6. Dynamic Interpretation

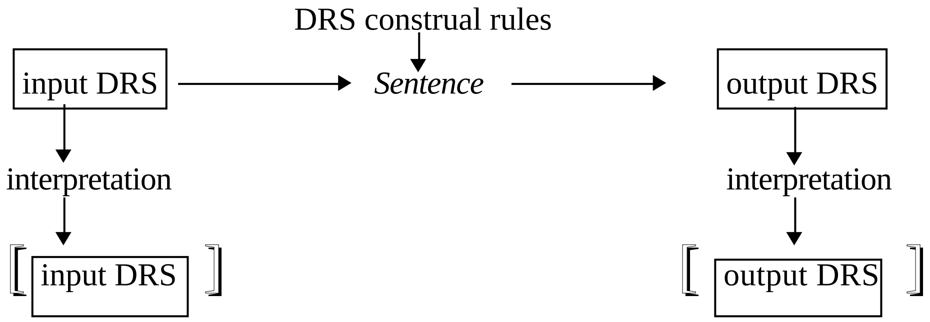
6.1 Introduction

6.1.1 Theoretical Options for Dynamic Interpretation

In the preceding chapter we studied one theory that attempts to handle anaphoric reference and other textual phenomena, namely, DRT. The crucial property of DRT was the introduction of a separate level in semantic theory, namely, a level of **representation** (discourse representation structures). Discourse representation structures were then interpreted in the usual, truth-conditional way. In this chapter we will discuss alternative theories that deal with such phenomena in a more direct way, by changing the way how we see semantic **interpretation**.

In a graph, DRT offered the following model of interpretation:

(1)



That is, the meaning of a sentence changes semantic representations. Specifically, a sentence changes an input DRS to an output DRS. DRSs in turn are expressions in a complex formal language that can be interpreted with respect to a model.

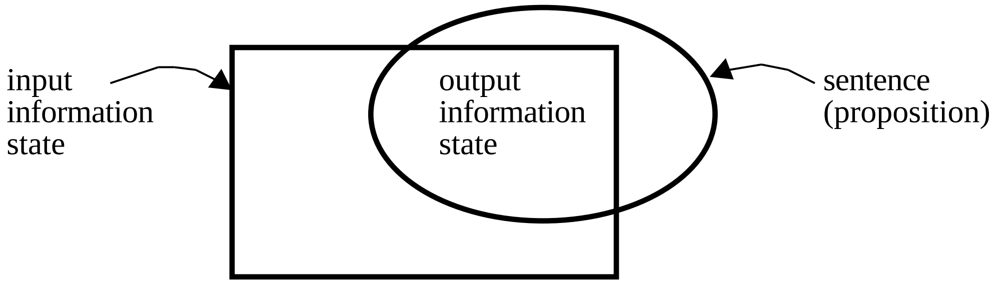
The crucial new thing here in comparison with classical model-theoretic semantics is that sentences are not interpreted directly in terms of truth and falsity. Rather, they are seen as **devices to change information states**. This is a very natural and convincing picture, when you consider the main uses of language, namely, the transfer of information.

However, it is not clear whether a separate level of discourse representation structures is necessary for this purpose. We could incorporate the idea that sentences are information changers in a more direct way, without making reference to a level of semantic representation:

(2)

We can think of the representation of information states in a number of ways. For example, as early as 1972, Robert Stalnaker proposed to model information states as **sets of possible worlds** (the set of possible worlds that are compatible with what a hearer knows). The interpretation of a sentence then adds new information to that state. When we reconstruct information states as sets of possible worlds, then this means that this set is reduced. We can illustrate this easily by Venn-diagrams when we analyze information states and the meaning of sentences both as sets of possible worlds:

(3)



What could be the advantage of this type of direct dynamic interpretation over DRT?

· The architecture of semantic interpretation is somewhat simpler.

· DRT allows for the semantic interpretation of a discourse only after the DRS is constructed. However, there are many cases in which we need "online" semantic information. For example, take a discourse like *Pedro owns a donkey. He beats the poor animal*. In order for *the poor animal* to select the right antecedent, we must assume that the information that donkeys are animals, and that beating is an act that inspires compassion that can be expressed by the adjective *poor*, is available. But this is **semantic** information, for which we probably have to interpret the sentence *Pedro owns a donkey* first, and perhaps even the verb *beats*. One way to avoid that is to assume that all of our semantic knowledge is encoded in DRSs as well.

· Proponents of direct dynamic interpretation point out to the fact that DRT suggests a non-compositional interpretation. Sentences change syntactic objects (DRS's) in rather unrestricted ways. Direct dynamic interpretation, on the other hand, is forced to assume compositional interpetations.

6.1.2 The Notion of Common Ground

Above we have characterized information states as representing what the participants in a conversation know. This is not quite right; a speaker never is fully informed about what the hearer knows. Let us have a closer look at the information states that are relevant in a conversation. We can distinguish between various types of information:

a) The knowledge of the speaker S.

b) The knowledge of the hearer H.

c) The knowledge that S thinks is shared between S and H.

d) The knowledge that H thinks is shared between S and H.

It is important to note that (a), (b), (c) and (d) can be quite different. This is obvious for (a) and (b), (a) and (c), (a) and (d) etc. But even (c) and (d) might differ. If this becomes obvious during a conversation, the participants have the possibility to protest against such assumptions, and to readjust their concepts of mutual knowledge. A case in point is illustrated by the following dialogue:

(3) Speaker A: *I will have a party tomorrow. Do you want to come? Sue's husband will come as well.*
Speaker B: *I didn't know that Sue was married!*
Speaker A: *Yes, they are, they had their wedding a week ago.*

But we can idealize the situation for the time being and assume that (c) and (d) are identical. Note that they are conceived by S and H to be identical, and if it turns out that they aren't, something went wrong and has to be fixed.

The information present in (c) and (d) has been called the **common ground** of speaker and hearer by Robert Stalnaker (1974, 1979); related notions are "pragmatic presupposition", "speaker's presupposition", or simply "context". We can distinguish between at least three parts of a common ground:

a) The common world knowledge of S and H.

b) The common situational knowledge of S and H.

c) The knowledge that has been conveyed in the current discourse up to the current point.

If S utters a declarative sentence Φ with respect to a common ground c, then S tries to modify, typically to enlarge, the common ground c. If H accepts Φ, that is, does not show any sign of protest, S can assume that H accepted the information conveyed by Φ as being part of a new common ground c′. More precisely, c′ will consist of c and the content of Φ (we skip over additional information that might be conveyed by that act, e.g. the information that S uttered the sentence Φ, that H did not protest against it, etc.).

Stalnaker says about this notion of common ground (1979, p. 321, emphasis mine):

> This, I want to suggest, is the central concept needed to characterize speech contexts. Roughly speaking, the presuppositions of a speaker are the propositions whose truth he takes for granted as part of the background of the conversation. A proposition is presupposed if the speaker is disposed to act as if he assumes or believes that the proposition is true, and as if he assumes or believes that his audience assumes or believes that it is true as well. Presuppositions are what is taken by the speaker to be the **common ground** of the participants in the conversation, what is treated as their **common knowledge** or **mutual knowledge**. The propositions presupposed in the intended sense need not really be common or mutual knowledge; the speaker need not even believe them. He may presuppose any proposition that he finds convenient to assume for the purpose of the conversation, provided he is prepared to assume that his audience will assume it along with him.

