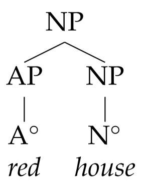
F16-Exercise

Course in Semantics · Ling 531 / 731
McKenzie · University of Kansas

1 Walkthrough

(1) *red house*

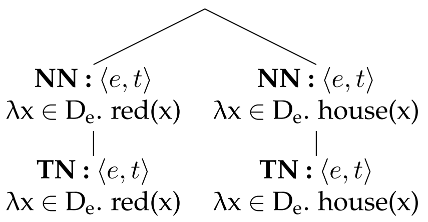
First the LF: The adjective projects to AP, which adjoins to the NP (2). We use the more recent theories of projection, which don't have $\overline{X}$ labels, but which generally do distinguish between medial and maximal XP projections (*e.g.*, Kayne 1994). So N° → NP can be one step if it's just the noun (like we saw), or more steps if there are adjuncts or specifiers.[1]

(2)


Then the lexical entries: Both *red* and *house* denote properties of individuals. Neither is a referring expression.

(3) a. $[\![\textit{red}]\!] = \lambda x \in D_e.\ \text{red}(x) : \langle e,t\rangle$ (char. function of $\{ x \in D_e \mid x \text{ is red} \}$)
b. $[\![\textit{house}]\!] = \lambda x \in D_e.\ \text{house}(x) : \langle e,t\rangle$ (char. function of $\{ x \in D_e \mid x \text{ is a house} \}$)

Composition. The lexical entries are the denotations of the respective syntactic heads. They are inserted using the Terminal Nodes rule, then carried upward by the Non-branching nodes rule.

(4)


We cannot employ functional application here. Why not? Check out the input types for each.

Check out the input type by looking at the leftmost λ-argument's domain condition. $[\![\textit{red}]\!]$'s domain is the set D_e, which we have defined as the set of individuals. So any object that *red* can be about is going to be an individual, of type e. So we can say that $[\![\textit{red}]\!]$'s input is of type e. We can start writing $[\![\textit{red}]\!]$'s type like this: $\langle e,$

(5) $$[\![\textit{red}]\!] = \lambda x \underbrace{\in D_e}_{\text{domain}} .\ \text{red}(x)$$

[1] Going even further, the development of "Bare phrase structure" (Chomsky 1995) does away with labels altogether, and makes no concrete distinction between heads and phrases; the latter are simply what results from merging new elements. Ultimately, under our model of the grammar, where the syntax feeds the semantics, none of these crucial syntactic issues matter for the ultimate result of the interpretation.

While we're at it, let's stop a moment and find the output type of $[\![\textit{red}]\!]$. We can find the output type of the function by ascertaining the type of everything after its leftmost λ-argument. In this case, there is only one λ-argument. After that is red(x), which we have adopted as an abbreviation of 1 if x is red, 0 if not. That is, the part after $\lambda x \in D_e$. denotes 1 or 0... these are the two truth-values. We named D_t as the set of truth-values, so $[\![\textit{red}]\!]$'s output is in D_t. Truth-values are of type t. So, the output of $[\![\textit{red}]\!]$ is of type t. We can finish writing the type of $[\![\textit{red}]\!]$: $\langle e, t \rangle$

The same method applies to $[\![\textit{house}]\!]$.

Now, coming back to our input— the input to $[\![\textit{red}]\!]$ needs to be of type e. If something's not of type e, we can't plug it in to $[\![\textit{red}]\!]$. In essence, if it isn't an individual, it can't have the property of being red (or the property of not being red). Since $[\![\textit{house}]\!]$ is of type $\langle e, t \rangle$, we can't plug it into $[\![\textit{red}]\!]$. And *vice versa*.

To get around this block, and to account for the other observed features of modification, we employ a new rule of composition, called Predicate Modification.

Rule 4:

Predicate Modification (PM):
If α is a branching node, whose daughters are $\{\beta, \gamma\}$, and if both $[\![\beta]\!]$ and $[\![\gamma]\!]$ are of type $\langle \sigma, \tau \rangle$, then
$[\![\alpha]\!] = \lambda a \in D_\sigma.\ [\![\beta]\!](a) = 1 \ \&\ [\![\gamma]\!](a) = 1$

Note: this varies slightly from the rule in the lecturelet. That one had a typo, which is corrected here.

In tree form, this rule looks like this:

$$\begin{tikzcd}
& \lambda a \in D_\sigma.\ [\![\beta]\!](a) = 1 \ \&\ [\![\gamma]\!](a) = 1 : \langle \sigma, \tau \rangle \arrow[dl, no head] \arrow[dr, no head] & \\
[\![\beta]\!] : \langle \sigma, \tau \rangle & & [\![\gamma]\!] : \langle \sigma, \tau \rangle
\end{tikzcd}$$

Let's replace this abstract thing with our tree. Since $[\![\beta]\!] = \lambda x \in D_e.\ \text{red}(x)$, we can replace the one with the other.

$$\begin{tikzcd}
& \begin{array}{c}\textbf{PM :}\ \langle e, t \rangle \\ \lambda a \in D_\sigma.\ [\![\beta]\!](a) = 1 \ \&\ [\![\gamma]\!](a) = 1\end{array} \arrow[dl, no head] \arrow[dr, no head] & \\
\begin{array}{c}\textbf{NN :}\ \langle e, t \rangle \\ \lambda x \in D_e.\ \text{red}(x)\end{array} & & [\![\gamma]\!] : \langle \sigma, \tau \rangle
\end{tikzcd}$$

