirable deductions

- important generalizations about all and some individuals (which have property P)
- *‘all P $\rightarrow$ some P’*

Some desirable deductions

- important generalizations about all and some individuals (which have property P)
- *‘all P $\rightarrow$ some P’*
- *‘Martin P $\rightarrow$ some P’*

The construction of PC

Atoms of PC

- individual variables: $x, y, z, x_1, x_2 \ldots$

Atoms of PC

- individual variables: $x, y, z, x_1, x_2 \ldots$
- individual constants: $a, b, c, \ldots$

Atoms of PC

- individual variables: $x, y, z, x_1, x_2 \ldots$
- individual constants: $a, b, c, \ldots$
- variables and constants: terms

Atoms of PC

- individual variables: $x, y, z, x_1, x_2 \ldots$
- individual constants: $a, b, c, \ldots$
- variables and constants: terms
- predicate symbols (taking individual symbols or tuples of them): $A, B, C, \ldots$

Atoms of PC

- individual variables: $x, y, z, x_1, x_2 \ldots$
- individual constants: $a, b, c, \ldots$
- variables and constants: terms
- predicate symbols (taking individual symbols or tuples of them): $A, B, C, \ldots$
- quantifiers: existential $\exists$ (or $\bigvee$) and universal $\forall$ (or $\bigwedge$)

Atoms of PC

- individual variables: $x, y, z, x_1, x_2 \ldots$
- individual constants: $a, b, c, \ldots$
- variables and constants: terms
- predicate symbols (taking individual symbols or tuples of them): $A, B, C, \ldots$
- quantifiers: existential $\exists$ (or $\bigvee$) and universal $\forall$ (or $\bigwedge$)
- plus the connectives of SL

- for an n-ary predicate P and terms $t_1 \dots t_n$, $P(t_1 \dots t_n)$ or $Pt_1 \dots t_n$ is a wff.

Some syntax

- for an n-ary predicate P and terms $t_1 \ldots t_n$,
  $P(t_1 \ldots t_n)$ or $Pt_1 \ldots t_n$ is a wff.
- possible prefix, function (bracket) and infix notation:
  Pxy, $P(x, y)$, xPy

Some syntax

- for an n-ary predicate P and terms $t_1 \dots t_n$, $P(t_1 \dots t_n)$ or $Pt_1 \dots t_n$ is a wff.
- possible prefix, function (bracket) and infix notation: Pxy, $P(x, y)$, xPy
- syntax for connectives from SL

Some syntax

- for an n-ary predicate P and terms $t_1 \dots t_n$, $P(t_1 \dots t_n)$ or $Pt_1 \dots t_n$ is a wff.
- possible prefix, function (bracket) and infix notation: Pxy, $P(x, y)$, xPy
- syntax for connectives from SL
- for any wff ϕ and any variable x, $(\exists x)\phi$ and $(\forall x)\phi$ are wff's

Semantic for individual constants

- denote individuals

Semantic for individual constants

- denote individuals
- a model $\mathcal{M}$ contains a set of individuals D

Semantic for individual constants

- denote individuals
- a model $\mathcal{M}$ contains a set of individuals D
- the valuation function V (or F): from constants to individuals in D

Semantic for individual constants

- denote individuals
- a model $\mathcal{M}$ contains a set of individuals D
- the valuation function V (or F): from constants to individuals in D
- for some $\mathcal{M}_1$: $D = \{Martin, Kilroy, Scully\}$

Semantic for individual constants

- denote individuals
- a model $\mathcal{M}$ contains a set of individuals D
- the valuation function V (or F): from constants to individuals in D
- for some $\mathcal{M}_1$: $D = \{Martin, Kilroy, Scully\}$
- $V_{\mathcal{M}_1}(m) = Martin$

Semantic for individual constants

- denote individuals
- a model $\mathcal{M}$ contains a set of individuals D
- the valuation function V (or F): from constants to individuals in D
- for some $\mathcal{M}_1$: $D = \{Martin, Kilroy, Scully\}$
- $V_{\mathcal{M}_1}(m) = Martin$
- $V_{\mathcal{M}_1}(k) = Kilroy$, $V_{\mathcal{M}_1}(s) = Scully$

Semantics for predicate symbols

- denote relations (sets of n-tuples)