Stalnaker suggested this type of interpretation to model the **factual content** only. Irene Heim (1982, 1983) and, later, Jeroen Groenendijk & Martin Stokhof, Mats Rooth, and others proposed ways to integrate the notion of **accessible discourse referents** into dynamic interpretation as well.

We will first discuss a framework that models just the interpretation of the factual content. In particular, this will provide us with a way to represent so-called presuppositions. Then we will turn to a framework that models the bookkeeping of accessible discourse referents. Finally, we will develop a framework that combines both.

6.2 Dynamic Interpretation of Factual Contents

6.2.1 The Basics: Sentences, Sentential Connectives, Texts

Let us assume that declarative sentences (and texts consisting of declarative sentences) are functions that map input information states to output information states. Let us also assume that information states are modeled as sets of possible worlds (the set of possible worlds that is compatible with respect to the information state).

I will use S to denote the set of information states, and s, s′ etc. as variables for the elements of S. If information states are modeled as sets of possible worlds, then S is the powerset of W, the set of all possible worlds, and s, s′ are subsets of W:

Modeling of information states:

(4) a. W: Set of possible worlds

b. S = pow(W): set of information states

c. s, s′ etc....: information states (note that s, s′...   W)

d. W: The most general information state
(compatible with everything, i.e. knows nothing at all)

e.   : The absurd information state
(compatible with nothing, i.e. contains contradictory information)

The meaning of a sentence Φ, ⟦Φ⟧, then is a function from information states to information states. We can write this with the usual function notation (a); it turns out that the notation (b) is more convenient. The notation (c) is also quite common; it is to be read as "s updated with the sentence Φ".

(5) Notational variants;
s: input information state, s′: output information state, ⟦Φ⟧: meaning of sentence Φ
a. ⟦Φ⟧(s) = s′
b. s ⟦Φ⟧ = s′
c. s + Φ = s′

We have seen above that there is an easy way to “translate” the old, truth-conditional interpretation of sentences, in which sentences were interpreted as sets of possible worlds, to the new, dynamic interpretation:

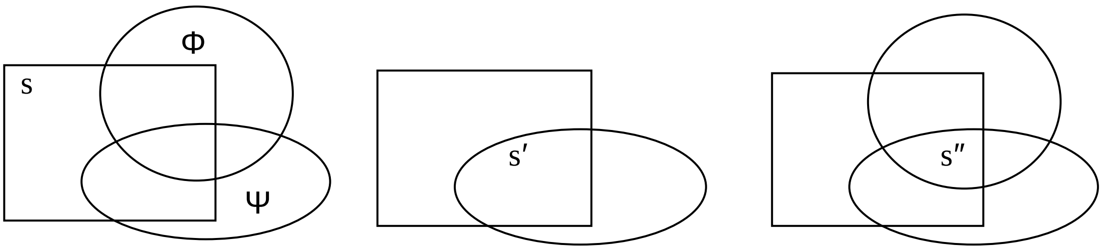
(6) If ⟦Φ⟧* = {w|Φ is true at w}, then ⟦Φ⟧ = λp[p ∩⟦Φ⟧*]

Now we have to specify what the usual sentential connectives mean in this new interpretation framework, where sentences are functions from input states to output states. Conjunction obviously should be defined in the following way:

(7) s + [Φ Ψ] = (s + Φ) + Ψ (simplified: s + Φ + Ψ)

That is, in order to update s with [Φ Ψ], we first update it with Φ, and then with Ψ. Example:

(9)



Input s, sentences Φ, Ψ   s + Φ = s′   s′ + Ψ = s″

We can have the same rule for texts, which can be seen as sequences of sentences:

(8) $s + \Phi_1; \Phi_2; ... \Phi_n = s + \Phi_1 + \Phi_2 + ... + \Phi_n$

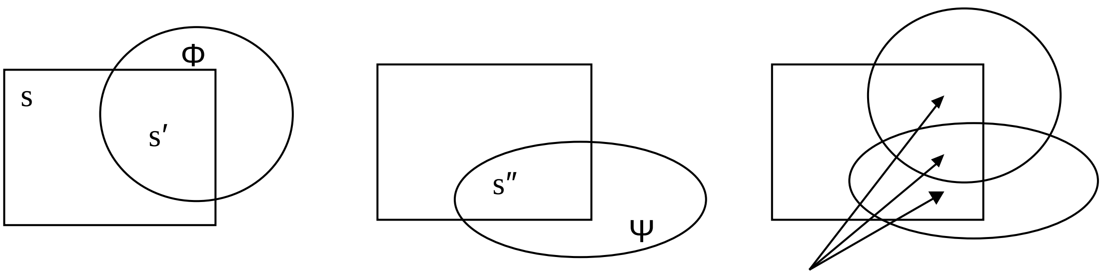
Conjunction under this interpretation is commutative, that is, (Φ Ψ) = (Ψ Φ).

How should we treat **disjunction**? One plausible rule is the following:

(9) s + [φ y] = [s + φ] [s + y]

This assumes that we can form unions on information states. This is certainly possible under the current representation of information states as sets of possible worlds. Example:

(9)

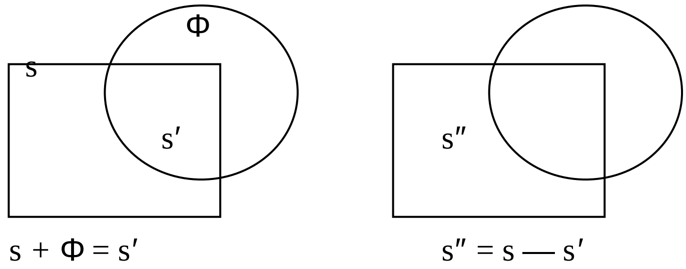


s + Φ = s′   s + Ψ = s″   s′ s″

We can treat **negation** in a similar way, assuming that information states allow for set-theoretic subtraction:

(10) s + [¬Φ] = s — [s + Φ]

(11)



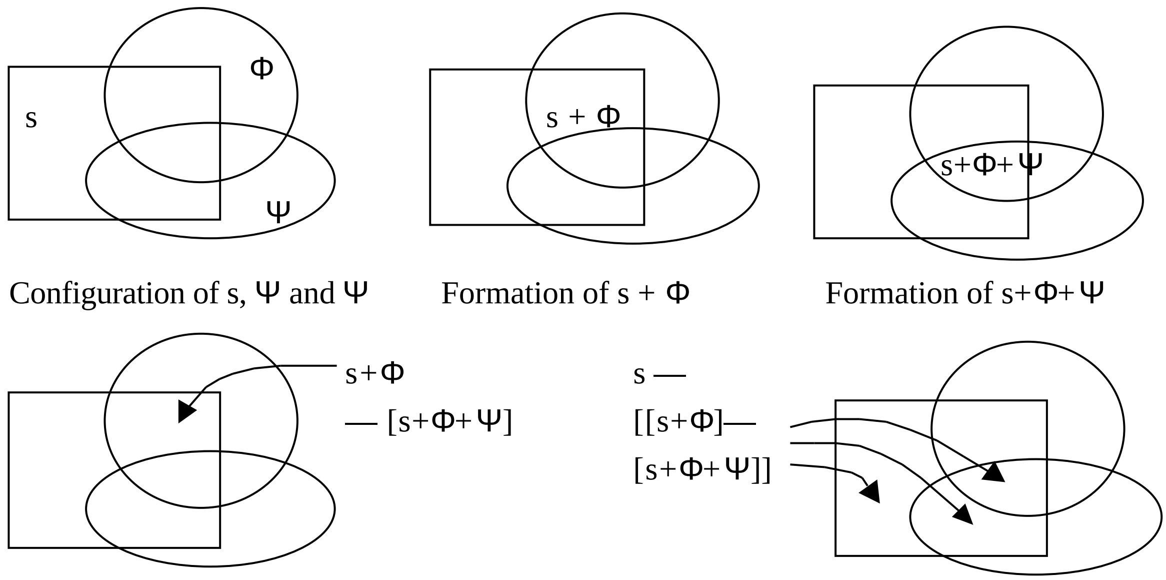
The most complex rule is the one for **conditionals**. It seems that we expect the following result from the conditional interpretation: If s is updated with [$\Phi \rightarrow \Psi$], we get an information state s′ with the following property: Whenever we update s′ further with Φ, the information Ψ is already present. (This is the "Ramsey test for conditionals", after the mathematician Frank Ramsey):

