CHAPTER FIVE: INTENSIONAL TYPE LOGIC

We will be short about the discussion of intensionality itself in this chapter, presupposing the discussion in Foundations and providing a more systematic setting for the operations of intensionalization and extensionalisation introduced there. We will introduce TY_2, two sorted type theory as our basic model and after that introduce Montague's IL for comparison.

5.1. Sorted Type theory

Sorted predicate logic is a subsystem of predicate logic.
Suppose that our domain D is sorted into two domains: $D = M \cup F$, where $M \cap F = \emptyset$.
Suppose we have two types of variables: $VAR_m = \{m_1, m_2\ldots\}$ and $VAR_f = \{f_1, f_2\ldots\}$.
And suppose we introduce sorted quantification:

If $m \in VAR_m$ and $\varphi \in FORM$ then $\forall m\varphi \in FORM$
If $f \in VAR_f$ and $\varphi \in FORM$ then $\forall f\varphi \in FORM$

Fact: Sorted quantification can be expressed in unsorted predicate logic.

We do this by introducing two special predicates MOUSE, FOX and set (for every model):

$F_M(MOUSE) = M$ and $F_M(FOX) = F$

Let φ be a formula and let x be a variable that does not occur in φ.
Then: $\forall m\varphi \Leftrightarrow \forall x[MOUSE(x) \rightarrow \varphi[x/m]]$
(where $\varphi[x/m]$ is the result of replacing every free occurrence of m in φ by x)

Thus, if we allow predicate denotations to be set in this way, introducing sorts in predicate logic does not influence the expressive power of predicate logic.

Obviously, the same is true in type logic. So we can unproblematically add more basic types. and get a sorted type theory.

Example: numbers.

$TYPE_{TLn}$ is the smallest set such that:
1. $e, t, n \in TYPE_{TLn}$
2. If $a,b \in TYPE_{TLn}$ then $<a,b> \in TYPE_{TLn}$

The type logic is exactly the same. I only add the clauses and structure that is new, the clauses are only sample clauses (i.e. you may want to add real numbers, arithmetic operations, measure functions, etc. etc.)

1. If $\alpha \in EXP_{<a,t>}$ then $|\alpha| \in EXP_n$
2. $<, > \in CON_{<n,<n,t>>}$

3. A model for TL_n is a structure $M = <D, <\mathbb{N},<_{\mathbb{N}}>, F_M>$,
   $<\mathbb{N},<_{\mathbb{N}}>$ is the natural numbers ordered by smaller than.

4. $D_{n,M} = \mathbb{N}$

5. If $\alpha \in EXP_{<a,t>}$ then $[\![|\alpha|]\!]_{M,g} = |ch([\![\alpha]\!]_{M,g})|$,
   where ch(f) is the set characterized by f
6. $F_M(<) = curry(<_{\mathbb{N}})$,
   $F_M(>) = curry(>_{\mathbb{N}})$
   where $curry(<_{\mathbb{N}})$ is curried relation of type <n,<n,t>>that corresponds to $<_{\mathbb{N}}$.

Important: the relations between types n, <n,t>, and <<n,t>,t>> are similar to those between e, <e,t>, and <<e,t>,t> . i.e. $D_{<<n,t>,t>}$ is also a freely generated complete atomic Boolean algebra.

In this language we can express the determiner meaning of *most* directly as:

$$\lambda Q \lambda P. |\lambda x.Q(x) \wedge P(x)| > | \lambda x.Q(x) \wedge \neg P(x)|$$

5.2. Two sorted type theory TY_2

Possible world semantics is concerned with modality, the variation of extension across possibilities, possible situations, times, world-time pairs,…
I will call them *worlds*, for short (in the literature they are also called *indices*).
We are now extending type theory with a type for possible worlds.
The logical language in question is called TY_2, two sorted type theory, where the types are individuals and worlds. The type for worlds is called s.

$TYPE_{TY2}$ is the smallest set such that:
1. $e,t,s \in TYPE_{TY2}$
2. if $a,b \in TYPE_{TY2}$ then $<a,b> \in TYPE_{TY2}$

The language of TY_2 is just the same language as TL, except that there are now also variables of type s (and possibly constants).
And we include in TY_2 a special relation of type <s,<s,t>>:

$R \in CON_{<s,<s,t>>}$

A *model* for TY_2 is a structure: $M = <D, <W,R_W>, F>$,
where D is a non-empty set of individuals,
W a non-empty set of possible worlds,
$R \subseteq W \times W$ the accessibility relation on W, a two place relation between worlds.
and F an interpretation function for the logical constants such that:
$F(R) = curry(R_W)$, the appropriate curried version of R_w

And we set:

$D_{s,M} = W$

This is all there is to TY_2, the rest is all the same as in TL.

Some sample new types:
<s,e>, the type of expressions denoting functions from worlds into individuals: **individual concepts**.

<s,t>, the type of expressions denoting functions from worlds into truth values: **propositions**.
These functions are characteristic functions of sets of possible worlds.
Hence we identify propositions with sets of possible worlds.

<s,<e,t>>, the type of expressions denoting functions from worlds into sets of individuals: **intensional properties**.

<s,<<e,t>,t>>, the type of expressions denoting functions from worlds into generalized quantifiers **intensional generalized quantifiers**.

5.3. Variation of extension across worlds

Possible world semantics for modality is concerned with variation of extension across worlds. In the modal predicate logic that we discussed in Foundations we dealt with that by interpreting expressions relative to possible worlds. We defined:

$[\![\alpha]\!]_{M,w,g}$, the extension of α ***in world*** w relative to model M and assignment g

e.g. $[\![PURR(RONYA)]\!]_{M,w,g} = 1$ iff $F_{M,w}(RONYA) \in F_{M,w}(PURR)$

and added modal operators to the language:

$[\![\Diamond\varphi]\!]_{M,w,g} = 1$ iff for some $v \in W$: $R_W(w,v)$ and $[\![\varphi]\!]_{M,v,g} = 1$

A characteristic property of this approach is that the reference to worlds and quantification over worlds never happens in the object language.

This is different in TY_2.
In TY_2 all reference to worlds and quantification over worlds is represented explicitly in the object language TY_2. And this is because TY_2 is just TL, you only evaluate relative to a model and assignment function, not relative to a world:

$[\![\alpha]\!]_{M,\,g}$, the extension of α relative to model M and assignment g

So what do you do if you want to express that Ronya purrs in world w but not in world v, i.e that *purr* varies its denotation from world w to v?
This:

1. Since the *purr* needs to vary its extension from world to world,
you interpret *purr* as a TY_2 predicate PURR $\in CON_{<s,<e,t>>}$, an *intension*, a function from worlds to extensions.

So: PURR $\in CON_{<s,<e,t>>}$ and $F_M(\text{PURR}) \in (W \rightarrow D_{<e,t>})$

This is different from what we did in modal logic before, where we interpreted *purr* as PURR $\in CON_{<e,t>}$, but assigned different extensions in different worlds:

$F_{M,w}(\text{PURR}_{<e,t>}) \in D_{<e,t>}$

The strategy of TY_2 is: every expressions that varies its interpretation across worlds is interpreted as the intension that expresses that variation.
To get the extension, we need to apply that intension to a world variable.
It will now be useful to have a notational convention:

Notation: If $\alpha \in EXP_{<s,a>}$ and $w \in VAR_s$ then: $\alpha_w := (\alpha(w))$

So we write the world-indices as subscripts on the expressions that vary their extension along worlds.

If we want to express in modal predicate logic that Ronya purrs in world w and doesn't purr in world v we do that by assuming PURR $\in CON_{<e,t>}$ and:

$[\![\text{PURR(RONYA)}]\!]_{M,w,g} = 1$ and $[\![\text{PURR(RONYA)}]\!]_{M,v,g} = 0$

In TY_2 we express the same with PURR $\in CON_{<s,<e,t>>}$, using mnemonic variables:

$[\![\text{PURR}_w(\text{RONYA}) \wedge \neg\text{PURR}_v(\text{RONYA})]\!]_{M,g} = 1$
where $g(w) = w$ and $g(v) = v$

This is true if Ronya purrs in w, but not in v.

