Semantics and Binding

Greg Kobele

Summer Semester, 2024

1 Lexical Context Sensitivity

In order to deal with context sensitivity, we have given each lexical item a special *context sensitive* meaning, possibly in addition to its regular meaning. For ℓ a lexical item, $[\![\ell]\!]^S$ is its context sensitive meaning. For most lexical items, the context sensitive meaning is derived from its normal meaning in a regular way. (This is because most lexical items are not inherently context sensitive.) The table below shows how the context sensitive meaning relates to the more basic non-context sensitive meaning for expressions of different types.

Type	Basic Meaning	Context-Sensitive Meaning
t	ϕ	$\lambda g.\phi$
et	P	$\lambda X, g.P\ (X\ g)$
eet	R	$\lambda X, Y, g.R\ (X\ g)\ (Y\ g)$
tet	R	$\lambda \Phi, X, g.R\ (\Phi\ g)\ (X\ g)$
(et)t	F	$\lambda \mathcal{P}, g.F\ (\lambda y.\mathcal{P}\ (\lambda h.y)\ g)$
(et)et	F	$\lambda \mathcal{P}, X, g.F\ (\lambda y.\mathcal{P}\ (\lambda h.y)\ g)\ (X\ g)$

While there is a systematic way of deriving the context-sensitive meaning from the basic meaning, that is not relevant here.

Importantly, for *intrinsically* context-sensitive items, that is, those which manipulate contexts in a non-trivial way, we must assign them meanings directly, as per the table below.

lexical item	Basic Meaning	Context-Sensitive Meaning
Pronoun	`N/A`	$\lambda \mathcal{P}, g.\mathcal{P}\ \mathsf{sel}\ g$
Determiner	D	$\lambda \mathcal{P}, \mathcal{Q}, g.D\ (\lambda y.\mathcal{P}\ (\lambda h.y)\ (y+g))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+g))$

2 Context sensitive operations

These context-sensitive meanings make use of fundamental operations sel and $+$. Operation $\mathsf{sel} : \gamma e$ is a function taking a context and returning an individual – it is a pronoun resolution algorithm. Operation $+ : e\gamma\gamma$ is a function taking an individual and a context and returning a context – a context update operator.

Different theories of what contexts are exactly will influence how these operations work, however, this is not (only) the job of linguists. Certainly, it is not our concern here. For our purposes, we will assume that contexts are just sets of individuals, $x + g = \{x\} \cup g$, and $\mathsf{sel}\ g \in g$. That is, context update is just set union, and, given a context, an individual in that context is returned.

3 Examples

With these defintions, we can compute the context sensitive meaning of the sentence *every boy laughed*, with basic structure [[*every boy*] *laughed*].

1. $[\![DP]\!]^S$

$$
\begin{aligned}
&[\![every]\!]^S[\![boy]\!]^S \\
&\quad =(\lambda\mathcal{P},\mathcal{Q},g.[\![every]\!]\ (\lambda y.\mathcal{P}\ (\lambda h.y)\ (y+g))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+g)))\ [\![boy]\!]^S \\
&\quad =\lambda\mathcal{Q},g.[\![every]\!]\ (\lambda y.[\![boy]\!]^S\ (\lambda h.y)\ (y+g))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+g)) \\
&\quad =\lambda\mathcal{Q},g.[\![every]\!]\ (\lambda y.(\lambda X,i.[\![boy]\!]\ (X\ i))\ (\lambda h.y)\ (y+g))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+g)) \\
&\quad =\lambda\mathcal{Q},g.[\![every]\!]\ (\lambda y.(\lambda i.[\![boy]\!]\ ((\lambda h.y)\ i)))\ (y+g))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+g)) \\
&\quad =\lambda\mathcal{Q},g.[\![every]\!]\ (\lambda y.(\lambda i.[\![boy]\!]\ y)\ (y+g))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+g)) \\
&\quad =\lambda\mathcal{Q},g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+g))
\end{aligned}
$$