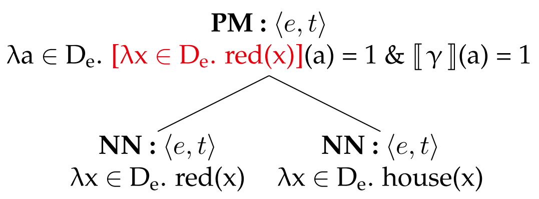
Same goes for $[\![\gamma]\!]$ and $\lambda x \in D_e.\ \text{house}(x)$.

$$\begin{tikzcd}
& \begin{array}{c}\textbf{PM :}\ \langle e, t \rangle \\ \lambda a \in D_\sigma.\ [\![\beta]\!](a) = 1 \ \&\ [\![\gamma]\!](a) = 1\end{array} \arrow[dl, no head] \arrow[dr, no head] & \\
\begin{array}{c}\textbf{NN :}\ \langle e, t \rangle \\ \lambda x \in D_e.\ \text{red}(x)\end{array} & & \begin{array}{c}\textbf{NN :}\ \langle e, t \rangle \\ \lambda x \in D_e.\ \text{house}(x)\end{array}
\end{tikzcd}$$

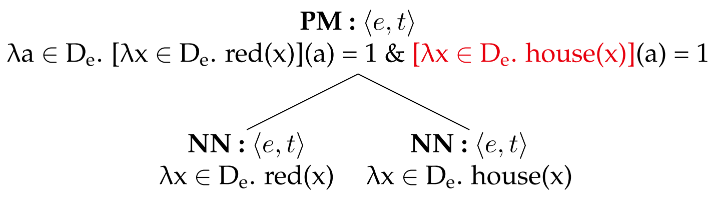
Now for the mother node. Change the domain condition so it's the same as that of the daughters.

$$\begin{tikzcd}
& \begin{array}{c}\textbf{PM :}\ \langle e, t \rangle \\ \lambda a \in D_e.\ [\![\beta]\!](a) = 1 \ \&\ [\![\gamma]\!](a) = 1\end{array} \arrow[dl, no head] \arrow[dr, no head] & \\
\begin{array}{c}\textbf{NN :}\ \langle e, t \rangle \\ \lambda a \in D_e.\ \text{red}(x)\end{array} & & \begin{array}{c}\textbf{NN :}\ \langle e, t \rangle \\ \lambda a \in D_e.\ \text{house}(x)\end{array}
\end{tikzcd}$$

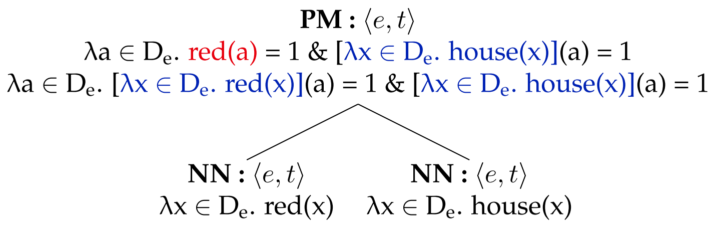
We can replace $[\![\beta]\!]$ again in the mother node.


PM : $\langle e,t\rangle$
λa ∈ D_e. [λx ∈ D_e. red(x)](a) = 1 & $[\![\gamma]\!]$(a) = 1
NN : $\langle e,t\rangle$
λx ∈ D_e. red(x)
NN : $\langle e,t\rangle$
λx ∈ D_e. house(x)

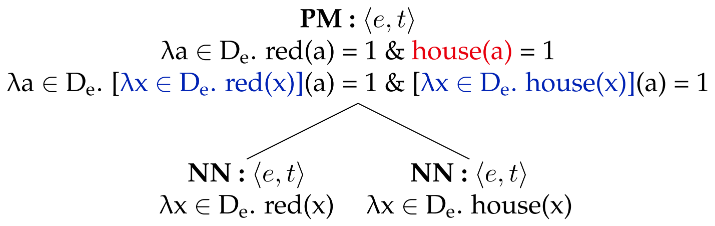
We can also replace $[\![\gamma]\!]$ again in the mother node.


PM : $\langle e,t\rangle$
λa ∈ D_e. [λx ∈ D_e. red(x)](a) = 1 & [λx ∈ D_e. house(x)](a) = 1
NN : $\langle e,t\rangle$
λx ∈ D_e. red(x)
NN : $\langle e,t\rangle$
λx ∈ D_e. house(x)

Now, we have two λ-function/argument structures that we can reduce. For $[\![\textit{red}]\!]$, plug in a for every instance of x, and *voilà*!


PM : $\langle e,t\rangle$
λa ∈ D_e. red(a) = 1 & [λx ∈ D_e. house(x)](a) = 1
λa ∈ D_e. [λx ∈ D_e. red(x)](a) = 1 & [λx ∈ D_e. house(x)](a) = 1
NN : $\langle e,t\rangle$
λx ∈ D_e. red(x)
NN : $\langle e,t\rangle$
λx ∈ D_e. house(x)

For $[\![\textit{house}]\!]$, do the same: plug in a for every instance of x, and *voilà*!


PM : $\langle e,t\rangle$
λa ∈ D_e. red(a) = 1 & house(a) = 1
λa ∈ D_e. [λx ∈ D_e. red(x)](a) = 1 & [λx ∈ D_e. house(x)](a) = 1
NN : $\langle e,t\rangle$
λx ∈ D_e. red(x)
NN : $\langle e,t\rangle$
λx ∈ D_e. house(x)

And there you have it. $[\![\textit{red house}]\!]$ = λa ∈ D_e. red(a) = 1 & house(a) = 1

Recall that we can use any variable we like; here we've used a, but most people will still use x, y, z.

2 Exercise

Now you try it!

(1) *happy dog*

Now, plug your NP into a DP.

(2) *the happy dog*

Try a few more NPs

(3) *persnickety neighbor*

(4) *enormous victory*

(5) *trifling matter*

(6) *big, dumb ox*