Variable-free Discourse Representation Structures *

Johan Bos
University of Groningen

Abstract If we want to develop open-domain, robust semantic parsers for Discourse Representation Theory, a formalism that received considerable attention in formal semantics, then we need to produce large annotated corpora that pair sentences and texts with Discourse Representation Structures. To reduce the manual and mental effort required to construct such a corpus, we propose a simplified notation for meaning representations that gives precedence to simplicity, without explicit discourse referents and variables, while maintaining the same expressive power.

Keywords: Discourse Representation Theory, Discourse Representation Structure, Occam's razor, event semantics, negation, presupposition

1 Introduction

Discourse Representation Theory, or DRT, has made a big impact in the development of formal semantics (Kamp 1981, Heim 1982, Klein 1987, Zeevat 1989, Van der Sandt 1992, Asher 1993, Kamp & Reyle 1993, Muskens 1996, Van Eijck & Kamp 1997, Kadmon 2001, Kamp et al. 2011, Geurts et al. 2020). Because DRT works with explicit meaning representations (similar in structure to models of interpretation employed in model-theoretic semantics), that are meant to capture the meaning as interpreted by a language user when exposed to linguistic input, the formalism is also of increased interest to computational semantics and artificial intelligence. In the area of computational linguistics, semantic parsers have been developed for meaning representations based on DRT (Wada & Asher 1986, Johnson & Klein 1986, Bos et al. 1994, Bos 2001, 2008).

In contrast to the first generation of parsers, that were based on manually-crafted grammars, the current generation of parsers for DRT are data-driven (Curran et al. 2007, Le & Zuidema 2012, Bos 2015). The most recent ones are based on neural network technology, and are able to learn meaning representations directly from

* An earlier reincarnation of this paper was submitted to the Journal of Semantics & Pragmatics. It was not accepted for publication, and perhaps rightly so, because it put emphasis on reducing annotation effort for which the paper had no substantial claims to offer. The three reviewers kindly gave detailed feedback that I incorporated in the current version of the paper.

large quantities of sentences paired with the desired meaning representations. This latter line of research has also been applied to DRT (Van Noord et al. 2018, Liu et al. 2019, Evang 2019), using English data from the Groningen Meaning Bank (Bos et al. 2017) and later the Parallel Meaning Bank (Abzianidze et al. 2017), that supports also Dutch, German, Italian, Japanese and Chinese.

The meaning representation proposed in DRT are Discourse Representation Structures, DRSs for short. DRSs are complex structures: they contain discourse referents represented by variables, have various kind of logical operators, and are recursive of nature, so DRSs can be embedded inside other DRSs. They are therefore hard to annotate manually, and also challenging for machine learning approaches. The question that we will be addressing is whether we can simplify the nature of DRS in such a way that redundancies in representation and the use of variables is minimized.

The motivation for simplification is many-fold. From a theoretical perspective, the possibilitity of rewriting a complex expression into a simpler form is not only interesting but could also lead to further insight in the expressive power of a logical language. From a practical perspective, simpler representations are often more readable and perspicuous, making manual annotation or correction of DRSs less cumbersome. A meaning representation without variables is also easier to integrate in the syntax-semantics interface, which is desirable if one wants to convert output of a syntactic parser into formal meaning representations. Finally, variable-free semantic representations are suitable for machine learning applications in computational semantics, such as wide-coverage semantic parsing.

In a nutshell, I am proposing a notational variant of DRS, that simplifies the original representation considerably, without having to give up much of its expressive power.[1] In Section 2, we show how some redundancies in DRS can be removed, and also discuss the role of *order* in DRS elements. In Section 3, we propose a representational variant of DRS where explicit variables are eliminated from the meaning representation. This results in Simplified Box Notation (SBN), a notation that not only facilitates alignment between meaning representations and sentences (Section 4), but can also be extended to deal with discourse relations in the style of Segmented DRT (Section 8). Finally, in Section 5, we show how meanings in SBN are interpreted, and how we can translate them to fully-fledged DRSs.

1 This is, obviously, a vague claim. This is done on purpose. There are many variants of DRT and DRS languages that differ with respect to how they deal with phenomena such as plurals, tense, and generalized quantifiers. To add precision to this claim one needs to fix the DRS language and show that it holds. This is what we do in Section 5 for a simple DRS language.

2 Simplifying Discourse Representation Structures

A DRS is a composite of two objects: a set of discourse referents, and a set of conditions applied to these discourse referents. A discourse referent denotes an entity of the domain that we want to model. These may be concrete things (people, animals, artifacts, locations, organisations), but can also be of abstract nature: ideas, events, or time periods. The DRS-conditions supply this information, and they also state how the different entities are semantically related to each other. Figure 1 shows some examples.

x y
cat(x)
chase(x,y)
dog(y)

x e y t
cat(x)
chase(e,x,y)
dog(y)
time(t)
$e \subseteq t$
$t \prec$ now

x e y t
cat(x)
chase(e)
Agent(e,x)
Theme(e,y)
Time(e,t)
dog(y)
time(t)
$t \prec$ now

Figure 1 Discourse Representation Structures (DRSs) for "a cat chased a dog", left as in Kamp & Reyle (Chapter 5), right in a neo-Davidsonian style.

In our quest to simplification we make three assumption about the nature of DRSs in the context of natural language semantics. Firstly, we assume that every discourse referent has a corresponding one-place predicate that classifies the entity within some background knowledge ontology.[2] From a formal point of view, this is not an explicit restriction, but it makes sense to make this assumption. For what sense would it make to model an entity that cannot be identified in some part of an ontology? Secondly, we assume that this identifying predicate is declared locally, i.e., in the same DRS as the discourse referent was introduced. This second assumption also makes sense: what's the point of providing more specific information about a discoures referent in another context? The third assumption is that exactly one

2 In a way, this is equivalent to a many-sorted first-order logic, where sorts, sometimes called types, denote subsets of the domain. Instead of assigning a sort to a variable directly, we do this by adding a one-place predicate.

predicate is needed to classify an entity. Also this makes sense: entities belong to one type, and when modelling, you want to be as specific as you can.[3][4]

These three assumptions are key to a first straightforward simplification in DRS. As a declaration of a discourse referent and the introduction of its predicate concept seem to go hand in hand, they can be represented together, instead of separately. This also means that we do not need the two-fold partition of DRS into domain and condition-set anymore. So we can rewrite the DRS that we examined before as a DRS in which the discourse referents are aligned with their predicates. Now, it doesn't take much thought to see another obvious simplification. The one-place predicates that introduce discourse referents and declare predicate concepts, contain redundant information: the variable occurs twice in each occasion. Applying Occam's razor once more gives a simplified representation as exemplified in Figure 2.[5]

x e y t
cat(x)
chase(e)
Agent(e,x)
Theme(e,y)
Time(e,t)
dog(y)
time(t)
$t \prec$ now

x	-	cat(x)
e	-	chase(e)
		Agent(e,x)
		Theme(e,y)
		Time(e,t)
y	-	dog(y)
t	-	time(t)
		$t \prec$ now

x	-	cat
e	-	chase
		Agent(e,x)
		Theme(e,y)
		Time(e,t)
y	-	dog
t	-	time
		$t \prec$ now

Figure 2 Evolution from a standard DRS (left) to reduced notation. First discourse referents are aligned with their conceptual predicates (middle). Then redundant arguments are discarded (right).

At this point the reader might be sceptical and think of counterexamples to the above simplification procedure. Of course, the proposal so far is based on certain assumptions — the three assumptions outlined above. But one might want to bring in a counter-instance that violates one of the assumptions, and a good candidate would be a DRS embedded inside another DRS, where the discourse referent is

3 Here, we don't mean that the meaning of a single noun is always mapped to a single predicate. A case in point are agent nouns, such as *actress*, *butcher* or *victim*, that following recent proposals for lexical decomposition (Bos & Abzianidze 2019) would introduce two predicates: one denoting a person, and one denoting the role that is performed by that person.

4 We are ignoring compound entities here for the moment, entities that pair single entities into composites, resulting in ontologically complex objects with prototypical examples *book*, *bottle* and *university*, and we assume a solution along the lines of Pustejovsky (1995) and Pinkal & Kohlhase (2000).

5 This representation of referents and concepts is strongly reminiscent of, and inspired by, the PENMAN notation (Kasper 1989, Bateman et al. 1989, Langkilde & Knight 1998), where entities are introduced and at the same time given a conceptual declaration.

not in the same DRS as its conceptual predicate. This is, actually, a natural thing to occur when we take negation or disjunction into account. For instance, someone might see a glimpse of an animal, and state that it was no fox, or that it was a fox or cat. Figure 3 shows corresponding DRSs for this situation. Here, the discourse referents and their conceptual predicates are placed in different boxes. But we can still maintain our requirement for reduced notation by introducing a predicate that subsumes all others (the top concept of the ontology, here, and throughout the rest of this article, we assume this to be "entity"), introduce an extra discourse referent, and an equality condition. The semantically equivalent DRSs that result from this modification satisfy the three assumptions (see Figure 3), and can be notationally reduced in the form that we propose in this article.

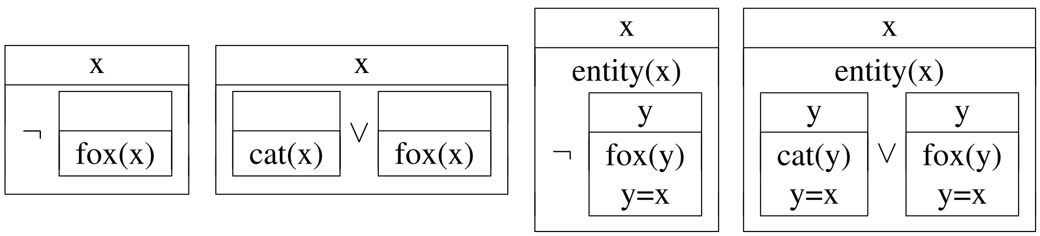


Figure 3 Negation and disjunction could give rise to discourse referents that are separated from their conceptual predicates (left). But this can be fixed (right) by making use of the axiom: $\forall x(P(x) \leftrightarrow \exists y(P(y) \wedge y{=}x))$.

