We now know the denotations of proper names, intransitive and transitive verbs and connectives. In this lecture, we will discuss the denotations of nouns and adjectives. Before that, there are several caveats:

From now on, we will not be explicit about exactly how the syntax works. So we will not formulate syntactic rules that generate nouns and adjectives. Instead, we'll just assume structures that are reasonable, according to modern syntactic theory. However, we will also often simplify structures by omitting non-branching nodes. This is harmless, because by assumption, non-branching nodes do not change the denotations. If syntacticians found what we were doing and told us to put the non-branching nodes back, we could always do so without changing our semantic analysis substantially (although these days, many syntacticians are also skeptical about non-branching nodes, so they probably won't tell us to have them). Also, we will often omit the syntactic labels like VP, NP, S, etc., because they play no semantic role in our semantic theory. That is, all we need to know is the hierarchical structure, and indeed, the compositional rules do not mention syntactic labels (e.g. we don't have rules that say 'if A is a VP and B is a DP, ...').

Also, we will not be explicit about how many verbs, nouns and adjectives there are in the lexicon. As should be clear by now, our analysis of a small set of intransitive and transitive verbs is generalisable to any other intransitive and transitive verbs. And as we will see below, this is also the case for nouns and adjectives.

1 Nouns

What do nouns denote? The first thing to notice is that just like verbs, nouns can be used to describe properties that people have. For example, compare the following two sentences.

(1) a. John smokes.
    b. John is a smoker.

These two sentences mean very similar things. We have an analysis of the first sentence (1a), according to which 'smokes' denotes a function of type $\langle e, t\rangle$. More specifically:

(2) For any model M,
$[\![\text{smokes}]\!]^M = \lambda x \in D_e.\ 1 \text{ iff } x \text{ smokes in } M$

By analogy, let us assume that the noun 'smoker' in (1b) denotes the following function of type $\langle e, t\rangle$.

(3) For any model M,
$[\![\text{smoker}]\!]^M = \lambda x \in D_e.\ 1 \text{ iff } x \text{ is a smoker in } M$

If the sentence (1b) looked like 'John smoker', the analysis would be very easy. Bu there are words in (1b) that do not occur in (1a), namely, 'is' and 'a'.[1] We need to say something about them.

[1]However, in a number of languages, the sentence corresponding to (1b) in fact looks like 'John smoker'. English is, for some reason, not one of them.

1.1 Semantic Vacuity

Let us pursue the idea that 'is' and 'a' are **semantically vacuous**, which is another way of saying that they mean nothing. What does it mean to mean nothing? Here, we equate meaning nothing to denoting an **identity function**.

An identity function is a function that returns the exact same thing as the input. Identity functions come in many different semantic types, but since the input and the output are always the same, they all have a semantic type that looks like $\langle \tau, \tau \rangle$. The one we need in the analysis of (1b) is of type $\langle et, et \rangle$ (recall et is a shorthand for $\langle e, t \rangle$). Specifically, we assign the following denotations to 'is' and 'a'.

(4) For any model M,

a. $[\![\text{is}]\!]^M = [\lambda f_{\langle e,t \rangle}.\ f]$

b. $[\![\text{a}]\!]^M = [\lambda f_{\langle e,t \rangle}.\ f]$

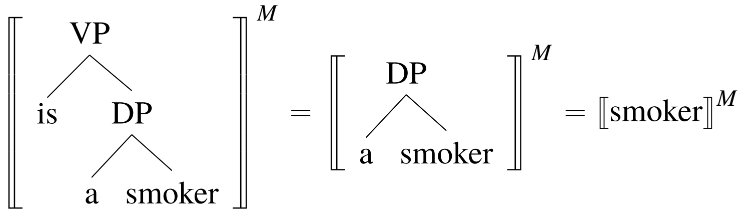
They have the same identity function as their denotations. They take a function of type $\langle e, t \rangle$ and return the same function. You will see why this leads to 'mean nothing', when you compute the denotations. First, let's compute $[\![\text{a smoker}]\!]^M$.

