8. Intro. to Quantifier Logic

8. Intro. to Quantifier Logic

a. The goals of QL

Limits of symbolization in SL

- Consider the argument:

Limits of symbolization in SL

- Consider the argument:

  Greta is a hero.

Limits of symbolization in SL

- Consider the argument:

  Greta is a hero.

  ∴ There is a hero.

Limits of symbolization in SL

- Consider the argument:

  Greta is a hero.

  $\therefore$ There is a hero.

- It's clearly valid: in any case in which Greta is a hero, someone (or something, at least) is a hero, so there must be a hero.

Limits of symbolization in SL

- Consider the argument:

  Greta is a hero.

  ∴ There is a hero.

- It’s clearly valid: in any case in which Greta is a hero, someone (or something, at least) is a hero, so there must be a hero.

- But its symbolization in SL is invalid in SL:

Limits of symbolization in SL

- Consider the argument:

  Greta is a hero.

  $\therefore$ There is a hero.

- It's clearly valid: in any case in which Greta is a hero, someone (or something, at least) is a hero, so there must be a hero.

- But its symbolization in SL is invalid in SL:

  G

Limits of symbolization in SL

- Consider the argument:

  Greta is a hero.

  $\therefore$ There is a hero.
- It's clearly valid: in any case in which Greta is a hero, someone (or something, at least) is a hero, so there must be a hero.
- But its symbolization in SL is invalid in SL:

  G

  $\therefore H$

The problem

- Symbolization in SL allows us to break down sentences containing “and,” “or,” “if-then” and determine validity in virtue of these **connectives**.

The problem

- Symbolization in SL allows us to break down sentences containing “and,” “or,” “if-then” and determine validity in virtue of these **connectives**.
- Anything that can’t be further broken down must be symbolized by sentence letters.

The problem

- Symbolization in SL allows us to break down sentences containing “and,” “or,” “if-then” and determine validity in virtue of these **connectives**.
- Anything that can’t be further broken down must be symbolized by sentence letters.
- That includes basic sentences like “Greta is a hero,” but also:

The problem

- Symbolization in SL allows us to break down sentences containing “and,” “or,” “if-then” and determine validity in virtue of these **connectives**.
- Anything that can’t be further broken down must be symbolized by sentence letters.
- That includes basic sentences like “Greta is a hero,” but also:
  - Everyone is a hero.

The problem

- Symbolization in SL allows us to break down sentences containing "and," "or," "if-then" and determine validity in virtue of these **connectives**.
- Anything that can't be further broken down must be symbolized by sentence letters.
- That includes basic sentences like "Greta is a hero," but also:
  - Everyone is a hero.
  - No one is a hero.

The problem

- Symbolization in SL allows us to break down sentences containing "and," "or," "if-then" and determine validity in virtue of these **connectives**.
- Anything that can't be further broken down must be symbolized by sentence letters.
- That includes basic sentences like "Greta is a hero," but also:
  - Everyone is a hero.
  - No one is a hero.
  - All heroes wear capes.

The goals of Quantifier Logic (QL)

- Finer-grained symbolization

The goals of Quantifier Logic (QL)

- Finer–grained symbolization
- Augments SL (all of SL and *more*!)

The goals of Quantifier Logic (QL)

- Finer-grained symbolization
- Augments SL (all of SL and *more*!)
- Allows for precise semantics (like truth tables for SL)

The goals of Quantifier Logic (QL)

- Finer-grained symbolization
- Augments SL (all of SL and *more*!)
- Allows for precise semantics (like truth tables for SL)
- Works with natural deduction (add new rules!)

The goals of Quantifier Logic (QL)

- Finer-grained symbolization
- Augments SL (all of SL and *more*!)
- Allows for precise semantics (like truth tables for SL)
- Works with natural deduction (add new rules!)
- Be simple & expressive (only a few new symbols!)

The goals of QL

- Consider the valid argument:

The goals of QL

- Consider the valid argument:

  Greta is a hero.

The goals of QL

- Consider the valid argument:

  Greta is a hero.

  Greta does not wear a cape.

The goals of QL

- Consider the valid argument:

  Greta is a hero.

  Greta does not wear a cape.

  $\therefore$ Not all heroes wear capes.

The goals of QL

- Consider the valid argument:

  Greta is a hero.

  Greta does not wear a cape.

  $\therefore$ Not all heroes wear capes.
- We'll need to connect the occurrences of the name "Greta" in the premises

The goals of QL

- Consider the valid argument:

  Greta is a hero.

  Greta does not wear a cape.

  ∴ Not all heroes wear capes.
- We'll need to connect the occurrences of the name "Greta" in the premises
- We'll need to connect "hero" in the premise and conclusion

The goals of QL

- Consider the valid argument:

  Greta is a hero.

  Greta does not wear a cape.

  $\therefore$ Not all heroes wear capes.
- We'll need to connect the occurrences of the name "Greta" in the premises
- We'll need to connect "hero" in the premise and conclusion
- We want to retain using the symbol '$\sim$' for "not"

The goals of QL

- Consider the valid argument:

  Greta is a hero.

  Greta does not wear a cape.

  $\therefore$ Not all heroes wear capes.
- We'll need to connect the occurrences of the name "Greta" in the premises
- We'll need to connect "hero" in the premise and conclusion
- We want to retain using the symbol '$\sim$' for "not"
- Ultimately, we'll want our argument–symbolization to have a proof

8. Intro. to Quantifier Logic

b. Beginning symbolization in QL

First steps: names (a.k.a. ‘constants’)

- Purpose of a **proper name**: to pick out a single, specific thing.

First steps: names (a.k.a. ‘constants’)

- Purpose of a **proper name**: to pick out a single, specific thing.
- (Contrast with common nouns like “hero” or “rock” which pick out collections of things)

First steps: names (a.k.a. ‘constants’)

- Purpose of a **proper name**: to pick out a single, specific thing.
- (Contrast with common nouns like “hero” or “rock” which pick out collections of things)
- For simplicity, we’ll only consider names that pick out a **specific object** (often within a hypothetical case we’re considering)

First steps: names (a.k.a. ‘constants’)

- Purpose of a **proper name**: to pick out a single, specific thing.
- (Contrast with common nouns like “hero” or “rock” which pick out collections of things)
- For simplicity, we’ll only consider names that pick out a **specific object** (often within a hypothetical case we’re considering)
- Later on, we’ll be able to deal with other expressions that play a similar role to names, e.g., “the president of the USA”

First steps: names (a.k.a. 'constants')

- Purpose of a **proper name**: to pick out a single, specific thing.
- (Contrast with common nouns like "hero" or "rock" which pick out collections of things)
- For simplicity, we'll only consider names that pick out a **specific object** (often within a hypothetical case we're considering)
- Later on, we'll be able to deal with other expressions that play a similar role to names, e.g., "the president of the USA"
- In QL, names are symbolized by lowercase letters a–v (allowing natural number sub-scripts, e.g. m_1, t_{2022})

First steps: predicates (including properties and relations)

- Remove a name from a sentence. What's left over is a **predicate**:

First steps: predicates (including properties and relations)

- Remove a name from a sentence. What's left over is a **predicate**:

  Greta is a hero

First steps: predicates (including properties and relations)

- Remove a name from a sentence. What's left over is a **predicate**:

  Greta is a hero

$\Rightarrow$ $\underline{\qquad}_x$ is a hero. (i.e. property of being a hero)

First steps: predicates (including properties and relations)

- Remove a name from a sentence. What's left over is a **predicate**:

  Greta is a hero

  $\Rightarrow$ $\_\_\_\_\__x$ is a hero. (i.e. property of being a hero)

  Greta admires Autumn

First steps: predicates (including properties and relations)

- Remove a name from a sentence. What's left over is a **predicate**:

  Greta is a hero

  $\Rightarrow$ $\_\_\_\_\__x$ is a hero. (i.e. property of being a hero)

  Greta admires Autumn

  $\Rightarrow$ $\_\_\_\_\__x$ admires $\_\_\_\_\__y$. (i.e. relation of x admiring y)

First steps: predicates (including properties and relations)

- Remove a name from a sentence. What's left over is a **predicate**:

  Greta is a hero

  $\Rightarrow$ $\underline{\qquad}_x$ is a hero. (i.e. property of being a hero)

  Greta admires Autumn

  $\Rightarrow$ $\underline{\qquad}_x$ admires $\underline{\qquad}_y$. (i.e. relation of x admiring y)
- In QL, predicates are symbolized using uppercase letters *A*-*Z* plus a number of argument slots (marked with variables), e.g., Hx or Axy.

First steps: predicates (including properties and relations)

- Remove a name from a sentence. What's left over is a **predicate**:

  Greta is a hero

  $\Rightarrow$ $\_\_\_\_\__x$ is a hero. (i.e. property of being a hero)

  Greta admires Autumn

  $\Rightarrow$ $\_\_\_\_\__x$ admires $\_\_\_\_\__y$. (i.e. relation of x admiring y)
- In QL, predicates are symbolized using uppercase letters *A*-*Z* plus a number of argument slots (marked with variables), e.g., Hx or Axy.
- Argument slots correspond to blanks.