This was a little excursion into boolean operators. Let's now turn to back to our initial example, because we're not done yet. From an annotator's point of view, a rather unpleasant task is the blessing of discourse referents with unique names. They are artificial, arbitrarily chosen by the formal semanticist when they make their analysis of a sentence. Instead of "x" we could have picked a "z", or something else that could go by as a variable. For relatively short sentences this is still routine for a practitioner of formal semantics, but for longer sentences one runs out of variables very quickly and needs to resort to naming tricks that involve single accents, double accents, or even wilder diacritical signs. For representing reasonably sized texts, arguably one of the key objectives of Discourse Representation Theory, one needs quickly over a hundred different discourse referents. So it would be utterly attractive to get rid of explicit discourse referents—not just for human annotators, but also for automated semantic parsers, especially those based on deep learning. Is this possible?

This *is* possible, but before we show how to do so, we should note something important. In the original definition of DRS, discourse referents and DRS-conditions are sets, and their elements are, mathematically speaking, *unordered*. This means that we could list the discourse referents and DRS-conditions in any order we fancy, as Figure 4 demonstrates. In practice, this is not what usually happens. For readability and verification purposes, the order of discourse referents and the order of DRS-conditions reflect the surface order of the words of the sentence (or text) that is analysed by the DRS. This is important because in our representational reduction order will play a crucial role, as we will see in the next section.

And there is another point to make here concerning order. In a classical DRS, the discourse referents are placed on top, ordered from left to right, and the DRS-conditions are listed from top to bottom. These two orthogonal directions put another unnecessary mental load on a human interpreter of DRSs. Our simplified notation (that at this stage still employs variables, but got rid of the distinction between discourse referents and DRS-conditions) does not suffer from this (Figure 4), because the reading direction is going in one way.

x	-	cat			$t \prec$ now			$t \prec$ now
e	-	chase	x	-	cat			Agent(e,x)
		Agent(e,x)			Agent(e,x)			Time(e,t)
		Theme(e,y)	y	-	dog			Theme(e,y)
		Time(e,t)			Time(e,t)	t	-	time
y	-	dog			Theme(e,y)	e	-	chase
t	-	time	t	-	time	x	-	cat
		$t \prec$ now	e	-	chase	y	-	dog

Figure 4 In theory, DRS-conditions are sets, so their order has no impact on meaning. These three DRSs (in reduced notation) are all semantically equivalent. In practice, the order reflects by and large the order of the sentence that it models.

In other words, even though DRS-conditions are unordered sets of objects, in practice, a certain order is maintained when they are visualised. Indeed, from an annotation point of view, and also from a computational learning perspective, it makes sense to let the order of DRS-conditions reflect the surface word order of the sentence that is analysed. Moreover, a fixed order of DRS-conditions is crucial for the main simplification step we are going to apply: changing absolute referents to relative referents. This will give us means to get rid of those, from the perspectives of human text analysts and deep learning algorithms, terribly annoying variable names.

3 Freeing Discourse Representation Structures from Variables

The technique we are going to use for eliminating variables from the meaning representation is based on de Bruijn-indices (de Bruijn 1972), but with a bi-directional twist. de Bruijn-indices are a clever way to stipulate co-indexing between variables. Instead of using names to distinguish between different variables, *physical distance* between binder and variable occurrence is used to distinguish bound variables. For instance, instead of ∃x∃y P(x,y), we use the variable-free notation ∃∃P(2,1), where the index “2” is short for a variable introduced by the second binder going from right to left in the formula, and the index "1" is bound the first binder (both binders are existential quantifiers in this case). Van Eijck (2001) and Dekker (2016) also apply indexing to DRT. Our approach is more radical, and permits indices that can bind in two directions, and covers discourse structure as well.

In DRT, the discourse referents are the binders. In our new notation, however, where discourse referents are conflated with the DRS-conditions, binding can go in two directions: backwards, from bottom to top, also across DRSs that are in a proper subordination relation; and forwards, from top to bottom, until the end of a DRS. For backward references, we will use indices represented by negative integers. For forward reference, we will use positive integers. We will use 0 for the current discourse referent: the referent that is lastly introduced. So, after a discourse referent is introduced, we can refer to it by 0. When a second discourse referent is introduced, we refer to it by 0, and to the first discourse referent by -1. If a discourse referent is introduced *after* a condition that needs it, we refer to it by a positive index (e.g., +2 or +1). The schema in Figure 5 illustrates this idea.

		Ordered DRS	x	e	y	t
x	-	cat	0	+1	+2	+3
e	-	chase	-1	0	+1	+2
		Agent(e,x)	-1	0	+1	+2
		Theme(e,y)	-1	0	+1	+2
		Time(e,t)	-1	0	+1	+2
y	-	dog	-2	-1	0	+1
t	-	time	-3	-2	-1	0
		t $\prec$ now	-3	-2	-1	0

Figure 5 Bi-directional de Bruijn-indices exemplified for four variables. The columns on the right show to which variables the indices are replacing when used in a certain position in the DRS. For instance, the DRS-condition Time(e,t) is equicalent to the indexed Time(0,+2), where 0 denotes the current discourse referent, and +2 to the second discourse referent in the future.

Applying the indices to variables used in the DRS relations we arrive at DRS-conditions that do not show absolute references to variables anymore (Figure 6). And, consequently, the names of the discourse referents do not play any role anymore, so we may leave them out as well. This will yield a variable-free DRS with positive, negative, or zero-indices.

And this brings us to one more simplification step we can perform. The two-place relations have two variables that are declared as discourse referent in the same DRS or in a DRS that subordinates this DRS. In the former case, when its first argument is placed directly after the concept that introduces it, the index is always 0. And because it is always 0, we can as well apply another representational simplification: leaving out the zero-indices. We refer to this as "0-drop" (Figure 6) a representational technique that is part and parcel of the PENMAN notation (Kasper 1989, Bateman et al. 1989).

x	-	cat	cat			
e	-	chase	chase		cat	
		Agent(0,-1)		Agent(0,-1)	chase	Agent -1
		Theme(0,+1)		Theme(0,+1)		Theme +1
		Time(0,+2)		Time(0,+2)		Time +2
y	-	dog	dog		dog	
t	-	time	time		time	$\prec$ now
		0 $\prec$ now		0 $\prec$ now		

Figure 6 From ordered DRS to variable-free DRS, also known as Simplified Box Notation, shown here without and with 0-drop.

4 Simplified Box Notation

We call this new notation of DRS the Simplified Box Notation (SBN). A DRS in SBN is a sequence of concepts (one-place predicates), relations (two-place predicates), or structural constraints. The concepts are binders for first-order entities, and they have a dual purpose: (1) introduce a discourse referent, and (2) establish the type of the discourse referent with a conceptual one-place predicate. The relations comprise the classical thematic roles found in a neo-Davidsonian analysis and comparison constraints, including equality (=), inequality ($\neq$), and temporal precedence ($\prec$) and its inverse ($\succ$). The structural constraints provide information about discourse structure.

There are two ways of representing discourse structure. One approach is to follow the recursive nature of DRS, and form complex conditions by combining negation, implication or disjunction with a DRS in SBN, as Figure 7 shows. However, in Section 8, we will adopt a method that flattens discourse structure, fully integrating

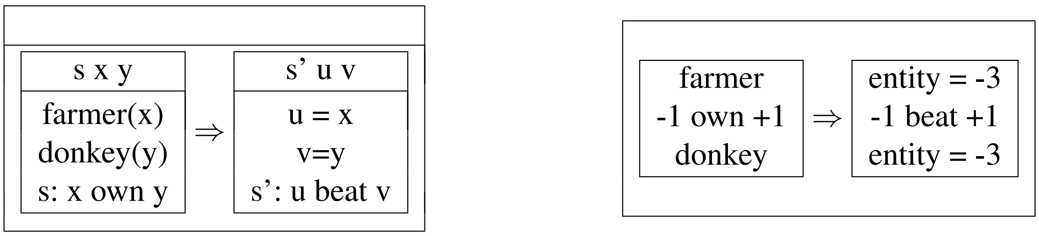


Figure 7 In an article about DRT, an analysis of Geach's (1962) donkey sentence, "Every farmer who owns a donkey beats it", should certainly not be missing. Shown here on the left as in Kamp & Reyle (1993), and on the right in (recursive) SBN.

it within SBN. For local discourse structure we will make use of the propositional equivalences $(p \rightarrow q) \leftrightarrow \neg(p \wedge \neg q)$ and $(p \vee q) \leftrightarrow \neg(\neg p \wedge \neg q)$, that can also be transferred to DRT, and gives us a single operator (negation) for local discourse structure in SBN.[6] This gives us an advantage when flattening DRSs, because no additional constraints on well-formedness of DRSs with implication or disjunction will be required.[7]

We use several notational conventions in the examples of SBN that follow, and these are roughly based on the DRS-language of the Parallel Meaning Bank (Abzianidze et al. 2017). Concepts (one-place predicates) are always in lowercased symbols. Relations, or roles, (two-place predicates) start with an uppercase character, and may have the suffix "Of". Roles without the suffix "Of" are read as "event X has as R Y". For instance, in Figure 6, the "chase" event has as *Agent* a cat. Roles with the suffix "Of" are inversed roles, and are paraphrased as X is R of y, for instance, the cat is the *Agent* of the chasing event. Although a neo-Davidsonian analysis is best suited for this notation, because events are introduced as a one-place predicate, this is strictly speaking not required, as Figure 7 demonstrates.

