Logical form tutorial

`http://www-rohan.sdsu.edu/~gawron/semantics`

Jean Mark Gawron

San Diego State University, Department of Linguistics

2010-08-19

Overview

Outline

Logical Form

Goal:

- A few simple rules to help the beginner get the hang of translating into logic

Logical Form

Goal:

- A few simple rules to help the beginner get the hang of translating into logic
- Problems

Logical Form

Goal:

- A few simple rules to help the beginner get the hang of translating into logic
- Problems
  - There are a LOT of things to cover

Logical Form

Goal:

- A few simple rules to help the beginner get the hang of translating into logic
- Problems
  - There are a LOT of things to cover
  - The rules can't be complete.

Logical Form

Goal:

- A few simple rules to help the beginner get the hang of translating into logic
- Problems
  - There are a LOT of things to cover
  - The rules can't be complete.
  - Ambiguity of English

Outline

Outline

Principles

The logical form of an English sentence is a decomposition of the sentence into predicates and connectives. The predicates capture the concepts being expressed. The connectives caoture how the concepts are related.

1. Almost every noun, verb, and adjectives corresponds to a predicate.

Principles

The logical form of an English sentence is a decomposition of the sentence into predicates and connectives. The predicates capture the concepts being expressed. The connectives caoture how the concepts are related.

1. Almost every noun, verb, and adjectives corresponds to a predicate.
2. Exceptions: No auxiliary verb, including the verb *be* (*is*, *are*, *was*, *being*, *been*), corresponds to a predicate.

Principles

The logical form of an English sentence is a decomposition of the sentence into predicates and connectives. The predicates capture the concepts being expressed. The connectives caoture how the concepts are related.

1. Almost every noun, verb, and adjectives corresponds to a predicate.
2. Exceptions: No auxiliary verb, including the verb *be* (*is, are, was, being, been*), corresponds to a predicate.
3. Make sure each predicate word is accounted for in your translation.

Principles

The logical form of an English sentence is a decomposition of the sentence into predicates and connectives. The predicates capture the concepts being expressed. The connectives caoture how the concepts are related.

1. Almost every noun, verb, and adjectives corresponds to a predicate.
2. Exceptions: No auxiliary verb, including the verb *be* (*is*, *are*, *was*, *being*, *been*), corresponds to a predicate.
3. Make sure each predicate word is accounted for in your translation.
4. Use your predicates consistently

Principles

The logical form of an English sentence is a decomposition of the sentence into predicates and connectives. The predicates capture the concepts being expressed. The connectives caoture how the concepts are related.

1. Almost every noun, verb, and adjectives corresponds to a predicate.
2. Exceptions: No auxiliary verb, including the verb *be* (*is, are, was, being, been*), corresponds to a predicate.
3. Make sure each predicate word is accounted for in your translation.
4. Use your predicates consistently
    1. Same arity (same number of arguments)

Principles

The logical form of an English sentence is a decomposition of the sentence into predicates and connectives. The predicates capture the concepts being expressed. The connectives caoture how the concepts are related.

1. Almost every noun, verb, and adjectives corresponds to a predicate.
2. Exceptions: No auxiliary verb, including the verb *be* (*is, are, was, being, been*), corresponds to a predicate.
3. Make sure each predicate word is accounted for in your translation.
4. Use your predicates consistently
   1. Same arity (same number of arguments)
   2. Arguments are in a consistent order

Connectives: Quantifiers, negation, and sentential

Universals ($\forall$), Existentials ($\exists$), and negation $\sim$ correspond to appropriate English words, and each quantifier goes with its appropriate sentential connective:

every, all, any	$\forall$	$\forall$	$\rightarrow$	$\forall x$ dog(x) $\rightarrow$ bark(x)
some, a, a certain	$\exists$	$\exists$	&	$\exists x$ dog(x) & bark(x)
not, n't	$\sim$	$\sim \exists$	&	$\sim \exists x$ dog(x) & bark(x)
no	$\sim \exists$			

Ambiguity

(1) a. Every prize was won by some high school kid.