Semantics for predicate symbols

- denote relations (sets of n-tuples)
- $[\![P]\!]^{\mathcal{M}_1} = \{Martin, Kilroy\}$ or $V_{\mathcal{M}_1}(P) = \{Martin, Kilroy\}$

Semantics for predicate symbols

- denote relations (sets of n-tuples)
- $[\![P]\!]^{\mathcal{M}_1} = \{Martin, Kilroy\}$ or $V_{\mathcal{M}_1}(P) = \{Martin, Kilroy\}$
- $V_{\mathcal{M}_1}(Q) = \{\langle Martin, Kilroy\rangle, \langle Martin, Scully\rangle, \langle Kilroy, Kilroy\rangle, \langle Scully, Scully\rangle\}$

Semantics for predicate symbols

- denote relations (sets of n-tuples)
- $[\![P]\!]^{\mathcal{M}_1} = \{Martin, Kilroy\}$ or $V_{\mathcal{M}_1}(P) = \{Martin, Kilroy\}$
- $V_{\mathcal{M}_1}(Q) = \{\langle Martin, Kilroy\rangle, \langle Martin, Scully\rangle, \langle Kilroy, Kilroy\rangle, \langle Scully, Scully\rangle\}$
- s.t. $[\![P(m)]\!]^{\mathcal{M}_1} = [\![P]\!]^{\mathcal{M}_1}([\![m)]\!]^{\mathcal{M}_1}) = 1$ iff $[\![m]\!]^{\mathcal{M}_1} \in [\![P]\!]^{\mathcal{M}_1}$

Semantics for connectives and quantifiers

- connectives: ‘apply to’ formulas (semantically truth-valued), semantics as in SL

Semantics for connectives and quantifiers

- connectives: ‘apply to’ formulas (semantically truth-valued), semantics as in SL
- $(\forall x)\phi$ = 1 iff ϕ is true for every $d \in D$
  assigned to every occurence of x in ϕ

Semantics for connectives and quantifiers

- connectives: ‘apply to’ formulas (semantically truth-valued), semantics as in SL
- $(\forall x)\phi$ = 1 iff ϕ is true for every $d \in D$
  assigned to every occurence of x in ϕ
- $(\exists x)\phi$ = 1 iff ϕ is true for at least one $d \in D$
  assigned to every occurence of x in ϕ

Semantics for connectives and quantifiers

- connectives: ‘apply to’ formulas (semantically truth-valued), semantics as in SL
- $(\forall x)\phi = 1$ iff ϕ is true for every $d \in D$
  assigned to every occurence of x in ϕ
- $(\exists x)\phi = 1$ iff ϕ is true for at least one $d \in D$
  assigned to every occurence of x in ϕ
- algorithmic instruction to check wff’s containing Q’s

Semantics for connectives and quantifiers

- connectives: ‘apply to’ formulas (semantically truth-valued), semantics as in SL
- $(\forall x)\phi = 1$ iff ϕ is true for every $d \in D$ assigned to every occurence of x in ϕ
- $(\exists x)\phi = 1$ iff ϕ is true for at least one $d \in D$ assigned to every occurence of x in ϕ
- algorithmic instruction to check wff’s containing Q’s
- check outside-in (unambiguous scoping)

Dependencies

- universal quantifiers can be swapped:
  $(\forall x)(\forall y)\phi \Leftrightarrow (\forall y)(\forall x)\phi$

Dependencies

- universal quantifiers can be swapped:
  $(\forall x)(\forall y)\phi \Leftrightarrow (\forall y)(\forall x)\phi$
- same for existential quantifiers:
  $(\exists x)(\exists y)\phi \Leftrightarrow (\exists y)(\exists x)\phi$

Dependencies

- universal quantifiers can be swapped:
  $(\forall x)(\forall y)\phi \Leftrightarrow (\forall y)(\forall x)\phi$
- same for existential quantifiers:
  $(\exists x)(\exists y)\phi \Leftrightarrow (\exists y)(\exists x)\phi$
- whereas: $(\exists x)(\forall y)\phi \Rightarrow (\forall y)(\exists x)\phi$