As remarked earlier, in the original DRS language, discourse referents and DRS conditions are formally not ordered. In practice, however, a certain order is maintained for readability purposes. This ordering reflects the order of the words in the phrase, sentence, or text that is under analysis. Usually, in formal semantics, the

6 For a modal extension of the DRS language, one could extend the local discourse relation with $\Diamond$ and $\Box$ in addition to $\neg$.

7 This is because negation is a unary operator, and therefore forms a relationship between *two* objects: the DRS in which the negation appears, and the DRS that forms its argument. However, implication and disjunction, being binary operators, form a relationship between three objects: the DRS in which they appear, plus their two DRS arguments. Flattening a binary relation is easier than flattening a ternary relation. The donkey sentence in Figure 7 is expressed in SBN as: NEGATION -1 farmer own Pivot -1 Theme +1 donkey NEGATION -1 entity = -3 beat Agent -1 Patient +1 entity = -3.

alignment between input natural language and output DRS is never made explicit, although one could see derivations of compositional versions of DRT doing this implicitly (Muskens 1996). But alignment plays an important role in manual annotation (readability and verification) and also for machine learning approaches for natural language processing, both for parsing and generation.

It is important to note that there isn't always a perfect alignment between DRS-conditions and words — one DRS-condition might have been triggered by a single word, or by multiple words. And a single word may introduce one or more DRS-conditions, or none at all. In what follows here, we have a glimpse at how SBN affects alignment. We consider three cases: SBN with only negative indices and 0-drop, SBN with positive and negative indices and zero drop, and SBN with bidirectional indices without 0-drop. The first is the most restrictive; the last the most permissive.

Although we have been using positive indices in SBN, they are strictly speaking not needed and we could ban them from SBN. This is possible because relations are tied to two conceptual predicates, and we can just order them in such a way that the predicates come first, followed by the relation instantiated with negative indices (Figure 8). This puts a serious constraint on the ordering, but we are helped by the fact that we can inverse roles: for any role R, we have that $\forall x \forall y (R(x,y) \leftrightarrow R\mathrm{Of}(y,x))$. Obviously, all comparison constraints can be inversed, too. Equality and inequality are symmetric, so their arguments can be flipped anyway. The inverse of $\prec$ is $\succ$.

This gives us flexibility, because even with 0-drop we can order the two predicates in any order, and still refrain from using positive indices: c1 c2 Role -1 would be equivalent to c2 c1 RoleOf -1. Figure 8 illustrates role inversion and its impact on alignment. Nonetheless, if we want to give more weight to alignment than expressive power, we need to use positive indices, and that's exactly why we have introduced them already in Section 4.

x e t y
male(x)
Name(x,"Tim")
time(t)
t $\prec$ now
hit(e)
Theme(e,y)
Patient(e,x)
Time(e,t)
baseball(y)

male	Name "Tim"	% Tim
time	$\prec$ now	% was
hit	Time -1 Patient -2	% hit
	Theme +1	% by
baseball		% a baseball.

male	Name "Tim"	% Tim
time	$\prec$ now	% was
hit	Time -1 Patient -2	% hit
baseball	ThemeOf -1	% by a baseball.

Figure 8 Positive indices can always be eliminated from SBN using role inversion, but this comes at the cost of alignment precision.

Once we give up 0-drop, or make it optional, we gain even more freedom for improving alignment. Giving up 0-drop means that roles will be decorated by two (non-zero) indices. This will enable us to get a better match between meaning and expression for sentences that show PP-fronting, topicalisation, or other long-distance phenomena. An example is given in Figure 9. Giving up 0-drop comes at the cost of readability. So there is a trade-off between alignment precision and readability. In the rest of this article we will assume 0-drop.

t x e
time(t)
Year(t, "1968")
t ≺ now
person(x)
Name(x,"JFK")
assassinate(e)
Patient(e,x)
Time(e,t)

```
       time Year "1968" TimeOf +3 % In 1968,
     person Name "JFK"            % JFK
       time ≺ now                 % was
assassinate Time -1 Patient -2    % assassinated.

     person Name "JFK"            % JFK
       time ≺ now                 % was
assassinate Time -1 Patient -2    % assassinated
            Time +1               % in
       time Year "1968"           % 1968.
```

Figure 9 PP-fronting is a troublemaker for aligners. At the top we need to inverse the role of the preposition *in*.

The importance of order in SBN gives rise to an interesting side-effect. It gives us a way to represent information structure. Information structure, in linguistics, is the way information is packaged inside a sentence (Vallduví 2016). For instance, even though an active sentence is truth-conditionally equivalent to its passive variant, the order in which entities are introduced is different. Because order of clauses is important in SBN, it provides us with a way to make information structure explicit in a meaning representation. This also has consequences for multi-lingual perspectives. Word order might differ among different languages, and SBN can reflect these differences. An example is shown in Figure 10.

x s
dress.n.01(x)
Colour(x,s)
red.a.01(s)

```
red.a.01 ColourOf +1 % red      dress.n.01           % vestito
dress.n.01           % dress    red.a.01 ColourOf -1 % rosso
```

Figure 10 Representation for colour adjectives, aligned with English and Italian.

The ability to extend SBN to represent information structure goes beyond the scope of this paper. A critical reader might even suggest that the difference between

DRS and SBN is superficial, and that DRS can easily be extended to perform a similar purpose by specifying a DRS as a pair of *ordered* discourse referents and *ordered* conditions. There are crucial differences, though. Order is part and parcel of SBN—there is no way you can shuffle around items without changing its interpretation (you would need to update the indices, too). The explict recursive structure of DRS, and the dichotomy between discourse referents and conditions in DRS make it less straightforward to explicate information structure.

5 Interpreting Simplified Box Notation

A DRS in SBN can be converted to an ordinary DRS using a procedure that takes an SBN expression and an *index register* for indices. The registers are similar to those introduced by Muskens (1996), in the sense that they give a "dynamic" interpretation to DRSs. But where the registers of Muskens contain the things denoted by discourse referents, in our case the registers contain the discourse referents denoted by the indices.

An index register is a partial function from indices to variables. Given a register ρ, the value of an index i is given by the term $\rho(i)$. New index-value pairs can be pushed on a register where all variables are shifted by decreasing their corresponding index value by one, and assigning a fresh (previously unused variable) onto the register: $\rho \Downarrow \rho'$ iff $\forall i(i \in dom(\rho) \rightarrow \rho'(i-1) = \rho(i))$. We first consider the translation of SBN to a DRS without any structural constraints. Recall that a basic SBN is a sequence of concepts and relations (we assume 0-drop for all relations). The following five clauses specify the translation steps for $[.]^{\rho_{in}}_{\rho_{out}}$, with ρ_{in} an ingoing register, and ρ_{out} an outgoing register.

i. $[P]^{\rho}_{\rho''} \stackrel{\text{def}}{=} \begin{array}{|c|}\hline \text{x} \\ \hline P(\text{x}) \\ \hline \end{array}$ iff $\rho \Downarrow \rho'$ and $0 \notin dom(\rho')$ and $\rho''[0]\rho'$ and $\rho''(0)$=x

ii. $[P]^{\rho}_{\rho'} \stackrel{\text{def}}{=} \begin{array}{|c|}\hline \rho'(0) \\ \hline P(\rho'(0)) \\ \hline \end{array}$ iff $\rho \Downarrow \rho'$ and $0 \in dom(\rho')$

iii. $[Rc]^{\rho}_{\rho} \stackrel{\text{def}}{=} \begin{array}{|c|}\hline \\ \hline R(\rho(0), c) \\ \hline \end{array}$ iff c is a constant, and $0 \in dom(\rho)$

iv. $[Ri]^{\rho}_{\rho} \stackrel{\text{def}}{=} \begin{array}{|c|}\hline \\ \hline R(\rho(0), \rho(i)) \\ \hline \end{array}$ iff $\{0, i\} \subseteq dom(\rho)$

v. $[Ri]^{\rho}_{\rho'} \stackrel{\text{def}}{=} \begin{array}{|c|}\hline \\ \hline R(\rho(0), x) \\ \hline \end{array}$ iff $0 \in dom(\rho)$ and $i \notin dom(\rho)$ and $\rho[i]\rho'$ and $\rho'(i)$=x

Clauses (i) and (ii) deals with a concepts. Recall that a concept has a dual function of introducing a discourse referent (so the register needs to be shifted) and introducing a conceptual predicate. Clause (i) covers new discourse referents, whereas clause (ii) handles the case where a discourse referent has already been introduced on the register via a positive index. The remaining clauses cover the relations. Clause (iii) deals with constants, clause (iv) translates variables that are already present in the register, and clause (v) handles positive indices. Given an SBN S=$[s_1,\ldots,s_n]$, its DRS translation is defined as $([s_1]^{\rho_0}_{\rho_1} \oplus [s_2]^{\rho_1}_{\rho_2} \oplus, \ldots, \oplus [s_n]^{\rho_{n-1}}_{\rho_n})$, where $\oplus$ is the DRS merge operation, and ρ_0 is the initial (empty) register. Appendix B illustrates the translation to DRS for an SBN with negative and positive indices.

As free variables may occur in a standard DRS, indices without a destination may occur in SBN. But because of the bi-directional nature of the indices that are employed in SBN, the concept of free and bound variables is slightly more complex. Let's first consider negative indices. Assuming that, as in standard DRT, subordination between DRSs is given shape in the form of a tree, then a negative index -*n* is free if *n* is greater or equal than the number of binders on its subordination path. In sum, the negative indices behave as ordinary variables in DRS. But this is not the case for positive indices. Because they count "downwards" in the subordination tree formed by a complex DRS, granting positive indices uncontrolled binding power could lead to ambiguous bindings (because there could be several branches for subordination formed by sub-DRSs). They cannot refer to discourse referents that are in subordinated DRSs, and this also means that they can never occur free, in the sense that they might be bound when embedded in a larger piece of DRS. In other words, a positive index without a referent is lost forever. This becomes evident when we consider cases of negation in SBN.