Ambiguity

(2) a. Every prize was won by some high school kid.

b. For every prize, x, there was some high school kid, y, such that y won x.

b. $\forall x[\text{prize}(x) \rightarrow \exists y[\text{high-school-kid}(y) \; \& \; \text{win}(y, x)]\,]$

Ambiguity

(3) a. Every prize was won by some high school kid.

b. For every prize, x, there was some high school kid, y, such that y won x.

c. Some particular high school kid y won every prize, x.

b. $\forall x[\text{prize}(x) \rightarrow \exists y[\text{high-school-kid}(y) \mathbin{\&} \text{win}(y, x)]]$

c. $\exists y[\text{high-school-kid}(y) \mathbin{\&} \forall x[\text{prize}(x) \rightarrow \text{win}(y, x)]]$

Ambiguity

(4) a. Every prize was won by some high school kid.

b. For every prize, x, there was some high school kid, y, such that y won x.

c. Some particular high school kid y won every prize, x.

b. $\forall x[\text{prize}(x) \rightarrow \exists y[\text{high-school-kid}(y)\ \&\ \text{win}(y, x)]\,]$

c. $\exists y[\text{high-school-kid}(y)\ \&\ \forall x[\text{prize}(x) \rightarrow \text{win}(y, x)]\,]$

The two translations share all the same predicates, and even the arguments of the predicates are the same. All that differs is the way the predications are **connected**.

Outline

Noun, Adj, Prep

	Arity		Comments
Nouns	1-place	man(x)	except relational nouns (*husband, father*)
Adjectives	1-place	happy(x)	except relational adjectives (*fond+for, certain+to*)
Prepositions	2-place	from(x, Spain)	except sometimes part of verb meaning (*rely+on*), object of prep is arg2

Verbs

	Arity		Comments
intransitive	1-place	walk(j)	*walk, faint, sleep, fall, ...* Ignore tense.
transitive	2-place	hit(j,f) love(m,j)	*hit, kill, kick, eat, ...* unpassivize passive sentences (*John was loved by Mary* $\rightarrow$ *Mary loved John*)
ditransitive	3-place	give(m,the book, j)	*give, send, cost, charge, ...*
Auxiliaries	syncategorematic		*be, do*, have*, may, might, can, could, should, shall, will, would*

*: *do* and *have* are ambiguous. They are also transitive verbs.

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists\, x\ \text{kid}(x)$

NP modifiers

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists\, x\ \text{kid}(x)$
a tall kid_x	

NP modifiers

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists x\, \text{kid}(x)$
a tall kid_x	$\exists x\, \text{tall}(x)\ \&\ \text{kid}(x)$

NP modifiers

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists x\ \text{kid}(x)$
a tall kid_x	$\exists x\ \text{tall}(x)\ \&\ \text{kid}(x)$
a kid_x from Spain	

NP modifiers

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists x\ \text{kid}(x)$
a tall kid_x	$\exists x\ \text{tall}(x)\ \&\ \text{kid}(x)$
a kid_x from Spain	$\exists x\ \text{kid}(x)\ \&\ \text{from}(x, \textsc{s})$

NP modifiers

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists\, x\ \text{kid}(x)$
a tall kid_x	$\exists\, x\ \text{tall}(x)\ \&\ \text{kid}(x)$
a kid_x from Spain	$\exists\, x\ \text{kid}(x)\ \&\ \text{from}(x, \textsc{s})$
a tall kid_x from Spain	

NP modifiers

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists\, x\ \text{kid}(x)$
a tall kid_x	$\exists\, x\ \text{tall}(x)\ \&\ \text{kid}(x)$
a kid_x from Spain	$\exists\, x\ \text{kid}(x)\ \&\ \text{from}(x, \textsc{s})$
a tall kid_x from Spain	$\exists\, x\ \text{tall}(x)\ \&\ \text{kid}(x)\ \&\ \text{from}(x, \textsc{s})$