Now, when we take a normal non-modal sentence like *Ronya purred* (ignoring the tense), then we assumed in modal logic that this sentence is evaluated relative to a world of evaluation, the real world, the world in which the context locates itself: w_0.

We make the same assumption in TY_2, but it takes a slightly different form.
In predicate logic, variables that are free in a formula are assigned a value by the outside assignment function g, and we think of those as being deictically linked.
In TY_2 we assume the same for linking to the real world. We choose one variable, say, w, to be the variable that the outside assigment function g links to the real world:

$g(w) = w_0$

This means that *if* variable w occurs *free* in φ, it is linked 'deictically'; by g to w_0.
But, w is just a variable like others.
If w is bound in φ it doesn't link to w_0 any more.

Since by default we start out by assuming that expressions are interpreted relative to the some world w_0 unless there is modality involved,
we will interpret an expression like *Ronya purrs and Emma does not*, by using *the same* free world variable:

$$PURR_w(RONYA) \wedge \neg PURR_w(EMMA)$$

We don't have modal operators in the language, but we *do* have quantification over worlds and abstraction over worlds.
And that means that we can express the semantic effects of modal operators directly in the language.

Now look at the sentences (1a,b):

(1) a. Virginia Woolf could have been the author of Ulysses.
b. Virginia Woolf could not have been the author of Finnegans wake.

author of Ulysses $\rightarrow$ AU $\in CON_{<s,<e,t>>}$
author of Finnegans wake $\rightarrow$ AF $\in CON_{<s,<e,t>>}$
Virginia Woolf $\rightarrow$ WOOLF $\in CON_e$

In modal predicate logic we translated these sentences as in (2):
We choose predicates *AU*, *AF* $\in CON_{<e,t>}$ and wrote:

(2) a. $\Diamond(WOOLF = \sigma(AU))$
b. $\neg\Diamond(WOOLF = \sigma(AF))$

with truth conditions:

(3) a. $[\![\Diamond(WOOLF = \sigma(AU))]\!]_{M,w,g} = 1$ iff
$\exists v \in W[\ R(w,v) \text{ and } F_{M,v}(WOOLF) = \sigma(F_{M,v}(AU))$
b. $[\![\neg\Diamond(WOOLF = \sigma(AF))]\!]_{M,w,g} = 1$ iff
$\forall v \in W[\text{if } R(w,v) \text{ then } F_{M,v}(WOOLF) \neq \sigma(F_{M,v}(AF))$

In TY_2 we represent the sentences in (1) as:

(4) a. $\exists v[R(\mathbf{w},v) \wedge (WOOLF = \sigma(AU_{v)})] \wedge \neg\exists v[R(\mathbf{w},v) \wedge (WOOLF = \sigma(AF_{v)}))$

What we see is:
The fact that the expression *the author of Ulysses* occurs in the scope of modal operator is interpreted as the assumption that the world variable relative to which the extension of *author of Ulysses* varies is ***bound*** by the quantifier over possible worlds.
Thus, *the author of Ulysses* is not itself evaluated relative to world w_0.
The modal operator itself though *is* evaluated relative to w_0, because the modality is restricted by the accessibility relation which is in this example linked to w_0 via the free variable w.

Abstraction over possible worlds brings you from extensions to intensions:

(5) a. [$_S$ Ronya purrs]
b. $(PURR_w(RONYA))$ — true relative to M,g if in w_0 Ronya purrs

(6) a. [$_{CP}$ That Ronya purrs]
b. $\lambda w.(PURR_w(RONYA))$ =
$\lambda v.(PURR_v(RONYA))$ — the set of worlds v where ronya purrs. <s,t>

This means that the CP denotes the proposition that Ronya purrs, and note that the expression in (6b) *no longer* has a free world variable. This is essential in dealing with the intensionality puzzles.

I will now make a typographical assumption that will make the notation a bit more readable and a bit closer to Montague's intensional logic.

Notation convention: If $w \in VAR_s$ then $^{\wedge w}\beta := \lambda w.\beta$

So the notation is just lambda abstraction over a world variable, but it bring out the fact that we are dealing with an intension.
[Maybe I should have written $^{\lambda w}\beta$. If you prefer that, you can think of $^{\wedge w}\beta$ as being just that, but λ written with a capital letter Λ.]
So:

(6) a. [$_{CP}$ That Ronya purrs]
b. $^{\wedge w}(PURR_w(RONYA))$ =
$^{\wedge v}(PURR_v(RONYA))$ — the set of worlds v where ronya purrs. <s,t>

5.4. Propositional attitudes

(1) and (2) do not entail (3):
(1) Fred believes that the author of Ulysses is the author of Ulysses.
(2) The author of Ulysses is the author of Finnegans Wake.
(3) Fred believes that the author of Ulysses is the author of Finnegans Wake.

Analysis:
BELIEVE is a constant of type <s, <<s,t>,<e,t>>>.
This means that $BELIEVE_w$ is a relation between individuals and propositions (x believes p in w).

(1a) $BELIEVE_w(FRED, {}^{\wedge w}\sigma(AU_w) = \sigma(AU_w))$
(2a) $\sigma(AU_w) = \sigma(AF_w)$
(3a) $BELIEVE_w(FRED, {}^{\wedge w}\sigma(AU_w) = \sigma(AF_w))$

When we clarify the nature of bound variables, we see that the pattern is clearly invalid (even if we don't specify more about the possible world semantics of *believe*):

(1a) $BELIEVE_w(FRED, {}^{\wedge v}\sigma(AU_v) = \sigma(AU_v))$

Fred stands in w in the believe relation to the sets of worlds where the author of Ulysses is self identical. This is the set of all possible worlds!
(2a) $\sigma(AU_w) = \sigma(AF_w)$
The author of Ulysses in w is the author of Finnegans wake in w (this is true!)

(3a) $BELIEVE_w(FRED, {}^{\wedge v}\sigma(AU_v) = \sigma(AF_v))$
Fred stands in w in the believe relation to the sets of worlds where the author of Ulysses is the author of Finnegans wake.
This is not the set of all possible worlds, because there may be worlds in which FW wasn't written, or written by somebody else (unlikely), or Ulysses wasn't written, or written by somebody else.

So there is no reason that (1a) and (2a) entail (3a).

Similarly for the modal inference pattern:

(4) Virginia Woolf could have been the author of Ulysses.
(5) The author of Ulysses is the author of Finnegans wake.
(6) Virginia Woolf could have been the author of Finnegans wake.

(4a) $\exists v[R(\mathbf{w},v) \wedge (WOOLF = \sigma(AU_v))]$
(5a) $\sigma(AU_w) = \sigma(AF_w)$
(6a) $\exists v[R(\mathbf{w},v) \wedge (WOOLF = \sigma(AF_v))]$

(4a) and (5a) for the same reason do not entail (6a): there might be an accessible world where Virginia Woolf wrote Ulysses, even if there is no accessible world where she wrote Finnegans wake.

A technical remark. Compositionally, the fact that modal operators are variable binding operators means that they don't operate on the extension of their complement, but on the intension.

Thus, if we compositionally want to divide $\exists w[R(w,v) \wedge \varphi[w/v]]$ into the modal and non-modal meaning that are combined, we do that by backwards λ-conversion: with
$p \in VAR_{<s,t>}$:

(1) $\exists w[R(w,v) \wedge \varphi[v/w]]$
(2) $\exists w[R(w,v) \wedge {}^{\wedge w}\varphi[w](v)]$
(3) $\lambda p \exists w[R(w,v) \wedge p(v)]\ ({}^{\wedge w}\varphi[w])$

This means that we can *define* the modals as:

$\Diamond = \lambda p^{\wedge w}\exists v[R(w,v) \wedge p(v)]$
$\Box = \lambda p^{\wedge w}\forall v[R(w,v) \rightarrow p(v)]$

But as operations they don't operate on φ_w of type t but on φ, which is ${}^{\wedge v}\varphi[v/w]$ of type

<s,t>
Thus, modal operators are analyzed as functions from propositions to propositions, from sets of worlds to sets of worlds.
So let $\varphi \in EXP_{<s,t>}$.
φ is true in a world w iff $w \in \varphi$.