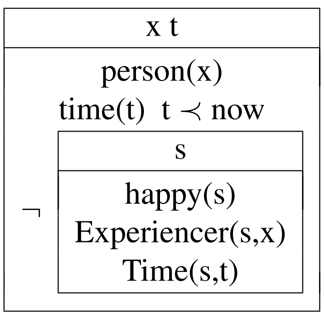
6 Negation

In DRT, negation is one of the logical operators that makes DRS recursive. So far we have only hinted at how negation will be inserted into SBN. Although this could be done using explicit recursion (in the way done in Figure 7), we prefer a simple, flat representational alternative, which is in line with the overall motivation of SBN: an easier text-based annotation scheme, a possibly simpler syntax-semantics interface, and supporting sequence-based machine learning approaches.

As negation is sensitive to scope, we need a way of saying which parts of the meaning representation belong to the scope of a negation operator. The technique used in SBN is one based on introducing explicit discourse structure markers in the representation, dividing the sequence of concepts and roles into two discourse units. In SBN we write these discourse structure indicators in all-uppercase characters, to avoid confusion with concepts and relations. In the case of negation, we have

NEGATION, that divides the meaning represented in SBN in two parts: the sequence before, and the sequence after the marker. In general, a representation in SBN with n markers divides the meaning representation into $n+1$ discourse units.

It is important to realise that discourse structure markers such as NEGATION (we will introduce more later) connect *two* contexts. In the case of negation it is the argument of the negation (its scope), *and* the context in which the negation holds. Usually, this is the previous context, but not always. The way we represent discourse structure in SBN is such that the second argument is always the direct argument. Hence, we don't need to specify this explicitly. However, the first argument specifies the context it relates to. This is usually the previous context, and we express this by the context index -1. Figure 11 illustrates the general idea.

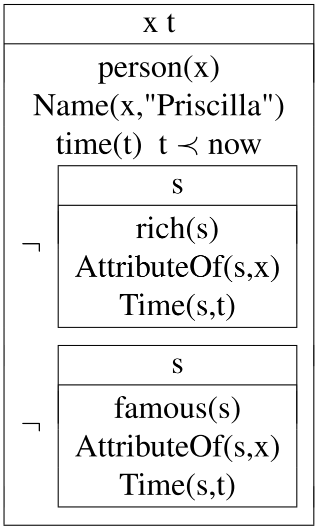


person		`% Someone`
time	$\prec$ now	`% was`
NEGATION -1		`% not`
happy	Experiencer -2 Time -1	`% happy.`

Figure 11 Negation in DRS and corresponding meaning in SBN.

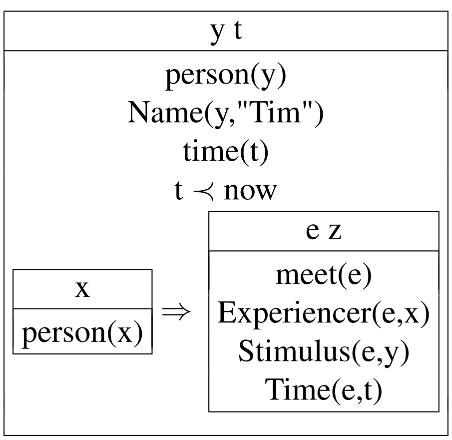
But even in the case of negation, it is not always the previous context in which negation is interpreted. A case in point is shown in Figure 12, where there is a sequence of two negation operators. As both are part of the same context, their indices cannot be the same: for the first negation it is -1 (the previous context), but for the second negation an index with value of -1 would place the second negation in the scope of the first negation, and -2 is the index that places the negation in the desired context.

Universal quantification and disjunction is also analysed with the help of negation in SBN. For universal quantifiction we make use of the logical equivalence $(p \rightarrow q) \leftrightarrow \neg(p \wedge \neg q)$, as Figure 13 shows. At first this might be perceived as counterintuitive and unlike the principles of DRT. The alternative would be introduce two new discourse relations in the SBN language, say ANTECEDENT and CONSEQUENCE. Then a DRS B0 with a conditional B1 $\Rightarrow$ B2 would be of the form B0 ANTECEDENT -1 B1 CONSEQUENCE -1 B2, instead of the structurally equivalent B0 NEGATION -1 B1 NEGATION -1 B2. However, this option would require additional well-formedness dependencies (a ANTECEDENT always needs a CONSEQUENCE) and is therefore less preferred.



person	Name "Priscilla"	`% Priscilla`
time	≺ now	`% was`
-1	NEGATION	`% neither`
rich	AttributeOf -2 Time -1	`% rich`
-2	NEGATION	`% nor`
famous	AttributeOf -2 Time -1	`% famous.`

Figure 12 Negation coordination in DRS and translation in SBN.



time	≺ now	
-1	NEGATION	
person		`% Everyone`
-1	NEGATION	
meet	Experiencer -1	`% met`
	Stimulus +1	
	Time -2	
person		`% someone.`

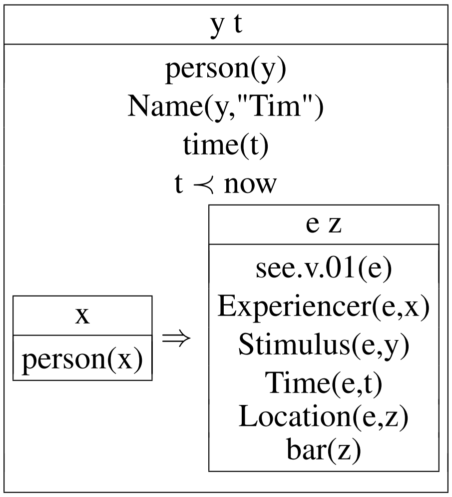
Figure 13 Analysis of "Everyone met someone." in DRS and in SBN, using the equivalence $(p \rightarrow q) \leftrightarrow \neg(p \wedge \neg q)$.

Discourse referents in subordinated DRSs are not accessible in DRT. This principe should be reflected in the interpretation of SBN. This means that (a) neither positive nor negative indices can enter negated contexts, and (b) only negative indices in negated contexts can refer to discourse referents outside the negation.

For the interpretation of SBN with negation we need to extend the translation procedure presented above. Recall that we used registers to map indices to variables. Because now we need to deal with several DRSs, we will associate each DRS with its own register. For convenience, we will label each DRS, and associate each DRS index with one of these labels. What we end up with is a system with two levels of registers: the registers for indices for first-order entities that we had before, and registers for indices for DRSs. The catch is that the latter type of register is of recursive nature, and not only links a box index to a DRS label, but also to the variable index belonging to that DRS. Appendix C gives an example translation.

7 Presupposition

Embedded structure is directly connected to another semantic phenomenon: presupposition. One of the key insights of DRT is the interaction of noun phrases and their availability as antecedent for pronouns, in particular in context triggered by conditionals and negation. A consequence of these observations is that proper names introduce their discourse referent and accompanying conditions in the outermost level of DRS, rather than in situ, say inside a box in the scope of a negation. This non-compositional aspect for proper names has haunted DRT for years, and more principled mechanism to deal with presupposition triggers, including proper names, have been proposed (Van der Sandt 1992, Geurts 1999, Beaver 2002, Bos 2003). Because proper names are added to the outermost level of DRS, they usually mess up the alignment order. Venhuizen et al. (2018) solve this by adding an index to every DRS-condition, that tells us in which DRS they ought to be interpreted. If we would follow this route in SBN, the indices would be substantially hard to interpret. Instead, we propose accommodation of semantic content at the appropriate level of DRS, and a "trace" of the trigger of the DRS it got projected from, but only in cases where the two levels of DRS are different. This could be done by creating a complete copy of the accommodated material, as in Figure 14. Both approaches improve alignment between linguistic input and meaning representation; the former is preferred for annotation purposes, the latter might be advantageous for machine learning applications.




```
person  Name "Tim"
        PRESUPPOSITION -1
  time  ≺ now
        NEGATION -1
person                      % Everyone
        NEGATION -1
   see  Experiencer -1      % saw
        Stimulus +1
        Time -2
person  Name "Tim" = -4     % Tim
   bar  LocationOf -2       % in a bar.
```

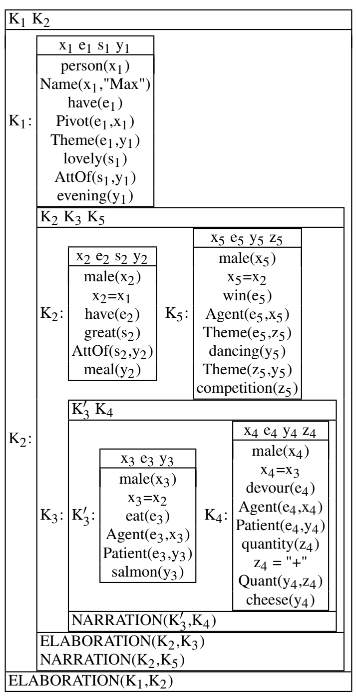
Figure 14 Analysis of "Everyone saw Tim in a bar." in DRS and in SBN. Here we make use of the equivalence $(p \to q) \leftrightarrow \neg(p \wedge \neg q)$. The proper name is projected to the main DRS, leaving a "copy" behind.