2. $[\![DP\ V]\!]^S$

$$
\begin{aligned}
&[\![every\ boy]\!]^S\ [\![laughed]\!]^S \\
&\quad =(\lambda\mathcal{Q}, g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+g)))\ [\![laughed]\!]^S \\
&\quad =\lambda g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ \lambda y.[\![laughed]\!]^S\ (\lambda h.y)\ (y+g) \\
&\quad =\lambda g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ \lambda y.(\lambda X, i.[\![laughed]\!]\ (X\ i))\ (\lambda h.y)\ (y+g) \\
&\quad =\lambda g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ \lambda y.(\lambda i.[\![laughed]\!]\ ((\lambda h.y)\ i))\ (y+g) \\
&\quad =\lambda g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ \lambda y.(\lambda i.[\![laughed]\!]\ y)\ (y+g) \\
&\quad =\lambda g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ \lambda y.[\![laughed]\!]\ y)
\end{aligned}
$$

Note that the computed meaning, $\lambda g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ \lambda y.[\![laughed]\!]\ y)$, once supplied with a context, yields the fully standard non-context sensitive meaning of this sentence. This is what we might expect, given that this sentence is in real life not sensitive to context.

We turn now to the sentence *every boy said he laughed*, with basic structure [[*every boy*] [*said* [*he laughed*]]].

1. $[\![he\ laughed]\!]^S$

$$
\begin{aligned}
&[\![he]\!]^S[\![laughed]\!]^S = (\lambda\mathcal{P}, g.\mathcal{P}\ \mathsf{sel}\ g)\ [\![laughed]\!]^S \\
&\quad =\lambda g.[\![laughed]\!]^S\ \mathsf{sel}\ g \\
&\quad =\lambda g.(\lambda X, h.[\![laughed]\!]\ (X\ h))\ \mathsf{sel}\ g \\
&\quad =\lambda g.(\lambda h.[\![laughed]\!]\ (\mathsf{sel}\ h))\ g \\
&\quad =\lambda g.[\![laughed]\!]\ (\mathsf{sel}\ g)
\end{aligned}
$$

2. $[\![said\ he\ laughed]\!]^S$

$$
\begin{aligned}
&[\![said]\!]^S[\![he\ laughed]\!]^S = (\lambda\Phi, X, g.[\![said]\!]\ (\Phi\ g)\ (X\ g))\ [\![he\ laughed]\!]^S \\
&\quad =(\lambda X, g.[\![said]\!]\ ([\![he\ laughed]\!]^S\ g)\ (X\ g)) \\
&\quad =(\lambda X, g.[\![said]\!]\ ((\lambda h.[\![laughed]\!]\ (\mathsf{sel}\ h))\ g)\ (X\ g)) \\
&\quad =(\lambda X, g.[\![said]\!]\ ([\![laughed]\!]\ (\mathsf{sel}\ g))\ (X\ g))
\end{aligned}
$$

3. $[\![every\ boy\ VP]\!]^S$

$$
\begin{aligned}
&[\![every\ boy]\!]^S[\![said\ he\ laughed]\!]^S \\
&\quad =(\lambda\mathcal{Q}, g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+g)))\ [\![said\ he\ laughed]\!]^S \\
&\quad =\lambda g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ (\lambda y.[\![said\ he\ laughed]\!]^S\ (\lambda h.y)\ (y+g)) \\
&\quad =\lambda g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ (\lambda y.(\lambda X, i.[\![said]\!]\ ([\![laughed]\!]\ (\mathsf{sel}\ i))\ (X\ g))\ (\lambda h.y)\ (y+g)) \\
&\quad =\lambda g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ (\lambda y.(\lambda i.[\![said]\!]\ ([\![laughed]\!]\ (\mathsf{sel}\ i))\ ((\lambda h.y)\ g))\ (y+g)) \\
&\quad =\lambda g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ (\lambda y.(\lambda i.[\![said]\!]\ ([\![laughed]\!]\ (\mathsf{sel}\ i))\ y)\ (y+g)) \\
&\quad =\lambda g.[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ (\lambda y.[\![said]\!]\ ([\![laughed]\!]\ (\mathsf{sel}\ (y+g)))\ y)
\end{aligned}
$$

The resulting meaning term contains an instance of sel, which requires that we determine the referent of the pronoun in context. Whatever the actual context might be, we see that the variable y is added to it, making it possible that the pronoun be resolved to y. In this case, the sentence would mean:

$$[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ (\lambda y.[\![said]\!]\ ([\![laughed]\!]\ y)\ y)$$