$$\begin{aligned}
[\![\text{a smoker}]\!]^M &= [\![\text{a}]\!]^M([\![\text{smoker}]\!]^M) && \text{(BNR)}\\
&= [\lambda f_{\langle e,t \rangle}.\ f]([\![\text{smoker}]\!]^M) && \text{(Lexicon)}\\
&= [\lambda f_{\langle e,t \rangle}.\ f](\lambda x \in D_e.\ 1 \text{ iff } x \text{ is a smoker in } M) && \text{(Lexicon)}\\
&= [\lambda x \in D_e.\ 1 \text{ iff } x \text{ is a smoker in } M] && \text{(λ-conversion)}\\
&= [\![\text{smoker}]\!]^M
\end{aligned}$$

Note that you could go directly from the second line to the final line, because $[\![\text{smokes}]\!]^M$ is of type $\langle e, t \rangle$ and $[\lambda f_{\langle e,t \rangle}.\ f]$ is an identity function over type-$\langle e, t \rangle$ functions. Importantly, the above computation proves that $[\![\text{a smoker}]\!]^M = [\![\text{smokes}]\!]^M$. Now, let's compute $[\![\text{is a smoker}]\!]^M$.

$$\begin{aligned}
[\![\text{is a smoker}]\!]^M &= [\![\text{is}]\!]^M([\![\text{a smoker}]\!]^M) && \text{(BNR)}\\
&= [\lambda f_{\langle e,t \rangle}.\ f]([\![\text{a smoker}]\!]^M) && \text{(Lexicon)}\\
&= [\lambda f_{\langle e,t \rangle}.\ f]([\![\text{smoker}]\!]^M) && \text{(above computation)}\\
&= [\![\text{smoker}]\!]^M && \text{(λ-conversion)}
\end{aligned}$$

These computations demonstrate the following equality.



Therefore, the words 'a' and 'is', although syntactically and phonologically real, have no role to play in the compositional semantics. They just pass the meaning of their sister constituent up to the next level, without changing it. We'll encounter some more semantically vacuous words in this course. They all denote identity functions, although possibly of different semantic types.

1.2 Transitive Nouns

Nouns like ‘smoker’ are intransitive nouns, in the sense that they only take one argument. They are similar in meaning to intransitive verbs like ‘smokes’, which also only take one argument. Just like there are transitive verbs like ‘studies’, there are transitive nouns too. For example, ‘student’, as in (5b), takes two arguments, Mary and linguistics.

(5) a. Mary studies linguistics.
  b. Mary is a student of linguistics.

According to the analysis we developed in the previous lecture, the transitive verb ‘studies’ can be given the following function of type $\langle e, et\rangle$ as its denotation (again, recall $et = \langle e, t\rangle$, so $\langle e, \langle e, t\rangle\rangle$).

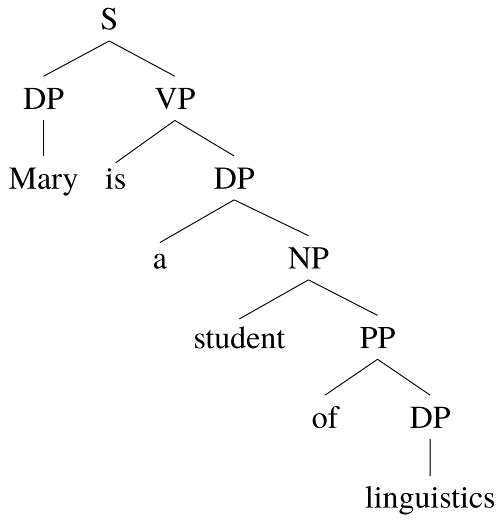
(6) For any model M,
$[\![\text{studies}]\!]^M = [\lambda x \in D_e.\ [\lambda y \in D_e.\ 1 \text{ iff } y \text{ studies } x \text{ in } M]]$

Here we assume (without argument) that ‘linguistics’ is a kind of proper name and denotes an individual. Of course the denotation of ‘linguistics’ is not a concrete person or object, but a more abstract kind of individual. What exactly such abstract individuals might be is an important question, but to simplify the discussion here, let’s not dwell on this issue.