8 Discourse Structure

So far, we have only considered cases of alignment of single sentences with meaning representations in SBN. Certainly, we can't ignore discourse structure, and what we present in this section is a way to include elements of Segmented Discourse Representation Theory, SDRT (Asher 1993, Asher & Lascarides 2003) in SBN. In SDRT, DRSs representing discourse units are labelled with variables, and discourse relations are imposed on these variables to model the rhetorical structure of a text. Viewed this way, we have two distinct sets of variables: variables ranging over entities (as in the previous section), and variables ranging over discourse units (as we did for negation). In the previous sections we already showed how to eliminate entity-variables using de Bruin-indices. We will do the same for the variables ranging over boxes. Hence, we will employ two types of indices in SBN.

Recall that de Bruijn-indices require *binders* to determine the identity of an index. For entity indices we use concepts as binders. We need a similar device for box indices. We could introduce explicit binders in SBN that would signal that a new DRS starts in a sequence of SBN elements. Alternatively, we could use the fact that there is, given a coherent piece of text, always a discourse relation between two discourse units in SDRT. What we will do is to use this insight and make discourse relations binders of box indices. For instance, given a discourse relation NARRATION, the SBN sequence $s_1 \ldots s_i$ NARRATION $s_{i+1} \ldots s_n$ would represent two discourse units connected by a discourse relation. Here every s_i is a concept or role, and the sequence $s_1 \ldots s_i$ represents the first box. As NARRATION is a binder, it will introduce a new discourse unit referent for the elements that follow, $s_{i+1} \ldots s_n$. Using de Bruijn-indexing, a more precise notation would be $s_1 \ldots s_i$ NARRATION(-1,0) $s_{i+1} \ldots s_n$ as the -1 index refers to the previous DRS, and the 0 to the current DRS. We can drop the brackets, and use 0-drop, resulting in $s_1 \ldots s_i$ NARRATION -1 $s_{i+1} \ldots s_n$. We also need a way to start the discourse, and we use the SBN element BOS (beginning of sequence) for this purpose. Figure 15 shows an SDRS (Segmented DRS) in SBN for the classic example of Max having an amazing night out from Asher & Lascarides (2003).

We will make a distinction between two sorts of discourse structure. On the one hand we have local, internal structure, as present in classical DRT, to wit negation, disjunction, and conditionals (and as before we continue to represent this with the NEGATION discourse relation). On the other hand we have global, external structure, as introduced in SDRT to represent rhetorical dimensions.




```
                BOS
     person Name "Max"          % Max
       have Pivot-1 Theme +2    % had
     lovely AttOf +1            % a lovely
    evening                     % evening.
                ELABORATION -1
       male = -4                % He
       have Pivot -1 Theme +2   % had
      great AttOf +1            % a great
       meal                     % meal.
                ELABORATION -1
       male = -4                % He
        eat Agent -1 Patient +1 % ate
     salmon                     % salmon.
                NARRATION -1
       male = -3                % He
     devour Agent -1 Patient +1 % devoured
   quantity = "+"               % lots
                QuantOf +1      % of
     cheese                     % cheese.
                NARRATION -3
       male = -4                % He
        win Agent -1 Theme +1   % won
    dancing ThemeOf +1          % a dancing
competition                     % competition.
```

Figure 15 Discourse relations exemplified. On the left in an SDRT-like manner, right in SBN. Temporal information is ignored here for reasons of space.

To translate an SBN with structural constraints, we need to extend the translation procedure to produce Segmented DRSs. We do so by defining another translation function that produces an SDRS, but that calls upon the earlier defined translation function to produce DRS pieces. In SDRT the boxes are labelled, so we need an additional register to keep track of the variables of boxes. For simplicity we only cover negative box indices in the following definitions (where K is a fresh, previously unused variable). We also include a way of performing DRS-merge in SBN.

To make the definition below more comprehensible, we will make use of two globally update functions: **B** and **R**. Firstly, **B** is a global update function, that keeps track of the DRSs for each DRS variable. It is a function from DRS labels to DRSs. The notation $\mathbf{B}\leftarrow$K: B_{new} is short for replacing the old value of K, B_{old}, by $\text{B}_{old} \oplus \text{B}_{new}$, that is, a merge of the old, earlier assigned DRS, and a new DRS. Secondly, **R** is a global update function that administers the register value for each DRS. This is required because discourse units are not always locally attached (e.g., the fifth discourse unit in Figure 15 is attached to the second one). The translation function from SBN to SDRS comprises six clauses:[8]

i. $[\text{BOS}]^{\sigma\rho}_{\sigma'\rho} \stackrel{\text{def}}{=} \begin{array}{|l|}\hline \text{K} \\ \hline \text{K: }\mathbf{B}(\text{K}) \\ \hline \\ \hline \end{array}$ iff $\sigma[0]\sigma'$ and $\sigma'(0)$=K and $\mathbf{B}\leftarrow$K: $\begin{array}{|c|}\hline \\ \hline \\ \hline \end{array}$

ii. $[P]^{\sigma\rho}_{\sigma\rho'} \stackrel{\text{def}}{=} \begin{array}{|c|}\hline \\ \hline \\ \hline \end{array}$ iff $0 \in dom(\sigma)$ and $\mathbf{B}\leftarrow \sigma(0)$: $[P]^{\rho}_{\rho'}$

iii. $[Rt]^{\sigma\rho}_{\sigma\rho'} \stackrel{\text{def}}{=} \begin{array}{|c|}\hline \\ \hline \\ \hline \end{array}$ iff $0 \in dom(\sigma)$ and $\mathbf{B}\leftarrow \sigma(0)$: $[Rt]^{\rho}_{\rho'}$

iv. $[\text{MERGE } i]^{\sigma\rho}_{\sigma'\mathbf{R}(\sigma'(i))} \stackrel{\text{def}}{=} \begin{array}{|c|}\hline \\ \hline \\ \hline \end{array}$ iff $\sigma \Downarrow \sigma'$ and $\sigma'(0)$=$\sigma'(i)$ and $\mathbf{R}\leftarrow\sigma(0) : \rho$

v. $[\text{NEG } i]^{\sigma\rho}_{\sigma'\mathbf{R}(\sigma'(i))} \stackrel{\text{def}}{=} \begin{array}{|c|}\hline \\ \hline \\ \hline \end{array}$ iff $\sigma \Downarrow \sigma'$, $\sigma'(0)$=K and $\mathbf{B}\leftarrow$K: $\begin{array}{|c|}\hline \\ \hline \\ \hline \end{array}$ and $\mathbf{B}\leftarrow\sigma'$(i): $\begin{array}{|l|}\hline \\ \hline \neg\,\mathbf{B}(\text{K}) \\ \hline \end{array}$ and $\mathbf{R}\leftarrow\sigma(0) : \rho$

vi. $[Si]^{\sigma\rho}_{\sigma'\mathbf{R}(\sigma'(i))} \stackrel{\text{def}}{=} \begin{array}{|l|}\hline \text{K} \\ \hline \text{K: }\mathbf{B}(\text{K}) \\ \hline S(\sigma'(i),\text{K}) \\ \hline \end{array}$ iff $\sigma \Downarrow \sigma'$ and $\sigma'(0)$=K and $\mathbf{B}\leftarrow$K: $\begin{array}{|c|}\hline \\ \hline \\ \hline \end{array}$ and $\mathbf{R}\leftarrow\sigma(0) : \rho$

8 This definition requires an extension that distinguishes between subordinating and coordinating discourse relations, following Asher & Vieu (2005). A subordinating discourse relation, like SUBORDINATION, will always introduce a new discourse unit. However, a coordinating discourse relation, like NARRATION, will create a new level of recursive SDRS structure if needed, as Figure 15 shows.

In clause (i) the BOS element triggers a new fresh box to be added to the SDRS, and therefore requires a shift for the box register (the entity register is not affected). Clause (ii) and (iii) translate concepts and relations with the help of the translation function defined earlier, and cause an update of the DRS they are part of, without changing the internal structure of the SDRS. Clause (iv) is the merge, that makes it possible to have discontinuous pieces of DRS within SBN. Note that it does perform a shift, but doesn't add a new DRS label, and instead picks an earlier declared box label of the DRS it s merging with. It also needs to get access to the entity register of that DRS. Clause (v) deals with negation. As negation affects local discourse structure, the SDRS remains unaltered, but the DRS in which the negation will be placed requires an update. Clause (vi) translates the discourse relations. Given an SBN S=[$s_1, \ldots, s_n$], its SDRS translation is defined as ($[\mathrm{BOS}]^{\rho\sigma}_{\rho_0\sigma_0} \otimes [s_1]^{\rho_0\sigma_0}_{\rho_1\sigma_1} \otimes [s_2]^{\rho_1\sigma_1}_{\rho_2\sigma_2} \otimes$, $\ldots, \otimes [s_n]^{\rho_{n-1}\sigma_{n-1}}_{\rho_n\sigma_n}$), where $\otimes$ is the SDRS merge operation, and σ and ρ are the initial (empty) registers.

9 Conclusion and Future Work

SBN, the notational variant of DRS that I introduced in this article, enables a way to manually code a DRS as a sequence of elements, without resorting to explicit variable names and removing several notational redundancies. It is similar in spirit to the formalisms of Van Eijck (2001) and Dekker (2016), who also apply indexing to DRT in order to eliminate variables, although our way of indexing is different (it is bi-directional, and is applied to DRS labels as well). SBN shares characteristics with the PENMAN notation (Kasper 1989), and formalisms based on that such as Abstract Meaning Representation (Langkilde & Knight 1998, Banarescu et al. 2013), but has three advantages: (1) the alignment between linguistic input and meaning representation components is easier; (2) there is a way to assign scope for negation and other structural relations; and (3) there is no need to assign variable names. The latter sets SBN aside from other meaning representations with flat structures (Schiehlen et al. 2000, Copestake et al. 2005).