Symbolization keys

- **Names/constants**: lowercase letters for proper names

Symbolization keys

- **Names/constants**: lowercase letters for proper names
- **Predicates**: uppercase letters with variables marking blanks

Symbolization keys

- **Names/constants**: lowercase letters for proper names
- **Predicates**: uppercase letters with variables marking blanks

a: Autumn

Symbolization keys

- **Names/constants**: lowercase letters for proper names
- **Predicates**: uppercase letters with variables marking blanks

  a: Autumn

  g: Greta

Symbolization keys

- **Names/constants**: lowercase letters for proper names
- **Predicates**: uppercase letters with variables marking blanks

a: Autumn

g: Greta

Hx: $\underline{\qquad}_x$ is a hero

Symbolization keys

- **Names/constants**: lowercase letters for proper names
- **Predicates**: uppercase letters with variables marking blanks

a: Autumn

g: Greta

Hx: $\underline{\qquad}_x$ is a hero

Vx: $\underline{\qquad}_x$ is a villain

Symbolization keys

- **Names/constants**: lowercase letters for proper names
- **Predicates**: uppercase letters with variables marking blanks

$$\begin{aligned} a&: \text{ Autumn} \\ g&: \text{ Greta} \\ Hx&: \text{ \_\_\_\_\_}_x \text{ is a hero} \\ Vx&: \text{ \_\_\_\_\_}_x \text{ is a villain} \\ Ix&: \text{ \_\_\_\_\_}_x \text{ inspires} \end{aligned}$$

Symbolization keys

- **Names/constants**: lowercase letters for proper names
- **Predicates**: uppercase letters with variables marking blanks

$$
\begin{aligned}
a&: \text{ Autumn} \\
g&: \text{ Greta} \\
Hx&: \text{ \_\_\_\_\_}_x \text{ is a hero} \\
Vx&: \text{ \_\_\_\_\_}_x \text{ is a villain} \\
Ix&: \text{ \_\_\_\_\_}_x \text{ inspires} \\
Cx&: \text{ \_\_\_\_\_}_x \text{ wears a cape}
\end{aligned}
$$

Symbolization keys

- **Names/constants**: lowercase letters for proper names
- **Predicates**: uppercase letters with variables marking blanks

$$
\begin{aligned}
a&: \text{ Autumn}\\
g&: \text{ Greta}\\
Hx&: \underline{\qquad}_x \text{ is a hero}\\
Vx&: \underline{\qquad}_x \text{ is a villain}\\
Ix&: \underline{\qquad}_x \text{ inspires}\\
Cx&: \underline{\qquad}_x \text{ wears a cape}\\
Wxy&: \underline{\qquad}_x \text{ welcomed } \underline{\qquad}_y
\end{aligned}
$$

Symbolization keys

- **Names/constants**: lowercase letters for proper names
- **Predicates**: uppercase letters with variables marking blanks

$$\begin{aligned}
a&: \text{ Autumn}\\
g&: \text{ Greta}\\
Hx&: \underline{\qquad}_x \text{ is a hero}\\
Vx&: \underline{\qquad}_x \text{ is a villain}\\
Ix&: \underline{\qquad}_x \text{ inspires}\\
Cx&: \underline{\qquad}_x \text{ wears a cape}\\
Wxy&: \underline{\qquad}_x \text{ welcomed } \underline{\qquad}_y\\
Axy&: \underline{\qquad}_x \text{ admires } \underline{\qquad}_y
\end{aligned}$$

Symbolization keys

- **Names/constants**: lowercase letters for proper names
- **Predicates**: uppercase letters with variables marking blanks

$$
\begin{aligned}
a&: \text{Autumn}\\
g&: \text{Greta}\\
Hx&: \underline{\qquad}_x \text{ is a hero}\\
Vx&: \underline{\qquad}_x \text{ is a villain}\\
Ix&: \underline{\qquad}_x \text{ inspires}\\
Cx&: \underline{\qquad}_x \text{ wears a cape}\\
Wxy&: \underline{\qquad}_x \text{ welcomed } \underline{\qquad}_y\\
Axy&: \underline{\qquad}_x \text{ admires } \underline{\qquad}_y\\
Yxy&: \underline{\qquad}_x \text{ is younger than } \underline{\qquad}_y
\end{aligned}
$$

Symbolization keys

- **Names/constants**: lowercase letters for proper names
- **Predicates**: uppercase letters with variables marking blanks

a: Autumn
g: Greta
Hx: $\_\_\_\_\__x$ is a hero
Vx: $\_\_\_\_\__x$ is a villain
Ix: $\_\_\_\_\__x$ inspires
Cx: $\_\_\_\_\__x$ wears a cape
Wxy: $\_\_\_\_\__x$ welcomed $\_\_\_\_\__y$
Axy: $\_\_\_\_\__x$ admires $\_\_\_\_\__y$
Yxy: $\_\_\_\_\__x$ is younger than $\_\_\_\_\__y$

- **Domain**: the non-predicate objects we're talking about in a context—also called grandly the '**Universe of Discourse**' (UD)

e.g., people alive in 2022

Symbolization of Sentences without Quantifiers

- Basic sentences: predicates with names replacing variables.

Symbolization of Sentences without Quantifiers

- Basic sentences: predicates with names replacing variables.
  - Greta is a hero: Hg

Symbolization of Sentences without Quantifiers

- Basic sentences: predicates with names replacing variables.
  - Greta is a hero: Hg
  - Greta admires Autum: Aga

Symbolization of Sentences without Quantifiers

- Basic sentences: predicates with names replacing variables.
  - Greta is a hero: Hg
  - Greta admires Autum: Aga
- Combinations using connectives:

Symbolization of Sentences without Quantifiers

- Basic sentences: predicates with names replacing variables.
  - Greta is a hero: Hg
  - Greta admires Autum: Aga
- Combinations using connectives:
  - Greta and Autumn are heroes: $Hg \mathbin{\&} Ha$

Symbolization of Sentences without Quantifiers

- Basic sentences: predicates with names replacing variables.
  - Greta is a hero: Hg
  - Greta admires Autum: Aga
- Combinations using connectives:
  - Greta and Autumn are heroes: $Hg \& Ha$
  - If Autumn admires Greta, then Autumn is a hero: $Aag \supset Ha$

Symbolization of Pronouns

- Replacing pronouns by antecedents:

Symbolization of Pronouns

- Replacing pronouns by antecedents:
  - If Autumn is a hero, Greta admires **her**: $Ha \supset Aga$

Symbolization of Pronouns

- Replacing pronouns by antecedents:
  - If Autumn is a hero, Greta admires **her**: $Ha \supset Aga$
  - Greta doesn't admire **herself**: ${\sim}Agg$ (but she should!)

Symbolization of Pronouns

- Replacing pronouns by antecedents:
  - If Autumn is a hero, Greta admires **her**: $Ha \supset Aga$
  - Greta doesn't admire **herself**: $\sim Agg$ (but she should!)
  - Greta and Autumn welcomed **each other**: $Wga \,\&\, Wag$

Symbolization of Pronouns

- Replacing pronouns by antecedents:
  - If Autumn is a hero, Greta admires **her**: $Ha \supset Aga$
  - Greta doesn't admire **herself**: $\sim Agg$ (but she should!)
  - Greta and Autumn welcomed **each other**: $Wga \,\&\, Wag$

- Modifiers:

Symbolization of Pronouns

- Replacing pronouns by antecedents:
  - If Autumn is a hero, Greta admires **her**: $Ha \supset Aga$
  - Greta doesn't admire **herself**: $\sim Agg$ (but she should!)
  - Greta and Autumn welcomed **each other**: $Wga \,\&\, Wag$
- Modifiers:
  - Autumn is an **inspiring hero**:
    i.e. Autumn inspires, and she is a hero: $Ia \,\&\, Ha$

Symbolization of Pronouns

- Replacing pronouns by antecedents:
  - If Autumn is a hero, Greta admires **her**: $Ha \supset Aga$
  - Greta doesn't admire **herself**: $\sim Agg$ (but she should!)
  - Greta and Autumn welcomed **each other**: $Wga \,\&\, Wag$

- Modifiers:
  - Autumn is an **inspiring hero**:
    i.e. Autumn inspires, and she is a hero: $Ia \,\&\, Ha$
  - Greta is a **hero who doesn't wear a cape**:
    Greta is a hero, and it's not the case that Greta wears a cape:
    $Hg \,\&\, \sim Cg$

Mind the Modifiers!

- ‘Greta is an international hero’:

Mind the Modifiers!

- ‘Greta is an international hero’:
    - Can’t be paraphrased as
      “Greta is international and a hero.”

Mind the Modifiers!

- ‘Greta is an international hero’:
  - Can’t be paraphrased as
    “Greta is international and a hero.”
  - So “_____ is an international hero” needs its own predicate

Mind the Modifiers!

- ‘Greta is an international hero’:
  - Can’t be paraphrased as
    “Greta is international and a hero.”
  - So “_____ is an international hero” needs its own predicate
- ‘The Piltdown Man is a fake fossil’

Mind the Modifiers!

- ‘Greta is an international hero’:
  - Can’t be paraphrased as
    “Greta is international and a hero.”

  - So “_____ is an international hero” needs its own predicate
- ‘The Piltdown Man is a fake fossil’
  - Can’t be paraphrased as
    “The Piltdown Man is fake and a fossil.”

Mind the Modifiers!

- ‘Greta is an international hero’:
  - Can’t be paraphrased as “Greta is international and a hero.”
  - So “_____ is an international hero” needs its own predicate
- ‘The Piltdown Man is a fake fossil’
  - Can’t be paraphrased as “The Piltdown Man is fake and a fossil.”
  - Since “fake” and other privative adjectives (“pretend,” “fictitious”) deny the property that they modify! (‘fake news’ isn’t news!)

Examples

- Autumn and Greta are inspiring heroes.

Examples

- Autumn and Greta are inspiring heroes.
  $(Ia \mathbin{\&} Ha) \mathbin{\&} (Ig \mathbin{\&} Hg)$

Examples

- Autumn and Greta are inspiring heroes.
  $(Ia \mathbin{\&} Ha) \mathbin{\&} (Ig \mathbin{\&} Hg)$
- Greta admires Autumn but not herself.

Examples

- Autumn and Greta are inspiring heroes.
  $(Ia \,\&\, Ha) \,\&\, (Ig \,\&\, Hg)$
- Greta admires Autumn but not herself.
  $Aga \,\&\, {\sim}Agg$

Examples

- Autumn and Greta are inspiring heroes.
  $(Ia \mathbin{\&} Ha) \mathbin{\&} (Ig \mathbin{\&} Hg)$
- Greta admires Autumn but not herself.
  $Aga \mathbin{\&} {\sim}Agg$
- Greta inspires only if Autumn does.

Examples

- Autumn and Greta are inspiring heroes.
  $(Ia \,\&\, Ha) \,\&\, (Ig \,\&\, Hg)$
- Greta admires Autumn but not herself.
  $Aga \,\&\, {\sim}Agg$
- Greta inspires only if Autumn does.
  $Ig \supset Ia$

Examples

- Autumn and Greta are inspiring heroes.
  $(Ia \,\&\, Ha) \,\&\, (Ig \,\&\, Hg)$
- Greta admires Autumn but not herself.
  $Aga \,\&\, {\sim}Agg$
- Greta inspires only if Autumn does.
  $Ig \supset Ia$
- Greta and Autumn welcomed each other.