By the same token, we analyze transitive nouns as denoting functions of type $\langle e, et\rangle$, i.e.

(7) For any model M,
$[\![\text{student}]\!]^M = [\lambda x \in D_e.\ [\lambda y \in D_e.\ 1 \text{ iff } y \text{ is a student of } x \text{ in } M]]$

Again there are a couple of words in (5b) that do not occur in (5a), namely, ‘is’, ‘a’ and ‘of’. We discussed ‘is’ and ‘a’ above and analyzed them as semantically vacuous. It seems that ‘of’ is also a kind of word that has no meaning. So let us analyze it as semantically vacuous too. However, it’s semantic type should be different from the semantic type of ‘is’ and ‘a’. To see this, consider the following tree diagram.

(8)


By assumption ‘linguistics’ denotes an individual, so $[\![\text{linguistics}]\!]^M$ is of type e. Then, in order to combine $[\![\text{of}]\!]^M$ and $[\![\text{linguistics}]\!]^M$ via the Branching Node Rule, $[\![\text{of}]\!]^M$ needs to be a kind of function that can take $[\![\text{linguistics}]\!]^M$ as its argument. This means its semantic type will look like $\langle e, \_\ \rangle$. Now, we are analyzing $[\![\text{of}]\!]^M$ to be semantically vacuous, which is to say it denotes an identity function. Since an identity function simply returns the input, its semantic type will be $\langle e, e \rangle$. Therefore, the denotation of 'of' is as in (9).

(9) For any model M,
$[\![\text{of}]\!]^M = [\lambda x \in D_e.\ x]$

1.3 Implicit Arguments

You might be wondering what happens in sentences like (10).

(10) Mary is a student.

This sentence does not explicitly say what Mary is a student of. There are a couple of analytic possibilities here.

One possibility is that there actually are intransitive and transitive versions of the noun 'student', and what occurs in (10) is the intransitive version, unlike the transitive version used in (5b). The denotations of these two nouns can be written as follows.

(11) a. $[\![\text{student}_{\text{intr}}]\!]^M = [\lambda y \in D_e.\ 1 \text{ iff } y \text{ is a student in } M]$
b. $[\![\text{student}_{\text{tr}}]\!]^M = [\lambda x \in D_e.\ [\lambda y \in D_e.\ 1 \text{ iff } y \text{ is a student of } x \text{ in } M]]$

Another possibility is that one of the meanings is the basic one (perhaps the transitive use, because it's more complex), and the other use is derived by some general mechanism. Notice that some verbs show similar transitive-intransitive alternation.

(12) a. Mary studied at UCL.
b. Mary studied linguistics at UCL.

(13) a. John ate.
b. John ate pizza.

That transitive-intransitive alternation is observed widely suggests that there indeed is some general mechanism behind it. This is a very interesting topic, but it is beyond the scope of this course. We leave this issue open.

1.4 Summary: Nouns

To summarize, we analyzed nouns on a par with verbs. Just like there are intransitive and transitive verbs, there are intransitive and transitive nouns. Intransitive verbs and intransitive nouns denote functions of type $\langle e, t \rangle$.

(14) For any model M,
a. $[\![\text{smokes}]\!]^M = \lambda x \in D_e.\ 1 \text{ iff } x \text{ smokes in } M$
b. $[\![\text{smoker}]\!]^M = \lambda x \in D_e.\ 1 \text{ iff } x \text{ is a smoker in } M$

It is easy to come up with similar lexical entries for other intransitive nouns.

(15) For any model M,
a. $[\![\text{boy}]\!]^M = \lambda x \in D_e.\ 1 \text{ iff } x \text{ is a boy in } M$
b. $[\![\text{linguist}]\!]^M = \lambda x \in D_e.\ 1 \text{ iff } x \text{ is a linguist in } M$
c. $[\![\text{table}]\!]^M = \lambda x \in D_e.\ 1 \text{ iff } x \text{ is a table in } M$
d. $[\![\text{city}]\!]^M = \lambda x \in D_e.\ 1 \text{ iff } x \text{ is a city in } M$

Transitive verbs and transitive nouns denote functions of type $\langle e, et\rangle$.