Dependencies

- universal quantifiers can be swapped: $(\forall x)(\forall y)\phi \Leftrightarrow (\forall y)(\forall x)\phi$
- same for existential quantifiers: $(\exists x)(\exists y)\phi \Leftrightarrow (\exists y)(\exists x)\phi$
- whereas: $(\exists x)(\forall y)\phi \Rightarrow (\forall y)(\exists x)\phi$
- example in $\mathcal{M}_1$:
  - $[\![\underline{(\forall x)\underline{(\exists y)Qxy}}]\!]^{\mathcal{M}_1} = 1$

Dependencies

- universal quantifiers can be swapped:
  $(\forall x)(\forall y)\phi \Leftrightarrow (\forall y)(\forall x)\phi$
- same for existential quantifiers:
  $(\exists x)(\exists y)\phi \Leftrightarrow (\exists y)(\exists x)\phi$
- whereas: $(\exists x)(\forall y)\phi \Rightarrow (\forall y)(\exists x)\phi$
- example in $\mathcal{M}_1$:
  - $[\![\underline{(\forall x)\underline{(\exists y)Qxy}}]\!]^{\mathcal{M}_1} = 1$
  - but: $[\![\underline{(\exists y)\underline{(\forall x)Qxy}}]\!]^{\mathcal{M}_1} = 0$

Dependencies

- universal quantifiers can be swapped:
  $(\forall x)(\forall y)\phi \Leftrightarrow (\forall y)(\forall x)\phi$
- same for existential quantifiers:
  $(\exists x)(\exists y)\phi \Leftrightarrow (\exists y)(\exists x)\phi$
- whereas: $(\exists x)(\forall y)\phi \Rightarrow (\forall y)(\exists x)\phi$
- example in $\mathcal{M}_1$:
  - $[\![\underline{(\forall x)\underline{(\exists y)Qxy}}]\!]^{\mathcal{M}_1} = 1$
  - but: $[\![\underline{(\exists y)\underline{(\forall x)Qxy}}]\!]^{\mathcal{M}_1} = 0$
  - direct consequence of algorithmic definition

Dependencies

- universal quantifiers can be swapped:
  $(\forall x)(\forall y)\phi \Leftrightarrow (\forall y)(\forall x)\phi$
- same for existential quantifiers:
  $(\exists x)(\exists y)\phi \Leftrightarrow (\exists y)(\exists x)\phi$
- whereas: $(\exists x)(\forall y)\phi \Rightarrow (\forall y)(\exists x)\phi$
- example in $\mathcal{M}_1$:
  - $[\![(\forall x)\underline{\underline{(\exists y)Qxy}}]\!]^{\mathcal{M}_1} = 1$
  - but: $[\![(\exists y)\underline{\underline{(\forall x)Qxy}}]\!]^{\mathcal{M}_1} = 0$
  - direct consequence of algorithmic definition
  - if $\exists\forall$ is true, $\forall\exists$ follows

Hints on quantifiers

- domain of quantifiers: D (universe of discourse)

Hints on quantifiers

- domain of quantifiers: D (universe of discourse)
- $\forall x$ checks for truth of some predication for all individuals

Hints on quantifiers

- domain of quantifiers: D (universe of discourse)
- $\forall x$ checks for truth of some predication for all individuals
- $\exists x(Px \land \neg Px)$ is a contradiction

Hints on quantifiers

- domain of quantifiers: D (universe of discourse)
- $\forall x$ checks for truth of some predication for all individuals
- $\exists x(Px \land \neg Px)$ is a contradiction
- $\forall x(Wx \land \neg Wx)$ is a contradiciton,
  $\forall x$ ‘checks’ for an empty set by def.

Hints on quantifiers

- domain of quantifiers: D (universe of discourse)
- $\forall x$ checks for truth of some predication for all individuals
- $\exists x(Px \wedge \neg Px)$ is a contradiction
- $\forall x(Wx \wedge \neg Wx)$ is a contradiciton,
  $\forall x$ ‘checks’ for an empty set by def.
- standard form of NL quantification:
  $\forall x(Wx \rightarrow Bx)$ ‘All women are beautiful.’

Hints on quantifiers

- domain of quantifiers: D (universe of discourse)
- $\forall x$ checks for truth of some predication for all individuals
- $\exists x(Px \wedge \neg Px)$ is a contradiction
- $\forall x(Wx \wedge \neg Wx)$ is a contradiciton,
  $\forall x$ 'checks' for an empty set by def.
- standard form of NL quantification:
  $\forall x(Wx \rightarrow Bx)$ 'All women are beautiful.'
- standard form of NL existential quantification:
  $\exists x(Wx \wedge Bx)$ 'Some woman is beautiful.'