Examples

- Autumn and Greta are inspiring heroes.
  $(Ia \,\&\, Ha) \,\&\, (Ig \,\&\, Hg)$
- Greta admires Autumn but not herself.
  $Aga \,\&\, {\sim}Agg$
- Greta inspires only if Autumn does.
  $Ig \supset Ia$
- Greta and Autumn welcomed each other.
  $Wga \,\&\, Wag$

Examples

- Autumn and Greta are inspiring heroes.
  $(Ia \,\&\, Ha) \,\&\, (Ig \,\&\, Hg)$
- Greta admires Autumn but not herself.
  $Aga \,\&\, \sim Agg$
- Greta inspires only if Autumn does.
  $Ig \supset Ia$
- Greta and Autumn welcomed each other.
  $Wga \,\&\, Wag$
- Greta is older than Autumn.

Examples

- Autumn and Greta are inspiring heroes.
  $(Ia \& Ha) \& (Ig \& Hg)$
- Greta admires Autumn but not herself.
  $Aga \& {\sim}Agg$
- Greta inspires only if Autumn does.
  $Ig \supset Ia$
- Greta and Autumn welcomed each other.
  $Wga \& Wag$
- Greta is older than Autumn.
  Yag (i.e. Autumn is *younger than* Greta)

Examples

- Autumn and Greta are inspiring heroes.
  $(Ia \,\&\, Ha) \,\&\, (Ig \,\&\, Hg)$
- Greta admires Autumn but not herself.
  $Aga \,\&\, {\sim}Agg$
- Greta inspires only if Autumn does.
  $Ig \supset Ia$
- Greta and Autumn welcomed each other.
  $Wga \,\&\, Wag$
- Greta is older than Autumn.
  Yag (i.e. Autumn is *younger than* Greta)
- One of Greta and Autumn welcomed the other.

Examples

- Autumn and Greta are inspiring heroes.
  $(Ia \,\&\, Ha) \,\&\, (Ig \,\&\, Hg)$
- Greta admires Autumn but not herself.
  $Aga \,\&\, {\sim}Agg$
- Greta inspires only if Autumn does.
  $Ig \supset Ia$
- Greta and Autumn welcomed each other.
  $Wga \,\&\, Wag$
- Greta is older than Autumn.
  Yag (i.e. Autumn is *younger than* Greta)
- One of Greta and Autumn welcomed the other.

  At least one: $Wga \vee Wag$

  Exactly one: $(Wga \vee Wag) \,\&\, {\sim}(Wga \,\&\, Wag)$

8. Intro. to Quantifier Logic

c. The existential quantifier

Existential quantifier (something or other)

- In English: “something,” “someone,” “there is . . . ”

Existential quantifier (something or other)

- In English: “something,” “someone,” “there is . . . ”
- For instance:

Existential quantifier (something or other)

- In English: “something,” “someone,” “there is . . . ”
- For instance:
  - **Someone** wears a cape.

Existential quantifier (something or other)

- In English: “something,” “someone,” “there is . . . ”
- For instance:
  - **Someone** wears a cape.
  - **There is** a hero.

Existential quantifier (something or other)

- In English: "something," "someone," "there is . . . "
- For instance:
  - **Someone** wears a cape.
  - **There is** a hero.
  - **Something** inspires.

Existential quantifier (something or other)

- In English: “something,” “someone,” “there is . . . ”
- For instance:
  - **Someone** wears a cape.
  - **There is** a hero.
  - **Something** inspires.
- Note: often goes where names and pronouns are placed

Existential quantifier (something or other)

- In English: “something,” “someone,” “there is . . . ”
- For instance:
  - **Someone** wears a cape.
  - **There is** a hero.
  - **Something** inspires.

- Note: often goes where names and pronouns are placed
- But works differently from names (“something” doesn’t pick out a unique, specific object).

How (not) to symbolize “something”

- Idea(?): introduce a special term ‘*sg*’ for ‘a something’?

How (not) to symbolize “something”

- Idea(?): introduce a special term ‘sg’ for ‘a something’?
- Problem: now we can’t distinguish between

  as both would be symbolized by ‘$H(sg) \,\&\, C(sg)$’.

How (not) to symbolize "something"

- Idea(?): introduce a special term 'sg' for 'a something'?
- Problem: now we can't distinguish between
  - Someone is a hero and wears a cape.

  as both would be symbolized by '$H(sg) \& C(sg)$'.

How (not) to symbolize “something”

- Idea(?): introduce a special term ‘*sg*’ for ‘a something’?
- Problem: now we can’t distinguish between
  - Someone is a hero and wears a cape.
  - Someone is a hero and someone wears a cape.

  as both would be symbolized by ‘$H(sg) \& C(sg)$’.

How (not) to symbolize "something"

- Idea(?): introduce a special term 'sg' for 'a something'?
- Problem: now we can't distinguish between
  - Someone is a hero and wears a cape.
  - Someone is a hero and someone wears a cape.

  as both would be symbolized by '$H(sg) \& C(sg)$'.
- Better idea: symbolize (complex) **properties** and introduce a notation for expressing that properties are **instantiated**

Expressing properties (and relations)

- One-place predicates **express** properties, e.g.,

Expressing properties (and relations)

- One-place predicates **express** properties, e.g.,
  - Hx expresses property “being a hero”

Expressing properties (and relations)

- One-place predicates **express** properties, e.g.,
  - Hx expresses property "being a hero"
  - Ix expresses "is inspiring" ('x' is a variable)

Expressing properties (and relations)

- One-place predicates **express** properties, e.g.,
  - Hx expresses property "being a hero"
  - Ix expresses "is inspiring" ('x' is a variable)

- Combinations of predicates (with connectives, names) can express **derived** properties, e.g.,

Expressing properties (and relations)

- One-place predicates **express** properties, e.g.,
  - Hx expresses property “being a hero”
  - Ix expresses “is inspiring” (‘x’ is a variable)

- Combinations of predicates (with connectives, names) can express **derived** properties, e.g.,
  - Axg expresses “admires Greta”

Expressing properties (and relations)

- One–place predicates **express** properties, e.g.,
  - Hx expresses property "being a hero"
  - Ix expresses "is inspiring" ('x' is a variable)
- Combinations of predicates (with connectives, names) can express **derived** properties, e.g.,
  - Axg expresses "admires Greta"
  - Wax expresses "is welcomed by Autumn"

Expressing properties (and relations)

- One-place predicates **express** properties, e.g.,
  - Hx expresses property “being a hero”
  - Ix expresses “is inspiring” (‘x’ is a variable)
- Combinations of predicates (with connectives, names) can express **derived** properties, e.g.,
  - Axg expresses “admires Greta”
  - Wax expresses “is welcomed by Autumn”
  - $Hx \,\&\, Cx$ expresses “is a hero who wears a cape”

Expressing properties (and relations)

- One-place predicates **express** properties, e.g.,
  - Hx expresses property "being a hero"
  - Ix expresses "is inspiring" ('x' is a variable)

- Combinations of predicates (with connectives, names) can express **derived** properties, e.g.,
  - Axg expresses "admires Greta"
  - Wax expresses "is welcomed by Autumn"
  - $Hx \& Cx$ expresses "is a hero who wears a cape"

- Note: all contain a **single** variable x

The existential quantifier $\exists$

- Symbol for "there is": $\exists$

The existential quantifier ∃

- Symbol for "there is": ∃
- Combine '∃' with an expression for a property (e.g., $(Hx \,\&\, Cx)$) to say "something (or someone) has that property"

The existential quantifier $\exists$

- Symbol for "there is": $\exists$
- Combine '$\exists$' with an expression for a property (e.g., $(Hx \,\&\, Cx)$) to say "something (or someone) has that property"
- Put the variable that serves as a marker for the gap after $\exists$. E.g.,

$$(\exists x)\,(Hx \,\&\, Cx)$$

  says "Someone is a hero and wears a cape"

The existential quantifier ∃

- Symbol for "there is": ∃
- Combine '∃' with an expression for a property (e.g., $(Hx \mathbin{\&} Cx)$) to say "something (or someone) has that property"
- Put the variable that serves as a marker for the gap after ∃. E.g.,

  $$(\exists x)\,(Hx \mathbin{\&} Cx)$$

  says "Someone is a hero and wears a cape"
- MUST always wrap a quantifier and the variable it 'binds' within **parentheses**: $(\exists y)$; *Carnap* will require this!!!

Quantifiers and variables

$$\text{Compare: } (\exists x)(Hx \,\&\, Cx) \text{ to}$$
$$(\exists x)\, Hx \,\&\, (\exists x)\, Cx$$

- In first case, *the same person* must be a hero and wear a cape.

Quantifiers and variables

$$\text{Compare: } (\exists x)\,(Hx \,\&\, Cx) \text{ to } (\exists x)\,Hx \,\&\, (\exists x)\,Cx$$

- In first case, *the same person* must be a hero and wear a cape.
- In second case, one person can be the hero and another (possibly different) person wears a cape.

Quantifiers and variables

Compare: $(\exists x)\,(Hx \mathbin{\&} Cx)$ to

$(\exists x)\,Hx \mathbin{\&} (\exists x)\,Cx$

- In first case, *the same person* must be a hero and wear a cape.
- In second case, one person can be the hero and another (possibly different) person wears a cape.
- Instances of '$(\exists x)$' separated by other connectives are independent, even if they bind the same variable x.
  e.g. there's no difference in meaning between the following:

$$(\exists x)\,Hx \mathbin{\&} (\exists x)\,Cx \quad \textbf{vs.}$$

$$(\exists x)\,Hx \mathbin{\&} (\exists y)\,Cy$$

Quantifiers and variables

Compare: $(\exists x)(Hx \mathbin{\&} Cx)$ to

$(\exists x)\, Hx \mathbin{\&} (\exists x)\, Cx$

- In first case, *the same person* must be a hero and wear a cape.
- In second case, one person can be the hero and another (possibly different) person wears a cape.
- Instances of '$(\exists x)$' separated by other connectives are independent, even if they bind the same variable x.
  e.g. there's no difference in meaning between the following:

  $$(\exists x)\, Hx \mathbin{\&} (\exists x)\, Cx \quad \textbf{vs.}$$

  $$(\exists x)\, Hx \mathbin{\&} (\exists y)\, Cy$$

- But we'll never write '$(\exists x)(\exists x)(Hx \mathbin{\&} Cx)$'

The domain (UD) and quantifiers

- Symbolization key gives a domain of objects being talked about.