$\Diamond$ maps the set of worlds where φ is true onto
the set of worlds from which a world is accessible where φ is true.

With this it is indeed the case that
$\Diamond\varphi$ is true in world w iff $w \in \Diamond\varphi$ iff w is a world from which a world is accessible where φ is true, i.e. from which a world v is accessible such that $v \in \varphi$.

Notationally, you will find also analyses that stress the analogy with generalized quanfiers, that define $\Box$ and $\Diamond$ just as generalized quantifers EVERY and SOME but as relations between sets of worlds.
Then, if we set:
$R_w = \lambda v.R(w,v)$, we will write:

Instead of $\Box\varphi$:	$EVERY[R_w, {}^{\wedge v}\varphi[v/w]]$or	$R_w \subseteq {}^{\wedge v}\varphi[v/w]$
Instead of $\Diamond\varphi$:	$SOME[R_w, {}^{\wedge v}\varphi[v/w]]$	or $R_w \cap {}^{\wedge v}\varphi[v/w] = \emptyset$

5.5. Seek and find

In TY_2 we can also analyze intensional verbs that are not express propositional attitudes.

(4) Mary seeks the author of Ulysses.
(5) The author of Ulysses is the author of Finnegans Wake.
(6) Mary seeks the author of Finnegans Wake.

(4) and (5) do not entail (6):

When we say that (4) and (5) do not entail (6), we mean that there is a reading of (4) and a reading of (6) on which (4) and (5) do not entail (6).
At the same time, we have to recognize that there is also a reading of (4) and of (6) on which the inference is perfectly valid.
That is, (4) and (6) are ambiguous between a **de dicto** and a **de re** reading.

On the **de re** reading of (4), the interpretation of the description *the author of Ulysses* comes to the account of the speaker, that is, we interpret this expression as an object that is the author of Ulysses according to the speaker. This reading can be paraphrased by (7):

(7) There is a person, the author of Ulysses, and Mary seeks that person.

The inference from (7) and (5) to (8) is perfectly valid:
(7) There is a person, the author of Ulysses, and Mary seeks that person.
(5) The author of Ulysses is the author of Finnegans Wake.
(8) There is a person, the author of Finnegans Wake, and Mary seeks that person.

On the **de dicto** reading of (4), the interpretation of the description *the author of Ulysses* comes to the account of Mary. That is, Mary does not necessarily stand in a relation to any actual object at all, or she may stand in relation to the wrong object, an object who she thinks is the author of Ulysses.
What plays a role in the interpretation of (4) on this reading is not the **person** who is the author of Ulysses, but rather the **role** that that person plays, the role of being the author of Ulysses.
We can think of the difference in terms of when Mary would be satisified to say that she found the author of Ulysses.
On the **de re** reading, she would be satisfied if she believed she had found James Joyce, who happens to be the author of Ulysses.
On the **de dicto** reading, she would be satisfied if believed she had found an object which she believed to fill the role of being the author of Ulysses.

Now, we will say more about the semantics of intensional verbs like *seek* below.
But we can show here that the assumption that *seek* creates an intensional context deals with the puzzle.
We will make an assumption for the semantics of *seek* here that will only deal with this example.

Let SEEK be a constant of type <s, <<s,e>,<e,t>>

(4a) $\text{SEEK}_w(\text{MARY}, {}^{\wedge v}\sigma(\text{AU}_v))$
(5a) $\sigma(\text{AU}_w) = \sigma(\text{AF}_w)$
(6a) $\text{SEEK}_w(\text{MARY}, {}^{\wedge v}\sigma(\text{AF}_v))$

The fact that Mary stands in w in the seek relation to the individual concept *the author of Ulysses*, does not entail that she stands in the seek relation to the individual concept *the author of Finnegans wake*, and the fact that both concepts happen to have the same value for the real world doesn't change that.

For the *de re*-readings we will need to find a grammar mechanism that derives them. We will do that later. But the semantic idea is as follows.
The complement of SEEK_w in this analysis needs to be of type <s,e>, an individual concept.
Let x be a free variable. then ${}^{\wedge v}x$ is the function that maps every world onto g(x), the constant function on g(x).
Suppose that we interpret *Mary seeks individual x* as: $\text{SEEK}_w(\text{MARY}, {}^{\wedge v}x)$,
i.e. Mary stands in the seek relation to the individual concept that corresponds one-one with the individual x.
Then we can analyze the *de re* reading via abstraction over x:

(7a) $\lambda x.\text{SEEK}_w(\text{MARY}, {}^{\wedge v}x)(\sigma(\text{AU}_w))$
(5a) $\sigma(\text{AU}_w) = \sigma(\text{AF}_w)$
(6a) $\lambda x.\text{SEEK}_w(\text{MARY}, {}^{\wedge v}x)(\sigma(\text{AF}_w))$

We can simplify this to:

(7a) $\text{SEEK}_w(\text{MARY}, {}^{\wedge v}\sigma(\text{AU}_w))$
(5a) $\sigma(\text{AU}_w) = \sigma(\text{AF}_w)$
(6a) $\text{SEEK}_w(\text{MARY}, {}^{\wedge v}\sigma(\text{AF}_w))$

Crucially, the complement stays a constant function. This has nothing to do with:

(7b) $SEEK_w(MARY, {}^{\wedge v}\sigma(AU_v))$
(5a) $\sigma(AU_w) = \sigma(AF_w)$
(6b) $SEEK_w(MARY, {}^{\wedge v}\sigma(AF_v))$

The b-pattern is *de dicto* and invalid. But the a-pattern is perfectly valid.

Montague and Zimmermann

Analyzing *seek* as a relation between individuals and individual concepts is ok for the above examples, but not for other examples.

Let us add a second piece of data: (1) and (2) on the *de dicto* reading:

(1) Mary seeks a griffin.
(2) Mary seeks a centaur.

In the real world there are no griffins, nor are there centaurs.
The non-equivalence of (1) and (2) shows very clearly that the type of the object of SEEK needs to be intensional.

In the real world w: $F(GRIFFIN_w) = F(CENTAUR_w) = \emptyset$.

Any extensional translation of the object of *seek*, resp. *a griffin* and *a centaur*, will be compositionally built from the extensions of $GRIFFIN_w$ and $CENTAUR_w$.
Since these are the same in w, these extensional translations of *a griffin* and *a centaur* will also be the same, and (1) and (2) will become equivalent.

So we need an interpretation that uses the intension of *griffin* and *centaur*.
The above interpretation via individual concepts does that, but is tailored to definite DPs.
It is not clear how you interpret (1) and (2) on that model, because it is not clear *what* individual concept the interpretations of *a griffin* and *a unicorn* would be.
Notice too that Mary need three giffins and five centaurs for the spell she has in mind, so we also need *de dicto* interpretations for (3) and (4):

(3) Mary seeks three griffins and five unicorns.
(4) Mary seeks five unicorns,

Montague argued that this argument applied to DPs in general, and that we need to deal with *de dicto* interpretations of (5) and (6) as well:

(5) Mary seeks every unicorn.
(6) Mary seeks most griffins.

Montague 1974 assumes that in w SEEK is a relation between individuals and intensions of generalized quantifiers: $<s,<<e,t>,t>>$ (actually, for independent reasons he assumes an even higher, more intensional type). In TY_2 this becomes:

SEEK is a constant of type <s, < <s,<<e,t>,t>>,<e,t>> >

This gives:

(7) Mary seeks a griffin $\rightarrow$ $SEEK_w(MARY, {}^{\wedge v}\lambda P.\exists x[GRIFFIN_v(x) \wedge P(x)])$
(8) Mary seeks a centaur $\rightarrow$ $SEEK_w(MARY, {}^{\wedge v}\lambda P.\exists x[CENTAUR_v(x) \wedge P(x)])$

And these are indeed *de dicto* and not equivalent:
if Mary stands in w in the seek-relation to the function that maps every world onto the set of all sets of individuals that intersect the set of griffins in that world,
it doesn't follow that Mary stands in w in the seek-relation to the function that maps every world onto the set of all sets of individuals that intersect the set of centaurs in that world,
nor does it say anything about the existence of griffins or centaurs in this world.