Functor/quantifier practice

- by def., functors take formulas, not terms:

Functor/quantifier practice

- by def., functors take formulas, not terms:
  - $\neg Wm$ ‘Mary doesn’t weep.’

Functor/quantifier practice

- by def., functors take formulas, not terms:
  - $\neg Wm$ ‘Mary doesn’t weep.’
  - $(\exists x)(Gx \land Wx)$ ‘Some girl weeps.’

Functor/quantifier practice

- by def., functors take formulas, not terms:
  - $\neg Wm$ 'Mary doesn't weep.'
  - $(\exists x)(Gx \wedge Wx)$ 'Some girl weeps.'
  - $^{*}W\neg x$

Functor/quantifier practice

- by def., functors take formulas, not terms:
  - $\neg Wm$ 'Mary doesn't weep.'
  - $(\exists x)(Gx \wedge Wx)$ 'Some girl weeps.'
  - $^{*}W\neg x$
  - $^{*}(\exists\neg x)(Gx)$

Functor/quantifier practice

- by def., functors take formulas, not terms:
  - $\neg Wm$ ‘Mary doesn’t weep.’
  - $(\exists x)(Gx \wedge Wx)$ ‘Some girl weeps.’
  - $^{*}W\neg x$
  - $^{*}(\exists\neg x)(Gx)$
- quantifiers take variables, not constants:

Functor/quantifier practice

- by def., functors take formulas, not terms:
  - $\neg Wm$ 'Mary doesn't weep.'
  - $(\exists x)(Gx \wedge Wx)$ 'Some girl weeps.'
  - $^{*}W\neg x$
  - $^{*}(\exists\neg x)(Gx)$
- quantifiers take variables, not constants:
  - $(\forall x)(Ox \rightarrow Wx)$ 'All ozelots are wildcats.'

Functor/quantifier practice

- by def., functors take formulas, not terms:
  - $\neg Wm$ 'Mary doesn't weep.'
  - $(\exists x)(Gx \wedge Wx)$ 'Some girl weeps.'
  - $^{*}W\neg x$
  - $^{*}(\exists\neg x)(Gx)$
- quantifiers take variables, not constants:
  - $(\forall x)(Ox \rightarrow Wx)$ 'All ozelots are wildcats.'
  - $^{*}(\forall o)(Wo)$

Functor/quantifier practice

- by def., functors take formulas, not terms:
  - $\neg Wm$ ‘Mary doesn’t weep.’
  - $(\exists x)(Gx \wedge Wx)$ ‘Some girl weeps.’
  - $^{*}W\neg x$
  - $^{*}(\exists\neg x)(Gx)$
- quantifiers take variables, not constants:
  - $(\forall x)(Ox \rightarrow Wx)$ ‘All ozelots are wildcats.’
  - $^{*}(\forall o)(Wo)$
- $\neg$ negates the wff, not the q:
  $^{*}(\neg\forall x)Px$ but $\neg(\forall x)Px$

Scope

- quantifiers bind variables

Scope

- quantifiers bind variables
- free variables (constants) are unbound

Scope

- quantifiers bind variables
- free variables (constants) are unbound
- no double binding $*(\forall x \exists x)Px$

Scope

- quantifiers bind variables
- free variables (constants) are unbound
- no double binding $*(\forall x \exists x)Px$
- Q scope: only the first wff to its right:

Scope

- quantifiers bind variables
- free variables (constants) are unbound
- no double binding $*(\forall x \exists x)Px$
- Q scope: only the first wff to its right:
  - $\underline{(\forall x)Px} \vee Qx$

Scope

- quantifiers bind variables
- free variables (constants) are unbound
- no double binding $^*(\forall x \exists x)Px$
- Q scope: only the first wff to its right:
  - $\underline{(\forall x)Px} \vee Qx$
  - $\underline{(\forall x)(Px \vee Qx)} = \underline{(\forall x)Px} \vee \underline{(\forall x)Qx}$