The domain (UD) and quantifiers

- Symbolization key gives a domain of objects being talked about.
- Quantifier **ranges over** this ‘universe of discourse’ (UD).

The domain (UD) and quantifiers

- Symbolization key gives a domain of objects being talked about.
- Quantifier **ranges over** this 'universe of discourse' (UD).
- That means: $(\exists x) \ldots x \ldots$ is true iff some object **in the domain** has the property expressed by $\ldots x \ldots$.

The domain (UD) and quantifiers

- Symbolization key gives a domain of objects being talked about.
- Quantifier **ranges over** this ‘universe of discourse’ (UD).
- That means: $(\exists x)\ \ldots x \ldots$ is true iff some object **in the domain** has the property expressed by $\ldots x \ldots$.
- Domain makes a difference: Consider $(\exists x)\, Wxg$.

The domain (UD) and quantifiers

- Symbolization key gives a domain of objects being talked about.
- Quantifier **ranges over** this 'universe of discourse' (UD).
- That means: $(\exists x)\ \ldots x \ldots$ is true iff some object **in the domain** has the property expressed by $\ldots x \ldots$.
- Domain makes a difference: Consider $(\exists x)\ Wxg$.
  - True if someone welcomed Greta (say, Autumn did).

The domain (UD) and quantifiers

- Symbolization key gives a domain of objects being talked about.
- Quantifier **ranges over** this 'universe of discourse' (UD).
- That means: $(\exists x) \ldots x \ldots$ is true iff some object **in the domain** has the property expressed by $\ldots x \ldots$.
- Domain makes a difference: Consider $(\exists x)\, Wxg$.
  - True if someone welcomed Greta (say, Autumn did).
  - Now take the domain to include only Greta.

The domain (UD) and quantifiers

- Symbolization key gives a domain of objects being talked about.
- Quantifier **ranges over** this ‘universe of discourse’ (UD).
- That means: $(\exists x)\ \ldots x \ldots$ is true iff some object **in the domain** has the property expressed by $\ldots x \ldots$.
- Domain makes a difference: Consider $(\exists x)\, Wxg$.
  - True if someone welcomed Greta (say, Autumn did).
  - Now take the domain to include only Greta.
  - Relative to that domain, $(\exists x)\, Wxg$ is true iff Greta welcomed herself (e.g. to the left-over chocolate fondue!)

Quantifier restriction in English

- “something” and “someone” work grammatically like singular terms (they can go where names can also go).

Quantifier restriction in English

- “something” and “someone” work grammatically like singular terms (they can go where names can also go).
- “some” (on its own) does not: it is a **determiner** and needs a **complement**, e.g.,

Quantifier restriction in English

- “something” and “someone” work grammatically like singular terms (they can go where names can also go).
- “some” (on its own) does not: it is a **determiner** and needs a **complement**, e.g.,
  - a common noun (“some hero”), or

Quantifier restriction in English

- “something” and “someone” work grammatically like singular terms (they can go where names can also go).
- “some” (on its own) does not: it is a **determiner** and needs a **complement**, e.g.,
  - a common noun (“some hero”), or
  - a noun phrase (“some admirer of Greta”).

Quantifier restriction in English

- "something" and "someone" work grammatically like singular terms (they can go where names can also go).
- "some" (on its own) does not: it is a **determiner** and needs a **complement**, e.g.,
  - a common noun ("some hero"), or
  - a noun phrase ("some admirer of Greta").
- "some" + complement works grammatically like "someone", e.g., "**Some hero** wears a cape"

Quantifier restriction in English

- “something” and “someone” work grammatically like singular terms (they can go where names can also go).
- “some” (on its own) does not: it is a **determiner** and needs a **complement**, e.g.,
  - a common noun (“some hero”), or
  - a noun phrase (“some admirer of Greta”).
- “some” + complement works grammatically like “someone”, e.g., “**Some hero** wears a cape”
- General form: “Some F is G.”

Quantifier restriction in QL

- "Some F is G" **restricts** the "something" quantifier to Fs.

Quantifier restriction in QL

- "Some F is G" **restricts** the "something" quantifier to Fs.
- We could (and linguists often do) mark restrictions in the quantifier, e.g., $((\exists x) : Fx)Gx$

Quantifier restriction in QL

- "Some F is G" **restricts** the "something" quantifier to Fs.
- We could (and linguists often do) mark restrictions in the quantifier, e.g., $((\exists x): Fx)Gx$
- We won't because we can do without this additional notation

Quantifier restriction in QL

- "Some F is G" **restricts** the "something" quantifier to Fs.
- We could (and linguists often do) mark restrictions in the quantifier, e.g., $((\exists x) : Fx)Gx$
- We won't because we can do without this additional notation
- "Some F is G" is true iff there is something which is both F and also G, so:

Quantifier restriction in QL

- "Some F is G" **restricts** the "something" quantifier to Fs.
- We could (and linguists often do) mark restrictions in the quantifier, e.g., $((\exists x) : Fx)Gx$
- We won't because we can do without this additional notation
- "Some F is G" is true iff there is something which is both F and also G, so:
- "Some F is G" can be symbolized as

$$(\exists x)(Fx \,\&\, Gx)$$

Quantifier restriction in QL

- "Some F is G" **restricts** the "something" quantifier to Fs.
- We could (and linguists often do) mark restrictions in the quantifier, e.g., $((\exists x) : Fx)Gx$
- We won't because we can do without this additional notation
- "Some F is G" is true iff there is something which is both F and also G, so:
- "Some F is G" can be symbolized as

$$(\exists x)(Fx \,\&\, Gx)$$

- We'll also symbolize the plural form this way ("Some Fs are Gs").

Quantifier restriction in QL

- “Some F is G” **restricts** the “something” quantifier to Fs.
- We could (and linguists often do) mark restrictions in the quantifier, e.g., $((\exists x) : Fx)Gx$
- We won’t because we can do without this additional notation
- “Some F is G” is true iff there is something which is both F and also G, so:
- “Some F is G” can be symbolized as

$$(\exists x)(Fx \,\&\, Gx)$$

- We’ll also symbolize the plural form this way (“Some Fs are Gs”).
- And more generally (most) sentences of the form: “G(some F)” or “G(something that Fs)”.

Examples

- Some hero wears a cape.
  Some heroes wear capes.

Examples

- Some hero wears a cape.
  Some heroes wear capes.

  $(\exists x)(Hx \,\&\, Cx)$

Examples

- Some hero wears a cape.
  Some heroes wear capes.

  $(\exists x)(Hx \mathbin{\&} Cx)$

- Someone who wears a cape welcomed Greta.

Examples

- Some hero wears a cape.
  Some heroes wear capes.

  $(\exists x)(Hx \,\&\, Cx)$

- Someone who wears a cape welcomed Greta.

  $(\exists x)(Cx \,\&\, Wxg)$

Examples

- Some hero wears a cape.
  Some heroes wear capes.

  $(\exists x)(Hx \,\&\, Cx)$
- Someone who wears a cape welcomed Greta.

  $(\exists x)(Cx \,\&\, Wxg)$
- Greta admires some hero who wears a cape.

Examples

- Some hero wears a cape.
  Some heroes wear capes.

  $(\exists x)(Hx \,\&\, Cx)$

- Someone who wears a cape welcomed Greta.

  $(\exists x)(Cx \,\&\, Wxg)$

- Greta admires some hero who wears a cape.

  $(\exists x)((Hx \,\&\, Cx) \,\&\, Agx)$

Examples

- Some hero wears a cape.
  Some heroes wear capes.

  $(\exists x)(Hx \,\&\, Cx)$

- Someone who wears a cape welcomed Greta.

  $(\exists x)(Cx \,\&\, Wxg)$

- Greta admires some hero who wears a cape.

  $(\exists x)((Hx \,\&\, Cx) \,\&\, Agx)$

- Autumn welcomed someone who welcomed Greta.

Examples

- Some hero wears a cape.
  Some heroes wear capes.

  $(\exists x)(Hx \,\&\, Cx)$
- Someone who wears a cape welcomed Greta.

  $(\exists x)(Cx \,\&\, Wxg)$
- Greta admires some hero who wears a cape.

  $(\exists x)((Hx \,\&\, Cx) \,\&\, Agx)$
- Autumn welcomed someone who welcomed Greta.

  $(\exists x)(Wxg \,\&\, Wax)$

8. Intro. to Quantifier Logic

d. The universal quantifier

Universal quantifier

- "**Something** is F" is true iff **at least one** element of domain is F.

Universal quantifier

- "**Something** is F" is true iff **at least one** element of domain is F.
- "**Everything** is F" is true iff **every element** of the domain is F.

Universal quantifier

- "**Something** is F" is true iff **at least one** element of domain is F.
- "**Everything** is F" is true iff **every element** of the domain is F.
- In QL: $(\forall x)\, Fx$.

Universal quantifier

- "**Something** is F" is true iff **at least one** element of domain is F.
- "**Everything** is F" is true iff **every element** of the domain is F.
- In QL: $(\forall x)\ Fx$.
- E.g.:

Universal quantifier

- “**Something** is F” is true iff **at least one** element of domain is F.
- “**Everything** is F” is true iff **every element** of the domain is F.
- In QL: $(\forall x)\, Fx$.
- E.g.:
  - “Everyone wears a cape”: $(\forall x)\, Cx$

Universal quantifier

- "**Something** is F" is true iff **at least one** element of domain is F.
- "**Everything** is F" is true iff **every element** of the domain is F.
- In QL: $(\forall x)\, Fx$.
- E.g.:
  - "Everyone wears a cape": $(\forall x)\, Cx$
  - "Everyone welcomed Greta or Autumn": $(\forall x)(Wxg \vee Wxa)$

Universal determiners: all, every, any

- Determiners with universal meaning: **all, every, any**.

Universal determiners: all, every, any

- Determiners with universal meaning: **all, every, any**.
- Take complements (just like “some” does), e.g.,

Universal determiners: all, every, any

- Determiners with universal meaning: **all, every, any**.
- Take complements (just like “some” does), e.g.,
  - Every hero inspires.

Universal determiners: all, every, any

- Determiners with universal meaning: **all, every, any**.
- Take complements (just like "some" does), e.g.,
  - Every hero inspires.
  - All heroes inspire.