This can be made clearer by trying to work out more details of the lexical semantics of *seek* (as *try to find*): here is a stab:

$S_{w,x,B,\alpha}$ is the set of success worlds in w for *individual x, behaviour B and property* α
$S_{w,x,B,\alpha} = {}^{\wedge v,t}\ \lambda x.T(\lambda y FIND_{v,t}(x,y))$ (with T a generalized quantifer):
the set of world-time pairs <v,t> where if x shows B at time t in v,
then at some time t' >t $T(\lambda y FIND_{v,t'}(x,y))$ holds.

$B_{w,x}$ is the set of worlds compatible with what Mary believes.

Mary seeks T in w if Mary shows in w behaviour B at time t and $B_{w,x} \subseteq S_{w,x,B,\alpha}$

So: Mary seeks a griffin if Mary shows behaviour B and in Mary's belief worlds, showing behaviour B leads to finding a griffin at some later time.

See Sharvit 2003 for a more subtle analysis.

Even so: *Seeking a griffin* and *seeking a centaur* are not equivalent, because in Mary's belief worlds the behaviour shown may well lead to finding a griffin, but not to finding a centaur.

There is an alternative to Montague's analysis by Ede Zimmermann in the proceedings of SALT1992 which analyzes *seek* at type <s, <<s,<e,t>> ,<e,t>> instead.

(9) $SEEK_w(MARY, {}^{\wedge w}GRIFFIN_w)$
(10) $SEEK_w(MARY, {}^{\wedge w}CENTAUR_w)$

The difference between these approaches has to do with quantificational DPs.

Zimmerman argues that *seek* only has a **de dicto** reading with object DPs that can have a predicative reading (definite and indefinite DPs, but not quantificational DPs like *every cat* or *most girls*).
Zimmermann assumes that predicative DPs can get an interpretation at the type of intensional properties <s,<e,t>>.
A wide-scope mechanism provides the *de re* analysis that sentences with quantificational

DPs as the object of *seek* have
On Montague's analysis *seek* has a **de dicto** reading for all object DPs.

Thus the difference of opinion concerns the analysis of (12):
Zimmerman generates only a **de re** reading, roughly (12),
Montague generates roughly(12) **and** the **de dicto** reading (13):

(11) Mary seeks every griffin.
(12) $\forall x[\text{GRIFFIN}_w(x) \rightarrow \text{SEEK}_w(\text{MARY}^{\wedge\vee}\lambda z.z{=}x)]$
(13) $\text{SEEK}_w(\text{MARY}, {}^{\wedge\vee}\lambda P.\forall x[\text{GRIFFIN}_v(x) \rightarrow P(x)])$

A situation to test the analyses on is the following.
Suppose Mary believes there are griffins, and moreover she believes that she has caught all of them the previous weeks,
but she believes that in the meantime they have escaped.
She wants to recapture them and hence she starts seeking.
Of course, there are no griffins.

If you feel that in this situation you can felicitously and truthfully say (11),
Montague's analysis might be the one for you.
If you can't, you might want to adopt Zimmerman's.

5.6. Rigid designators

Notice that I have made a difference between the translations of proper names and definite descriptions: JOYCE , $\sigma(AU_w) \in EXP_e$.
I have not interpreted JOYCE as a constant of type <s,e> and written $JOYCE_w$.

The reason is the assumption stemming from Kripke 1972 that proper names are **rigid designators:** proper names do not vary their extension from world to world: a proper name refers to the same individual in every possible world.
This is a difference between proper names and definite descriptions.

Kripke 1972 argues that proper names are rigid in the context of modals and counterfactual conditionals.
Look at the definite description *the president* in sentence (1):

(1) If the president had not been a republican, there would have been a democratic administration.

Sentence (1) is ambiguous.
On the one reading, it means that if the current president, Dixon, had not been a republican he would have been a democrat.
The semantics for counterfactuals instructs us here to take the current president from our world to close worlds where he is not a republican: it says that he's a democrat there.

On the other reading, the sentence expresses something like:
if the expression *the president* had referred to somebody who isn't a republican, it would have referred to somebody who is a democrat.
This instructs us to look at the reference of *the president* in different worlds: go to close worlds where whoever is the president there is not a republican, that person is a democrat there.

Now look at the proper name *Dixon* in sentence (2):

(2) If Dixon had not been a republican he would have been a democrat.

Sentence (2) is **not** ambiguous. It only has a reading corresponding to the first reading (1) discussed above:
look at close worlds where Dixon is not a republican, he's a democrat there.
Sentence (2) does not mean: look at close worlds where the name *Dixon* does not refer to somebody who is a republican: it refers there to a democrat.

Kripke argues (and Montague follows him in this) that this means that while definite descriptions can refer to different individuals in different worlds, proper names cannot, proper names are rigid designators, they refer to the same individual in every world where they refer at all.
This is what the condition of rigidity (or the meaning postulate) guarantees.

5.7. Montague's Intensional Logic IL

Montague's IL is like the Modal predicate logic of Foundations in that it does not have variables over worlds in the object languages. His types are:

$TYPE_{IL}$ is the smallest set such that:

1. $e,t \in TYPE_{IL}$
2. If $a,b \in TYPE_{IL}$ then $<a,b> \in TYPE_{IL}$
3. If $a \in TYPE_{IL}$ then $<s,a> \in TYPE_{IL}$

IL is an intensional language unlike TL and TY_2.
The interpretation function changes for that reason to:

$$F: CON_a \to (W \to D_a)$$

And the interpretation clause for constants changes accordingly:

If $c \in CON_a$ then $[\![c]\!]_{M,w,g} = (F(c))(w)$ which we can write as $F_w(c)$.

Montague's language has modal operators with the obvious interpretation, and an intension and extension operator:

Intension: If $\alpha \in EXP_a$ then $^\wedge\alpha \in EXP_{<s,a>}$
Extension: If $\beta \in EXP_{<s,a>}$ then $^\vee\beta \in EXP_a$

with interpretation:

$[\![^\wedge\alpha]\!]_{Mw,g} =$ h: $W \to D_a$ such that for every $v \in W$: $h(v) = [\![\alpha]\!]_{Mv,g}$
$[\![^\vee\beta]\!]_{Mw,g} = [\![\beta]\!]_{Mw,g}(w)$

Clearly this corresponds to abstraction over and application to worlds in TY_2

$^\wedge\alpha$	corresponds to	$\lambda v.\ \alpha[v/w]$
$^\vee\beta$	corresponds to	$\beta(w)$

From this you can directly derive the fact of cup-cap elimination in IL:

Cup cap elimination
$^{\vee\wedge}\alpha = \alpha$
$\lambda v.\alpha[v/w]\ (w) = \alpha[w]$

Intensionality in IL:

(1) and (2) do not entail (3):
(1) Fred believes that the author of Ulysses is the author of Ulysses.
(2) The author of Ulysses is the author of Finnegans Wake.
(3) Fred believes that the author of Ulysses is the author of Finnegans Wake.

Analysis:

BELIEVE is a constant of type <<s,t>,<e,t>>.

This means that relative to world w: BELIEVE is a relation between individuals and propositions (x believes p (evaluated in w)).

(1a) BELIEVE (FRED, $^\wedge(\sigma(AU) = \sigma(AU))$

(2a) $\sigma(AU) = \sigma(AF)$

(3a) BELIEVE (FRED, $^\wedge(\sigma(AU) = \sigma(AF))$

This pattern is as invalid in IL as the corresponding pattern is in TY_2.

While TY_2 is an extensional language in which the intensionality is made explicit in world variables, IL is an intensional language:

-substitution of expressions with the same extension is not valid in IL, only substitution of expressions with the same intension is.