NP modifiers

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists\, x\ \text{kid}(x)$
a tall kid_x	$\exists\, x\ \text{tall}(x)\ \&\ \text{kid}(x)$
a kid_x from Spain	$\exists\, x\ \text{kid}(x)\ \&\ \text{from}(x,\ \text{S})$
a tall kid_x from Spain	$\exists\, x\ \text{tall}(x)\ \&\ \text{kid}(x)\ \&\ \text{from}(x,\ \text{S})$
a high school kid_x	

NP modifiers

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists x\ \text{kid}(x)$	
a tall kid_x	$\exists x\ \text{tall}(x)\ \&\ \text{kid}(x)$	
a kid_x from Spain	$\exists x\ \text{kid}(x)\ \&\ \text{from}(x,\ \text{S})$	
a tall kid_x from Spain	$\exists x\ \text{tall}(x)\ \&\ \text{kid}(x)\ \&\ \text{from}(x,\ \text{S})$	
a high school kid_x	$\exists x\ \text{high-school}(x)\ \&\ \text{kid}(x)$	Wrong!

NP modifiers

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists x\ \text{kid}(x)$	
a tall kid_x	$\exists x\ \text{tall}(x)\ \&\ \text{kid}(x)$	
a kid_x from Spain	$\exists x\ \text{kid}(x)\ \&\ \text{from}(x,\ \textsc{s})$	
a tall kid_x from Spain	$\exists x\ \text{tall}(x)\ \&\ \text{kid}(x)\ \&\ \text{from}(x,\ \textsc{s})$	
a high school kid_x	$\exists x\ \text{high-school}(x)\ \&\ \text{kid}(x)$	Wrong!
a high school kid_x		

NP modifiers

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists x\ \text{kid}(x)$	
a tall kid_x	$\exists x\ \text{tall}(x)\ \&\ \text{kid}(x)$	
a kid_x from Spain	$\exists x\ \text{kid}(x)\ \&\ \text{from}(x,\ \text{S})$	
a tall kid_x from Spain	$\exists x\ \text{tall}(x)\ \&\ \text{kid}(x)\ \&\ \text{from}(x,\ \text{S})$	
a high school kid_x	$\exists x\ \text{high-school}(x)\ \&\ \text{kid}(x)$	Wrong!
a high school kid_x	$\exists x\ \text{high-school-kid}(x)$	Right!

NP modifiers

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists x\ \text{kid}(x)$	
a tall kid_x	$\exists x\ \text{tall}(x)\ \&\ \text{kid}(x)$	
a kid_x from Spain	$\exists x\ \text{kid}(x)\ \&\ \text{from}(x,\ \textsc{s})$	
a tall kid_x from Spain	$\exists x\ \text{tall}(x)\ \&\ \text{kid}(x)\ \&\ \text{from}(x,\ \textsc{s})$	
a high school kid_x	$\exists x\ \text{high-school}(x)\ \&\ \text{kid}(x)$	Wrong!
a high school kid_x	$\exists x\ \text{high-school-kid}(x)$	Right!

NP modifiers

A	kid_x		from Spain
$\exists x$	$\text{kid}(x)$	&	from(, S)
			↑

Noun phrases

Translate each Noun phrase (NP) in isolation.

Some examples of noun phrase translations

a kid_x	$\exists x\ kid(x)$	
a tall kid_x	$\exists x\ tall(x)\ \&\ kid(x)$	
a kid_x from Spain	$\exists x\ kid(x)\ \&\ from(x, \textsc{s})$	
a tall kid_x from Spain	$\exists x\ tall(x)\ \&\ kid(x)\ \&\ from(x, \textsc{s})$	
a high school kid_x	$\exists x\ high\text{-}school(x)\ \&\ kid(x)$	Wrong!
a high school kid_x	$\exists x\ high\text{-}school\text{-}kid(x)$	Right!