(16) For any model M,

a. $[\![\text{studies}]\!]^M = [\lambda x \in D_e.\ [\lambda y \in D_e.1 \text{ iff } y \text{ studies } x \text{ in } M]]$

b. $[\![\text{student}]\!]^M = [\lambda x \in D_e.\ [\lambda y \in D_e.\ 1 \text{ iff } y \text{ is a student of } x \text{ in } M]]$

It is routine to generalise this analysis to other transitive nouns (can you come up with more transitive nouns?).

Sentences containing nouns often contain 'is', 'a' and 'of'. We analyzed them as semantically vacuous, meaning they denote identity functions. Identity functions simply return their inputs without changing them.

(17) For any model M,

a. $[\![\text{is}]\!]^M = [\lambda f_{\langle e,t\rangle}.\ f]$

b. $[\![\text{a}]\!]^M = [\lambda f_{\langle e,t\rangle}.\ f]$

c. $[\![\text{of}]\!]^M = [\lambda x \in D_e.\ x]$

2 Adjectives

Just like nouns and verbs, adjectives can be used to talk about properties of individuals. They also have the intransitive-transitive distinction, as demonstrated by (18).

(18) a. John is blond.

b. John is fond of Mary.

As for nouns and verbs, we analyze intransitive adjectives to denote type-$\langle e, t\rangle$ functions and transitive adjectives to denote type-$\langle e, et\rangle$ functions.

(19) For any model M,

a. $[\![\text{blond}]\!]^M = [\lambda x \in D_e.1 \text{ iff } x \text{ is blond in } M]$

b. $[\![\text{fond}]\!]^M = [\lambda x \in D_e.\ [\lambda y \in D_e.\ 1 \text{ iff } y \text{ is fond of } x \text{ in } M]]$

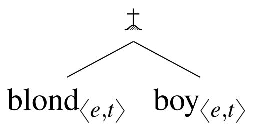
As before, 'is' and 'of' are semantically vacuous. With these assumptions, we can easily compute the denotations of the sentences in (18) (which is left as an exercise for you).

3 Modification

In the above examples, adjectives function as predicates, but importantly, adjectives can also function as modifiers, as in (20).

(20) John is a blond boy.

How is this possible? Notice that if both $[\![\text{blond}]\!]^M$ and $[\![\text{boy}]\!]^M$ are functions of type $\langle e, t\rangle$, we cannot use our Branching Node Rule, because neither of them are of the right semantic type to serve as the argument of the other! Such a situation is called a **type mismatch**.



In what follows, we will discuss two ways to account for the modifier use of adjectives like 'blond' in sentences like (20).

3.1 Lexical Ambiguity

The first approach is to postulate a separate lexical entry for the modifier use of ‘blond’. To remind you, the predicational use is:

(21) For any model M,
$[\![\text{blond}_{\text{pred}}]\!]^M = [\lambda x \in D_e.1 \text{ iff } x \text{ is blond in } M]$

This is a function of type $\langle e, t\rangle$ and used in sentences like (22).

(22) John is blond$_{\text{pred}}$.

For the modifier use, as in (23), we need a separate lexical entry, as ‘blond$_{\text{pred}}$’ would result in a type-mismatch in this context.

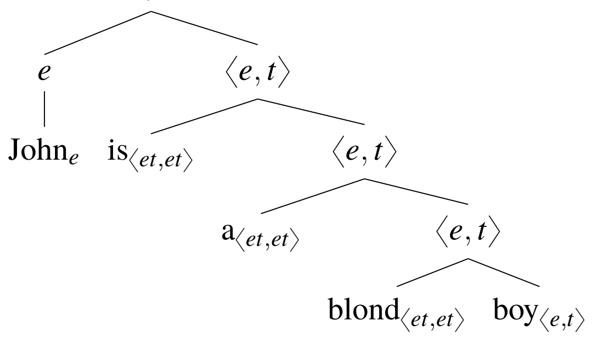
(23) John is a blond$_{\text{mod}}$ boy.