(12) If $s + [\Phi \rightarrow \Psi] = s'$, then s′ contains the following information: $s' + \Phi = [s' + \Phi] + \Psi$

Of course, s′ should not contain more information than just that. The following meaning rule gives us the expected result:

(13) $s + [\Phi \rightarrow \Psi] = s — \big[[s + \Phi] — [[s + \Phi] + \Psi]\big]$

(14)



Note that the resulting context s′ is the **biggest** subset of the original context that satisfies the Ramsey test: $s' + \Phi = s' + \Phi + \Psi$. This means that the resulting information state does not contain any additional information beyond the information contained in the conditional.

So far dynamic interpretation doesn't seem to be all that different from static interpretation. However, dynamic interpretation offers a natural way to capture the semantic impact of **presuppositions**. We will be concerned with presuppositions in the next chapter. Here, we will concentrate on how we can extend the notion of an information state to cover accessible discourse referents as well. We know that this is necessary, for otherwise we could not treat cases like Partee's marble example.

6.3 Dynamic Interpretation and Anaphoric Relationships

6.3.1 Introduction

We have seen that a model of semantic interpretation can be developed in which the truthconditional content of sentences is interpreted dynamically, as mappings from input information states to output information states. Can we also design a dynamic interpretation that deals with the introduction of discourse referents as antecedents of pronominal elements? Various theories of this type have been developed, in particular by Heim (1982, Chapter 3), Heim (1983), Rooth (1987), Staudacher (1987), Groenendijk & Stokhof (1990, 1991), and others. Heim's theory is sometimes called "File Change Semantics", Groenendijk & Stokhof's systems are known as "Dynamic Predicate Logic" and "Dynamic Montague Grammar".

These theories differ in certain features. Here I will discuss one such system that illustrates the essential techniques. The data coverage will not be as broad as in our discussion of Discourse Representation Theory, but it will cover the central phenomena of quantifiers, indefinite NPs, and anaphoric reference. It is quite close to Rooth (1987).

For simplicity I will concentrate on the component that deals with discourse referents. We can later combine dynamic interpretation of the truthconditional content with systems that deal with discourse referents.

6.3.2 Information States with Discourse Referents

So far we have defined information states as sets of possible worlds; this covers the truth-conditional content of sentences. If we want to model the accessibility of discourse referents, we need a different notion of information states. It should provide the following information:

- Which discourse referents are already introduced (and hence can be picked up by pronouns?)
- Which entities are these discourse referents anchored to?

The first requirement suggests that information states contain a **set of accessible discourse referents**. This set can grow larger during a discourse when more discourse referents become accessible. The second requirement suggests that information states contain information about the **entities** these discourse referents are anchored to.

Now, if a discourse referent could be anchored to exactly one entity, then the right way to model information states would be functions from accessible discourse referents to their anchors. But often, there are many possible anchors for a discourse referent. For example, a sentence like *A man came in* is unspecific about the person that the discourse referent introduced by *a man* is anchored to. Therefore we need a set of such functions to model an information state. These functions should all have the same domain, namely, the set of accessible discourse referents. This suggests the following definition:

(15) D is the set of **discourse referents**. D is a countably infinite set.

(16) A, the universe of the model, is a set of **individuals**.

(17) An **information state** with respect to a set of discourse referents D and a universe A
is a set of functions from a set D′, D′ D, to A.
D′ is the set of **accessible discourse referents** of the information state.

Let us assume that we use the set of natural numbers 1, 2, 3... as the set D of discourse referents, and let us assume that the set of entities A is {j, m, c1, c2, p1, p2}. For a function like { 1, j , 2, m } I will write [1 → j, 2 → m]. Then the following set of functions is an example of an information state with respect to D and A.

(18) s = {[1 → j, 2 → m, 3 → c1], [1 → j, 2 → m, 3 → c2]}

The set of accessible discourse referents for s is {1, 2, 3}. The state s represents the state where the discourse referent 1 is mapped to j, the discourse referent 2 is mapped to m, and the discourse referent 3 is mapped to either c1 or c2. If we interpret j as "John", m as "Mary", c1, c2 as cats, and assume that j gave c1 to m and that j gave c2 to m, then this information state could be the output after we have processed the sentence *John*$_1$ *gave Mary*$_2$ *a cat*$_3$ (where syntactic indices are identical to the discourse elements that the pronouns are associated with). That is, when we start with an empty information state , we should get the following result:

(19) + *John*$_1$ *gave Mary*$_2$ *a cat*$_3$ = {[1 → j, 2 → m, 3 → c1], [1 → j, 2 → m, 3 → c2]}

Sentences should be able to change input information states to output states, as before. In this case this can mean one of two things:

· Certain assignment functions can be **eliminated**. Let us assume that c1 is black, and c2 isn't. Then we expect the following change:

(20) s + *It*$_3$ *was black* = {[1 → j, 2 → m, 3 → c1]}

· The domain of an assignment function can be increased. Let us assume that p1, p2 are parrots, and that m has p1 and m has p2. Then we expect the following change:

(21) s + *She*$_2$ *owns a parrot*$_4$ =
{[1 → j, 2 → m, 3 → c1, 4 → p1], [1 → j, 2 → m, 3 → c1, 4 → p2],
[1 → j, 2 → m, 3 → c2, 4 → p1], [1 → j, 2 → m, 3 → c3, 4 → p2]}

Notice that discourse referent 4 can be anchored to either p1 or p2. We have to add each of these two possibilities to the two assignment functions in s, which leads to an information state that contains twice as many elements as before.

One important difference between the information states that just model factual information and the ones that model the increase of the set of discourse referents is that the former ones "decrease" during update (the set of possible worlds gets smaller), whereas the latter ones "increase" during update (the domain of accessible discourse referents gets bigger). However, either one of these processes mean that the updated information state contains more information: it identifies a more specific set of possible worlds, and it contains more discourse referents.

6.3.3 The Interpretation of Sentences

As before, we want to interpret sentences as functions from information states to information states. There are various options how this idea can be implemented. They differ in technical matters but not in their spirit.

The most natural option seems to be to model sentences directly as functions from information states to information states. It turns out that this leads to a rather clumsy notation when we are concerned with the semantic representation of sub-sentential expressions.

Another option is to analyze sentences as **relations from input assignment functions to output assignment functions**. When we update an information state s with a sentence Φ, we effectively have to apply this relation to all the elements of s and collect their output in a new set. That is, we apply the relation element-wise to the input state.