NP modifiers

A	kid_x		from Spain
$\exists x$	$kid(x)$	&	$from(x, \textsc{s})$
			↑

The NP variable always occurs in the translations of its modifiers.

Outline

Procedure: Simple version

Initial sentence

Some young woman arrived.

Procedure: Simple version

Initial sentence

Some young woman arrived.

1. Find each quantified NP, and choose a variable for it.

Procedure: Simple version

Initial sentence

Some young woman arrived.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

[Some young woman_x] arrived.

Procedure: Simple version

Initial sentence

Some young woman arrived.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

[Some young woman$_x$] arrived.

2. Move each quantified NP out of the sentence, leaving the variable you chose behind.

Procedure: Simple version

Initial sentence

Some young woman arrived.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

[Some young woman_x] arrived.

2. Move each quantified NP out of the sentence, leaving the variable you chose behind.

2. Remove Quantified NPs

Moved out	Sentence
	x arrived.

Procedure: Simple version

Initial sentence

Some young woman arrived.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

[Some young woman$_x$] arrived.

2. Move each quantified NP out of the sentence, leaving the variable you chose behind. But keep it around for later!

2. Remove Quantified NPs

Moved out	Sentence
[Some young woman$_x$]	x arrived.

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

3. NPs → logic

Moved out	Sentence
A young woman	x arrived.

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

3. NPs → logic

Moved out	Sentence
$\exists x$ woman(x) & young(x)	x arrived.

4. Turn the sentence into logic, using predicate principles.

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

3. NPs $\rightarrow$ logic

Moved out	Sentence
$\exists\, x$ woman(x) & young(x)	x arrived.

4. Turn the sentence into logic, using predicate principles.

4. Sentence $\rightarrow$ logic

Moved out	Sentence
$\exists\, x$ woman(x) & young(x)	arrive(x).

Procedure concluded

5. Add each NP translation back *one at a time*, using the right sentential connective:

Procedure concluded

5. Add each NP translation back *one at a time*, using the right sentential connective:

5. Move NP's back

Moved out	Sentence
	$\exists\, x$ woman(x) & young(x) & arrive(x)

Find Qtfd NPs

1.	Some young woman$_x$ arrived.
2.	
3.	
4.	
5.	

Summary

Remove Qtfd NPs

1.	Some young woman_x arrived.
2.	x arrived.
3.	
4.	
5.	

Summary

Remove Qtfd NPs

1.		Some young woman$_x$ arrived.
2.	Some young woman$_x$	x arrived.
3.		
4.		
5.		

Summary

NPs → logic

1.		[Some young woman$_x$] arrived.
2.	[Some young woman$_x$]	x arrived.
3.	$\exists x$ young(x) & woman(x)	x arrived.
4.		
5.		

Summary

S → logic

1.		Some young woman_x arrived.
2.	Some young woman_x	x arrived.
3.	$\exists x$ young(x) & woman(x)	x arrived.
4.	$\exists x$ young(x) & woman(x)	arrive(x)
5.		

Summary

Move NPs back

1.		Some young woman$_x$ arrived.
2.	Some young woman$_x$	x arrived.
3.	$\exists x$ young(x) & woman(x)	x arrived.
4.	$\exists x$ young(x) & woman(x)	arrive(x)
5.		$\exists x$ young(x) & woman(x) & arrive(x)

Outline

A different example

Initial sentence

Utopia welcomes every traveler from Spain.

A different example

Initial sentence

Utopia welcomes every traveler from Spain.

1. Find each quantified NP, and choose a variable for it.

A different example

Initial sentence

Utopia welcomes every traveler from Spain.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

Utopia welcomes [every traveler$_x$ from Spain].

A different example

Initial sentence

Utopia welcomes every traveler from Spain.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

Utopia welcomes [every traveler$_x$ from Spain].