Universal determiners: all, every, any

- Determiners with universal meaning: **all, every, any**.
- Take complements (just like “some” does), e.g.,
  - Every hero inspires.
  - All heroes inspire.
  - Any hero inspires.

Universal determiners: all, every, any

- Determiners with universal meaning: **all, every, any**.
- Take complements (just like “some” does), e.g.,
  - Every hero inspires.
  - All heroes inspire.
  - Any hero inspires.
- These are true in the same cases (i.e. they are synonymous).

Universal determiners: all, every, any

- Determiners with universal meaning: **all, every, any**.
- Take complements (just like “some” does), e.g.,
  - Every hero inspires.
  - All heroes inspire.
  - Any hero inspires.
- These are true in the same cases (i.e. they are synonymous).
- “Every F is G” is true iff everything which is an F is G.

Universal determiners: all, every, any

- Determiners with universal meaning: **all, every, any**.
- Take complements (just like "some" does), e.g.,
  - Every hero inspires.
  - All heroes inspire.
  - Any hero inspires.
- These are true in the same cases (i.e. they are synonymous).
- "Every F is G" is true iff everything which is an F is G.
- Watch out for "any": not always universal.

Restricted ∀ in QL

- Suppose we can symbolize two properties ‘F’ and ‘G’.

Restricted $\forall$ in QL

- Suppose we can symbolize two properties 'F' and 'G'.
- How do we symbolize "Every F is G"?

Restricted ∀ in QL

- Suppose we can symbolize two properties 'F' and 'G'.
- How do we symbolize "Every F is G"?
- Initial Ideas (only one of which is correct):

Restricted $\forall$ in QL

- Suppose we can symbolize two properties ‘F’ and ‘G’.
- How do we symbolize “Every F is G”?
- Initial Ideas (only one of which is correct):
  - $(\forall x)(Fx \& Gx)$

Restricted $\forall$ in QL

- Suppose we can symbolize two properties 'F' and 'G'.
- How do we symbolize "Every F is G"?
- Initial Ideas (only one of which is correct):
  - $(\forall x)(Fx \,\&\, Gx)$
  - $(\forall x)(Fx \vee Gx)$

Restricted $\forall$ in QL

- Suppose we can symbolize two properties 'F' and 'G'.
- How do we symbolize "Every F is G"?
- Initial Ideas (only one of which is correct):
  - $(\forall x)(Fx \,\&\, Gx)$
  - $(\forall x)(Fx \vee Gx)$
  - $(\forall x)(Fx \supset Gx)$

Restricted $\forall$ in QL

- Suppose we can symbolize two properties 'F' and 'G'.
- How do we symbolize "Every F is G"?
- Initial Ideas (only one of which is correct):
  - $(\forall x)(Fx \,\&\, Gx)$
    If true, everything must be F.
    So can be false when "Every F is G" is true.
  - $(\forall x)(Fx \vee Gx)$
  - $(\forall x)(Fx \supset Gx)$

Restricted $\forall$ in QL

- Suppose we can symbolize two properties 'F' and 'G'.
- How do we symbolize "Every F is G"?
- Initial Ideas (only one of which is correct):
  - $(\forall x)(Fx \mathbin{\&} Gx)$
    If true, everything must be F.
    So can be false when "Every F is G" is true.
  - $(\forall x)(Fx \vee Gx)$
    True if everything is F (without being G).
    So can be true when "Every F is G" is false.
  - $(\forall x)(Fx \supset Gx)$

Restricted $\forall$ in QL

- Suppose we can symbolize two properties 'F' and 'G'.
- How do we symbolize "Every F is G"?
- Initial Ideas (only one of which is correct):
  - $(\forall x)(Fx \,\&\, Gx)$
    If true, everything must be F.
    So can be false when "Every F is G" is true.
  - $(\forall x)(Fx \vee Gx)$
    True if everything is F (without being G).
    So can be true when "Every F is G" is false.
  - $(\forall x)(Fx \supset Gx)$
    If x is F, x must also be G.
    (If x is not F, doesn't matter if it's G or not.)

Symbolizing “all Fs are Gs” (memorize this!)

Symbolize the following as

$$(\forall x)(Fx \supset Gx)$$

- All Fs are Gs.
- Every F is G.
- Any F is G.

Examples

- **Every hero** wears a cape.
  All heroes wear capes.

Examples

- **Every hero** wears a cape.
  All heroes wear capes.
  $(\forall x)(Hx \supset Cx)$

Examples

- **Every hero** wears a cape.
  All heroes wear capes.
  $(\forall x)(Hx \supset Cx)$
- **Every hero who wears a cape** welcomed Greta.

Examples

- **Every hero** wears a cape.
  All heroes wear capes.
  $(\forall x)(Hx \supset Cx)$
- **Every hero who wears a cape** welcomed Greta.
  $(\forall x)((Hx \,\&\, Cx) \supset Wxg)$

Examples

- **Every hero** wears a cape.
  All heroes wear capes.
  $(\forall x)(Hx \supset Cx)$
- **Every hero who wears a cape** welcomed Greta.
  $(\forall x)((Hx \,\&\, Cx) \supset Wxg)$
- Greta and Autumn admire **anyone who wears a cape**.

Examples

- **Every hero** wears a cape.
  All heroes wear capes.
  $(\forall x)(Hx \supset Cx)$
- **Every hero who wears a cape** welcomed Greta.
  $(\forall x)((Hx \,\&\, Cx) \supset Wxg)$
- Greta and Autumn admire **anyone who wears a cape**.
  $(\forall x)(Cx \supset (Agx \,\&\, Aax))$

Examples

- **Every hero** wears a cape.
  All heroes wear capes.
  $(\forall x)(Hx \supset Cx)$
- **Every hero who wears a cape** welcomed Greta.
  $(\forall x)((Hx \,\&\, Cx) \supset Wxg)$
- Greta and Autumn admire **anyone who wears a cape**.
  $(\forall x)(Cx \supset (Agx \,\&\, Aax))$
- Autumn welcomed **everyone who welcomed Greta**.

Examples

- **Every hero** wears a cape.
  All heroes wear capes.
  $(\forall x)(Hx \supset Cx)$
- **Every hero who wears a cape** welcomed Greta.
  $(\forall x)((Hx \,\&\, Cx) \supset Wxg)$
- Greta and Autumn admire **anyone who wears a cape**.
  $(\forall x)(Cx \supset (Agx \,\&\, Aax))$
- Autumn welcomed **everyone who welcomed Greta**.
  $(\forall x)(Wxg \supset Wax)$

Examples

- **Every hero** wears a cape.
  All heroes wear capes.
  $(\forall x)(Hx \supset Cx)$
- **Every hero who wears a cape** welcomed Greta.
  $(\forall x)((Hx \,\&\, Cx) \supset Wxg)$
- Greta and Autumn admire **anyone who wears a cape**.
  $(\forall x)(Cx \supset (Agx \,\&\, Aax))$
- Autumn welcomed **everyone who welcomed Greta**.
  $(\forall x)(Wxg \supset Wax)$
- All heroes and villains welcomed Greta (a tricky one!).

Examples

- **Every hero** wears a cape.
  All heroes wear capes.
  $(\forall x)(Hx \supset Cx)$
- **Every hero who wears a cape** welcomed Greta.
  $(\forall x)((Hx \mathbin{\&} Cx) \supset Wxg)$
- Greta and Autumn admire **anyone who wears a cape**.
  $(\forall x)(Cx \supset (Agx \mathbin{\&} Aax))$
- Autumn welcomed **everyone who welcomed Greta**.
  $(\forall x)(Wxg \supset Wax)$
- All heroes and villains welcomed Greta (a tricky one!).
  $(\forall x)((Hx \vee Vx) \supset Wxg)$

Examples

- **Every hero** wears a cape.
  All heroes wear capes.
  $(\forall x)(Hx \supset Cx)$
- **Every hero who wears a cape** welcomed Greta.
  $(\forall x)((Hx \,\&\, Cx) \supset Wxg)$
- Greta and Autumn admire **anyone who wears a cape**.
  $(\forall x)(Cx \supset (Agx \,\&\, Aax))$
- Autumn welcomed **everyone who welcomed Greta**.
  $(\forall x)(Wxg \supset Wax)$
- All heroes and villains welcomed Greta (a tricky one!).
  $(\forall x)((Hx \vee Vx) \supset Wxg)$
  equivalent to $(\forall x)((Hx \supset Wxg) \,\&\, (Vx \supset Wxg))$

8. Intro. to Quantifier Logic

e. ‘No’, ‘only’, ‘a’, ‘some’, and ‘any’

No F is G

- "**No Fs are Gs**" can be paraphrased as

No F is G

- “**No *F*s are *G*s**” can be paraphrased as
  - “**Every** F is **not**–G,” or as

No F is G

- “**No *F*s are *G*s**” can be paraphrased as
  - “**Every** F is **not**–G,” or as
  - “**Not: some** F is G.”

No F is G

- “**No Fs are Gs**” can be paraphrased as
  - “**Every** F is **not**–G,” or as
  - “**Not: some** F is G.”
- So symbolize it using:

No F is G

- "**No Fs are Gs**" can be paraphrased as
  - "**Every** F is **not**–G," or as
  - "**Not: some** F is G."
- So symbolize it using:
  - $(\forall x)(Fx \supset \sim Gx)$ or

No F is G

- "**No Fs are Gs**" can be paraphrased as
  - "**Every** F is **not**–G," or as
  - "**Not: some** F is G."
- So symbolize it using:
  - $(\forall x)(Fx \supset {\sim}Gx)$ or
  - ${\sim}(\exists x)(Fx \mathbin{\&} Gx)$ (i.e. 'it is not the case that there is something that is both an F and a G')

Examples

- **No hero** wears a cape.
  No heroes wear capes.

Examples

- **No hero** wears a cape.
  No heroes wear capes.

  $(\forall x)(Hx \supset \sim Cx)$

Examples

- **No hero** wears a cape.
  No heroes wear capes.

  $(\forall x)(Hx \supset \sim Cx)$

- **No hero who wears a cape** welcomed Greta.

Examples

- **No hero** wears a cape.
  No heroes wear capes.

  $(\forall x)(Hx \supset \sim Cx)$

- **No hero who wears a cape** welcomed Greta.

  $(\forall x)((Hx \,\&\, Cx) \supset \sim Wxg)$

Examples

- **No hero** wears a cape.
  No heroes wear capes.