(22) a. A sentence Φ is interpreted as a relation R(Φ) between partial assignments.

b. s + Φ = {g | f[f s f, g R(Φ)]}

Let us illustrate this procedure with the two examples we used above. Recall that out model fixes the meaning of the predicate *black* to {c1}. Let us write g_d for the value of the application of an assignment function g to a discourse referent d, instead of g(d).

(23) a. R(*It*$_3$ *was black*) = { g, g | 3 DOM(g) g_3 F(*black*)}

b. s + *It*$_3$ *was black* = {g | f[f s f, g { g, g | 3 DOM(g) g_3 = c1}]}
= {[1 → j, 2 → m, 3 → c1]}

In this case the input assignment and the output assignment are the same (g), but the restriction is that g maps the index 3 to c1. This applies to only one of the elements of s, hence s is reduced. Sentences that do not increase the number of accessible discourse referents, but possibly reduce the number of assignment functions, are called **tests**.

In the second case, the set s is increased, and the individual assignment functions are enlarged as well. Let us introduce the following notation:

(24) If f, g are two partial assignment functions and d is a discourse referent, then f $<_d$ g iff
(i) d DOM(f);
(ii) DOM(g) = DOM(f) {d}
(iii) For all d′ DOM(g): f(d′) = g(d′).

That is, f $<_d$ g expresses that f and g differ only insofar as g maps an additional discourse referent d to some object.

Let us assume as before that our model specifies that Mary has two parrots, p1 and p2.

(25) a. R(*She*$_1$ *owns a parrot*$_4$)
= { f, g | 1 DOM(f) f $<_4$ g g_4 F(*parrot*) g_1, g_4 F(*owns*)}

b. s + *She*$_1$ *owns a parrot*$_4$
= {g | f[f s f, g { f, g | 1 DOM(f) f $<_4$ g g_4 F(*parrot*)
g_1, g_4 F(*owns*)}]}
= {[1 → j, 2 → m, 3 → c1, 4 → p1], [1 → j, 2 → m, 3 → c1, 4 → p2],
[1 → j, 2 → m, 3 → c2, 4 → p1], [1 → j, 2 → m, 3 → c3, 4 → p2]}

It seems that we have found a satisfying semantic interpretation scheme, at least as far as discourse referents are concerned. Now let us discuss the ways how complex sentences are build up by predicates, indefinite NPs, pronouns, and quantificational elements.

6.3.4 Syntactic Rules and Semantic Types

I will illustrate the principles of dynamic interpretation with a very small toy grammar. It has the following syntactic rules, specified as phrase structure rules:

(26) S → NP[s] VP   VP → V NP[o]   NP[α] → DET$_n$[α] N
N → N PP   PP → P NP[o]
D → S D   D → S
where n ranges over natural numbers.

D stands for “discourse”; a discourse consists of a sentence followed by a discourse, or (as the limiting case) just a sentence. The attached features [s], [o] stand for “subject” (or nominative) and “object” (or accusative); I suppressed features when not needed.

We assume lexical entries of the following type. Here, the index n stands for any natural number; this index will be related to the discourse referent.

(27) [$_{NPn}$*Pedro*], [$_{NPn}$*Chiquita*], [$_{NPn}$*he*], [$_{NPn}$*she*], [$_{NPn}$*it*],
[$_{VP}$*walked in*], [$_{VP}$*sat down*], [$_V$*kicked*], [$_V$*beat*],
[$_N$*farmer*], [$_N$*donkey*], [$_N$*widow*]
[$_P$*with*], [$_{DETn}$*a*], [$_{DETn}$*every*], [$_{DETn}$*no*]

This syntax will generate texts like the following, which illustrates all the relevant phenomena (indefinites, pronouns, quantifiers, indefinites in the restrictor of quantifiers, pronouns in the nuclear scope of quantifiers). In the following representation, indices are attached to the expressions that introduce them.

(28) A$_1$ donkey walked in. It$_1$ sat down. Every$_2$ farmer with a$_3$ donkey beat it$_3$.

This fragment contains prepositional phrases like *with a donkey*, not relative clauses like *that has a donkey*, as they are syntactically simpler than relative clauses.

Let us turn to the semantic interpretation. We intend to build a compositional semantic interpretation for this grammar. That is, the meaning rules will have, in general, the following format (illustrated with a binary branching construction):

(29) If α has as a meaning α',
and β has a meaning β',
and there is a syntactic rule SYN of the form $\gamma \rightarrow \alpha\ \beta$,
then there is a corresponding semantic rule SEM that specifies the meanings of the complex expression $[\![[_\gamma\ \alpha\ \beta]]\!]$ as a function of α', β' and SYN.

The toy grammar we have introduced above is not ambiguous, which simplifies things considerably, as every expression will have a unique interpretation. Furthermore, the semantic combination rule usually follows from the semantic types of the meanings to be combined. This allows us to write down interpretation rules in the following format:

(30) $[\![[_\gamma\ \alpha\ \beta]]\!]$ = SEM($[\![\alpha]\!]$, $[\![\beta]\!]$),
where SEM is functional composition, in most cases.

The **semantic type system** is defined as follows. We distinguish between predicate types (this includes sentences, as 0-place predicates), and functional types.

(31) a. t is a predicate type (sentence).
b. If τ is predicate type, then eτ is a predicate type.
c. If σ and τ are types, then $\sigma\tau$ is functional type.

The denotations of a certain type are constructed in the following way. Let us start with predicate types:

· The possible denotations of type t (sentences) are two-place relations between assignments. They are typically denoted in the format { f, g | ...}).

· The possible denotations of type et (one-place predicates) are three-place relations between two assignments and an entity. They are typically denoted by { f, x, g | ...}.

· The possible denotations of type *eet* (two-place predicates) are four-place relations between two assignments and two entities. They are typically denoted in the format { f, x, y, g | ...}.

The denotations of functional types are defined as usual:

(32) If D_σ is the set of denotations of type σ, and D_τ is the set of denotations of type τ,
then $[D_\sigma \rightarrow D_\tau]$, the set of functions from D_σ to D_τ, is the set of denotations of type $\sigma\tau$.

Denotations of functional types are typically rendered by lambda expressions. For example, expressions of type et, t (subject NPs) will be given in the format λP{ f, g |...P...}, where P is a variable of the predicate type et. And expressions of type (*eet*)*et* (object NPs) will be given in the format λR{ f, x, g |...R...}, where R is a variable of type *eet*.

As usual, we assume a model M = A, F , where A is a universe, and F is a semantic interpretation function for constants

Now we are equipped for the first semantic rules. We start with sentences without quantifiers.

6.3.5 Pronouns, Names, and Simple Sentences

Nouns and intransitive verbs (type et) are interpreted as follows:

(33) If α is a lexical item of categories N or VP, then $[\![\alpha]\!]$ = { g, x, g | x F(α)}

Transitive verbs are interpreted in this way:

(34) If α is a lexical item of category V, then $[\![\alpha]\!]$ = { g, x, y, g | x, y F(α)}

Notice that the input assignment and the output assignment are identical (g). This reflects the fact that nouns or verbs do not introduce any discourse referents.

Notice, also, that we distinguish between a "dynamic" meaning and a "static" meaning of basic expressions. $[\![\alpha]\!]$ is the dynamic meaning, F(α) the static meaning. Instead of F(α), I will write $\underline{\alpha}$, and instead of x F(α), I will write $\underline{\alpha}$(x).