2. Move each quantified NP out of the sentence, leaving the variable you chose behind.

A different example

Initial sentence

Utopia welcomes every traveler from Spain.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

Utopia welcomes [every traveler$_x$ from Spain].

2. Move each quantified NP out of the sentence, leaving the variable you chose behind.

2. Remove Quantified NPs

Moved out	Sentence
	Utopia welcomes x.

A different example

Initial sentence

Utopia welcomes every traveler from Spain.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

Utopia welcomes [every traveler$_x$ from Spain].

2. Move each quantified NP out of the sentence, leaving the variable you chose behind. But keep it around for later!

2. Remove Quantified NPs

Moved out	Sentence
every traveler from Spain$_x$	Utopia welcomes x.

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

3. NPs → logic

Moved out	Sentence
$\forall\, x$ traveler(x) & from(x, S)	Utopia welcomes x.

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

3. NPs → logic

Moved out	Sentence
$\forall$ *x* traveler(*x*) & from(*x*, S)	Utopia welcomes x.

4. Turn the sentence into logic, using predicate principles.

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

3. NPs → logic

Moved out	Sentence
$\forall\, x$ traveler(x) & from(x, S)	Utopia welcomes x.

4. Turn the sentence into logic, using predicate principles.

4. Sentence → logic

Moved out	Sentence
$\forall\, x$ traveler(x) & from(x, S)	welcome(U, x).

Procedure concluded

5. Add each NP translation back *one at a time*, using the right sentential connective:

Procedure concluded

5. Add each NP translation back *one at a time*, using the right sentential connective:

5. Move NP's back

Moved out	Sentence
	$\forall x$ traveler(x) & from(x, S)→welcome(U, x)

Outline

Two qtfd NPs

(5) a. Every prize was won by some high school kid.

Two qtfd NPs

(6) a. Every prize was won by some high school kid.
b. Some high school kid won every prize. (unpassivized form)

Step 0. Unpassivize.

Two qtfd NPs

(7) a. Every prize was won by some high school kid.
    b. Some high school kid won every prize. (unpassivized form)

Step 0. Unpassivize.

1. Find each quantified NP, and choose a variable for it.

Two qtfd NPs

(8) a. Every prize was won by some high school kid.
b. Some high school kid won every prize. (unpassivized form)

Step 0. Unpassivize.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

[Some high school kid$_x$] won [every prize$_y$].

Two qtfd NPs

(9) a. Every prize was won by some high school kid.
b. Some high school kid won every prize. (unpassivized form)

Step 0. Unpassivize.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

[Some high school kid_x] won [every $prize_y$].

2. Move each quantified NP out of the sentence, leaving the variable you chose behind.

Two qtfd NPs

(10) a. Every prize was won by some high school kid.

b. Some high school kid won every prize. (unpassivized form)

Step 0. Unpassivize.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

Some high school kid_x won every $prize_y$.

2. Move each quantified NP out of the sentence, leaving the variable you chose behind.

2. Remove Quantified NPs

Moved out	Sentence
Some high school kid_x	x won y.
every $prize_y$	

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

3. NPs → logic

Moved out	Sentence
$\exists x$ high-school-kid(x)	x won y.
$\forall y$ prize(y)	

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

3. NPs → logic

Moved out	Sentence
$\exists x$ high-school-kid(x)	x won y.
$\forall y$ prize(y)	

4. Turn the sentence into logic, using predicate principles.

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

3. NPs → logic

Moved out	Sentence
$\exists x$ high-school-kid(x)	x won y.
$\forall y$ prize(y)	

4. Turn the sentence into logic, using predicate principles.

4. Sentence → logic

Moved out	Sentence
$\exists x$ high-school-kid(x)	win(x, y)
$\forall y$ prize(y)	

Procedure concluded, rdg 1

5. Add each NP translation back *one at a time*, using the right sentential connective:

Procedure concluded, rdg 1

5. Add each NP translation back *one at a time*, using the right sentential connective:

5a. Move first NP back, using right connective (&).