Scope

- quantifiers bind variables
- free variables (constants) are unbound
- no double binding $^*(\forall x \exists x)Px$
- Q scope: only the first wff to its right:
  - $\underline{(\forall x)Px} \vee Qx$
  - $\underline{(\forall x)(Px \vee Qx)} = \underline{(\forall x)Px} \vee \underline{(\forall x)Qx}$
  - $\underline{(\exists x)Px} \rightarrow \underline{(\forall y)(Qy \wedge Ry)}$

Scope

- quantifiers bind variables
- free variables (constants) are unbound
- no double binding $^*(\forall x \exists x)Px$
- Q scope: only the first wff to its right:
  - $\underline{(\forall x)Px} \vee Qx$
  - $\underline{(\forall x)(Px \vee Qx)} = \underline{(\forall x)Px} \vee \underline{(\forall x)Qx}$
  - $\underline{(\exists x)Px} \rightarrow \underline{(\forall y)(Qy \wedge Ry)}$
  - $\underline{(\exists x)Px} \wedge Qx$ (second x is a unbound)

Scope

- quantifiers bind variables
- free variables (constants) are unbound
- no double binding $^*(\forall x \exists x)Px$
- Q scope: only the first wff to its right:
  - $\underline{(\forall x)Px} \lor Qx$
  - $\underline{(\forall x)(Px \lor Qx)} = \underline{(\forall x)Px} \lor \underline{(\forall x)Qx}$
  - $\underline{(\exists x)Px} \rightarrow \underline{(\forall y)(Qy \land Ry)}$
  - $\underline{(\exists x)Px} \land Qx$ (second x is a unbound)
- no double-naming

Laws of PC

Universal $\vee$ and $\wedge$

- $\exists$ and $\forall$ ‘or’ and ‘and’ over the universe of discourse (hence: $\bigvee$ and $\bigwedge$)

Universal $\vee$ and $\wedge$

- $\exists$ and $\forall$ 'or' and 'and' over the universe of discourse (hence: $\bigvee$ and $\bigwedge$)
- $(\forall x)Px \Leftrightarrow Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n$ for all x_n assigned to $d_n \in D$

Universal $\vee$ and $\wedge$

- $\exists$ and $\forall$ 'or' and 'and' over the universe of discourse (hence: $\bigvee$ and $\bigwedge$)
- $(\forall x)Px \Leftrightarrow Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n$ for all x_n assigned to $d_n \in D$
- $(\exists x)Px \Leftrightarrow Px_1 \vee Px_2 \vee \ldots \vee Px_n$ for all x_n assigned to $d_n \in D$

Universal $\vee$ and $\wedge$

- $\exists$ and $\forall$ 'or' and 'and' over the universe of discourse (hence: $\bigvee$ and $\bigwedge$)
- $(\forall x)Px \Leftrightarrow Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n$ for all x_n assigned to $d_n \in D$
- $(\exists x)Px \Leftrightarrow Px_1 \vee Px_2 \vee \ldots \vee Px_n$ for all x_n assigned to $d_n \in D$
- hence: $\neg(\forall x)Px \Leftrightarrow \neg(Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n)$

Universal $\vee$ and $\wedge$

- $\exists$ and $\forall$ 'or' and 'and' over the universe of discourse (hence: $\bigvee$ and $\bigwedge$)
- $(\forall x)Px \Leftrightarrow Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n$ for all x_n assigned to $d_n \in D$
- $(\exists x)Px \Leftrightarrow Px_1 \vee Px_2 \vee \ldots \vee Px_n$ for all x_n assigned to $d_n \in D$
- hence: $\neg(\forall x)Px \Leftrightarrow \neg(Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n)$
- with DeM: $\overline{Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n}$

Universal $\vee$ and $\wedge$

- $\exists$ and $\forall$ 'or' and 'and' over the universe of discourse (hence: $\bigvee$ and $\bigwedge$)
- $(\forall x)Px \Leftrightarrow Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n$ for all x_n assigned to $d_n \in D$
- $(\exists x)Px \Leftrightarrow Px_1 \vee Px_2 \vee \ldots \vee Px_n$ for all x_n assigned to $d_n \in D$
- hence: $\neg(\forall x)Px \Leftrightarrow \neg(Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n)$
- with DeM: $\overline{Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n}$
- $\Leftrightarrow \overline{Px_1} \vee \overline{Px_2} \vee \ldots \vee \overline{Px_n}$