What remains to be investigated is whether relative indices for co-reference, in the way as introduced in this article, are really easier to annotate by humans than absolute ones, and whether machine learning benefits from the notation as well. Indeed, large amounts of labelled data is the key for development of state-of-the-art semantic interpretation technology. Currently there are two extreme ways of annotating meaning representations for open domain applications. The first one is the grammar-based approach, where a computational grammar and parser assist the human annotator to select the appropriate semantic analysis for a sentence (Basile et al. 2012, Bender et al. 2015). The second approach ignores the aid of a computational grammar or parser, and requires trained human annotators to provide the meaning representation directly on the basis of the input string (Banarescu et al. 2013, Bunt 2020). The advocates of the grammar-based approach claim a more controlled way of producing meaning representations, ensuring consistency among annotators. The supporters of the grammar-denial approach claim more flexibility and independence of grammar, unhindered by compositionality constraints.

References

Abzianidze, Lasha, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen & Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th conference of the european chapter of the association for computational linguistics*, 242–247. Valencia, Spain.

Abzianidze, Lasha, Johan Bos & Stephan Oepen. 2020. DRS at MRP 2020: Dressing up discourse representation structures as graphs. In *Proceedings of the conll 2020 shared task: Cross-framework meaning representation parsing*, 23–32. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.conll-shared.2. https://www.aclweb.org/anthology/2020.conll-shared.2.

Asher, Nicholas. 1993. *Reference to Abstract Objects in Discourse*. Kluwer Academic Publishers.

Asher, Nicholas & Alex Lascarides. 2003. *Logics of conversation* Studies in natural language processing. Cambridge University Press.

Asher, Nicholas & Laure Vieu. 2005. Subordinating and coordinating discourse relations. *Lingua* 115(4). 591–610. https://doi.org/https://doi.org/10.1016/j.lingua.2003.09.017. https://www.sciencedirect.com/science/article/pii/S0024384103001475.

Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer & Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, 178–186. Sofia, Bulgaria.

Basile, Valerio, Johan Bos, Kilian Evang & Noortje Venhuizen. 2012. A platform for collaborative semantic annotation. In *Proceedings of the demonstrations at the 13th conference of the european chapter of the association for computational linguistics (eacl)*, 92–96. Avignon, France.

Bateman, John A., Robert T. Kasper, Joerg F. L. Schuetz & Erich H. Steiner. 1989. A new view on the process of translation. In *Proceedings of the fourth conference on european chapter of the association for computational linguistics*, 282–290.

Beaver, David I. 2002. Presupposition Projection in DRT: A Critical Assesment. In David Beaver, Luis Casillas, Brady Clark & Stefan Kaufmann (eds.), *The construction of meaning*, 23–43. Stanford University.

Bender, Emily M., Dan Flickinger, Stephan Oepen, Woodley Packard & Ann Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proceedings of the 11th international conference on computational semantics*, 239–249. London, UK: Association for Computational Linguistics. https://www.aclweb.org/anthology/W15-0128.

Bos, J. 2003. Implementing the Binding and Accommodation Theory for Anaphora Resolution and Presupposition Projection. *Computational Linguistics* 29(2). 179–210.

Bos, Johan. 2001. DORIS 2001: Underspecification, Resolution and Inference for Discourse Representation Structures. In Patrick Blackburn & Michael Kohlhase (eds.), *Icos-3, inference in computational semantics*, 117–124.

Bos, Johan. 2008. Wide-Coverage Semantic Analysis with Boxer. In J. Bos & R. Delmonte (eds.), *Semantics in text processing. step 2008 conference proceedings*, vol. 1 Research in Computational Semantics, 277–286. College Publications.

Bos, Johan. 2015. Open-domain semantic parsing with boxer. In Beáta Megyesi (ed.), *Proceedings of the 20th nordic conference of computational linguistics (nodalida 2015)*, 301–304.

Bos, Johan & Lasha Abzianidze. 2019. Thirty musts for meaning banking. In *Proceedings of the first international workshop on designing meaning representations*, 15–27. Florence, Italy: Association for Computational Linguistics. https://www.aclweb.org/anthology/W19-3302.

Bos, Johan, Valerio Basile, Kilian Evang, Noortje Venhuizen & Johannes Bjerva. 2017. The Groningen Meaning Bank. In Nancy Ide & James Pustejovsky (eds.), *Handbook of linguistic annotation*, vol. 2, 463–496. Springer.

Bos, Johan, Elsbeth Mastenbroek, Scott McGlashan, Sebastian Millies & Manfred Pinkal. 1994. A Compositional DRS-Based Formalism for NLP-Applications. In Harry Bunt, Reinhard Muskens & Gerrit Rentier (eds.), *International workshop on computational semantics*, University of Tilburg, The Netherlands.

de Bruijn, Nicolaas Govert. 1972. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the church-rosser theorem. In *Indagationes mathematicae (proceedings)*, vol. 75, 381–392. Elsevier.

Bunt, Harry. 2020. Annotation of quantification: The current state of ISO 24617-12. In *16th joint acl - iso workshop on interoperable semantic annotation proceedings*, 1–12. Marseille: European Language Resources Association.

Copestake, Ann, Dan Flickinger, Ivan Sag & Carl Pollard. 2005. Minimal recursion semantics: An introduction. *Journal of Research on Language and Computation*

3(2–3). 281–332.

Curran, James, Stephen Clark & Johan Bos. 2007. Linguistically Motivated Large-Scale NLP with C&C and Boxer. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, 33–36. Prague, Czech Republic.

Dekker, Paul. 2016. Exclusively indexical deduction. *Review of Symbolic Logic* 9(3). 603–637.

Evang, Kilian. 2019. Transition-based DRS parsing using stack-LSTMs. In *Proceedings of the IWCS shared task on semantic parsing*, Gothenburg, Sweden: Association for Computational Linguistics.

Fellbaum, Christiane (ed.). 1998. *Wordnet. an electronic lexical database*. The MIT Press.

Geach, Peter. 1962. *Reference and generality: An examination of some medieval and modern theories.* Ithaca, New York: Cornell University Press.

Geurts, Bart. 1999. *Presuppositions and pronouns*. Elsevier, London.

Geurts, Bart, David I. Beaver & Emar Maier. 2020. Discourse Representation Theory. In Edward N. Zalta (ed.), *The Stanford encyclopedia of philosophy*, Metaphysics Research Lab, Stanford University spring 2020 edn.

Heim, Irene. 1982. *The semantics of definite and indefinite noun phrases*: University of Massachusetts dissertation.

Johnson, Mark & Ewan Klein. 1986. Discourse, anaphora and parsing. In *11th international conference on computational linguistics. proceedings of coling '86*, 669–675. University of Bonn.

Kadmon, Nirit. 2001. *Formal pragmatics*. Blackwell.

Kamp, Hans. 1981. A Theory of Truth and Semantic Representation. In Jeroen Groenendijk, Theo Janssen & Martin Stokhof (eds.), *Mathematical Center Tracts*, vol. 131, 277–322. Amsterdam: Mathematisch Centrum.

Kamp, Hans, Josef van Genabith & Uwe Reyle. 2011. Discourse Representation Theory. In Dov M. Gabbay & Franz Guenthner (eds.), *Handbook of philosophical logic*, vol. 15, 125–394. Elsevier, MIT.

Kamp, Hans & Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Dordrecht: Kluwer.

Kasper, Robert T. 1989. A flexible interface for linking applications to penman's sentence generator. In *Proceedings of the darpa speech and natural language workshop*, 153–158. Philadelphia.

Kipper, Karin, Anna Korhonen, Neville Ryant & Martha Palmer. 2008. A large-scale classification of English verbs. *Language Resources and Evaluation* 42(1). 21–40.

Klein, Ewan. 1987. VP Ellipsis in DR Theory. In Jeroen Groenendijk et al. (eds.), *Studies in Discourse Representation Theory and the Theory of Generalised Quantifiers*, vol. 8, 161–187. Dordrecht: FLORIS.

Langkilde, Irene & Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *COLING 1998 volume 1: The 17th international conference on computational linguistics*, 704–710.

Le, Phong & Willem Zuidema. 2012. Learning compositional semantics for open domain semantic parsing. In *Proceedings of coling 2012*, vol. 1 COLING 2012, 33–50. Association for Computational Linguistics.

Liu, Jiangming, Shay B. Cohen & Mirella Lapata. 2019. Discourse representation structure parsing with recurrent neural networks and the transformer model. In *Proceedings of the IWCS shared task on semantic parsing*, 24–29. Gothenburg, Sweden: Association for Computational Linguistics.

de Marneffe, Marie-Catherine, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre & Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the ninth international conference on language resources and evaluation (LREC'14)*, 4585–4592. Reykjavik, Iceland: European Language Resources Association (ELRA).

Muskens, Reinhard. 1996. Combining Montague Semantics and Discourse Representation. *Linguistics and Philosophy* 19. 143–186.

Pinkal, Manfred & Michael Kohlhase. 2000. Feature logic for dotted types: A formalism for complex word meanings. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, 521–528. Hong Kong: Association for Computational Linguistics.

Pustejovsky, James. 1995. *The generative lexicon*. Cambridge, Massachusetts / London: The MIT Press.

Van der Sandt, Rob A. 1992. Presupposition Projection as Anaphora Resolution. *Journal of Semantics* 9. 333–377.

Schiehlen, Michael, Johan Bos & Michael Dorna. 2000. Verbmobil interface terms (vits). In Wolfgang Wahlster (ed.), *Verbmobil: Foundations of speech-to-speech translation*, Springer.

Vallduví, Enric. 2016. Information structure. In Maria Aloni & Paul Dekker (eds.), *The cambridge handbook of formal semantics*, chap. 23, 728–755. Cambridge: Cambridge University Press.

Van Eijck, Jan. 2001. Incremental dynamics. *Journal of Logic, Language and Information* 10. 319–351.

Van Eijck, Jan & Hans Kamp. 1997. Representing discourse in context. In J. van Benthem & A. ter Meulen (eds.), *Handbook of logic and language*, 179–237. Elsevier.