Moved out	Sentence
$\forall y$ prize(y)	$\exists x$ high-school-kid(x) & win(x, y)

Procedure concluded, rdg 1

5. Add each NP translation back *one at a time*, using the right sentential connective:

5a. Move first NP back, using right connective (&).

Moved out	Sentence
$\forall y$ prize(y)	$\exists x$ high-school-kid(x) & win(x, y)

5b. Move second NP back, using right connective ($\rightarrow$).

Moved out	Sentence
	$\forall y$ prize(y) $\rightarrow$ ($\exists x$ high-school-kid(x) & win(x, y))

Second reading

Hold on! We've only got ONE of the two readings!

(11) a. For every prize, x, there was some high school kid, y, such that y won x.

a. $\forall x[\text{prize}(x) \rightarrow \exists y[\text{high-school-kid}(y) \;\&\; \text{win}(y, x)]\,]$

b.

Second reading

Hold on! We've only got ONE of the two readings!

(12) a. For every prize, x, there was some high school kid, y, such that y won x.

b. Some particular high school kid, y, won every prize, x.

a. $\forall x[\text{prize}(x) \rightarrow \exists y[\text{high-school-kid}(y) \,\&\, \text{win}(y, x)]\,]$

b. $\exists y[\text{high-school-kid}(y) \,\&\, \forall x[\text{prize}(x) \rightarrow \text{win}(y, x)]\,]$

Procedure concluded, rdg 2

5. Add each NP translation back *one at a time*, using the right sentential connective:

Procedure concluded, rdg 2

5. Add each NP translation back *one at a time*, using the right sentential connective:

5a. Move second NP back, using right connective ($\rightarrow$).

Moved out	Sentence
$\exists x$ high-school-kid(x)	$\forall y$ prize(y) $\rightarrow$ win(x, y)

Procedure concluded, rdg 2

5. Add each NP translation back *one at a time*, using the right sentential connective:

5a. Move second NP back, using right connective ($\rightarrow$).

Moved out	Sentence
$\exists x$ high-school-kid(x)	$\forall y$ prize$(y) \rightarrow$ win(x, y)

5b. Move first NP back, using right connective (&).

Moved out	Sentence
	$\exists x$ high-school-kid(x) & $(\forall y$ prize$(y) \rightarrow$ win$(x, y))$

Consequences

Consequences

Consequences

What we did

We were able to capture the ambiguity by allowing the quantified NPs to recombine with the main sentence translation in either order.

Consequences

What we did

We were able to capture the ambiguity by allowing the quantified NPs to recombine with the main sentence translation in either order.

Consequences

What we did

We were able to capture the ambiguity by allowing the quantified NPs to recombine with the main sentence translation in either order.

A new kind of ambiguity

Consequences

What we did

We were able to capture the ambiguity by allowing the quantified NPs to recombine with the main sentence translation in either order.

A new kind of ambiguity

1. Not lexical ambiguity

Consequences

What we did

We were able to capture the ambiguity by allowing the quantified NPs to recombine with the main sentence translation in either order.

A new kind of ambiguity

1. Not lexical ambiguity
2. Not syntactic ambiguity.

Consequences

What we did

We were able to capture the ambiguity by allowing the quantified NPs to recombine with the main sentence translation in either order.

A new kind of ambiguity

1. Not lexical ambiguity
2. Not syntactic ambiguity.
3. What is it?

Summary

Separating NP meanings from sentences

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.
- Find all Qtfd NPs and choose variable for each.

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.
- Find all Qtfd NPs and choose variable for each.
- Remove Qtfd NPs, leaving behind their variables.

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.
- Find all Qtfd NPs and choose variable for each.
- Remove Qtfd NPs, leaving behind their variables.
- Translate each NP into logic.