Universal $\vee$ and $\wedge$

- $\exists$ and $\forall$ ‘or’ and ‘and’ over the universe of discourse (hence: $\bigvee$ and $\bigwedge$)
- $(\forall x)Px \Leftrightarrow Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n$ for all x_n assigned to $d_n \in D$
- $(\exists x)Px \Leftrightarrow Px_1 \vee Px_2 \vee \ldots \vee Px_n$ for all x_n assigned to $d_n \in D$
- hence: $\neg(\forall x)Px \Leftrightarrow \neg(Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n)$
- with DeM: $\overline{Px_1 \wedge Px_2 \wedge \ldots \wedge Px_n}$
- $\Leftrightarrow \overline{Px_1} \vee \overline{Px_2} \vee \ldots \vee \overline{Px_n}$
- $\Leftrightarrow (\exists x)\neg Px$

Quantifier negation (QN)

- $\neg(\forall x)Px \Leftrightarrow (\exists x)\neg Px$

Quantifier negation (QN)

- $\neg(\forall x)Px \Leftrightarrow (\exists x)\neg Px$
- $\neg(\exists x)Px \Leftrightarrow (\forall x)\neg Px$

Quantifier negation (QN)

- $\neg(\forall x)Px \Leftrightarrow (\exists x)\neg Px$
- $\neg(\exists x)Px \Leftrightarrow (\forall x)\neg Px$
- $\neg(\forall x)\neg Px \Leftrightarrow (\exists x)Px$

Quantifier negation (QN)

- $\neg(\forall x)Px \Leftrightarrow (\exists x)\neg Px$
- $\neg(\exists x)Px \Leftrightarrow (\forall x)\neg Px$
- $\neg(\forall x)\neg Px \Leftrightarrow (\exists x)Px$
- $\neg(\exists x)\neg Px \Leftrightarrow (\forall x)Px$

The distribution laws

- the conjunction of universally quantified formulas:
  $(\forall x)(Px \land Qx) \Leftrightarrow (\forall x)Px \land (\forall x)Qx$

The distribution laws

- the conjunction of universally quantified formulas:
  $(\forall x)(Px \wedge Qx) \Leftrightarrow (\forall x)Px \wedge (\forall x)Qx$
- the disjunction of existentially quantified formulas:
  $(\exists x)(Px \vee Qx) \Leftrightarrow (\exists x)Px \vee (\exists x)Qx$

The distribution laws

- the conjunction of universally quantified formulas: $\underline{(\forall x)(Px \wedge Qx)} \Leftrightarrow \underline{(\forall x)Px} \wedge \underline{(\forall x)Qx}$
- the disjunction of existentially quantified formulas: $\underline{(\exists x)(Px \vee Qx)} \Leftrightarrow \underline{(\exists x)Px} \vee \underline{(\exists x)Qx}$
- not v.v.: $(\forall x)Px \vee (\forall x)Qx \Rightarrow (\forall x)(Px \vee Qx)$

The distribution laws

- the conjunction of universally quantified formulas: $\underline{(\forall x)(Px \wedge Qx)} \Leftrightarrow \underline{(\forall x)Px} \wedge \underline{(\forall x)Qx}$
- the disjunction of existentially quantified formulas: $\underline{(\exists x)(Px \vee Qx)} \Leftrightarrow \underline{(\exists x)Px} \vee \underline{(\exists x)Qx}$
- not v.v.: $(\forall x)Px \vee (\forall x)Qx \Rightarrow (\forall x)(Px \vee Qx)$
- why?

Quantifier movement (QM)

- desirable format: prefix + matrix

Quantifier movement (QM)

- desirable format: prefix + matrix
- Movement Laws for antecedents of conditionals:
  $(\exists x)Px \rightarrow \phi \Leftrightarrow (\forall x)(Px \rightarrow \phi)$
  $(\forall x)Px \rightarrow \phi \Leftrightarrow (\exists x)(Px \rightarrow \phi)$

Quantifier movement (QM)

- desirable format: prefix + matrix
- Movement Laws for antecedents of conditionals:
  $(\exists x)Px \rightarrow \phi \Leftrightarrow (\forall x)(Px \rightarrow \phi)$
  $(\forall x)Px \rightarrow \phi \Leftrightarrow (\exists x)(Px \rightarrow \phi)$
- Movement Laws for Q's in disjunction, conjunction, and the consequent of conditionals: Just move them to the prefix!