-λ-conversion of an expression α that is not rigid (of which the intension is not a constant function) where α lands in an intensional context is not valid in IL:

e.g.: (1) and (2) are not equivalent:

(1) $\lambda x.\Diamond(\sigma(AU) = x)\ (\sigma(AF))$

(2) $\Diamond(\sigma(AU) = \sigma(AF))$

But (3) and (4) are equivalent, because variable y is rigid:

(3) $\lambda x.\Diamond(\sigma(AU) = x)\ (y)$

(4) $\Diamond(\sigma(AU) = y)$

We usually add a meaning postulate to IL to the effect that proper names are rigid so that (5) and (6) *are* equivalent:

(5) $\lambda x.\Diamond(\sigma(AU) = x)\ (JOYCE)$

(6) $\Diamond(\sigma(AU) = JOYCE)$

And we note that (7) and (8) and (9) **are** equivalent:

Let $\mathbf{x} \in VAR_{<s,e>}$

(7) $\lambda \mathbf{x}.\Diamond(\sigma(AU) = {}^\vee\mathbf{x})\ ({}^\wedge\sigma(AF))$

(8) $\Diamond(\sigma(AU) = {}^\vee{}^\wedge\sigma(AF))$ by the rigidity of $^\wedge\sigma(AF)$

(9) $\Diamond(\sigma(AU) = \sigma(AF))$ by cup-cap elimination

That this is so is shown easily by translating this into TY_2: in TY_2 you only need to do the λ-conversions to show that this is valid:

Let $\mathbf{x} \in VAR_{<s,e>}$

(7) $\lambda \mathbf{x}.\exists v[R(w,v) \wedge (\sigma(AU_v) = \mathbf{x}_v]\ (\lambda u\,\sigma(AF_u))$

(8) $\exists v[R(w,v) \wedge (\sigma(AU_v) = \lambda u\,\sigma(AF_u)_v\ ]$

(9) $\exists v[R(w,v) \wedge (\sigma(AU_v) = \sigma(AF_v))]$

So which language should you use as a presentation language?

In practice, people use IL when the intensionality doesn't matter much, but they want to use the operations $^\wedge$ and $^\vee$. These operations are visually nicer that yet another abstraction, this time over worlds.

On the other hand, often it is useful to have the world variable explicitly marked on expressions that vary their extension relative to worlds, and then TY_2 is called for.
Also, more complex modal relations are much easier to deal with in TY_2 (e.g. the work on the semantics of questions in Groenendijk and Stokhof 1983).

When working on topics that involves the world parameter I will in these notes use TY_2. But I like the visual cleanliness of the operators $^\wedge$ and $^\vee$, and so I introduce TY_2 variants of them, besides the indexed cap I may also sometimes use the indexed cup:

Let $v \in VAR_s$.
$^{\wedge_w}\alpha \; := \; \lambda w.\alpha$
$^{\vee_w}\beta \; := \; \beta(w)$ or we use the notation β_w for this

5.8. Adding intensional expressions to the grammar

We will now extend the grammar with the intentsional transitive verb *seek*, the complementizer *that*, and the propositional attitude verb *believe*. We will follow Montague's strategy for interpreting *seek* here.

$V \rightarrow$ <<s,<<e,t>,t>>,<e,t>>, <<s,t>,<e,t>>
$C \rightarrow$ <<s,t>,<s,t>>

And we add: $\text{COMP}[C,I] =_E I$

Lexical item: seek
Category: V
Interpretation: $\text{SEEK} \in \text{CON}_{<s,<<s,<<e,t>,t>>,<e,t>>>}$

Lexical item: believe
Category: V
Interpretation: $\text{BELIEVE} \in \text{CON}_{<s,<<s,t>,<e,t>>>}$
Lexical item: that
Category: C
Interpretation: $\lambda p.p$ with $p \in \text{VAR}_{<s,t>}$

We extend the type shifting theory with two operations:

Intensionalization:
LIFT: $a \rightarrow <s,a>$
$\text{LIFT}[\alpha] = {}^{\wedge v}\alpha[v/w]$ with w the designated world variable

Intensional argument lift:
LIFT: $e \rightarrow <s,<<e,t>,t>>$
$\text{LIFT}[\alpha] = {}^{\wedge w}\lambda P.P(\alpha)\ [v/w]$
with w the designated world variable

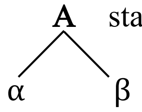
SEEK is of type <s,<<s,<<e,t>,t>>,<e,t>>>.
This means that SEEK_w is of type ,<<s,<<e,t>,t>>,<e,t>>, a relation between individuals and intensions of generalized quantifiers.

It will be useful to define in terms of this relation a relation of seeking between individuals. This relation is called SEEK_* and it is defined as:

$\text{SEEK}_{*,w} = \lambda y\lambda x.\text{SEEK}_w(x, {}^{\wedge w}\lambda P.P(y))$
the relation that holds between individuals x and y iff x stands in the seek relation to function that maps every world onto the set of sets in D_e that y is in.

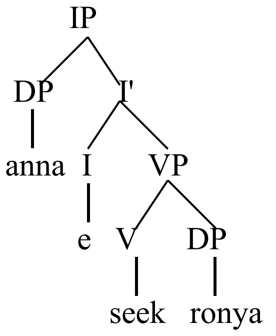
This definition will allow us to represent certain formulas in a simpler way below.

We will look at some examples. I will now use the format in which next to the syntactic tree, I write a semantic tree which shows the corresponding compositional semantics.

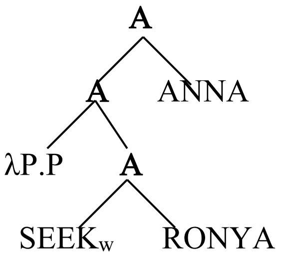
Here  stands for APPLY[α, β], so the semantic function is written on the left.

(1) Anna seeks Ronya.

SYNTAX



SEMANTICS



APPLY[$SEEK_w$, RONYA]

RONYA is of type e. The complement of seek is of type <s,<<e,t>,t>>.
This is a type mismatch that is resolved with the new type shifting rule LIFT[α] = $^{\wedge w}\lambda P.P(\alpha)$
So:

APPLY[$SEEK_w$, RONYA]
= $SEEK_w$(LIFT[RONYA]) [def. APPLY]
= $SEEK_w$($^{\wedge w}\lambda P.P(RONYA)$) [def. LIFT]

APPLY[$\lambda P.P$, $SEEK_w$($^{\wedge w}\lambda P.P(RONYA)$)]
= [$\lambda P.P$]($SEEK_w$($^{\wedge w}\lambda P.P(RONYA)$)) [def. APPLY]
= $SEEK_w$($^{\wedge w}\lambda P.P(RONYA)$)

APPLY[$SEEK_w$($^{\wedge w}\lambda P.P(RONYA)$), ANNA]
= [$SEEK_w$($^{\wedge w}\lambda P.P(RONYA)$))](ANNA) [def. APPLY]
= $SEEK_w$(ANNA, $^{\wedge w}\lambda P.P(RONYA)$)) [rel. notation]

This expresses that Anna stands in the seek relation between the function that maps every world onto the set of properties that Ronya has. We will now use $SEEK_{*,w}$ to express that this is a relation between individuals.