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.
- Find all Qtfd NPs and choose variable for each.
- Remove Qtfd NPs, leaving behind their variables.
- Translate each NP into logic.
- Translate the main S into logic.

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.
- Find all Qtfd NPs and choose variable for each.
- Remove Qtfd NPs, leaving behind their variables.
- Translate each NP into logic.
- Translate the main S into logic.
- Move the NPs back. Putting the NPs back in different orders will capture different readings.

Outline

Relative clause

Replace pronouns with their antecedents.

(13) a. Maxine sent every letter John had written to her to Ruth.

Relative clause

Replace pronouns with their antecedents.

(14) a. Maxine sent every letter John had written to her to Ruth.

b. Maxine sent every letter John had written to Maxine to Ruth.

Relative clause

Replace pronouns with their antecedents.

(15) a. Maxine sent every letter John had written to her to Ruth.

b. Maxine sent every letter John had written to Maxine to Ruth.

1. Find each quantified NP, and choose a variable for it.

Relative clause

Replace pronouns with their antecedents.

(16) a. Maxine sent every letter John had written to her to Ruth.

b. Maxine sent every letter John had written to Maxine to Ruth.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

Maxine sent [every letter$_x$ John had written to Maxine] to Ruth.

Relative clause

Replace pronouns with their antecedents.

(17) a. Maxine sent every letter John had written to her to Ruth.

b. Maxine sent every letter John had written to Maxine to Ruth.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

Maxine sent [every letter_x John had written to Maxine] to Ruth.

2. Move each quantified NP out of the sentence, leaving the variable you chose behind.

Relative clause

Replace pronouns with their antecedents.

(18) a. Maxine sent every letter John had written to her to Ruth.

b. Maxine sent every letter John had written to Maxine to Ruth.

1. Find each quantified NP, and choose a variable for it.

1. Find Quantified NPs

Maxine sent [every letter_x John had written to Maxine] to Ruth.

2. Move each quantified NP out of the sentence, leaving the variable you chose behind.

2. Remove Quantified NPs

Moved out	Sentence
[every letter $_x$ John had written to Maxine]	Maxine sent x to Ruth.

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

3. NPs → logic

Moved out	Sentence
$\forall x$ letter(x) & write(J, x, M)	Maxine sent x to Ruth.

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

3. NPs → logic

Moved out	Sentence
$\forall\, x$ letter(x) & write(J, x, M)	Maxine sent x to Ruth.

4. Turn the sentence into logic, using predicate principles.

Continuing procedure

3. Translate each quantified NP into logic, replacing the head noun with a 1-place predicate whose argument is the NP variable:

3. NPs → logic

Moved out	Sentence
$\forall x$ letter(x) & write(J, x, M)	Maxine sent x to Ruth.

4. Turn the sentence into logic, using predicate principles.

4. Sentence → logic

Moved out	Sentence
$\forall x$ letter(x) & write(J, x, M)	send(M, x, R).

Procedure concluded

5. Add each NP translation back *one at a time*, using the right sentential connective:

Procedure concluded

5. Add each NP translation back *one at a time*, using the right sentential connective:

5. Move NP's back

Moved out	Sentence
	$\forall x\,(\text{letter}(x)\ \&\ \text{write}(\textsc{j},\ x,\ \textsc{m})) \rightarrow \text{send}(\textsc{m},\ x,\ \textsc{r})$.