Quantifier movement (QM)

- desirable format: prefix + matrix
- Movement Laws for antecedents of conditionals:
  $(\exists x)Px \rightarrow \phi \Leftrightarrow (\forall x)(Px \rightarrow \phi)$
  $(\forall x)Px \rightarrow \phi \Leftrightarrow (\exists x)(Px \rightarrow \phi)$
- Movement Laws for Q's in disjunction, conjunction, and the consequent of conditionals: Just move them to the prefix!
- condition: x must not be free in ϕ.

Quantifier movement (QM)

- desirable format: prefix + matrix
- Movement Laws for antecedents of conditionals:
  $(\exists x)Px \rightarrow \phi \Leftrightarrow (\forall x)(Px \rightarrow \phi)$
  $(\forall x)Px \rightarrow \phi \Leftrightarrow (\exists x)(Px \rightarrow \phi)$
- Movement Laws for Q's in disjunction, conjunction, and the consequent of conditionals: Just move them to the prefix!
- condition: x must not be free in ϕ.
- i.e.: Watch your variables!

Let’s formalize:

- *Paul Kalkbrenner is a musician and signed on bpitchcontrol.*

Let’s formalize:

- *Paul Kalkbrenner is a musician and signed on bpitchcontrol.*
- *Herr S. installed RedHat and not every Linux distribution is easy to install.*

Let's formalize:

- *Paul Kalkbrenner is a musician and signed on bpitchcontrol.*
- *Herr S. installed RedHat and not every Linux distribution is easy to install.*
- *All talkmasters are human and Harald Schmidt is a talkmaster.*

Let's formalize:

- *Paul Kalkbrenner is a musician and signed on bpitchcontrol.*
- *Herr S. installed RedHat and not every Linux distribution is easy to install.*
- *All talkmasters are human and Harald Schmidt is a talkmaster.*
- *Some talkmasters are not musicians.*

Let's formalize:

- *Paul Kalkbrenner is a musician and signed on bpitchcontrol.*
- *Herr S. installed RedHat and not every Linux distribution is easy to install.*
- *All talkmasters are human and Harald Schmidt is a talkmaster.*
- *Some talkmasters are not musicians.*
- *Heiko Laux owns Kanzleramt records and does not like any Gigolo artist.*

Let's formalize:

- *Paul Kalkbrenner is a musician and signed on bpitchcontrol.*
- *Herr S. installed RedHat and not every Linux distribution is easy to install.*
- *All talkmasters are human and Harald Schmidt is a talkmaster.*
- *Some talkmasters are not musicians.*
- *Heiko Laux owns Kanzleramt records and does not like any Gigolo artist.*
- *Some humans are neither talkmasters nor do they own Kanzleramt records.*

Natural deduction in PC

Universal instantiation ($-\forall$) and generalization ($+\forall$)

- $(\forall x)Px \rightarrow Pa$

Universal instantiation ($-\forall$) and generalization ($+\forall$)

- $(\forall x)Px \rightarrow Pa$
- always applies

Universal instantiation ($-\forall$) and generalization ($+\forall$)

- $(\forall x)Px \rightarrow Pa$
- always applies

Universal instantiation ($-\forall$) and generalization ($+\forall$)

- $(\forall x)Px \rightarrow Pa$
- always applies
- can use any variable/constant

Universal instantiation ($-\forall$) and generalization ($+\forall$)

- $(\forall x)Px \rightarrow Pa$
- always applies
- can use any variable/constant
- $Pa \rightarrow (\forall x)Px$

Universal instantiation ($-\forall$) and generalization ($+\forall$)

- $(\forall x)Px \rightarrow Pa$
- always applies
- can use any variable/constant
- $Pa \rightarrow (\forall x)Px$
- iff *Pa* was instantiated by $-\forall$

Existential generalization ($+\exists$) and instantiation ($-\exists$)

- $Pa \rightarrow (\exists x)Px$ for any individual constant a

Existential generalization ($+\exists$) and instantiation ($-\exists$)

- $Pa \rightarrow (\exists x)Px$ for any individual constant a
- always applies

Existential generalization ($+\exists$) and instantiation ($-\exists$)

- $Pa \rightarrow (\exists x)Px$ for any individual constant a
- always applies
- $(\exists x)Px \rightarrow Pa$ for some indiv. const.

