

Unterspezifikation in der Computationellen Semantik

Einführung

Laura Kallmeyer
Heinrich-Heine-Universität Düsseldorf

Wintersemester 2011/2012

Einführung	1	10. Oktober 2011
------------	---	------------------

Kallmeyer		Unterspezifikation
-----------	--	--------------------

Überblick

1. Skopusambiguitäten
2. Bedeutung von NPs
3. Quantifizierte NPs und Generalisierte Quantoren
4. Andere quantifizierende Elemente

Einführung	2	10. Oktober 2011
------------	---	------------------

Skopusambiguitäten (1)

Wir unterscheiden verschiedene Typen von Ambiguitäten:

- *lexikalische* Ambiguitäten
(1) Hans geht zur *Bank*
- *strukturelle/syntaktische* Ambiguitäten
(2) he sees the man *with the telescope*
- *Skopusambiguitäten*
(3) Some student likes every course.
(4) Alle Politiker sind nicht korrupt

Einführung	3	10. Oktober 2011
------------	---	------------------

Kallmeyer		Unterspezifikation
-----------	--	--------------------

Skopusambiguitäten (2)

(5) Some student likes every course.

Zwei Skopuslesarten:

- Es gibt (mindestens) einen Studenten, dem alle Kurse gefallen.
Some > every.
- Für jeden Kurs findet man (mindestens) einen Studenten, dem dieser Kurs gefällt. Es handelt sich aber nicht notwendig immer um den gleichen Studenten.

Die zweite Lesart ist schwächer.

Einführung	4	10. Oktober 2011
------------	---	------------------

Skopusambiguitäten (3)

(6) Every researcher of a company saw most samples

- Für jeden Wissenschaftler irgendeiner Firma gilt, dass er die meisten Beispiele gesehen hat.
- Es gibt eine Firma, so dass für alle Wissenschaftler dieser Firma gilt, dass sie die meisten Beispiele gesehen haben.
- Für die meisten Beispiele findet man eine Firma (nicht notwendig immer dieselbe), so dass alle Wissenschaftler dieser Firma das Beispiel gesehen haben.
- Für die meisten Beispiele gilt, dass sie von allen Wissenschaftlern irgendeiner Firma gesehen wurden.

ähnlich:

(7) Two politicians spy on someone from every city.

Einführung	5	10. Oktober 2011
------------	---	------------------

Skopusambiguitäten (4)

Es können sehr viele Quantoren miteinander interagieren ohne dass wir ein Problem damit haben, derartige Sätze zu verarbeiten:

(8) A politician can fool most voters on most issues most of the time, but no politician can fool all voters on every single issue all of the time.

Mögliche Lösungen:

- Unterspezifizierte Darstellung, die die Skopusreihenfolge offen lässt.
- Defaultreihenfolgen und Backtrackingmechanismen.

In der Computerlinguistik werden in der Regel unterspezifizierte Repräsentationen verwendet.

Einführung	6	10. Oktober 2011
------------	---	------------------

Skopusambiguitäten (5)

Quantifizierte NPs können mit anderen quantifizierenden Operatoren interagieren, was zu weiteren Ambiguitäten führen kann.

(9) John wants to meet every professor

- Für jeden Professor gilt, dass John ihn treffen möchte.
- John weiß nicht so genau, welche Professoren es eigentlich gibt, aber er möchte auf jeden Fall alle treffen.

ähnlich:

(10) John seems to have have met every professor

(11) John allegedly drives a cadillac

(12) John sometimes dates a girl

Einführung	7	10. Oktober 2011
------------	---	------------------

Bedeutung von NPs (1)

Mögliche NPs:

- *I, you, ...*
- *John, Mary, ...*
- *the president of the US, the Eiffel tower, the new student, ...*
- *every student, a cadillac*

Einführung	8	10. Oktober 2011
------------	---	------------------

Bedeutung von NPs (2)

- Deiktische Ausdrücke wie *I, you, ...* denotieren im Kontext gegebene Individuen.

(13) I am sleeping.

$$\text{sleep}'(x_{\text{speaker}})$$

$$x_{\text{speaker}} \in [[\text{sleep}']]$$

- Eigennamen denotieren ebenfalls im Kontext eindeutig gegebene Individuen.

(14) John is sleeping.

$$\text{sleep}'(\text{John})$$

$$\text{John} \in [[\text{sleep}']]$$
Bedeutung von NPs (3)

- Definite NPs setzen sich aus einem Nomen, das eine Menge denotiert, und dem Operator $[[\text{the}']]$ zusammen, der ausdrückt, dass diese Menge genau ein Element enthalten muss, und der genau dieses eine Element auswählt.