If the pronoun is resolved some other way, what we call the *free* (i.e. unbound) reading obtains. In a context $g = \{j\}$ where John is salient, the sentence could mean:

$$[\![every]\!]\ (\lambda y.[\![boy]\!]\ y)\ (\lambda y.[\![said]\!]\ ([\![laughed]\!]\ j)\ y)$$

4 Bindability

Binding is possible in this system iff a bound variable is retrieved a resolution operator. Bound variables are only ever inserted into contexts within the argument positions of determiners. Thus binding is only possible if a pronoun is in the semantic argument position of a determiner. If we use a LF syntax-semantics interface, the semantic arguments of a determiner are its NP as well as the S it raises to at LF. Considering only this second argument, we see that this S and everything in it is c-commanded by the DP. Thus the familiar c-command restriction on binding is actually derived from more basic semantic considerations.

We need to say something special to account for examples like the below, where it appears the Saxon genitive which binds the pronoun is contained within the subject DP.

- [[Every boy]'s mother] loves him.

Consider the following more detailed structural analysis of this sentence:

- [[Every boy]i ['s [mother ti]]] loves him

Here, *every boy* is base generated as an argument of *mother*, and then raises to the specifier of *s*. Let us assume the following basic semantic values for these lexical items:

LI	Basic Semantic Value	Type
s	the	(et)(et)t
mother	$\lambda x, y.\mathsf{mother}\ x\ y$	eet

As *mother* is not inherently context sensitive, its meaning can be automatically obtained as mentioned previously. However, the Saxon genitive *s*, as a determiner, requires the special context-sensitive determiner meaning discussed above. (I treat it here as synonymous with the word *the*, which isn't quite right, but is close enough.) Then the part of the DP below the *every boy* specifier has a meaning which can be computed as per the below.

1. $[\![[k[s\ [mother\ t_k]]]]\!]^S$

$$
\begin{aligned}
[\![[k[s\ [mother\ t_k]]]]\!]^S &= \lambda X_k.[\![s]\!]^S([\![mother]\!]^S\ X_k)\\
&=\lambda X_k.(\lambda \mathcal{P}, \mathcal{Q}, i.\mathsf{the}\ (\lambda y.\mathcal{P}\ (\lambda h.y)\ (y+i))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+i)))([\![mother]\!]^S\ X_k)\\
&=\lambda X_k.(\lambda \mathcal{Q}, i.\mathsf{the}\ (\lambda y.[\![mother]\!]^S\ X_k\ (\lambda h.y)\ (y+i))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+i)))\\
&=\lambda X_k, \mathcal{Q}, i.\mathsf{the}\ (\lambda y.[\![mother]\!]^S\ X_k\ (\lambda h.y)\ (y+i))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+i))\\
&=\lambda X_k, \mathcal{Q}, i.\mathsf{the}\ (\lambda y.(\lambda Y, Z, g.\mathsf{mother}\ (Y\ g)\ (Z\ g))\ X_k\ (\lambda h.y)\ (y+i))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+i))\\
&=\lambda X_k, \mathcal{Q}, i.\mathsf{the}\ (\lambda y.(\lambda Z, g.\mathsf{mother}\ (X_k\ g)\ (Z\ g))(\lambda h.y)\ (y+i))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+i))\\
&=\lambda X_k, \mathcal{Q}, i.\mathsf{the}\ (\lambda y.(\lambda g.\mathsf{mother}\ (X_k\ g)\ ((\lambda h.y)\ g))\ (y+i))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+i))\\
&=\lambda X_k, \mathcal{Q}, i.\mathsf{the}\ (\lambda y.(\lambda g.\mathsf{mother}\ (X_k\ g)\ y)\ (y+i))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+i))\\
&=\lambda X_k, \mathcal{Q}, i.\mathsf{the}\ (\lambda y.\mathsf{mother}\ (X_k\ g)\ y\ (y+i))\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+i))
\end{aligned}
$$

The semantic type of this term is $(\gamma e)((\gamma e)\gamma t)\gamma t$, or (abbreviating γe as E and γt as T) $E(ET)T$. This is not able to serve as an argument to the moved DP of type $(ET)T$. Let us change the term by flipping the order it takes its first two arguments in, as follows.

$$\lambda \mathcal{Q}, X_k, i.\mathsf{the}\ (\lambda y.\mathsf{mother}\ (X_k\ (y+i))\ y)\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+i))$$