$SEEK_w(ANNA,\ {}^{\wedge w}\lambda P.P(RONYA)))$

First we do backward λ-conversion on ANNA:

$$\begin{aligned} & SEEK_w(ANNA,\ {}^{\wedge w}\lambda P.P(RONYA))) \\ = & [\lambda x.SEEK_w(x,\ {}^{\wedge w}\lambda P.P(RONYA)](ANNA) \qquad \text{[backwards } \lambda\text{-con]} \end{aligned}$$

Then we do backward λ-conversion on RONYA.
RONYA is in the scope of the intensional operator ${}^{\wedge w}$ but it is rigid (doesn't have a world variable), so backward λ-conversion is allowed:

$$\begin{aligned} & [\lambda x.SEEK_w(x,\ {}^{\wedge w}\lambda P.P(RONYA)](ANNA) \\ = & [[\lambda y\lambda x.SEEK_w(x,\ {}^{\wedge w}\lambda P.P(y))](RONYA)\ ]\ (ANNA) \qquad \text{[backwards } \lambda\text{-con]} \end{aligned}$$

and with relational notation:

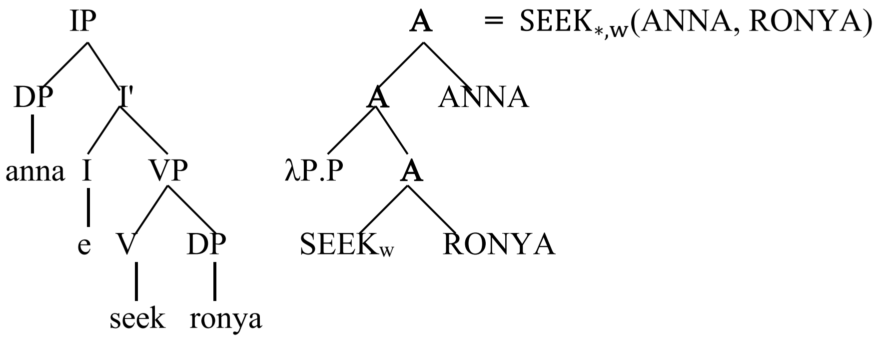
$$[\lambda y\lambda x.SEEK_w(x,\ {}^{\wedge w}\lambda P.P(y))](ANNA,\ RONYA)$$

Since $\lambda y\lambda x.SEEK_w(x,\ {}^{\wedge w}\lambda P.P(y)) = SEEK_{*,w}$ we derive:

$$SEEK_{*,w}(ANNA,\ RONYA) \qquad \text{[def } SEEK_{*,w}\text{]}$$

So we get:

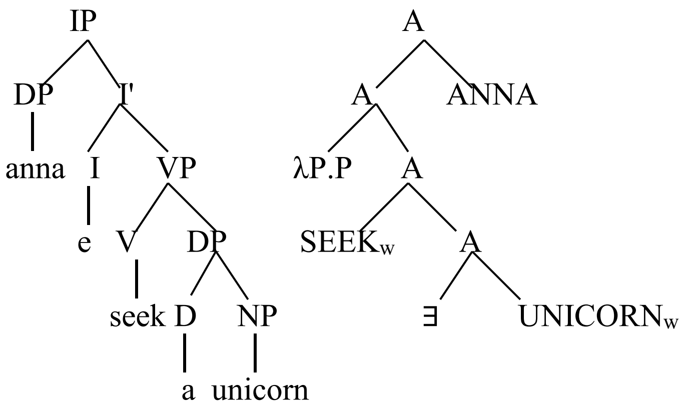
SYNTAX SEMANTICS



The sentence Anna seeks Ronya expresses, as it should, a relation between individuals.

(2) Anna seeks a unicorn.

We add the noun *unicorn* with interpretation UNICORN $\in$ $CON_{<s,<e,t>>}$
For simplicity we write in the tree $\exists$ for $\lambda Q\lambda P\exists x[Q(x) \wedge P(x)]$



$APPLY[SEEK_w, \lambda P.\exists z[UNICORN_w(z) \wedge P(z)]]$

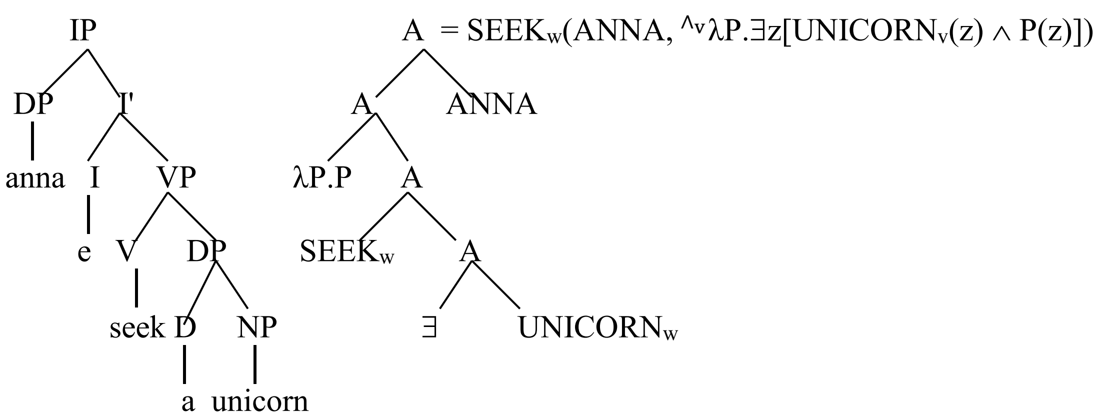
$\lambda P.\exists z[UNICORN_w(z) \wedge P(z)]$ is of type <<e,t>,t>, but $SEEK_w$ requires a complement of type <s,<<e,t>,t>>. The type mismatch is resolved with the new type shifting rule:
$LIFT[\alpha] = {}^{\wedge w}\alpha$

$APPLY[SEEK_w, \lambda P.\exists z[UNICORN_w(z) \wedge P(z)]]$
$= SEEK_w(LIFT[\lambda P.\exists z[UNICORN_w(z) \wedge P(z)]])$ [def APPLY]
$= SEEK({}^{\wedge w}\lambda P.\exists z[UNICORN_w(z) \wedge P(z)])$
$= SEEK({}^{\wedge v}\lambda P.\exists z[UNICORN_v(z) \wedge P(z)])$

$APPLY[\lambda P.P, SEEK_w({}^{\wedge v}\lambda P.\exists z[UNICORN_v(z) \wedge P(z)]]$
$= SEEK_w({}^{\wedge v}\lambda P.\exists z[UNICORN_v(z) \wedge P(z)])$

$APPLY[SEEK_w({}^{\wedge v}\lambda P.\exists z[UNICORN_v(z) \wedge P(z)]), ANNA]$
$= SEEK_w({}^{\wedge v}\lambda P.\exists z[UNICORN_v(z) \wedge P(z)]) (ANNA)$ [def. APPLY]
$= SEEK_w(ANNA, {}^{\wedge v}\lambda P.\exists z[UNICORN_v(z) \wedge P(z)])$ [rel. notation.]

So we get:



This expresses that Anna stands in the seek relation to the function that maps every world onto the set of properties that some unicorn has *in that world*. Since there are no unicorns in this world, this does not express a relation between Anna and any object in this world.

For clarity, we can point at the difference between *Anna seeks Ronya* and *Anna seeks the unicorn*.

Anna seeks ronya → $SEEK_{*,w}(ANNA, RONYA)$

Anna stands in w in the $SEEK_*$ relation to Ronya

Anna seeks the unicorn → $SEEK_w(ANNA,\ {}^{\wedge v}\lambda P.P(\sigma(UNICORN_v)))$

Anna stands in w in the SEEK to the function that maps every world onto the set of all properties that what is in that world the (contextually) unique unicorn has. This does not reduce to a relation in w between Anna and an individual.

Of course, these examples have *de re* readings too. We will deal with below and in the next chapter.

(3) Anna believes that a unicorn purred.

We now come to the complementizer *that*.
At this point we pause to explain something important about intensional contexts.
I have assumed:

that $\rightarrow \lambda p.p$ the identity function at type <<s,t>,<s,t>>

We know that propositional attitude verb *believe* takes a proposition of type <s,t> as object, so the output type <s,t> makes sense as the type of the CP. But the complement of complementizer *that* is the IP and the type of the IP is t.
So it would seem to make much more sense to assume that complementizer *that* is of type <t,<s,t>>, and that would give the following obvious translation:

that $\rightarrow \lambda\varphi.\ {}^{\wedge w}\varphi$ with φ a variable of type t.