NPs have the task of reducing an n-place predicate to an (n–1)-place predicate. Let us start with pronouns, in subject and object position:

(35) If $[_{NPn}\alpha]$ is a pronoun, then $[\![[_{NP[s]n}\alpha]]\!] = \lambda P\{\ f, g \mid f, f_n, g \quad P\}$
and $[\![[_{NP[o]n}\alpha]]\!] = \lambda R\{\ f, x, g \mid\ f, x, f_n, g \quad R\}$

Notice that pronouns do not introduce new discourse referents. They presuppose, however, that the index n of the pronoun is in the domain of g.

We can already derive a little sentence:

(36) $[\![[_S[_{NP[s]1}\textit{it}]\ [_{VP}\textit{sat down}]]]\!]$
$[_{NP[s]1}\textit{it}]$; $\lambda P\{\ f, g \mid\ f, f_1, g \quad P\}$
½
½ $[_{VP}\textit{sat down}]$; $\{\ g, x, g \mid \underline{\text{sat down}}(x)\}$
½
$[_S\textit{it sat down}]$;
$\lambda P\{\ f, g \mid\ f, f_1, g \quad P\}(\{\ g, x, g \mid \underline{\text{sat down}}(x)\})$
$= \{\ f, g \mid f, f_1, g \quad \{\ g, x, g \mid \underline{\text{sat down}}(x)\}\}$
$= \{\ g, g \mid \underline{\text{sat down}}(g_1)\}$

Neither the pronoun nor the predicate introduces new discourse referents. Hence we end up with a relation that holds for identical assignments g, g, provided that g_1 has the property *sat down*.

Names are slightly more complex than pronouns, as they do introduce new discourse entities.

(37) If $[_{NPn}\alpha]$ is a name, then $[\![[_{NP[s]n}\alpha]]\!] = \lambda P\{\ f, h \mid\ g[f <_n g \quad g_n = \underline{\alpha} \quad g, g_n, h \quad P]\}$
and $[\![[_{NP[o]n}\alpha]]\!] = \lambda R\{\ f, x, h \mid\ g[f <_n g \quad g_n = \underline{\alpha} \quad g, x, g_n, h \quad R]\}$

The following derivation contains a name, a pronoun, and a transitive verb:

(38) $[_{NP[o]}\ \textit{it}]$; $\lambda R\{\ f, x, g \mid\ f, x, f_1, g \quad R\}$
½
½ $[_V\textit{kicked}]$; $\{\ g, x, y, g \mid \underline{\text{kicked}}(x,y)\}$
½
$[_{VP}\textit{kicked it}_1]$;
$\lambda R\{\ f, x, g \mid\ f, x, f_1, g \quad R\}(\{\ g, x, y, g \mid \underline{\text{kicked}}(x, y)\})$
$= \{\ f, x, g \mid\ f, x, f_1, g \quad \{\ g, x, y, g \mid \underline{\text{kicked}}(x, y)\}\}$
$= \{\ g,x,g \mid \underline{\text{kicked}}(x,g_1)$
½
½ $[_{NP[s]4}\textit{Pedro}]$; $\lambda P\{\ f, h \mid\ g[f <_4 g \quad g_4 = \underline{\text{Pedro}} \quad g, g_4, h \quad P]\}$
½
$[_S\textit{Pedro}_4\ \textit{kicked it}_1]$;
$\lambda P\{\ f, h \mid\ g[f <_4 g \quad g_4 = \underline{\text{Pedro}} \quad g, g_4, h \quad P]\}(\{\ g, x, g \mid \underline{\text{kicked}}(x, g_1)\})$
$= \{\ f, h \mid\ g[f\ _4 g \quad g_4 = \underline{\text{Pedro}} \quad g, g_4, h \quad \{\ g, x, g \mid \underline{\text{kicked}}(x, g_1)\}]\}$
$=$ (as g = h): $\{\ f, g \mid f <_4 g \quad g_4 = \underline{\text{Pedro}} \quad \underline{\text{kicked}}(g_4, g_1)\}$

This relates input assignments f to output assignments g,

- where g is an extension of f for the discourse referent 4,
- g (and hence f) are defined for the discourse referent 1,
- g_4 is Pedro, and g_4 kicked g_1.

6.3.6 Indefinites

Let us now come to indefinite NPs. They are build by an indefinite determiner, which carries an index, and a noun.

(39) a. $[\![[_{DET[s]n}\textit{a}]]\!] = \lambda P'\lambda P\{\ f, i \mid\ g\ \ h[f <_n g \quad g, g_n, h \quad P' \quad h, h_n, i \quad P]\}$

b. $[\![[_{DET[o]n}\textit{a}]]\!] = \lambda P'\lambda R\{\ f, x, i \mid\ g\ \ h[f <_n g \quad [\ g, g_n, h \quad P' \quad h, x, h_n, i \quad R]\}$

Application of an indefinite article to a noun gives us the following result:

(40) [$_{DET[s]1}$*a*]; λP′λP{ f, i | g[f $<_1$ g h[g, g_1, h P′ h, h_1, i P]]}
½
½ [$_N$*donkey*]; { g, x, g | <u>donkey</u>(x)}
½
[$_{NP[s]}$*a*$_1$ *donkey*],
λP′λP{ f, i | g[f $<_1$ g h[g, g_1, h P′ h, h_1, i P]]}({ g, x, g | <u>donkey</u>(x)})
= λP{ f, i | g[f $<_1$ g h[g, g_1, h { g, x, g | <u>donkey</u>(x)} h, h_1, i P]]}
= (as g = h): λP{ f, i | g[f $<_1$ g <u>donkey</u>(g_1) g, g_1, i P]}

When we combine the resulting subject NP with an intransitive verb we end up with the following representation:

(41) [$_{NP[s]}$*a*$_1$ *donkey*]; λP{ f, h | g[f $<_1$ g <u>donkey</u>(g_1) g, g_1, h P]}
½
½ [$_{VP}$*came in*]; { g, x, g | <u>came in</u>(x)}
½
[$_S$*a*$_1$ *donkey came in*];
λP{ f, h | g[f $<_1$ g <u>donkey</u>(g_1) g, g_1, h P]}({ g, x, g | <u>came in</u>(x)})
= { f, h | g[f $<_1$ g <u>donkey</u>(g_1) g, g_1, h { g, x, g | <u>came in</u>(x)}]}
= (as g = h): { f, g | f $<_1$ g <u>donkey</u>(g_1) <u>came in</u>(g_1)}

This changes an input assignment f to an output assignment g, where g is defined for the discourse referent 1, and for which it holds that g_1 is a donkey and g_1 sat down.

6.3.7 Discourses

As with our fist model for dynamic interpretation, we can assume that a discourse (a sequence of sentences) leads to a gradual change of the input state. More precisely, when updating s with a discourse consisting of a sentence Φ and the following discourse Ψ (which may be complex again or consist of just one sentence), we first update s with Φ, and we update the result with Ψ:

(42) s + [$_D$ [$_S$ Φ] [$_D$ Ψ]] =
= s + [$_S$ Φ] + [$_D$ Ψ]
= {g | f[f s f, g R(Φ)]} + [$_D$ Ψ]
= {h | g[g {g | f[f s f, g R(Φ)]} g, h R(Ψ)}
= {h | f g[f s f, g R(Φ) g, h R(Ψ)}

Hence we get as output state the set of all assignments h such that there is an assignment f in the input state that is changed to an assignment g by the first sentence Φ, which in turn is changed to h by the second part Ψ. This represents the **incremental** interpretation of a discourse.