  $(\forall x)(Hx \supset \sim Cx)$

- **No hero who wears a cape** welcomed Greta.

  $(\forall x)((Hx \,\&\, Cx) \supset \sim Wxg)$

- Greta admires **no one who wears a cape**.

Examples

- **No hero** wears a cape.
  No heroes wear capes.

  $(\forall x)(Hx \supset {\sim}Cx)$
- **No hero who wears a cape** welcomed Greta.

  $(\forall x)((Hx \,\&\, Cx) \supset {\sim}Wxg)$
- Greta admires **no one who wears a cape**.

  ${\sim}(\exists x)(Cx \,\&\, Agx)$

Examples

- **No hero** wears a cape.
  No heroes wear capes.

  $(\forall x)(Hx \supset {\sim}Cx)$
- **No hero who wears a cape** welcomed Greta.

  $(\forall x)((Hx \,\&\, Cx) \supset {\sim}Wxg)$
- Greta admires **no one who wears a cape**.

  ${\sim}(\exists x)(Cx \,\&\, Agx)$
- Autumn welcomed **no one who welcomed Greta**.

Examples

- **No hero** wears a cape.
  No heroes wear capes.

  $(\forall x)(Hx \supset \sim Cx)$

- **No hero who wears a cape** welcomed Greta.

  $(\forall x)((Hx \,\&\, Cx) \supset \sim Wxg)$

- Greta admires **no one who wears a cape**.

  $\sim(\exists x)(Cx \,\&\, Agx)$

- Autumn welcomed **no one who welcomed Greta**.

  $\sim(\exists x)(Wxg \,\&\, Wax)$

Only Fs are G

- When is “Only Fs are Gs” false?

Only Fs are G

- When is "Only Fs are Gs" false?
- When there is a **non**−F that is a G.

Only Fs are G

- When is “Only Fs are Gs” false?
- When there is a **non**–F that is a G.
- So symbolize it as

$$\sim(\exists x)(\sim Fx \,\&\, Gx)$$

Only Fs are G

- When is "Only Fs are Gs" false?
- When there is a **non**–F that is a G.
- So symbolize it as

$$\sim(\exists x)(\sim Fx \mathbin{\&} Gx)$$

- Or, paraphrase it as: "Any x is G **only if** it is F"

Only Fs are G

- When is "Only Fs are Gs" false?
- When there is a **non**–F that is a G.
- So symbolize it as

$$\sim(\exists x)(\sim Fx \,\&\, Gx)$$

- Or, paraphrase it as: "Any x is G **only if** it is F"
- So another symbolization is:

$$(\forall x)(Gx \supset Fx)$$

Only *F*s are *G*

- When is "Only *F*s are *G*s" false?
- When there is a **non**–*F* that is a *G*.
- So symbolize it as

$$\sim(\exists x)(\sim Fx \,\&\, Gx)$$

- Or, paraphrase it as: "Any *x* is *G* **only if** it is *F*"
- So another symbolization is:

$$(\forall x)(Gx \supset Fx)$$

i.e. being an F is a necessary condition for being a G:
if it's not an F, then it can't be a G

Examples

- Only heroes wear capes:

Examples

- Only heroes wear capes:

  $(\forall x)(Cx \supset Hx)$ (being a hero is necessary for cape-wearing)

Examples

- Only heroes wear capes:

  $(\forall x)(Cx \supset Hx)$ (being a hero is necessary for cape-wearing)
- Only heroes who wear capes welcomed Greta.

Examples

- Only heroes wear capes:

  $(\forall x)(Cx \supset Hx)$ (being a hero is necessary for cape-wearing)
- Only heroes who wear capes welcomed Greta.

  $(\forall x)(Wxg \supset (Hx \,\&\, Cx))$

Examples

- Only heroes wear capes:

  $(\forall x)(Cx \supset Hx)$ (being a hero is necessary for cape–wearing)

- Only heroes who wear capes welcomed Greta.

  $(\forall x)(Wxg \supset (Hx \,\&\, Cx))$

- Greta admires only people who wear capes.

Examples

- Only heroes wear capes:

  $(\forall x)(Cx \supset Hx)$ (being a hero is necessary for cape–wearing)
- Only heroes who wear capes welcomed Greta.

  $(\forall x)(Wxg \supset (Hx \,\&\, Cx))$
- Greta admires only people who wear capes.

  $\sim(\exists x)(\sim Cx \,\&\, Agx)$

Examples

- Only heroes wear capes:

  $(\forall x)(Cx \supset Hx)$ (being a hero is necessary for cape-wearing)
- Only heroes who wear capes welcomed Greta.

  $(\forall x)(Wxg \supset (Hx \,\&\, Cx))$
- Greta admires only people who wear capes.

  $\sim(\exists x)(\sim Cx \,\&\, Agx)$
- Autumn welcomed only heroes and villains.

Examples

- Only heroes wear capes:

  $(\forall x)(Cx \supset Hx)$ (being a hero is necessary for cape–wearing)
- Only heroes who wear capes welcomed Greta.

  $(\forall x)(Wxg \supset (Hx \,\&\, Cx))$
- Greta admires only people who wear capes.

  $\sim(\exists x)(\sim Cx \,\&\, Agx)$
- Autumn welcomed only heroes and villains.

  $\sim(\exists x)(\sim(Hx \vee Vx) \,\&\, Wax)$

Examples

- Only heroes wear capes:

  $(\forall x)(Cx \supset Hx)$ (being a hero is necessary for cape–wearing)
- Only heroes who wear capes welcomed Greta.

  $(\forall x)(Wxg \supset (Hx \,\&\, Cx))$
- Greta admires only people who wear capes.

  $\sim(\exists x)(\sim Cx \,\&\, Agx)$
- Autumn welcomed only heroes and villains.

  $\sim(\exists x)(\sim(Hx \vee Vx) \,\&\, Wax)$

  $(\forall x)(Wax \supset (Hx \vee Vx))$

The indefinite article

- We use “is a” to indicate predication, e.g., “Greta is a hero.”

The indefinite article

- We use “is a” to indicate predication, e.g., “Greta is a hero.”
- Often “a” is used to claim existence, e.g.,

The indefinite article

- We use “is a” to indicate predication, e.g., “Greta is a hero.”
- Often “a” is used to claim existence, e.g.,

  Greta admires a hero.

The indefinite article

- We use “is a” to indicate predication, e.g., “Greta is a hero.”
- Often “a” is used to claim existence, e.g.,

  Greta admires a hero.

  $(\exists x)(Hx \mathbin{\&} Agx)$

The indefinite article

- We use "is a" to indicate predication, e.g., "Greta is a hero."
- Often "a" is used to claim existence, e.g.,

  Greta admires a hero.

  $(\exists x)(Hx \,\&\, Agx)$
- But a **generic** indefinite is closer to a universal quantifier:

The indefinite article

- We use "is a" to indicate predication, e.g., "Greta is a hero."
- Often "a" is used to claim existence, e.g.,

  Greta admires a hero.

  $(\exists x)(Hx \,\&\, Agx)$
- But a **generic** indefinite is closer to a universal quantifier:

  A hero is someone who inspires.

The indefinite article

- We use “is a” to indicate predication, e.g., “Greta is a hero.”
- Often “a” is used to claim existence, e.g.,

  Greta admires a hero.

  $(\exists x)(Hx \,\&\, Agx)$
- But a **generic** indefinite is closer to a universal quantifier:

  A hero is someone who inspires.

  $(\forall x)(Hx \supset Ix)$

The indefinite article

- We use “is a” to indicate predication, e.g., “Greta is a hero.”
- Often “a” is used to claim existence, e.g.,

  Greta admires a hero.

  $(\exists x)(Hx \,\&\, Agx)$
- But a **generic** indefinite is closer to a universal quantifier:

  A hero is someone who inspires.

  $(\forall x)(Hx \supset Ix)$
- Be careful if the indefinite article is in the antecedent of a conditional:

The indefinite article

- We use "is a" to indicate predication, e.g., "Greta is a hero."
- Often "a" is used to claim existence, e.g.,

  Greta admires a hero.

  $(\exists x)(Hx \,\&\, Agx)$
- But a **generic** indefinite is closer to a universal quantifier:

  A hero is someone who inspires.

  $(\forall x)(Hx \supset Ix)$
- Be careful if the indefinite article is in the antecedent of a conditional:

  If **a hero** wears a cape, **they** inspire.

The indefinite article

- We use "is a" to indicate predication, e.g., "Greta is a hero."
- Often "a" is used to claim existence, e.g.,

  Greta admires a hero.

  $(\exists x)(Hx \,\&\, Agx)$
- But a **generic** indefinite is closer to a universal quantifier:

  A hero is someone who inspires.

  $(\forall x)(Hx \supset Ix)$
- Be careful if the indefinite article is in the antecedent of a conditional:

  If **a hero** wears a cape, **they** inspire.

  That means: all heroes who wear capes inspire.

The indefinite article

- We use "is a" to indicate predication, e.g., "Greta is a hero."
- Often "a" is used to claim existence, e.g.,

  Greta admires a hero.

  $(\exists x)(Hx \,\&\, Agx)$
- But a **generic** indefinite is closer to a universal quantifier:

  A hero is someone who inspires.

  $(\forall x)(Hx \supset Ix)$
- Be careful if the indefinite article is in the antecedent of a conditional:

  If **a hero** wears a cape, **they** inspire.

  That means: all heroes who wear capes inspire.

  $(\forall x)((Hx \,\&\, Cx) \supset Ix)$

Universal “some”; existential “any”

- “Someone,” “something” can require a **universal** quantifier: if it’s in the antecedent of a conditional, with a pronoun in the consequent referring back to it, e.g.,

Universal "some"; existential "any"

- "Someone," "something" can require a **universal** quantifier: if it's in the antecedent of a conditional, with a pronoun in the consequent referring back to it, e.g.,

  If **someone** is a hero, Autum admires **them**.

Universal “some”; existential “any”

- “Someone,” “something” can require a **universal** quantifier: if it’s in the antecedent of a conditional, with a pronoun in the consequent referring back to it, e.g.,

  If **someone** is a hero, Autum admires **them**.

  Roughly: Autumn admires all heroes.

Universal “some”; existential “any”

- “Someone,” “something” can require a **universal** quantifier: if it’s in the antecedent of a conditional, with a pronoun in the consequent referring back to it, e.g.,

  If **someone** is a hero, Autum admires **them**.