As usual, we ask this question first: what semantic type should the denotation of ‘blond$_{\text{mod}}$’ be? Since $[\![\text{blond}_{\text{mod}}]\!]^M$ needs to combine with $[\![\text{boy}]\!]^M$, which is a function of type $\langle e, t\rangle$, it needs to be either (i) a kind of thing that can be the argument of $[\![\text{boy}]\!]^M$, or (ii) a function that can take $[\![\text{boy}]\!]^M$ as its argument. Let’s explore both possibilities.

(i) Since $[\![\text{boy}]\!]^M$ is a function of type $\langle e, t\rangle$, we can apply it to $[\![\text{blond}_{\text{mod}}]\!]^M$, if $[\![\text{blond}_{\text{mod}}]\!]^M$ is an individual. But this seems strange. What kind of individual should it be? Unlike for proper names like ‘John’, ‘Mary’, etc., it is not clear what ‘blond’ should refer to. Moreover, even if it were an individual, the meaning of $[\![\text{blond}_{\text{mod}} \text{ boy}]\!]^M$ would be $[\![\text{boy}]\!]^M([\![\text{blond}_{\text{mod}}]\!]^M)$, which is a truth-value!! Then it wouldn’t be able to combine with ‘is’ and ‘John’. So this is a dead-end.

(ii) The second possibility is more promising: $[\![\text{blond}_{\text{mod}}]\!]^M$ is a function that takes $[\![\text{boy}]\!]^M$ as its argument. Since the latter is of type $\langle e, t\rangle$, the semantic type of $[\![\text{blond}_{\text{mod}}]\!]^M$ should look like $\langle et, \_\_ \rangle$. In order for the rest of the composition to go through, the output also has to be of type $\langle e, t\rangle$. This is illustrated by the following tree diagram with semantic types.

(24)


t
e
$\langle e, t\rangle$
John$_e$
is$_{\langle et,et\rangle}$
$\langle e, t\rangle$
a$_{\langle et,et\rangle}$
$\langle e, t\rangle$
blond$_{\langle et,et\rangle}$
boy$_{\langle e,t\rangle}$

So $[\![\text{blond}_{\text{mod}}]\!]^M$ is of type $\langle et, et\rangle$, which means, its denotation looks like:

$$[\![\text{blond}_{\text{mod}}]\!]^M = [\lambda f_{\langle e,t\rangle}.\ [\lambda x \in D_e.\ 1 \text{ iff } ???]]$$

It takes a function of type $\langle e, t\rangle$ and returns another function of type $\langle e, t\rangle$. Because ‘blond’ is not a semantically vacuous word, it shouldn’t return the same function as the input. Rather, it should somehow add the meaning of ‘blond’. But how?

To answer this question, let’s contemplate what the result of the composition should be, i.e. $[\![\text{blond}_{\text{mod}} \text{ boy}]\!]^M$. In particular, we want to know how this is different $[\![\text{boy}]\!]^M$. Firstly, you should notice that $\llbracket \text{blond}_{\text{mod}}\ \text{boy} \rrbracket^M$ can be analyzed on a par with $\llbracket \text{boy} \rrbracket^M$, as a function of type $\langle e, t \rangle$. That is:

(25) For any model M,
$\llbracket \text{blond}_{\text{mod}}\ \text{boy} \rrbracket^M = [\lambda x \in D_e.\ 1 \text{ iff } x \text{ is a blond boy in } M]$

This is a function that maps everybody who is a blond boy to 1. Now recall that $\llbracket \text{boy} \rrbracket^M$ maps everybody who is a boy to 1.