This procedure could have been performed in a slightly different way, by first **composing** the two relations R(Φ) and R(Ψ) in the usual way:

(43) R(Φ) composed with R(Ψ), written perversely as R(Ψ) oR(Φ):
R(Ψ) oR(Φ) = { f, h | g[f, g R(Φ) g, h R(Ψ)]}

And then we could have applied the result to the input state s:

(44) s + { f, h | g[f, g R(Φ) g, h R(Ψ)]}
= {h | f[f s f, h { f, h | g[f, g R(Φ) g, h R(Ψ)]}]}
= {h | f g[f s f, g R(Φ) g, h R(Ψ)}

Notice that we arrive at the same result as before.

This allows us to give a semantic rule for discourses that interprets whole discourses as relations between input assignments and output assignments:

(45) a. ⟦[$_D$ Φ Ψ]⟧ = { f, h | g[f, g ⟦Φ⟧ g, h **[Ψ]**]}

b. ⟦[$_D$ Φ]⟧ = ⟦Φ⟧

Example derivation:

(46) ⟦[$_D$[$_S$*a*$_1$ *donkey came in.*] [$_D$[$_S$*it*$_1$ *sat down*]]]⟧

[$_S$*a*$_1$ *donkey came in.*];
{ f, g | f $<_1$ g donkey(g_1) came in(g_1)}
½
½ [$_S$*it*$_1$ *sat down*];
½ [$_D$*it*$_1$ *sat down*]; { g, g | sat down(g_1)}
½
[$_D$*a*$_1$ *donkey*$_2$ *came in. it*$_1$ *sat down.*];
= { f, h | g[f, g { f, g | f $<_1$ g donkey(g_1) came in(g_1)}
g, h { g, g | sat down(g_1)}]}
= { f, g | f $<_1$ g donkey(g_1) & came in(g_1) sat down(g_1)}

We arrive at a relation between input functions f and output functions g, where g differs from f only insofar as it maps the discourse referent 1 to an entity x, and it holds that x is a donkey, x came in, and x sat down.

6.3.8 Rules for PPs

The following rules give the interpretation of PPs and the way they are combined with Ns.

(47) a. ⟦[$_{PP}$ [$_P$α] [$_{NP}$β]]⟧ = ⟦α⟧(⟦β⟧)

b. ⟦[$_N$ [$_N$α] [$_{PP}$ β]]⟧ = ⟦β⟧(⟦α⟧)

The preposition *with* is interpreted as follows. T is a variable over object NP meanings (it stands for the governed NP), P is, as before, a variable over noun meanings (it stands for the noun the PP is attached to). Notice that the static meaning of *with* is similar to the static meaning of a transitive verb; it is roughly equivalent to *own*.

(48) ⟦*with*⟧ = λTλP{ f, x, h | g[f, x, g T({ f, x, y, f | with(x, y)}) g, x, h P]}

To see how things work out, consider the following derivation:

(49) ⟦[$_S$[$_{NP}$*a*$_1$ [$_N$*farmer* [$_{PP}$*with* [$_{NP}$*a*$_2$ *donkey*]]]] [$_{VP}$[$_V$*beat*][$_{NP}$*it*$_2$]]]]⟧

[$_{NP[o]}$*a*$_2$ *donkey*]; λR{ f, x, i | g[f$<_2$g donkey(g_2) g, x, g_2, i R]}
½
½[$_P$*with*]; λTλP{ f, x, h | g[f, x, g T({ f, x, y, f | with(x, y) g, x, h P]}
½

[$_{PP}$ *with a*$_2$ *donkey*];
λTλP{ f, x, h | g[f, x, g T({ f, x, y, f | with(x, y)}) g, x, h P]}
(λR{ f, x, i | g[f$<_2$g donkey(g_2) g, x, g_2, i R]})

= λP{ f, x, h | g[f, x, g [λR{ f, x, i | g[f$<_2$g donkey(g_2)
g, x, g_2, i R]}({ f, x, y, f | with(x, y)})] g, x, h P]}

= λP{ f, x, h | g[f$<_2$g donkey(g_2) with(x, g_2) g, x, h P]}
½
½ [$_N$*farmer*]; { g, x, g | farmer(x)}
½
[$_N$ *farmer with a*$_2$ *donkey*];
λP{ f, x, h | g[f$<_2$g donkey(g_2) with(x, g_2) g, x, h P]}({ g, x, g | farmer(x)})

= { f, x, g | f$<_2$g donkey(g_2) with(x, g_2) farmer(x)}
½
½ [$_{DET[s]1}$*a*]; λP′λP{ f, i | g[f$<_1$g h[g, g_1, h P′ h, h_1, i P]]}
½

[$_{NP}$ *a*$_1$ *farmer with a*$_2$ *donkey*];
λP′λP{ f, i | g[f$<_1$g h[g, g_1, h P′ h, h_1, i P]]}
({ f, x, g | f$<_2$g donkey(g_2) with(x, g_2) farmer(x)})

$= \lambda P\{\ f, h \mid\ \ g[f<_{1,2}g\ \ \underline{donkey}(g_2)\ \ \underline{with}(g_1, g_2)\ \ \underline{farmer}(g_1)\ \ \ g, g_1, h\ \ \ P]\}$

½

½ $[_{VP}$*beat it*$_2]$; $\{\ g, x, g\ \mid \underline{beat}(x, g_2)\}$

½

$[_S$ *a*$_1$ *farmer with a*$_2$ *donkey beat it*$_2]$;

$\lambda P\{\ f, h \mid\ \ g[f<_{1,2}g\ \ \underline{donkey}(g_2)\ \ \underline{with}(g_1, g_2)\ \ \underline{farmer}(g_1)$
$g, g_1, h\ \ \ P]\}(\{\ g, x, g\ \mid \underline{beat}(x, g_2)\})$

$= \{\ f, g \mid f<_{1,2}g\ \ \underline{donkey}(g_2)\ \ \underline{with}(g_1, g_2)\ \ \underline{farmer}(g_1)\ \ \underline{beat}(g_1, g_2)]\}$

We end up with a relation between input assignments f and output assignments g that extend f for the discourse referents 1 and 2 such that g_1 is a farmer, g_2 is a donkey, g_1 is with g_2, and g_1 beats g_2.

6.3.9 Quantificational NPs

In Generalized Quantifier theory, a quantificational determiner is analyzed as a two-place relation between sets, type (*et*)(*et*)*t*. Example:

(50) ⟦*every*⟧ = $\lambda X \lambda Y[X\ \ Y]$,

⟦*farmer*⟧ = {x | x is a farmer}

⟦*slept*⟧ = {x | x slept}

⟦*every*⟧(⟦*farmer*⟧)(⟦*slept*⟧) $= \lambda X \lambda Y[X\ \ Y](\{x \mid x \text{ is a farmer}\})(\{x \mid x \text{ slept}\})$
$= \{x \mid x \text{ is a farmer}\}\ \ \ \{x \mid\ x \text{ slept}\}$

This takes care of the truth-conditional aspects. Within dynamic interpretation, we also have to take care of the facts that:

- discourse referents introduced in the restrictor (the first argument) are accessible in the nuclear scope (the second argument);
- discourse referents introduced in the restrictor or the nuclear scope are not accessible from outside;
- previously introduced discourse referents are accessible in the restrictor and the nuclear scope;
- with nominal quantifiers, the variable associated with the head noun is the “boss”, that is, it is relevant to determine whether the quantified sentence is true or not.