Van Noord, Rik, Lasha Abzianidze, Antonio Toral & Johan Bos. 2018. Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics* 6. 619–633.

Venhuizen, Noortje, Johan Bos, Petra Hendriks & Harm Brouwer. 2018. Discourse semantics with information structure. *Journal of Semantics* 35(1). 127–169.

Wada, Hajime & Nicholas Asher. 1986. BUILDRS: An Implementation of DR Theory and LFG. In *11th international conference on computational linguistics. proceedings of coling '86*, 540–545. University of Bonn.

Zeevat, Henk. 1989. A compositional approach to discourse representation theory. *Linguistics and Philosophy* 12. 95–131.

Johan Bos
University of Groningen
Faculty of Arts
Oude Kijk in 't Jatstraat 26
9712 EK Groningen
The Netherlands
johan.bos@rug.nl

A A simple annotation example in five steps

Assume that our task is to provide SBN for "Tim was killed by an intruder." and its Dutch translation "Tim werd door een inbreker vermoord." The first step is to segment the sentences and produce the segments from top to bottom (Figure 16). The decisions made in this step are not definite—segmentation can always be revised in a later stage. The segmentation is rough and not part of the SBN representation itself; its sole purpose is to aid the annotator.

`% Tim`	`% Tim`
`% was`	`% werd`
`% killed`	`% door`
`% by`	`% een`
`% an`	`% inbreker`
`% intruder.`	`% vermoord.`

Figure 16 Step 1 of SBN annotation: segmentation. On the left we have an English text and on the right a Dutch translation. This segmentation might be revised later to yield a better alignment between meaning and text.

In Step 2 we identify the entities mentioned in the text (Figure 17). These can be objects, but also events and states. In this example we have three main entities. Here we follow the Parallel Meaning Bank (Abzianidze et al. 2017) and use WordNet (Fellbaum 1998) concepts as interlingual predicates (the first affix denotes the part of speech used in WordNet (noun, verb, adjective, adverb); the second affix is a sense number). Named entities get the concept describing its type. (We ignore tense here for the sake of clarity.)

person.n.01	`% Tim`	person.n.01	`% Tim`
	`% was`		`% werd`
kill.v.01	`% killed`		`% door`
	`% by`		`% een`
	`% an`	intruder.n.01	`% inbreker`
intruder.n.01	`% intruder.`	kill.v.01	`% vermoord.`

Figure 17 Step 2 of SBN annotation: identify the entities and represent them as WordNet concepts alligned with the text.

In Step 3 we add more information to the entities that require this (Figure 18). For named entities we add the name as a literal with the relation `Name`. Agent nouns are represented as a relational complex. Experienced annotators typically combine Step 2 and Step 3 in one go.

person.n.01	% Tim	person.n.01	% Tim
Name "Tim"	%	**Name "Tim"**	%
	% was		% werd
kill.v.01	% killed		% door
	% by	**person.n.01**	% een
person.n.01	% an	**Role +1**	%
Role +1	%	intruder.n.01	% inbreker
intruder.n.01	% intruder.	kill.v.01	% vermoord.

Figure 18 Step 3 of SBN annotation: refining the identified entities.

In Step 4 we add the relations between the entities (Figure 19). Here we use the inventory of VerbNet (Kipper et al. 2008) roles. The verb *to kill* makes the roles Agent and Patient explicit in this example. In the English case we can just add them after the kill.v.01 event, and because *Tim* plays the unlucky role of patient, the associated index is set to -1. The intruder receives the role of agent, but note that the index refers to the concept person, and therefore the index should be set to $+1$ and not $+2$. In the Dutch case we have the same roles, and we could add them to the kill.v.01 event, but as the word order is different the indices would require the values -3 and -2 repectively. Here we choose for a better alignment, and inverse the role of agent, and add it to the person.n.01 entity.

person.n.01	% Tim	person.n.01	% Tim
Name "Tim"	%	Name "Tim"	%
	% was		% werd
kill.v.01	% killed	person.n.01	%
Patient -1	%	**AgentOf +2**	% door
Agent +1	% by	Role +1	% een
person.n.01	% an	intruder.n.01	% inbreker
Role +1	%	kill.v.01	% vermoord.
intruder.n.01	% intruder.	**Patient -3**	%

Figure 19 Step 4 of SBN annotation: adding relations. In the Dutch case (right) the role is inversed in order to maximize alignment.

In the final step, Step 5, we add the structural relations. In this example there is no negation, and no discourse structure (because we are dealing with a single sentence), so there is nothing to add. Note that adding the structural relations to SBN will not affect the indices of the relations.

B Translation from SBN to DRS

Illustrated here for "Tom loves chocolate cake", with negative and positive de Bruijn-indices. The first column is the representation in SBN, the second the register, and the third the translated DRS.

[person Name "Tom" love Experiencer -1 Stimulus +1 cake MadeOf +1 chocolate]	$\emptyset$	
[Name "Tom" love Experiencer -1 Stimulus +1 cake MadeOf +1 chocolate]	$\{0 \to x\}$	x person(x)
[love Experiencer -1 Stimulus +1 cake MadeOf +1 chocolate]	$\{0 \to x\}$	x person(x) Name(x,"Tom")
[Experiencer -1 Stimulus +1 cake MadeOf +1 chocolate]	$\{0 \to e, -1 \to x\}$	x e person(x) Name(x,"Tom") love(e)
[Stimulus +1 cake MadeOf +1 chocolate]	$\{0 \to e, -1 \to x\}$	x e person(x) Name(x,"Tom") love(e) Experiencer(e,x)
[cake MadeOf +1, chocolate]	$\{+1 \to y, 0 \to e, -1 \to x\}$	x e person(x) Name(x,"Tom") love(e) Experiencer(e,x) Stimulus(e,y)
[MadeOf +1 chocolate]	$\{0 \to y, -1 \to e, -2 \to x\}$	x e y person(x) Name(x,"Tom") love(e) Experiencer(e,x) Stimulus(e,y) cake(y)
[chocolate]	$\{+1 \to z, 0 \to y, -1 \to e, -2 \to x\}$	x e y person(x) Name(x,"Tom") love(e) Experiencer(e,x) Stimulus(e,y) cake(y) MadeOf(y,z)
[ ]	$\{0 \to z, -1 \to y, -2 \to e, -3 \to x\}$	x e y z person(x) Name(x,"Tom") love(e) Experiencer(e,x) Stimulus(e,y) cake(y) MadeOf(y,z) chocolate(z)

C Translation from SBN with negation to DRS

Illustrated here for "Tom does not love chocolate cake", with negative and positive de Bruijn-indices. **B** is a global update function form DRS labels to DRSs. **R** is a global update function form DRS labels to registers.

[person Name "Tom" -1 NEGATION love Experiencer -1 Stimulus +1 cake MadeOf +1 chocolate]	$\{0 \rightarrow b1\}$	**R**(b1)=$\emptyset$	**B**(b1)= $\begin{array}{\|c\|}\hline \\ \hline \\ \hline\end{array}$
[-1 NEGATION love Experiencer -1 Stimulus +1 cake MadeOf +1 chocolate]	$\{0 \rightarrow b1\}$	**R**(b1)=$\{0 \rightarrow x\}$	**B**(b1)= $\begin{array}{\|c\|}\hline \text{x} \\ \hline \text{person(x)} \\ \text{Name(x,"Tom")} \\ \hline\end{array}$
[love Experiencer -1 Stimulus +1 cake MadeOf +1 chocolate]	$\{0 \rightarrow b2, -1 \rightarrow b1\}$	**R**(b1)=$\{0 \rightarrow x\}$	**B**(b1)= $\begin{array}{\|c\|}\hline \text{x} \\ \hline \text{person(x)} \\ \text{Name(x,"Tom")} \\ \neg \mathbf{B}\text{(b2)} \\ \hline\end{array}$
[Experiencer -1 Stimulus +1 cake MadeOf +1 chocolate]		**R**(b2)=$\{0 \rightarrow e, -1 \rightarrow x\}$	**B**(b2)= $\begin{array}{\|c\|}\hline \text{e} \\ \hline \text{love(e)} \\ \hline\end{array}$
[Stimulus +1 cake MadeOf +1 chocolate]		**R**(b2)=$\{0 \rightarrow e, -1 \rightarrow x\}$	**B**(b2)= $\begin{array}{\|c\|}\hline \text{e} \\ \hline \text{love(e)} \\ \text{Experiencer(e,x)} \\ \hline\end{array}$
[cake MadeOf +1, chocolate]		**R**(b2)=$\{+1 \rightarrow y, 0 \rightarrow e, -1 \rightarrow x\}$	**B**(b2)= $\begin{array}{\|c\|}\hline \text{e} \\ \hline \text{love(e)} \\ \text{Experiencer(e,x)} \\ \text{Stimulus(e,y)} \\ \hline\end{array}$
[MadeOf +1 chocolate]		**R**(b2)=$\{0 \rightarrow y, -1 \rightarrow e, -2 \rightarrow x\}$	**B**(b2)= $\begin{array}{\|c\|}\hline \text{e y} \\ \hline \text{love(e)} \\ \text{Experiencer(e,x)} \\ \text{Stimulus(e,y)} \\ \text{cake(y)} \\ \hline\end{array}$
[chocolate]		**R**(b2)=$\{+1 \rightarrow z, 0 \rightarrow y, -1 \rightarrow e, -2 \rightarrow x\}$	**B**(b2)= $\begin{array}{\|c\|}\hline \text{x e y} \\ \hline \text{love(e)} \\ \text{Experiencer(e,x)} \\ \text{Stimulus(e,y)} \\ \text{cake(y)} \\ \text{MadeOf(y,z)} \\ \hline\end{array}$
[ ]		**R**(b2)=$\{0 \rightarrow z, -1 \rightarrow y, -2 \rightarrow e, -3 \rightarrow x\}$	**B**(b2)= $\begin{array}{\|c\|}\hline \text{e y z} \\ \hline \text{love(e)} \\ \text{Experiencer(e,x)} \\ \text{Stimulus(e,y)} \\ \text{cake(y)} \\ \text{MadeOf(y,z)} \\ \text{chocolate(z)} \\ \hline\end{array}$

D Converting Universal Dependency Graphs to SBN

Universal Dependencies (UD) is a formalism that describes syntactic relations between words within a sentence, covering various (in fact, more than 100) languages (de Marneffe et al. 2014). Here we show how output of a UD-parser can be converted into SBN. The output shown below in Figure 20 is taken from the Stanford Parser.

```
nmod:poss(dog-2, My-1)              nmod:poss(+1, My-1)
nsubj(likes-4, dog-2)               nsubj(+2, dog-2)
advmod(likes-4, also-3)             advmod(+1, also-3)
root(ROOT-0, likes-4)               root(ROOT-0, likes-4)
xcomp(likes-4, eating-5)            xcomp(-1, eating-5)
obj(eating-5, sausage-6)            obj(-1, sausage-6)
```

Figure 20 Parser output (left) and the result of replacing absolute identifiers by relative ones (right).