(26) For any model M,
$\llbracket \text{boy} \rrbracket^M = [\lambda x \in D_e.\ 1 \text{ iff } x \text{ is a boy in } M]$

There is a particular relation between these two functions, namely, every individual who $\llbracket \text{blond boy} \rrbracket^M$ maps to 1 is a boy, so $\llbracket \text{boy} \rrbracket^M$ maps them to 1 as well. In fact, $\llbracket \text{blond boy} \rrbracket^M$ maps to 1 a subset of individuals that $\llbracket \text{boy} \rrbracket^M$ maps to 1, namely, those that are blond. The function of the adjectival modifier 'blond', therefore, can be thought of as restricting the meaning of 'boy' to the blond ones. We can represent this more explicitly by re-writing (25) as the final line of (27):

(27) For any model M,

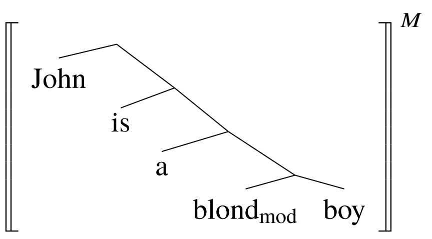
$$\begin{aligned}\llbracket \text{blond}_{\text{mod}}\ \text{boy} \rrbracket^M &= [\lambda x \in D_e.\ 1 \text{ iff } x \text{ is a blond boy in } M]\\ &= [\lambda x \in D_e.\ 1 \text{ iff } x \text{ is blond in } M \text{ and is a boy in } M]\\ &= [\lambda x \in D_e.\ 1 \text{ iff } x \text{ is blond in } M \text{ and } \llbracket \text{boy} \rrbracket^M(x) = 1]\end{aligned}$$

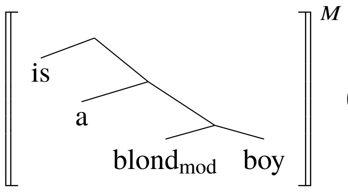
The final line clearly shows that $\llbracket \text{blond boy} \rrbracket^M$ is only true of those individuals that $\llbracket \text{boy} \rrbracket^M$ maps to 1 and also are blond in M.

Now, to arrive at the denotation of 'blond$_{\text{mod}}$' alone, all we need to know is what part of the meaning of (27) is coming from $\llbracket \text{boy} \rrbracket^M$. The rest is going to be the meaning of 'blond$_{\text{mod}}$'. It is quite clear. Replacing $\llbracket \text{boy} \rrbracket^M$ in (27) with a variable f, we get:

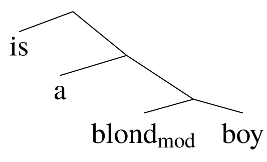
(28) For any model M,
$\llbracket \text{blond}_{\text{mod}} \rrbracket^M = [\lambda f \in D_{\langle e,t \rangle}.\ [\lambda x \in D_e.\ 1 \text{ iff } x \text{ is blond in } M \text{ and } f(x) = 1]]$

This denotation derives the correct meaning for sentences like 'John is a blond boy', as illustrated by a top-down computation (let's assume $\llbracket \text{John} \rrbracket^M = j$). (As mentioned at the beginning, the non-branching nodes and syntactic lables are omitted, because we know they don't change the denotations.)

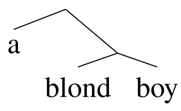


$=$ 

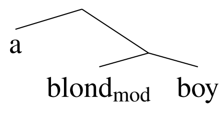
$(\llbracket \text{John} \rrbracket^M)$ (BNR)

$= [\![$ 

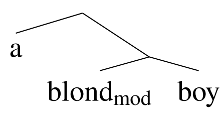
$]\!]^M (j)$ (Lexicon)

$= [\![\text{is}]\!]^M \Big([\![$ 

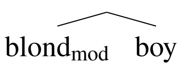
$]\!]^M \Big)(j)$ (BNR)

$= [\lambda f \in D_{\langle e,t\rangle}. f] \Big([\![$ 

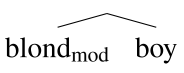
$]\!]^M \Big)(j)$ (Lexicon)

$= [\![$ 

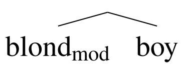
$]\!]^M (j)$ (λ-conv.)

$= [\![\text{a}]\!]^M \Big([\![$ 

$]\!]^M \Big)(j)$ (BNR)

$= [\lambda f \in D_{\langle e,t\rangle}. f] \Big([\![$ 

$]\!]^M \Big)(j)$ (Lexicon)

$= [\![$ 

$]\!]^M (j)$ (λ-conv.)