The following interpretation type (here exemplified with the subject quantifier *every*) gives us what we want. Here, EVERY stands for the non-dynamic quantifier $\lambda Y \lambda X[Y\ \ \ X]$. The rule corresponds to the DRT interpretation rule for asymmetric weak quantification (cf. the chapter on DRT).

(51) ⟦$[_{DET[s],\, n}$*every*⟧ =
$\lambda Q \lambda P\{\ f, f \mid \text{EVERY}(\{x \mid\ \ g\ \ h[f<_n g\ \ \ g_n = x\ \ \ g, g_n, h\ \ \ Q]\}$
$(\{x \mid\ \ g\ \ h[f<_n g\ \ \ g_n = x\ \ \ g, g_n, h\ \ \ Q\ \ \ i[\ h, g_n, i\ \ \ P]\}$

Notice, in particular, that the quantifier interpretation prevents the introduction of permanent discourse referents within its scope (as the input assignment and the output assignment are the same, f). Discourse referents introduced within the restrictor (that is, in the assignments g and h) are accessible for the consequent (P).

(52) every$_1$,
$\lambda Q \lambda P\{\ f, f \mid \text{EVERY}(\{x \mid\ \ g\ \ h[f<_1 g\ \ \ g_1 = x\ \ \ g, g_1, h\ \ \ Q]\})$
$(\{x \mid\ \ g\ \ h[f<_1 g\ \ \ g_1 = x\ \ \ g, g_1, h\ \ \ Q\ \ \ i[\ h, g_1, i\ \ \ P]\})$

½

½ $[_N$ *farmer with a*$_2$ *donkey*];

½ $\{\ f, x, g\ \mid f<_2 g\ \ \underline{donkey}(g_2)\ \ \underline{with}(x, g_2)\ \ \underline{farmer}(x)\}$

½

$[_{NP}$*every*$_1$ *farmer with a*$_2$ *donkey*]
λP{ f, f | EVERY({x | g h[f<$_1$g g$_1$ = x
g, g$_1$, h { f, x, g | f<$_2$g donkey(g$_2$) with(x, g$_2$) farmer(x)}]})
({x | g h[f<$_1$g g$_1$ = x g, g$_1$, h { f, x, g | f<$_2$g donkey(g$_2$) with(x, g$_2$)
farmer(x)} i[h, g$_1$, i P]})
½
½[$_{VP}$ *beat it*$_2$]
½ { g, x, g | beat(x, g$_2$)}
½
{ f, f | EVERY({x | g[f<$_{1,2}$g g$_1$ = x donkey(g$_2$) with(g$_1$, g$_2$) farmer(g$_1$)]})
({x | g[f<$_{1,2}$g g$_1$ = x donkey(g$_2$) with(g$_1$, g$_2$) farmer(g$_1$)} beat(g$_1$, g$_2$]})

We get the intended interpretation: Every x (x = g_1) such that there is a donkey g_2, where g_1 is with g_2 and g_1 is a farmer, is such that there is a donkey g_2 where g_1 is with g_2, and g_1 is a farmer, and g_1 beat g_2.

In case of a quantificational NP in object position or as an object of a preposition, we abstract over the subject argument and pass it along to the level of the VP:

(53) [[$_{DET[o], n}$*every*]] =
λQλR{ f, x, f | EVERY({y | g[f<$_n$g g$_n$ = y h[g, g$_n$, h Q]]}
{y | g[f<$_n$g g$_n$ = y h[g, g$_n$, h Q i[h, x, g$_n$, i R]]}

6.3.10 A new look at Conservativity

The interpretation rules for quantified statements we have discussed so far all make use of an important property of natural-languages quantifiers, namely, that they are **conservative**. Recall:

(54) A quantifier Q is conservative iff for all A, B it holds: $Q(A)(B) \Leftrightarrow Q(A)(A \cap B)$.

For example, *most* is conservative. We have, for example, *most girls sang* iff *most girls are girls that sang*. The role of conservativity is that it restricts the entities we have to consider in evaluating a quantifier to the entities that fall under the set A. For example, to evaluate *most girls sang* we just have to consider the girls in our model; non-girls are irrelevant for the truth-conditions of this sentence. An example of a non-conservative quantifier would be a hypothetical quantifier Q such that Q(A)(B) iff #(B — A) = 3. that is, if there are exactly three elements in B that are not A. But such quantifiers do not exist in natural languages.

The defining property for conservative qantifiers is that we can conjoin the restrictor set A to the nuclear scope set B: $Q(A)(B) \Leftrightarrow Q(A)(A \cap B)$. But this is precisely what we have done with interpretation rules like #). Notice that we refer to the antecedent Q in the restrictor argument of the quantifier, and to the conjunction of the antecedent Q and the consequent P in its nuclear scope.

Hence it seems that conservativity is not an accidental property of natural-language quantifiers. Rather, conservative quantifiers are inherently simpler than non-conservative ones when we consider their dynamic interpretation.

Exercises

1. Derive a step-by-step interpretation of the two sentences (a) *Pedro*$_1$ *owns a*$_2$ *donkey* and (b) *He*$_1$ *beats it*$_2$. in a framework with dynamic interpretation .

2. Assume the following model:
A = {p, d1, d2, d3},
F(*Pedro*) = p, F(*donkey*) = {d1, d2, d3},
F(*own*) = { p, d1 , p, d2 }, F(*beat*) = { p, d2 , p, d3 }.
And assume an empty input information state s, s = .

a) What is the output information state s′ if s is updated by (a), *Pedro*$_1$ *owns a*$_2$ *donkey*?
b) What is the output information state s″ if s′ is updated by (b), *He*$_1$ *beats it*$_2$?

3. In (51) we have given the interpretation rule of *every* for the weak interpretation. Give the interpretation rule for the strong interpretation.

6.4 Factual Content and Anaphoric Relationship Combined

In section 6.2, we have developed a dynamic interpretation that captured the factual content of sentences and texts. In section 6.3, we were concerned with a version of dynamic interpretation that captures the accessibility of discourse referents. The next obvious step is a combination of these two factors.

One way of doing this is as follows: The notion of an information state is a set of pairs of a possible world and a function from discourse referents to entities in the model. The possible worlds capture the factual information, and the function captures the accessible discourse referents and their interpretation. The meaning of a sentence is a relation between an input information state and an output information state, as before, but now the notion of an information state is slightly more complex.

The notions of a model of interpretation and of assignment functions is defined as follows:

(55) a. Model: A, W, F ,
where A: set of individuals, W: set of possible worlds,
F: static interpretation for basic expressions.

b. Assignment functions: partial functions from D to A,
where D is the set of discourse referents (here: natural numbers).

The static interpretation of a basic expression α is with respect to a possible world. For example, F(*farmer*)(w) gives us the set of all farmers in world w. I will write $\underline{\text{farmer}}_w$, for short.

Now we can define a new notion of information state and the notions of the domain of accessible discourse referents and of the factual content of an information state:

(56) a. An **information state** is a set s of pairs g, w ,
where g is an assignment function, and w is a set of possible worlds,
such that for all g, w , g′, w s, DOM(g) = DOM(g′).

b. The **domain** of an information state s is the domain of its assignment functions:
DOM(s) = DOM(g), if g, w s.

c. The **factual content** of an information state s is the set of its possible worlds:
FAC(s) = {w | g[g, w s]}

An information state is not just a set of assignment functions or a set of possible worlds, but a combination of assignment functions and possible worlds.