Existential generalization ($+\exists$) and instantiation ($-\exists$)

- $Pa \rightarrow (\exists x)Px$ for any individual constant a
- always applies
- $(\exists x)Px \rightarrow Pa$ for some indiv. const.
- always applies (there is a minimal individual for $\exists x$)

Existential generalization ($+\exists$) and instantiation ($-\exists$)

- $Pa \rightarrow (\exists x)Px$ for any individual constant a
- always applies
- $(\exists x)Px \rightarrow Pa$ for some indiv. const.
- always applies (there is a minimal individual for $\exists x$)
- for some $(\exists x)Px$ and $(\exists x)Qx$ the minimal individual might be different

Existential generalization ($+\exists$) and instantiation ($-\exists$)

- $Pa \rightarrow (\exists x)Px$ for any individual constant a
- always applies
- $(\exists x)Px \rightarrow Pa$ for some indiv. const.
- always applies (there is a minimal individual for $\exists x$)
- for some $(\exists x)Px$ and $(\exists x)Qx$ the minimal individual might be different
- hence: When you apply EI, always use fresh constants!

One sample task

- (1) Herr Keydana drives a Golf. (2) Anything that drives a golf is human or a complex program simulating an artificial neural net. (3) There are no programs s.a.a.n.n. which are complex enough to drive a Golf.

One sample task

- (1) Herr Keydana drives a Golf. (2) Anything that drives a golf is human or a complex program simulating an artificial neural net. (3) There are no programs s.a.a.n.n. which are complex enough to drive a Golf.
- Formalize and prove: At least one human exists.

One sample task

- (1) Herr Keydana drives a Golf. (2) Anything that drives a golf is human or a complex program simulating an artificial neural net. (3) There are no programs s.a.a.n.n. which are complex enough to drive a Golf.
- Formalize and prove: At least one human exists.
- (1) Dk

One sample task

- (1) Herr Keydana drives a Golf. (2) Anything that drives a golf is human or a complex program simulating an artificial neural net. (3) There are no programs s.a.a.n.n. which are complex enough to drive a Golf.
- Formalize and prove: At least one human exists.
- (1) Dk
- (2) $(\forall x)(Dx \rightarrow Hx \vee Px)$

One sample task

- (1) Herr <u>K</u>eydana <u>d</u>rives a Golf. (2) Anything that drives a golf is <u>h</u>uman or a complex <u>p</u>rogram simulating an artificial neural net. (3) There are no programs s.a.a.n.n. which are complex enough to drive a Golf.
- Formalize and prove: At least one human exists.
- (1) Dk
- (2) $(\forall x)(Dx \rightarrow Hx \vee Px)$
- (3) $\neg(\exists x)(Px \wedge Dx)$

One sample task

- (1) Herr Keydana drives a Golf. (2) Anything that drives a golf is human or a complex program simulating an artificial neural net. (3) There are no programs s.a.a.n.n. which are complex enough to drive a Golf.
- Formalize and prove: At least one human exists.
- (1) Dk
- (2) $(\forall x)(Dx \rightarrow Hx \vee Px)$
- (3) $\neg(\exists x)(Px \wedge Dx)$
- $(\exists x)Hx$

The proof

(1)	Dk	
(2)	$(\forall x)(Dx \rightarrow Hx \vee Px)$	
(3)	$\neg(\exists x)(Px \wedge Dx)$	
(4)	$(\forall x)\neg(Px \wedge Dx)$	3,QN
(5)	$(\forall x)(\neg Px \vee \neg Dx)$	4,DeM
(6)	$(\forall x)(Dx \rightarrow \neg Px)$	5,Comm,Impl
(7)	$Dk \rightarrow \neg Pk$	6,$-\forall$(1)
(8)	$\neg Pk$	1,7,MP
(9)	$Dk \rightarrow Hk \vee Pk$	2,$-\forall$(1)
(10)	$Hk \vee Pk$	1,9,MP
(11)	Hk	8,10,DS
$\therefore$	$(\exists x)Hx$	10,$+\exists$

Literatur I

Autor

Kontakt

Prof. Dr. Roland Schäfer
Institut für Germanistische Sprachwissenschaft
Friedrich-Schiller-Universität Jena
Fürstengraben 30
07743 Jena

https://rolandschaefer.net
roland.schaefer@uni-jena.de

Lizenz