(15) the girl is sleeping.

$$\text{sleep}'(\text{the}'(\text{girl}'))$$

$$x \in [[\text{sleep}']] \text{ where } x \text{ is the unique element in } [[\text{girl}']]$$

- Quantifizierte NPs setzen sich aus einem Determinierer und einem Nomen zusammen. Das Nomen schränkt die Menge ein, über die quantifiziert wird, und der Determinierer bestimmt, wie diese Menge mit dem Rest des Satzes zusammenhängt.

(16) every student is sleeping.

$$\text{for every } x, \text{ it holds that if } \text{student}'(x), \text{ then } \text{sleep}'(x)$$

$$[[\text{student}']] \subseteq [[\text{sleep}']]$$
Generalisierte Quantoren (1)

Frage: Was denotieren Quantoren, bzw. von welchem Typ sind sie?

Solange man es nur mit All- und Existenzquantifikation zu tun hat, könnte man diese als spezielle Mechanismen in die Syntax unserer Prädikatenlogik einbauen.

(17) every student is sleeping.

$$\forall x[\text{student}''(x) \rightarrow \text{sleep}''(x)]$$

(18) some student is sleeping.

$$\exists x[\text{student}''(x) \wedge \text{sleep}''(x)]$$

Problem: Was passiert mit anderen quantifizierenden Determinierern?

Generalisierte Quantoren (2)

Lösung: Generalisierte Quantoren.

(19) every student is sleeping.

- Nomen denotieren einstellige Prädikate, die man als Mengen von Individuen auffassen kann.

$$[[\text{student}']] = \{\text{Mary}, \text{Anna}, \text{Bill}, \text{Max}, \dots\}.$$

Typ: $\langle e, t \rangle$

- Einstellige (intransitive) Verben denotieren ebenfalls einstellige Prädikate.

$$[[\text{sleep}']] = \{\text{Mary}, \text{Anna}, \text{Bill}, \text{John}, \dots\}.$$

Typ: $\langle e, t \rangle$

Generalisierte Quantoren (3)

Der Determinierer beschreibt die Relation, die zwischen diesen beiden Mengen vorliegen muss. D.h., er nimmt zwei Argumente vom Typ $\langle e, t \rangle$ und liefert einen Wahrheitswert, ist also vom Typ

$$\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$$

$$\begin{aligned} & [[\text{every}'(\text{student}')(\text{sleep}')]] \\ = & [[\text{every}']]([\text{student}'])([\text{sleep}']) \\ = & t \text{ if } [[\text{student}']] \subseteq [[\text{sleep}']] \\ & f \text{ else} \end{aligned}$$

Solche Quantoren werden als *generalized quantifiers* bezeichnet.

Generalisierte Quantoren (4)

(20) some student is sleeping.

$$\begin{aligned} & [[\text{some}'(\text{student}')(\text{sleep}')]] \\ = & [[\text{some}']]([\text{student}'])([\text{sleep}']) \\ = & t \text{ if } [[\text{student}']] \cap [[\text{sleep}']] \neq \emptyset \\ & f \text{ else} \end{aligned}$$

(21) at least half of the students are sleeping.

$$\begin{aligned} & [[\text{at_least_half_of}'(\text{student}')(\text{sleep}')]] \\ = & [[\text{at_least_half_of}']]([\text{student}'])([\text{sleep}']) \\ = & t \text{ if } |[[\text{student}']] \cap [[\text{sleep}']]| \geq |[[\text{student}']] \setminus [[\text{sleep}']]| \\ & f \text{ else} \end{aligned}$$

Andere quantifizierende Elemente

- Quantifizierende Adverbien

(22) every professor sometimes teaches TAG

- $\text{every}'(\text{professor}')(\lambda x[\text{sometimes}'(\text{teach_TAG}'(x))])$
- $\text{sometimes}'(\text{every}'(\text{professor}')(\text{teach_TAG}'))$

$[[\text{sometimes}']](p) = t$ if there is a time/situation s such that p is true in s

- Kontrollverben

(23) John wants every professor to teach TAG

- $\text{every}'(\text{professor}')(\lambda x[\text{John_want}'(\text{teach_TAG}'(x))])$
- $\text{John_want}'(\text{every}'(\text{professor}')(\text{teach_TAG}'))$

$[[\text{John_want}']](p) = t$ if for all worlds w representing John's wishes in the actual world, it holds that p is true in w