Then we could argue:

that Ronya purrs $\rightarrow$ APPLY[$\lambda\varphi.\ {}^{\wedge w}\varphi$, $PURR_w(RONYA)$]

and reduce:

APPLY[$\lambda\varphi.\ {}^{\wedge w}\varphi$, $PURR_w(RONYA)$] [def. APPLY]
= [$\lambda\varphi.\ {}^{\wedge w}\varphi$]($PURR_w(RONYA)$) [$\lambda$-conversion]
= ${}^{\wedge w}PURR_w(RONYA)$ of type <s,t>

Unfortunately, we cannot do that, because the λ-conversion that we did here does not preserve meaning, because variable w is free in $PURR_w(RONYA)$, but bound by ${}^{\wedge w}$ in ${}^{\wedge w}PURR_w(RONYA)$.

[$\lambda\varphi.\ {}^{\wedge w}\varphi$]($PURR_w(RONYA)$) is equivalent to ${}^{\wedge v}PURR_w(RONYA)$ which denotes the constant function on 1 if Ronya purrs in w, characterizing {1}, and the constant function on 0 if Ronya doesn't purr in w, characterizing {0}, while what we want is ${}^{\wedge w}PURR_w(RONYA)$, the function characterizing the set of worlds where Ronya purrs.

The diagnosis of what goes wrong also suggests the solution:
You cannot take an **extensional non-rigid** input and expect to get the right intensional output, because that would require binding the world variable, which functional application doesn't do.
Insteas we take as input the **intension** of this extensional non-rigid input in which the world variable is bound, and that we map onto the right intensional output.

Wrong:	APPLY[$\lambda\varphi.\ {}^{\wedge w}\varphi$, $<t,<s,t>>$	$PURR_w(RONYA)$] t
Almost Right:	APPLY[$\lambda p.p$, $<<s,t>,<s,t>>$	$PURR_w(RONYA)$] t
Right:	APPLY[$\lambda p.p$, $<<s,t>,<s,t>>$	${}^{\wedge w}PURR_w(RONYA)$] $<s, t>$

So to solve the problem in the right way, we need to get from the IP interpretatation of type t $PURR_w(RONYA)$], to its intension ${}^{\wedge w}PURR_w(RONYA)$] of type $<s,t>$.

But, of course, we don't need to do extra work for that, because the *almost right* type mismatch resolves with

$$LIFT[\alpha] = {}^{\wedge w}\alpha.$$

This strategy is completely general. In fact, we applied it in the previous example as well: We didn't suggest to translate *seek* as:

$\lambda T\lambda x.SEEK_w(x, {}^{\wedge w}T)$

and try to apply λ-conversion in:

$\lambda T\lambda x.SEEK_w(x, {}^{\wedge w}T)(\lambda P.\exists x[UNICORN_w(x) \wedge P(x)])$

because λ-conversion is not valid here.

Instead we gave a translation of *seek* equivalent to:

$\lambda \mathit{T}\lambda x.SEEK_w(x,\mathit{T})$ with T a variable of type $<s,<<e,t>,t>>$

and derive:

APPLY[$\lambda \mathit{T}\lambda x.SEEK_w(x,\mathit{T})$, $\lambda P.\exists x[UNICORN_w(x) \wedge P(x)]$]

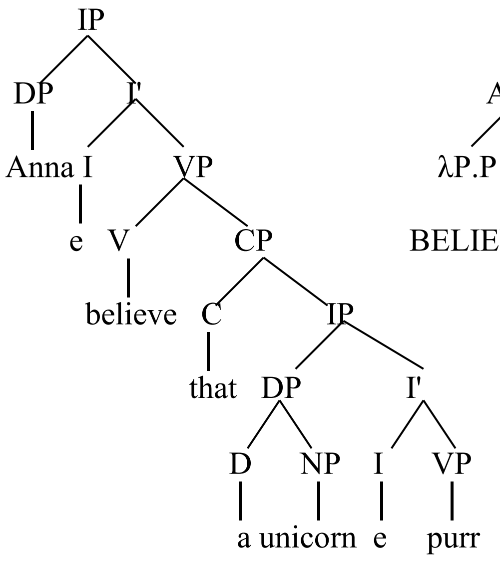
which resolves with the same type shifting rule as:

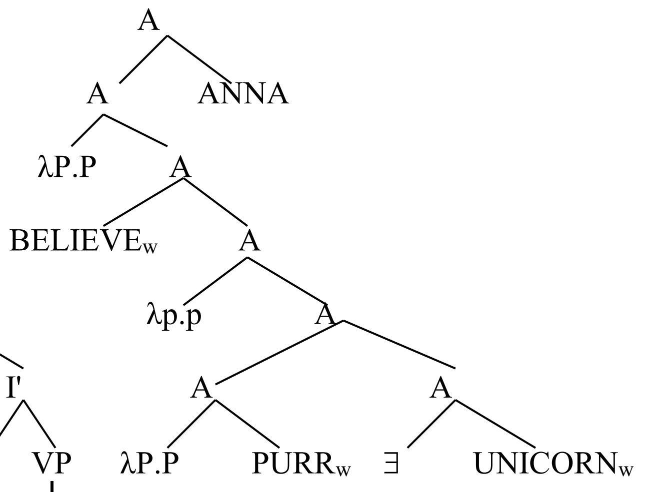
APPLY[$\lambda \mathit{T}\lambda x.SEEK_w(x,\mathit{T})$, ${}^{\wedge v}\lambda P,\exists x[UNICORN_v(x) \wedge P(x)]$]

and we derive with λ-conversion:

$\lambda x.SEEK_w(x, {}^{\wedge v}\lambda P,\exists x[UNICORN_v(x) \wedge P(x)])$

We do the derivation:





We derive at the lower IP level:

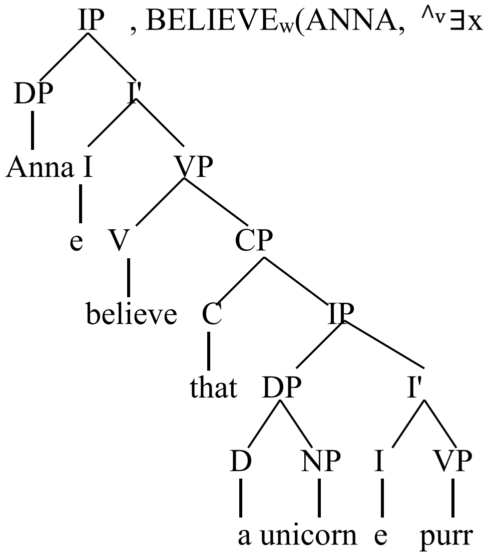
a unicorn purrs → $\exists x[\mathrm{UNICORN}_w(x) \wedge \mathrm{PURR}_w(x)]$

From here we continue with:

APPLY[$\lambda p.p$, $\exists x[\mathrm{UNICORN}_w(x) \wedge \mathrm{PURR}_w(x)]$]
= $\lambda p.p$ (LIFT[$\exists x[\mathrm{UNICORN}_w(x) \wedge \mathrm{PURR}_w(x)]$]) [def APPLY]
= $\lambda p.p({}^{\wedge v}\exists x[\mathrm{UNICORN}_v(x) \wedge \mathrm{PURR}_v(x)])$ [def LIFT]
= ${}^{\wedge v}\exists x[\mathrm{UNICORN}_v(x) \wedge \mathrm{PURR}_v(x)]$

APPLY[$\mathrm{BELIEVE}_w$, ${}^{\wedge v}\exists x[\mathrm{UNICORN}_v(x) \wedge \mathrm{PURR}_v(x)]$]
= $\mathrm{BELIEVE}_w({}^{\wedge v}\exists x[\mathrm{UNICORN}_v(x) \wedge \mathrm{PURR}_v(x)])$ [def APPLY]

APPLY[$\mathrm{BELIEVE}_w({}^{\wedge v}\exists x[\mathrm{UNICORN}_v(x) \wedge \mathrm{PURR}_v(x)])$, ANNA]
= $\mathrm{BELIEVE}_w({}^{\wedge v}\exists x[\mathrm{UNICORN}_v(x) \wedge \mathrm{PURR}_v(x)])(\mathrm{ANNA})$ [def APPLY]
= $\mathrm{BELIEVE}_w(\mathrm{ANNA}, {}^{\wedge v}\exists x[\mathrm{UNICORN}_v(x) \wedge \mathrm{PURR}_v(x)])$
So we get:



$, \mathrm{BELIEVE}_w(\mathrm{ANNA}, {}^{\wedge v}\exists x[\mathrm{UNICORN}_v(x) \wedge \mathrm{PURR}_v(x)])$

This is the reading on which *a unicorn* is interpreted *de dicto*: the claim that there is a unicorn comes for the account of Anna, not the speaker. We will take about *de re* readings in the next chapter.