The interpretation rules in this new framework are rather straightforward:

(57) a. If α is a lexical item of categories N or VP,
then ⟦α⟧ = { g,w , x, g,w | x F(α)(w)}

b. If α is a lexical item of category V,
then ⟦α⟧ = { g,w , x, y, g,w | x, y F(α)(w)}

c. If α is a pronoun,
then $[\![[_{NP[s],n}\ \alpha\ ]]\!]$ = λP{ f,w , g,w | f,w , f_n, g,w P}

d. If α is a name, then
$[\![[_{NP[s],n}\ \alpha\ ]]\!]$ = λP{ f,w , h,w | g[f $<_n$ g g_n = F(α)(w) g,w , g_n, h,w P]}

e. $[\![[_{DET[s]n}\ a\ ]]\!]$ = λP′λP{ f,w , i,w | g h[f $<_n$ g g,w , g_n, h,w P′
h,w , h_n, i,w P]}

f. ⟦*with*⟧ = λTλP{ f,w , x, h,w |
g[f,w , x, g,w T({ f,w , x, y, f,w | F(*with*)(w)})
g,w , x, h,w P]}

g. ⟦[$_{DET[s],n}$ *every*]⟧ =
λP′λP{ f,w , f,w |
EVERY({x | g h[f $<_n$ g g_n = x g,w , g_n, h,w P′]})
({x | g h i[f $<_n$ g g_n = x g,w , g_n, h,w P′
h,w , h_n, i,w P]})}

h. ⟦[$_D$ Φ Ψ]⟧ = { f,w , h,w | g[f,w , g,w ⟦Φ⟧ g,w , h,w ⟦Ψ⟧]}

The update of an information state s with a sentence or text Φ is defined as follows:

(58) s + Φ (with respect to an interpretation ⟦ . ⟧)
= { g,w | f[f,w s f, w , g,w ⟦Φ⟧]}

Let us consider a very simple example:

(59) [$_{NP[s],1}$ *Pedro*]; = λP{ f,w , h,w | g[f $<_1$ g g_1 = <u>Pedro</u>$_w$ g,w , g_1, h,w P}
½
½ [$_{VP}$*came in*]; { g,w , x, g,w | <u>came in</u>$_w$(x)}
½
[$_S$*Pedro*$_1$ *came in*];
λP{ f,w , h,w | g[f $<_1$ g g_1 = <u>Pedro</u>$_w$ g,w , g_1, h,w P]}
({ g,w , x, g,w | <u>came in</u>$_w$(x)})

= { f,w , h,w |
g[f $<_1$ g g_1 = <u>Pedro</u>$_w$ g,w , g_1, h,w { g,w , x, g,w | <u>came in</u>$_w$(x)}]}

= { f,w , h,w | g[f $<_1$ g g_1 = <u>Pedro</u>$_w$ g = h <u>came in</u>$_w$(g_1)]}

= { f,w , g,w | f $<_1$ g g_1 = <u>Pedro</u>$_w$ <u>came in</u>$_w$(g_1)}

When we apply this dynamic meaning to an information state s, we get the following result:

(60) s + [$_S$*Pedro*$_1$ *came in*]
= { g,w | f[f,w s
f,w , g,w { f,w , g,w | f $<_1$ g g_1 = <u>Pedro</u>$_w$ <u>came in</u>$_w$(g_1)}]}
= { g,w | f[f,w s f $<_1$ g g_1 = <u>Pedro</u>$_w$ <u>came in</u>$_w$(g_1)]}
= s′

That is, the assignment-world-pairs f, w in the input information state s are changed in the following way:

- If Pedro came in in w, then f is changed to a g, where g is defined for the argument 1, and g_1 is Pedro (in w).
- If Pedro did not come in in w, then the pair f, w is simply eliminated.

The first rule captures the increase in the set of accessible discourse referents triggered by the sentence. The second captures the increase in factual knowledge. The output information state s′ contains all the new pairs of increased assignments and worlds that are compatible with the information expressed in this sentence.

The sentence can be continued as follows:

(61) [$_{NP[s],1}$ *he*]; λP{ f,w , g,w | f,w , f_1, g,w P}
|
| [$_{VP}$ *sat down*]; { g,w , x, g,w | <u>sat down</u>$_w$(x)}
| /
[$_S$ *he*$_1$ *sat down*];
λP{ f,w , g,w | f,w , f_1, g,w P}({ g,w , x, g,w | <u>sat down</u>$_w$(x)}
= { g,w , g,w | <u>sat down</u>$_w$(g_1)}

If we update the output information state s′ of the previous sentence with this example, we get the following result:

(62) s′ + [$_S$ *he*$_1$ *sat down*]
= { g,w | f[f,w s′ f,w , g,w { g,w , g,w | $\underline{\text{sat down}}_w$($g_1$)}]}
= { g,w | g,w s′ $\underline{\text{sat down}}_w$($g_1$)}

This sentence restricts the input information state s′ to those assignment-world pairs g,w for which it holds that g_1 is defined, and for which g_1 sat down in w.

6.5 Summary

We have seen that we can develop a compositional semantics of direct dynamic interpretation that takes care of classical instances of anaphoric reference and quantification.

This theory is more restrictive than DRT. For example, we have discussed DRT rules for names or specific NPs in which their discourse referents were introduced in the "largest" box, not in the local box. The current framework of direct dynamic interpretation as developed here does not allow for this; every change of assignment functions is a "local" change. In order to deal with wide-scope indefinites, we would have to construct first a logical form of the sentence in question in which the indefinite has wide scope. Also, the current version does not capture phenomena like modal subordination.

Exercises

1. In the framework of dynamic interpretation in which we combined the two aspects of accessibility of discourse referents and factual information , we haven't specified the interpretation of pronouns, names, indefinite articles and the determiner *every* when they occur in object position. Do this!

2. Derive a step-by-step interpretation of the sentence *Pedro*$_1$ *owns a*$_2$ *donkey* in this framework, and describe the changes that the meaning of this sentence triggers in an input information state s.

3. Derive a step-by-step interpretation of the sentence *He*$_1$ *beats it*$_2$ in this framework, and describe the changes that the meaning of this sentence triggers in an input information state s.

4. Derive an interpretation of the discourse *Pedro*$_1$ *owns a*$_2$ *donkey*. *He*$_1$ *beats it*$_2$ in this framework, and describe the changes that it triggers in an input information state s.

5. Assume the following model:
A = {p, d1, d2, d3}, W = {w1, w2, w3},
F(*Pedro*)(w) = p, for w = w1, w2, w3
F(*donkey*)(w) = {d1, d2, d3}, for w = w1, w2, w3
F(*own*)(w) = { p, d1 , p, d2 }, for w = w1, w2, w3
F(*beat*)(w_1) = { p, d2 , p, d3 },
F(*beat*)(w_2) = { p, d1 , p, d2 },
F(*beat*)(w_3) = { p, d3 }
And assume the empty information state { , w1 , , w2 , , w3 } as input.
a) What is the information state after processing *Pedro*$_1$ *owns a*$_2$ *donkey*.?
b) What is the information state after processing *He*$_1$ *beats it*$_2$?

6. Derive a step-by-step interpretation of the sentence *Every*$_1$ *farmer with a*$_2$ *donkey beats it*$_2$ in this framework, and describe the changes that the meaning of this sentence triggers in an information state s