  Roughly: Autumn admires all heroes.

  $(\forall x)(Hx \supset Aax)$

Universal "some"; existential "any"

- "Someone," "something" can require a **universal** quantifier: if it's in the antecedent of a conditional, with a pronoun in the consequent referring back to it, e.g.,

  If **someone** is a hero, Autum admires **them**.

  Roughly: Autumn admires all heroes.

  $(\forall x)(Hx \supset Aax)$
- "Any" in antecedents but **without** pronouns referring back to them are **existential**:

Universal "some"; existential "any"

- "Someone," "something" can require a **universal** quantifier: if it's in the antecedent of a conditional, with a pronoun in the consequent referring back to it, e.g.,

  If **someone** is a hero, Autum admires **them**.

  Roughly: Autumn admires all heroes.

  $(\forall x)(Hx \supset Aax)$

- "Any" in antecedents but **without** pronouns referring back to them are **existential**:

  If **anyone** is a hero, Greta is.

Universal "some"; existential "any"

- "Someone," "something" can require a **universal** quantifier: if it's in the antecedent of a conditional, with a pronoun in the consequent referring back to it, e.g.,

  If **someone** is a hero, Autum admires **them**.

  Roughly: Autumn admires all heroes.

  $(\forall x)(Hx \supset Aax)$

- "Any" in antecedents but **without** pronouns referring back to them are **existential**:

  If **anyone** is a hero, Greta is.

  Roughly: if there are heroes (at all), Greta is a hero.

Universal “some”; existential “any”

- “Someone,” “something” can require a **universal** quantifier: if it’s in the antecedent of a conditional, with a pronoun in the consequent referring back to it, e.g.,

  If **someone** is a hero, Autum admires **them**.

  Roughly: Autumn admires all heroes.

  $(\forall x)(Hx \supset Aax)$

- “Any” in antecedents but **without** pronouns referring back to them are **existential**:

  If **anyone** is a hero, Greta is.

  Roughly: if there are heroes (at all), Greta is a hero.

  $(\exists x)\, Hx \supset Hg$

8. Intro. to Quantifier Logic

f. Mixed domains

Mixed domains

- Sometimes you want to talk about more than one kind of thing.

Mixed domains

- Sometimes you want to talk about more than one kind of thing.
- The domain can include any mix of things (e.g., people, animals, items of clothing, feelings)

Mixed domains

- Sometimes you want to talk about more than one kind of thing.
- The domain can include any mix of things (e.g., people, animals, items of clothing, feelings)
- Proper symbolization then needs predicates for these kinds, e.g.:

Mixed domains

- Sometimes you want to talk about more than one kind of thing.
- The domain can include any mix of things (e.g., people, animals, items of clothing, feelings)
- Proper symbolization then needs predicates for these kinds, e.g.:

Domain: people alive in 2022 and items of clothing

Mixed domains

- Sometimes you want to talk about more than one kind of thing.
- The domain can include any mix of things (e.g., people, animals, items of clothing, feelings)
- Proper symbolization then needs predicates for these kinds, e.g.:

Domain: people alive in 2022 and items of clothing

Px: $\_\_\_\_\__x$ is a person

Mixed domains

- Sometimes you want to talk about more than one kind of thing.
- The domain can include any mix of things (e.g., people, animals, items of clothing, feelings)
- Proper symbolization then needs predicates for these kinds, e.g.:

Domain: people alive in 2022 and items of clothing

Px: $\_\_\_\_\__x$ is a person

Lx: $\_\_\_\_\__x$ is an item of clothing.

Mixed domains

- Sometimes you want to talk about more than one kind of thing.
- The domain can include any mix of things (e.g., people, animals, items of clothing, feelings)
- Proper symbolization then needs predicates for these kinds, e.g.:

Domain: people alive in 2022 and items of clothing

Px: $\_\_\_\_\__x$ is a person

Lx: $\_\_\_\_\__x$ is an item of clothing.

Ex: $\_\_\_\_\__x$ is a cape (recall: ‘Cx’ is ‘wears a cape’)

Mixed domains

- Sometimes you want to talk about more than one kind of thing.
- The domain can include any mix of things (e.g., people, animals, items of clothing, feelings)
- Proper symbolization then needs predicates for these kinds, e.g.:

Domain: people alive in 2022 and items of clothing

Px: $\_\_\_\_\__x$ is a person

Lx: $\_\_\_\_\__x$ is an item of clothing.

Ex: $\_\_\_\_\__x$ is a cape (recall: 'Cx' is 'wears a cape')

Rxy: $\_\_\_\_\__x$ wears $\_\_\_\_\__y$

Quantification in mixed domains

- Not everyone is wearing a cape.

Quantification in mixed domains

- Not everyone is wearing a cape.
  - In domain of people only:

Quantification in mixed domains

- Not everyone is wearing a cape.
  - In domain of people only:
    $\sim(\forall x)\, Cx$

Quantification in mixed domains

- Not everyone is wearing a cape.
  - In domain of people only:
    $\sim(\forall x)\, Cx$
  - In mixed domain:

Quantification in mixed domains

- Not everyone is wearing a cape.
  - In domain of people only:
    $\sim(\forall x)\, Cx$
  - In mixed domain:
    $\sim(\forall x)\,(Px \supset Cx)$

Quantification in mixed domains

- Not everyone is wearing a cape.
  - In domain of people only:
    $\sim(\forall x)\, Cx$
  - In mixed domain:
    $\sim(\forall x)\,(Px \supset Cx)$
- Some people inspire.

Quantification in mixed domains

- Not everyone is wearing a cape.
  - In domain of people only:
    $\sim(\forall x)\, Cx$
  - In mixed domain:
    $\sim(\forall x)\,(Px \supset Cx)$
- Some people inspire.
  - In domain of people only:

Quantification in mixed domains

- Not everyone is wearing a cape.
  - In domain of people only:
    $\sim(\forall x)\, Cx$
  - In mixed domain:
    $\sim(\forall x)\, (Px \supset Cx)$
- Some people inspire.
  - In domain of people only:
    $(\exists x)\, Ix$

Quantification in mixed domains

- Not everyone is wearing a cape.
  - In domain of people only:
    $\sim(\forall x)\, Cx$
  - In mixed domain:
    $\sim(\forall x)\,(Px \supset Cx)$
- Some people inspire.
  - In domain of people only:
    $(\exists x)\, Ix$
  - In mixed domain:

Quantification in mixed domains

- Not everyone is wearing a cape.
  - In domain of people only:
    $\sim(\forall x)\ Cx$
  - In mixed domain:
    $\sim(\forall x)\ (Px \supset Cx)$
- Some people inspire.
  - In domain of people only:
    $(\exists x)\ Ix$
  - In mixed domain:
    $(\exists x)(Px \ \&\ Ix)$

Quantification in mixed domains

- Not everyone is wearing a cape.
  - In domain of people only:
    $\sim(\forall x)\, Cx$
  - In mixed domain:
    $\sim(\forall x)\,(Px \supset Cx)$
- Some people inspire.
  - In domain of people only:
    $(\exists x)\, Ix$
  - In mixed domain:
    $(\exists x)(Px \,\&\, Ix)$
- Greta wears something.

Quantification in mixed domains

- Not everyone is wearing a cape.
  - In domain of people only:
    $\sim(\forall x)\, Cx$
  - In mixed domain:
    $\sim(\forall x)\,(Px \supset Cx)$
- Some people inspire.
  - In domain of people only:
    $(\exists x)\, Ix$
  - In mixed domain:
    $(\exists x)(Px \,\&\, Ix)$
- Greta wears something.
  $(\exists x)(Lx \,\&\, Rgx)$

8. Intro. to Quantifier Logic

g. Captain Morgan’s (tele)scope!

(Spiced) de Morgan's for Quantifiers

- We can push negations through quantified expressions, flipping the quantifiers and negating what lies in their scope:

(Spiced) de Morgan's for Quantifiers

- We can push negations through quantified expressions, flipping the quantifiers and negating what lies in their scope:
- $\sim(\forall x)\Phi(x)$: "it's not the case that for everything, Phi"

  is equivalent to

(Spiced) de Morgan's for Quantifiers

- We can push negations through quantified expressions, flipping the quantifiers and negating what lies in their scope:
- $\sim(\forall x)\Phi(x)$: "it's not the case that for everything, Phi"
  is equivalent to
  $(\exists x)\sim\Phi(x)$: "there exists something such that not-Phi"

(Spiced) de Morgan's for Quantifiers

- We can push negations through quantified expressions, flipping the quantifiers and negating what lies in their scope:
- $\sim(\forall x)\Phi(x)$: "it's not the case that for everything, Phi"
  is equivalent to
  $(\exists x)\sim\Phi(x)$: "there exists something such that not-Phi"
- $\sim(\exists x)\Phi(x)$: "it's not the case that for something, Phi"
  is equivalent to

(Spiced) de Morgan's for Quantifiers

- We can push negations through quantified expressions, flipping the quantifiers and negating what lies in their scope:
- $\sim(\forall x)\Phi(x)$: "it's not the case that for everything, Phi"

  is equivalent to

  $(\exists x)\sim\Phi(x)$: "there exists something such that not-Phi"
- $\sim(\exists x)\Phi(x)$: "it's not the case that for something, Phi"

  is equivalent to

  $(\forall x)\sim\Phi(x)$: "for everything, not-Phi", i.e. Phi for-nothing!

(Spiced) de Morgan's for Quantifiers

- We can push negations through quantified expressions, flipping the quantifiers and negating what lies in their scope:
- $\sim(\forall x)\Phi(x)$: "it's not the case that for everything, Phi"

  is equivalent to

  $(\exists x)\sim\Phi(x)$: "there exists something such that not-Phi"
- $\sim(\exists x)\Phi(x)$: "it's not the case that for something, Phi"

  is equivalent to

  $(\forall x)\sim\Phi(x)$: "for everything, not-Phi", i.e. Phi for-nothing!