This term can be combined with the moved DP by means of *function composition*:

$$
\begin{aligned}
&[\![\textit{every boy}]\!]^S \circ \texttt{flip}\ [\![[k[s\ [\textit{mother}\ t_k]]]]\!]^S \\
=&\lambda\mathcal{P}.[\![\textit{every boy}]\!]^S\ (\texttt{flip}\ [\![\textit{'s mother}]\!]^S\ \mathcal{P}) \\
=&\lambda\mathcal{P}.(\lambda\mathcal{Q}, g.[\![\textit{every}]\!]\ (\lambda y.[\![\textit{boy}]\!]\ y)\ (\lambda y.\mathcal{Q}\ (\lambda h.y)\ (y+g)))\ (\texttt{flip}\ [\![\textit{'s mother}]\!]^S\ \mathcal{P}) \\
=&\lambda\mathcal{P}.(\lambda g.[\![\textit{every}]\!]\ (\lambda y.[\![\textit{boy}]\!]\ y)\ (\lambda y.((\texttt{flip}\ [\![\textit{'s mother}]\!]^S\ \mathcal{P}))\ (\lambda h.y)\ (y+g))) \\
=&\lambda\mathcal{P}, g.[\![\textit{every}]\!]\ (\lambda y.[\![\textit{boy}]\!]\ y)\ (\lambda y.((\texttt{flip}\ [\![\textit{'s mother}]\!]^S\ \mathcal{P}))\ (\lambda h.y)\ (y+g)) \\
=&\lambda\mathcal{P}, g.[\![\textit{every}]\!]\ (\lambda y.[\![\textit{boy}]\!]\ y)\ (\lambda y.((\texttt{flip}\ (\lambda X_k, \mathcal{Q}, i.\mathsf{the}\ (\lambda z.\mathsf{mother}\ (X_k\ (z+i))\ z)\ (\lambda z.\mathcal{Q}\ (\lambda h.z)\ (z+i) \\
=&\lambda\mathcal{P}, g.[\![\textit{every}]\!]\ (\lambda y.[\![\textit{boy}]\!]\ y)\ (\lambda y.(((\lambda\mathcal{Q}, X_k, i.\mathsf{the}\ (\lambda z.\mathsf{mother}\ (X_k\ (z+i))\ z)\ (\lambda z.\mathcal{Q}\ (\lambda h.z)\ (z+i)))\ \mathcal{P})) \\
=&\lambda\mathcal{P}, g.[\![\textit{every}]\!]\ (\lambda y.[\![\textit{boy}]\!]\ y)\ (\lambda y.(\lambda X_k, i.\mathsf{the}\ (\lambda z.\mathsf{mother}\ (X_k\ (z+i))\ z)\ (\lambda z.\mathcal{P}\ (\lambda h.z)\ (z+i)))\ (\lambda h.y)\ (\\
=&\lambda\mathcal{P}, g.[\![\textit{every}]\!]\ (\lambda y.[\![\textit{boy}]\!]\ y)\ (\lambda y.(\lambda i.\mathsf{the}\ (\lambda z.\mathsf{mother}\ ((\lambda h.y)\ (z+i))\ z)\ (\lambda z.\mathcal{P}\ (\lambda h.z)\ (z+i)))\ (y+g)) \\
=&\lambda\mathcal{P}, g.[\![\textit{every}]\!]\ (\lambda y.[\![\textit{boy}]\!]\ y)\ (\lambda y.(\lambda i.\mathsf{the}\ (\lambda z.\mathsf{mother}\ y\ z)\ (\lambda z.\mathcal{P}\ (\lambda h.z)\ (z+i)))\ (y+g)) \\
=&\lambda\mathcal{P}, g.[\![\textit{every}]\!]\ (\lambda y.[\![\textit{boy}]\!]\ y)\ (\lambda y.\mathsf{the}\ (\lambda z.\mathsf{mother}\ y\ z)\ (\lambda z.\mathcal{P}\ (\lambda h.z)\ (z+(y+g))))
\end{aligned}
$$

We see that in this expression, which has the type of a (context sensitive) generalized quantifier, the VP will saturate the $\mathcal{P}$ argument position, thereby bringing any pronouns it may contain into a position suitable for retrieving the variable y from the context.