A complicated NP

Every $letter_x$ John had sent to Maxine
$\forall x\ \text{letter}(x)\ \&\ \text{write}(\text{J},\ x,\ \text{M})$

1. Recognize this Noun phrase contains a SENTENCE (relative clause).
   every letter [$_S$ John had sent to Maxine]
   SUBJ VERB PP

A complicated NP

Every letter$_x$ John had sent to Maxine
$\forall\, x$ letter(x) & write(J, x, M)

1. Recognize this Noun phrase contains a SENTENCE (relative clause).
   every letter [$_S$ John had sent to Maxine]
   SUBJ VERB PP

That sentence says something about x

?	John had sent to Maxine.
every letter x	[John had sent x to Maxine]
	send(J, x, M)

A complicated NP

Every letter$_x$ John had sent to Maxine
$\forall\, x$ letter(x) & write(J, x, M)

1. Recognize this Noun phrase contains a SENTENCE (relative clause).

every letter	[$_S$ John	had sent	to Maxine]
	SUBJ	VERB	PP

That sentence says something about x

?	John had sent to Maxine.
every letter x	[John had sent x to Maxine]
	send(J, x, M)

2. Find where x goes, and translate the sentence on its own; add the translation to the translation of the NP:

$$\forall x \text{ letter}(x) \;\&\; \text{send}(\text{J}, x, \text{M})$$

Summary

Separating NP meanings from sentences

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.
- Remove pronouns, if necessary.

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.
- Remove pronouns, if necessary.
- Find all Qtfd NPs and choose variable for each.

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.
- Remove pronouns, if necessary.
- Find all Qtfd NPs and choose variable for each.
- Remove Qtfd NPs, leaving behind their variables.

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.
- Remove pronouns, if necessary.
- Find all Qtfd NPs and choose variable for each.
- Remove Qtfd NPs, leaving behind their variables.
- Translate each NP into logic, making sure you understand how each modifier relates to the NP variable.

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.
- Remove pronouns, if necessary.
- Find all Qtfd NPs and choose variable for each.
- Remove Qtfd NPs, leaving behind their variables.
- Translate each NP into logic, making sure you understand how each modifier relates to the NP variable.
- Translate the main S into logic.

Summary

Separating NP meanings from sentences

- Unpassivize the sentence, if necessary.
- Remove pronouns, if necessary.
- Find all Qtfd NPs and choose variable for each.
- Remove Qtfd NPs, leaving behind their variables.
- Translate each NP into logic, making sure you understand how each modifier relates to the NP variable.
- Translate the main S into logic.
- Move the NPs back. Putting the NPs back in different orders will capture different readings.

Translations discussed

(19) a. A young woman arrived.
b. Utopia welcomes every traveler from Spain.
c. Every prize was won by some high school student.
d. Maxine sent every letter John had written to her to Ruth.

Other translations

1. No spider plants dance.

Other translations

1. No spider plants dance.

$$\sim \exists x \; \text{spider-plant}(x) \;\&\; \text{dance}(x)$$

2. There is a Santa Claus. [Paraphrase this as *A Santa Claus exists*]

Other translations

1. No spider plants dance.

$$\sim \exists x \text{ spider-plant}(x) \mathbin{\&} \text{dance}(x)$$

2. There is a Santa Claus. [Paraphrase this as *A Santa Claus exists*]

$$\exists x \text{ Santa Claus}(x) \mathbin{\&} \text{exists}(x)$$

3. There's no business like show business. [Treat *show business* as a name; treat *there is* as before, paraphrase: *No business like show business exists*, treat *like* as a preposition]

Other translations

1. No spider plants dance.

$$\sim \exists x \text{ spider-plant}(x) \ \& \ \text{dance}(x)$$

2. There is a Santa Claus. [Paraphrase this as *A Santa Claus exists*]

$$\exists x \text{ Santa Claus}(x) \ \& \ \text{exists}(x)$$

3. There's no business like show business. [Treat *show business* as a name; treat *there is* as before, paraphrase: *No business like show business exists*, treat *like* as a preposition]

$$\sim \exists \text{ business}(x) \ \& \ \text{like}(x, \textsc{sb}) \ \& \ \text{exists}(x)$$

Other translations: Conjunction

Conjunction can often be treated by producing a paraphrase with two conjoined sentences:

4. Grammar A generates all and only well-formed formula. [paraphase this as *Grammar A generates all well-formed formula and Grammar A generates only well-formed formula.*; translate each conjoined sentence on its own.]