  A concrete example:

(Spiced) de Morgan's for Quantifiers

- We can push negations through quantified expressions, flipping the quantifiers and negating what lies in their scope:
- $\sim(\forall x)\Phi(x)$: "it's not the case that for everything, Phi"
  is equivalent to
  $(\exists x)\sim\Phi(x)$: "there exists something such that not-Phi"
- $\sim(\exists x)\Phi(x)$: "it's not the case that for something, Phi"
  is equivalent to
  $(\forall x)\sim\Phi(x)$: "for everything, not-Phi", i.e. Phi for-nothing!
  A concrete example:
- *It's not the case that some hero wears a cape*:

(Spiced) de Morgan's for Quantifiers

- We can push negations through quantified expressions, flipping the quantifiers and negating what lies in their scope:
- $\sim(\forall x)\Phi(x)$: "it's not the case that for everything, Phi"
  is equivalent to
  $(\exists x)\sim\Phi(x)$: "there exists something such that not–Phi"
- $\sim(\exists x)\Phi(x)$: "it's not the case that for something, Phi"
  is equivalent to
  $(\forall x)\sim\Phi(x)$: "for everything, not–Phi", i.e. Phi for–nothing!
  A concrete example:
- *It's not the case that some hero wears a cape*:
  $\sim(\exists x)(Hx \,\&\, Cx)$

(Spiced) de Morgan's for Quantifiers

- We can push negations through quantified expressions, flipping the quantifiers and negating what lies in their scope:
- $\sim(\forall \chi)\Phi(\chi)$: "it's not the case that for everything, Phi"
  is equivalent to
  $(\exists \chi)\sim\Phi(\chi)$: "there exists something such that not–Phi"
- $\sim(\exists \chi)\Phi(\chi)$: "it's not the case that for something, Phi"
  is equivalent to
  $(\forall \chi)\sim\Phi(\chi)$: "for everything, not–Phi", i.e. Phi for–nothing!
  A concrete example:
- *It's not the case that some hero wears a cape*:
  $\sim(\exists x)(Hx \,\&\, Cx)$
  $(\forall x)\sim(Hx \,\&\, Cx)$ (now apply regular de Morgan's!)

(Spiced) de Morgan's for Quantifiers

- We can push negations through quantified expressions, flipping the quantifiers and negating what lies in their scope:
- $\sim(\forall \chi)\Phi(\chi)$: "it's not the case that for everything, Phi"
  is equivalent to
  $(\exists \chi)\sim\Phi(\chi)$: "there exists something such that not-Phi"
- $\sim(\exists \chi)\Phi(\chi)$: "it's not the case that for something, Phi"
  is equivalent to
  $(\forall \chi)\sim\Phi(\chi)$: "for everything, not-Phi", i.e. Phi for-nothing!
  A concrete example:
- *It's not the case that some hero wears a cape*:
  $\sim(\exists x)(Hx \,\&\, Cx)$
  $(\forall x)\sim(Hx \,\&\, Cx)$ (now apply regular de Morgan's!)
  $(\forall x)(\sim Hx \vee \sim Cx)$

Quantifier Scope

- **Scope of a Quantifer**: the **sub**formula for which the quantifier is the main logical operator

Quantifier Scope

- **Scope of a Quantifer**: the **sub**formula for which the quantifier is the main logical operator

in '$(\exists x)(Lx \,\&\, Rgx)$', the scope of the existential is '$(Lx \,\&\, Rgx)$'

Quantifier Scope

- **Scope of a Quantifer**: the **sub**formula for which the quantifier is the main logical operator

in '$(\exists x)(Lx \,\&\, Rgx)$', the scope of the existential is '$(Lx \,\&\, Rgx)$'

in '$(\exists x)Lx \,\&\, (\forall y)Rgy$', the scope of the existential is 'Lx'

Quantifier Scope

- **Scope of a Quantifer**: the **sub**formula for which the quantifier is the main logical operator

  in '$(\exists x)(Lx \,\&\, Rgx)$', the scope of the existential is '$(Lx \,\&\, Rgx)$'

  in '$(\exists x)Lx \,\&\, (\forall y)Rgy$', the scope of the existential is 'Lx'

- A variable is **bound** if it lies within the scope of a quantifer

Quantifier Scope

- **Scope of a Quantifer**: the **sub**formula for which the quantifier is the main logical operator

  in '$(\exists x)(Lx \,\&\, Rgx)$', the scope of the existential is '$(Lx \,\&\, Rgx)$'

  in '$(\exists x)Lx \,\&\, (\forall y)Rgy$', the scope of the existential is 'Lx'

- A variable is **bound** if it lies within the scope of a quantifer
- By the recursive definition that follows, each variable is bound by at most one quantifier!

Recall our Recursive definition of SL wffs

1. Every atomic formula is a wff.

Recall our Recursive definition of SL wffs

1. Every atomic formula is a wff.
2. If Φ is a wff, then $\sim\Phi$ is a wff.

Recall our Recursive definition of SL wffs

1. Every atomic formula is a wff.
2. If Φ is a wff, then $\sim\Phi$ is a wff.
3. If Φ and Ψ are wffs, then $(\Phi \,\&\, \Psi)$ is a wff.

Recall our Recursive definition of SL wffs

1. Every atomic formula is a wff.
2. If Φ is a wff, then $\sim\Phi$ is a wff.
3. If Φ and Ψ are wffs, then $(\Phi \,\&\, \Psi)$ is a wff.
4. If Φ and Ψ are wffs, $(\Phi \vee \Psi)$ is a wff.

Recall our Recursive definition of SL wffs

1. Every atomic formula is a wff.
2. If Φ is a wff, then $\sim\Phi$ is a wff.
3. If Φ and Ψ are wffs, then $(\Phi \& \Psi)$ is a wff.
4. If Φ and Ψ are wffs, $(\Phi \vee \Psi)$ is a wff.
5. If Φ and Ψ are wffs, then $(\Phi \supset \Psi)$ is a wff.

Recall our Recursive definition of SL wffs

1. Every atomic formula is a wff.
2. If Φ is a wff, then $\sim\Phi$ is a wff.
3. If Φ and Ψ are wffs, then $(\Phi \mathbin{\&} \Psi)$ is a wff.
4. If Φ and Ψ are wffs, $(\Phi \vee \Psi)$ is a wff.
5. If Φ and Ψ are wffs, then $(\Phi \supset \Psi)$ is a wff.
6. If Φ and Ψ are wffs, then $(\Phi \equiv \Psi)$ is a wff.

Recall our Recursive definition of SL wffs

1. Every atomic formula is a wff.
2. If Φ is a wff, then $\sim\Phi$ is a wff.
3. If Φ and Ψ are wffs, then $(\Phi \& \Psi)$ is a wff.
4. If Φ and Ψ are wffs, $(\Phi \vee \Psi)$ is a wff.
5. If Φ and Ψ are wffs, then $(\Phi \supset \Psi)$ is a wff.
6. If Φ and Ψ are wffs, then $(\Phi \equiv \Psi)$ is a wff.

Nothing else is a wff of SL! (But some new things are wffs of QL!)

Extending our Recursive definition to QL wffs

1.* **Atomic formula of QL**: an n–place predicate followed by n terms (i.e. constants or variables), where $n \in \mathbb{N}$

Extending our Recursive definition to QL wffs

1.* **Atomic formula of QL**: an n−place predicate followed by n terms (i.e. constants or variables), where $n \in \mathbb{N}$

This includes the SL atomic wff, which are 0−place predicates

Extending our Recursive definition to QL wffs

1.* **Atomic formula of QL**: an *n*–place predicate followed by *n* terms (i.e. constants or variables), where $n \in \mathbb{N}$

   This includes the SL atomic wff, which are 0–place predicates

7. If Φ is a wff, χ is a variable, and Φ contains no χ–quantifiers, then $(\forall\chi)\Phi$ is a wff.

Extending our Recursive definition to QL wffs

1.* **Atomic formula of QL**: an *n*–place predicate followed by *n* terms (i.e. constants or variables), where $n \in \mathbb{N}$

   This includes the SL atomic wff, which are 0–place predicates

7. If Φ is a wff, χ is a variable, and Φ contains no χ–quantifiers, then $(\forall \chi)\Phi$ is a wff.

8. If Φ is a wff, χ is a variable, and Φ contains no χ–quantifiers, then $(\exists \chi)\Phi$ is a wff.

Extending our Recursive definition to QL wffs

1.* **Atomic formula of QL**: an *n*−place predicate followed by *n* terms (i.e. constants or variables), where $n \in \mathbb{N}$

    This includes the SL atomic wff, which are 0−place predicates

7. If Φ is a wff, χ is a variable, and Φ contains no χ−quantifiers, then $(\forall\chi)\Phi$ is a wff.

8. If Φ is a wff, χ is a variable, and Φ contains no χ−quantifiers, then $(\exists\chi)\Phi$ is a wff.

9. All and only wffs of QL come from the prior 8 rules.

QL Sentences: proper subset of QL wffs

- Recall: the wffs of SL *just are* the sentences of SL, i.e. the statements that are true or false under a truth–value assignment to atomic sentences

QL Sentences: proper subset of QL wffs

- Recall: the wffs of SL *just are* the sentences of SL, i.e. the statements that are true or false under a truth-value assignment to atomic sentences
- Not all wffs of QL are sentences! Watch out!

QL Sentences: proper subset of QL wffs

- Recall: the wffs of SL *just are* the sentences of SL, i.e. the statements that are true or false under a truth-value assignment to atomic sentences
- Not all wffs of QL are sentences! Watch out!
- Let L be a two-place predicate. Then Lxx is an atomic wff of QL, but NOT a sentence (‘Lxx’ is neither true nor false)

QL Sentences: proper subset of QL wffs

- Recall: the wffs of SL *just are* the sentences of SL, i.e. the statements that are true or false under a truth-value assignment to atomic sentences
- Not all wffs of QL are sentences! Watch out!
- Let L be a two-place predicate. Then Lxx is an atomic wff of QL, but NOT a sentence ('Lxx' is neither true nor false)
- the x's in 'Lxx' are **free variables**, i.e. unbound variables

QL Sentences: proper subset of QL wffs

- Recall: the wffs of SL *just are* the sentences of SL, i.e. the statements that are true or false under a truth-value assignment to atomic sentences
- Not all wffs of QL are sentences! Watch out!
- Let L be a two-place predicate. Then Lxx is an atomic wff of QL, but NOT a sentence ('Lxx' is neither true nor false)
- the x's in 'Lxx' are **free variables**, i.e. unbound variables
- **Sentence of QL**: a wff that has no free variables: i.e. any variable that occurs is bound by a quantifier