$= [\![\text{blond}_{\text{mod}}]\!]^M ([\![\text{boy}]\!]^M)(j)$ (BNR)

$= [\lambda f_{\langle e,t\rangle}. [\lambda x \in D_e. \text{ 1 iff } x \text{ is blond in } M \text{ and } f(x) = 1]]([\![\text{boy}]\!]^M)(j)$ (Lexicon)

$= [\lambda x \in D_e. \text{ 1 iff } x \text{ is blond in } M \text{ and } [\![\text{boy}]\!]^M(x) = 1](j)$ (λ-conv.)

$= 1 \text{ iff } j \text{ is blond in } M \text{ and } [\![\text{boy}]\!]^M(j) = 1$ (λ-conv.)

$= 1 \text{ iff } j \text{ is blond in } M \text{ and } [\lambda x \in D_e. \text{ 1 iff } x \text{ is a boy in } M](j) = 1$ (Lexicon)

$= 1 \text{ iff } j \text{ is blond in } M \text{ and } x \text{ is a boy in } M$ (λ-conv.)

The same denotation (28) will work when the noun is different, e.g. 'blond$_{\text{mod}}$ girl'.

3.2 A New Compositional Rule

In the above analysis, we postulated two lexical entries for the same word 'blond'. You might think that this is not a very attractive analysis, because you need to duplicate entries of lots of other adjectives as well, e.g. 'bald', 'Japanese', 'green', etc, because they also have a predicative and modificational use.

So let us consider an alternative analysis that avoids this redundancy. We will assume a single lexical entry for each adjective. Specifically, we take 'blond' to always denote the function of type $\langle e,t\rangle$ in (29).

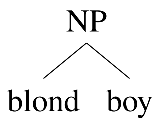
(29) For any model M,
$[\![\text{blond}]\!]^M = [\lambda x \in D_e. \text{ 1 iff } x \text{ is blond in } M]$

Then, how do we combine this with $[\![\text{boy}]\!]^M$, which is also of type $\langle e,t\rangle$? To this end, we postulate a new compositional rule that combines two type-$\langle e,t\rangle$ functions. Let's call this rule

Predicate Modification (following Heim & Kratzer 1998).

(30) *Predicate Modification*: For any model M,
if A is a branching node with B and C as its daughters and $[\![\text{B}]\!]^M$ and $[\![\text{C}]\!]^M$ are both of type $\langle e,t\rangle$, then $[\![\text{A}]\!]^M = [\lambda x \in D_e.\ 1 \text{ iff } [\![\text{B}]\!]^M(x) = 1 \text{ and } [\![\text{C}]\!]^M(x) = 1]$.

As we see now, this rule allows us to combine $[\![\text{blond}]\!]^M$ and $[\![\text{boy}]\!]^M$ and obtain the same result as in the previous analysis. Consider the constituent:



This is a branching node that has two type-$\langle e,t\rangle$ daughters, so Predicate Modification can apply to it. Predicate Modification says that the meaning of this whole constituent will be $[\lambda x \in D_e.\ 1 \text{ iff } [\![\text{blond}]\!]^M(x) = 1 \text{ and } [\![\text{boy}]\!]^M(x) = 1]$. We can now look up the lexicon and perform several λ-conversions, as demonstrated below. To avoid confusion, I will change the variable names each time.

$$\begin{aligned}
&[\lambda x \in D_e.\ 1 \text{ iff } [\![\text{blond}]\!]^M(x) = 1 \text{ and } [\![\text{boy}]\!]^M(x) = 1] \\
&= [\lambda x \in D_e.\ 1 \text{ iff } [\lambda y \in D_e.\ 1 \text{ iff } y \text{ is blond in } M](x) = 1 \text{ and } [\![\text{boy}]\!]^M(x) = 1] && \text{(Lexicon)} \\
&= [\lambda x \in D_e.\ 1 \text{ iff } x \text{ is blond in } M \text{ and } [\![\text{boy}]\!]^M(x) = 1] && (\lambda\text{-conv.}) \\
&= [\lambda x \in D_e.\ 1 \text{ iff } x \text{ is blond in } M \text{ and } [\lambda z \in D_e.\ 1 \text{ iff } z \text{ is boy in } M](x) = 1] && \text{(Lexicon)} \\
&= [\lambda x \in D_e.\ 1 \text{ iff } x \text{ is blond in } M \text{ and } x \text{ is boy in } M] && (\lambda\text{-conv.})
\end{aligned}$$

This is exactly the denotation we want for $[\![\text{blond boy}]\!]^M$.