(3) Anna seeks and finds a unicorn.

Look at (4a) and (4b)

(4) a. Anna sought and found a unicorn. She is grooming it now.
b. Anna sought and found a unicorn, thought it is not clear that what she found was exactly what she was looking for.

We will give in the next chapter a scope mechanism that will derive two readings for *Anna seeks a unicorn* that together with an interpretation for conjunction will derive the two readings for (4) as well. But even without that we can entertain a simple assumption that will derive these two readings even without a scope mechanism.

We specified in our grammar the interpretation of *seek* as:

seek → SEEK_w of type <<s,<<e,t>t>,<e,t>>

But we also defined $\text{SEEK}_{*,w}$ of type <e,<e,t>>

Let us make the assumption that $\text{SEEK}_{*,w}$ is also an available interpretation of *seek*:

seek → SEEK_w of type <<s,<<e,t>t>,<e,t>>
$\text{SEEK}_{*,w}$ of type <e,<e,t>>

In that case, we get a first reading of (3) by applying conjunction at type <e,<e,t>>, just as we have done before:

seek and find → $\lambda y \lambda x.\text{SEEK}_{*,w}(x,y) \wedge \text{FIND}_w(x,y)$ of type <e,<e,t>>

And this combines in the normal way with the object and the subject, and we derive a *de re* reading:

Anna seeks and finds a unicorn →
$\exists y[\text{UNICORN}_w(y) \wedge \text{SEEK}_{*,w}(\text{ANNA},y) \wedge \text{FIND}_w(\text{ANNA},y)]$

But, of course, this intepretation will also assign a *de re* reading in the simple case:

Anna seeks a unicorn → $\exists y[\text{UNICORN}_w(y) \wedge \text{SEEK}_{*,w}(\text{ANNA},y)]$

What about the other reading?
Here we assume that *seek* is interpreted as the intension type:

seek → $SEEK_w$ of type <<s,<<e,t>t>,<e,t>>
find → $FIND_w$ of type <e,<e,t>>

We have seen how to assign a lower interpretation to *seek*: it derives a *de re* reading.
But, of course, Montague's strategy is to assign a higher interpretation to FIND, and interpretation at the intensional type for *seek*: <<s,<<e,t>t>,<e,t>>.
How do we find that interpretation?
Answer: of course, with backward λ-conversion

Anna finds a unicorn → $\exists y[UNICORN_w(y) \wedge FIND_w(ANNA, y)]$

We abstract the subject out:

find a unicorn → $\lambda x.\exists y[UNICORN_w(y) \wedge FIND_w(x, y)]$

This time we want to solve the equation:

$V({}^{\wedge v}\lambda P.\exists y[UNICORN_v(y) \wedge P(y)]) = \lambda x.\exists y[UNICORN_w(y) \wedge FIND_w(x, y)]$

where V is of type <<s,<<e,t>t>,<e,t>>

The way to solve this is to look at what we had before in the similar extensional case:

$$V = \lambda T\lambda x.T(\lambda y.FIND_w(x,y))$$

In this case the input cannot be T of type <<e,t>,t>, but must be *T* of type <s,<<e,t>,t>>.
So write that:

$$V = \lambda T\, \lambda x.T(\lambda y.FIND_w(x,y))$$

The problem is that this isn't wellformed: *T* is of type <s,<<e,t>,t>>, but $\lambda y.FIND_w(x,y)$ is of type <e,t>.
So we need to add one more thing, to make it wellformed:
you cannot apply intensional function *T* to $\lambda y.FIND_w(x,y)$,
but you can apply the extension of *T* to $\lambda y.FIND_w(x,y)$, so we set:

$V = \lambda T\, \lambda x.T_w(\lambda y.FIND_w(x,y))$ of type <<s,<<e,t>t>,<e,t>>

or equivalently

$$V = \lambda T\, \lambda x.[{}^{\vee w}T]\, (\lambda y.FIND_w(x,y))$$

Thus we add a lifting rule:

> LIFT: $EXP_{<e,<e,t>>} \rightarrow EXP_{<<s,<<e,t>t>,<e,t>>}$
> LIFT[R] = $\lambda T\lambda x.[\ {}^{\vee w}T]\ (\lambda y.FIND_w(x,y))$

And with Boolean conjunction we now derive

seek and find $\rightarrow \lambda T\ \lambda x.SEEK_w(x,T) \wedge T_w(\lambda y.FIND_w(x,y))$

We derive:
seek and find a unicorn $\rightarrow$

$APPLY[\ \lambda T\ \lambda x.SEEK_w(x,T) \wedge T_w(\lambda y.FIND_w(x,y)),\ \lambda P.\exists y[UNICORN_w(y) \wedge P(y)]$

$= APPLY[\lambda T\ \lambda x.SEEK_w(x,T) \wedge T_w(\lambda y.FIND_w(x,y)),\ LIFT[\lambda P.\exists y[UNICORN_w(y) \wedge P(y)]]$

$= APPLY[\lambda T\ \lambda x.SEEK_w(x,T) \wedge T_w(\lambda y.FIND_w(x,y)),\ {}^{\wedge v}\lambda P.\exists y[UNICORN_v(y) \wedge P(y)]]$

$= \lambda T\ \lambda x.SEEK_w(x,T) \wedge T_w(\lambda y.FIND_w(x,y))(\ {}^{\wedge v}\lambda P.\exists y[UNICORN_v(y) \wedge P(y)])$

With λ-conversion we get:

$= \lambda T\ \lambda x.SEEK_w(x,T) \wedge T_w(\lambda y.FIND_w(x,y))(\ {}^{\wedge v}\lambda P.\exists y[UNICORN_v(y) \wedge P(y)])$
$= \lambda x.SEEK_w(x,\ {}^{\wedge v}\lambda P.\exists y[UNICORN_v(y) \wedge P(y)]) \wedge$
$\qquad {}^{\wedge v}\lambda P.\exists y[UNICORN_v(y) \wedge P(y)]_w(\lambda y.FIND_w(x,y))$

[$\quad {}^{\wedge v}\lambda P.\exists y[UNICORN_v(y) \wedge P(y)]_w \quad =$
$\quad \boldsymbol{\lambda v}.\lambda P.\exists y[(UNICORN\boldsymbol{(v)})(y) \wedge P(y)]\boldsymbol{(w)} \quad =$
$\quad \lambda P.\exists y[(UNICORN\boldsymbol{(w)})(y) \wedge P(y)] \quad =$
$\quad \lambda P.\exists y[(UNICORN_w(y) \wedge P(y)]$]

$= \lambda x.SEEK_w(x,\ {}^{\wedge v}\lambda P.\exists y[UNICORN_v(y) \wedge P(y)]) \wedge$
$\qquad \lambda P.\exists y[UNICORN_w(y) \wedge P(y)](\lambda y.FIND_w(x,y))$

The remaining λ-conversions are standard:

seek and find a unicorn $\rightarrow$
$\lambda x.SEEK_w(x,\ {}^{\wedge v}\lambda P.\exists y[UNICORN_v(y) \wedge P(y)]) \wedge \exists y[UNICORN_w(y) \wedge FIND_w(x,y)]$

Anna seeks and finds a unicorn $\rightarrow$
> $SEEK_w(ANNA,\ {}^{\wedge v}\lambda P.\exists y[UNICORN_v(y) \wedge P(y)]) \wedge$
> $\exists y[UNICORN_w(y) \wedge FIND_w(ANNA,y)]$

Anna sees a unicorn (de dicto) and Anna finds a unicorn (de re).