Notice that the new compositional rule, Predicate Modification, is a rule that applies to branching nodes. So it is not appropriate to call the old branching node rule the Branching Node Rule. Also, the old rule should be stated explicitly to exclude situations like ‘blond boy’ from its intended domain of application. From now on, we call it *Functional Application* and re-formulate it as follows.

(31) *Functional Application*: For any model M,
if A is a branching node with B and C as its daughters and $[\![\text{B}]\!]^M$ is of type $\langle \sigma,\tau\rangle$ and $[\![\text{C}]\!]^M$ is of type σ, then $[\![\text{A}]\!]^M = [\![\text{B}]\!]^M([\![\text{C}]\!]^M)$.

Since this rule does not mention the linear order between B and C, it can apply to situations where the function-denoting constituent B occurs to the left or to the right of C. Note that which of Functional Application and Predicate Modification should be used is solely determined by the semantic types of the daughter constituents.

3.3 Which Analysis Is Better?

So we have two analyses that both derive correct denotations for $[\![\text{blond boy}]\!]^M$. In the lecture we discuss some facts that might favour one over the other, but we will not be able to decide between them. For the rest of this course, we will assume the second approach with a new compositional rule.

4 Summary

Intransitive verbs, nouns and adjectives.

(32) For any model M,

a. $[\![\text{smokes}]\!]^M = \lambda x \in D_e.\ 1$ iff x smokes in M
b. $[\![\text{smoker}]\!]^M = \lambda x \in D_e.\ 1$ iff x is a smoker in M
c. $[\![\text{blond}]\!]^M = \lambda x \in D_e.\ 1$ iff x is blond in M

Transitive verbs, nouns and adjectives.

(33) For any model M,

a. $[\![\text{studies}]\!]^M = [\lambda x \in D_e.\ [\lambda y \in D_e.1$ iff y studies x in $M]]$
b. $[\![\text{student}]\!]^M = [\lambda x \in D_e.\ [\lambda y \in D_e.\ 1$ iff y is a student of x in $M]]$
c. $[\![\text{fond}]\!]^M = [\lambda x \in D_e.\ [\lambda y \in D_e.\ 1$ iff y is fond of x in $M]]$

Semantically vacuous words:

(34) For any model M,

a. $[\![\text{is}]\!]^M = [\lambda f_{\langle e,t\rangle}.\ f]$
b. $[\![\text{a}]\!]^M = [\lambda f_{\langle e,t\rangle}.\ f]$
c. $[\![\text{of}]\!]^M = [\lambda x \in D_e.\ x]$

Compositional rules:

(35) *Functional Application*: For any model M,
if A is a branching node with B and C as its daughters and $[\![\text{B}]\!]^M$ is of type $\langle \sigma, \tau\rangle$ and $[\![\text{C}]\!]^M$ is of type σ, then $[\![\text{A}]\!]^M = [\![\text{B}]\!]^M([\![\text{C}]\!]^M)$.

(36) *Predicate Modification*: For any model M,
if A is a branching node with B and C as its daughters and $[\![\text{B}]\!]^M$ and $[\![\text{C}]\!]^M$ are both of type $\langle e, t\rangle$, then $[\![\text{A}]\!]^M = [\lambda x \in D_e.\ 1$ iff $[\![\text{B}]\!]^M(x) = 1$ and $[\![\text{C}]\!]^M(x) = 1]$.

(37) *Non-Branching Node Rule*: For any model M,
If A is a non-branching node with B as its sole daughter, then $[\![\text{A}]\!]^M = [\![\text{B}]\!]^M$.