The conversion from an UD parse to SBN proceeds over various steps. The first step involves replacing the absolute identifiers to relative ones—the future SBN indices. Note that every relation consists of the positions of the words that it connects. Computing the absolute identifiers can be carried out by subtracting the index of the first argument of the UD relation with its second index, e.g., for `nmod:poss` in the example this is $2-1=+1$, and for `xcomp` it's $4-5=-1$ (see Figure 20).

In the next step we add a concept for each word and a relation (except for the root). Essentially this is a mapping from the syntactically inspired universal dependency relations to semantics relations (here we use the inventory of VerbNet as in the Parallel Meaning Bank (Abzianidze et al. 2017)). We then instantiate the arguments of each obtained relation: the first argument is the index assigned in the previous step; the second argument is always 0 (Figure 21).

```
nmod:poss(+1, My-1)      person  Owner(+1,0)        % My
nsubj(+2, dog-2)         dog     Experiencer(+2,0)  % dog
advmod(+1, also-3)       also    Manner(+1,0)       % also
root(ROOT-0, likes-4)    like                       % likes
xcomp(-1, eating-5)      eat     Stimulus(-1,0)     % eating
obj(-1, sausage-6)       sausage Patient(-1,0)      % sausage
```

Figure 21 Adding concepts and relations.

And this is, essentially, the end of the conversion. We could convert this SBN into DRS. Or we can we play around with 0-drop by inversing the relations to make the SBN more compact (Figure 22). Alternatively, we can move the relations to avoid inversion, and then apply 0-drop (Figure 23).

```
person  Owner(+1,0)         person  OwnerOf +1        % My
dog     Experiencer(+2,0)   dog     ExperiencerOf +2  % dog
also    Manner(+1,0)        also    MannerOf +1       % also
like                        like                      % likes
eat     Stimulus(-1,0)      eat     StimulusOf -1     % eating
sausage Patient(-1,0)       sausage PatientOf -1      % sausage
```

Figure 22 SBN without (left) and with 0-drop (right) after role inversion.

This was a simple, straightforward example. It goes without saying that not all UD parses are this easy to convert. Not all words introduce entities (e.g., prepositions, discourse connectors), and need a different treatment in the conversion. Negation particles (and universal quantigiers) need to introduce the local discourse relation NEGATION, but it is important where in SBN these relations are inserted, and there is probably no straightforward, systematic way to do so. Cases with coordination, most likely, require special rules.

```
person                      person                    % My
dog     Owner(0,-1)         dog     Owner -1          % dog
also                        also                      % also
like    Manner(0,-1)        like    Manner -1         % likes
        Experiencer(0,-2)           Experiencer -2    %
        Stimulus(0,+1)              Stimulus +1       %
eat     Patient(0,+1)       eat     Patient +1        % eating
sausage                     sausage                   % sausage
```

Figure 23 SBN without (left) and with 0-drop (right) after role shifting.

E Graphs

In this appendix I show how the DRS described by SBN can be visualised in directed graphs. The idea to visualise meaning representations as a directed graph is not new, obviously. One of the attractive features of AMR is their graph representation (Langkilde & Knight 1998). Because DRS are more expressive than AMR graph representations for DRS are not appealing because of the required reification steps (Abzianidze et al. 2020). But the simpliciation that SBN offers has also consequences for graph visualisation. Essentially, it is 0-drop in SBN that gives us the insight that DRS can be transformed to triples.

	BOS
male.n.02	Name "Tim"
time.n.08	$\prec$ now
hit.v.02	Time -1
	Patient -2
	Theme +1
baseball.n.02	

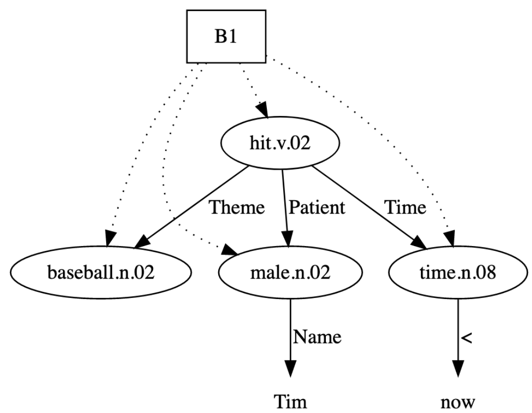


Figure 24 SBN in graphical notation, for “Tim was hit by a baseball” (Figure 8).

	BOS
male.n.02	Name "JFK"
time.n.08	$\prec$ now
assassinate.v.02	Patient -2
	Time -1
	Time +1
time.n.08	= -2
	YearOfCentury "1968"

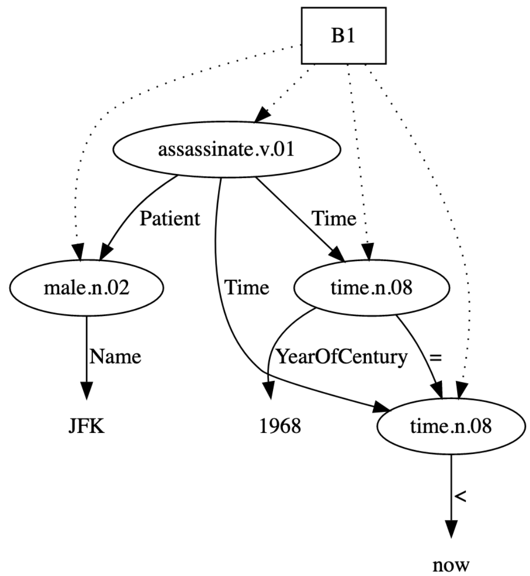


Figure 25 Graph for “JFK was assassinated in 1968” (Figure 9).

	BOS
male.n.02	Name "JFK"
time.n.08	$\prec$ now
	NEGATION -1
assassinate.v.02	Patient -2
	Time -1
	Time +1
time.n.08	= -2
	YearOfCentury "1969"

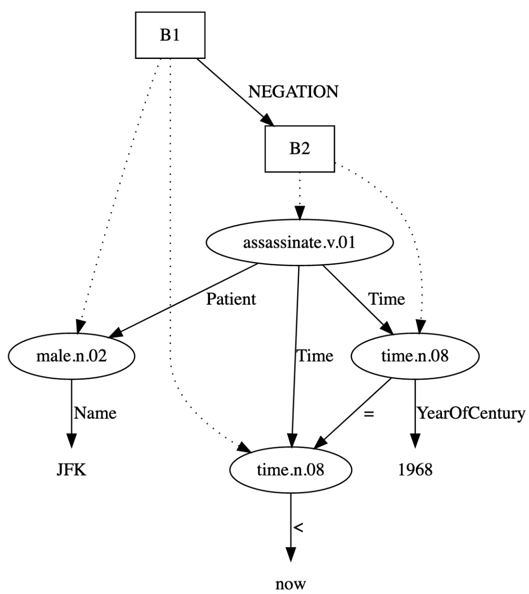


Figure 26 Graph for “JFK was not assassinated in 1969”

	NEGATION -1
farmer.n.01	
own.v.01	Pivot -1
	Theme +1
donkey.n.01	
	NEGATION -1
beat.v.01	Agent -3
	Patient +1
entity.n.01	= -2

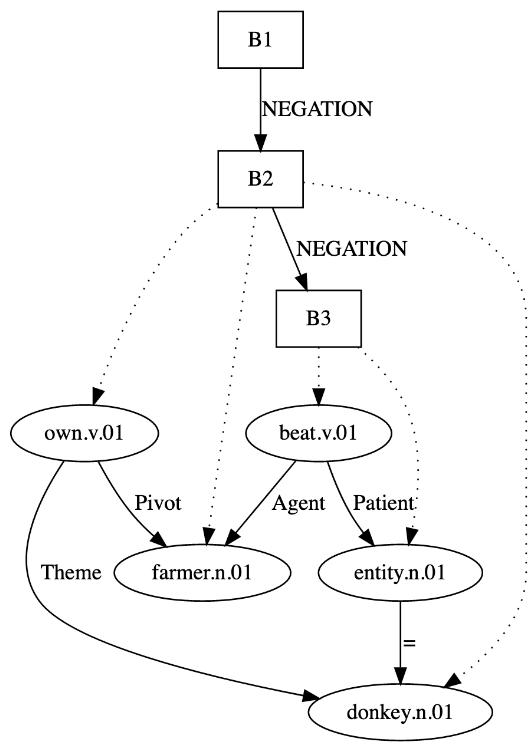


Figure 27 SBN and corresponding DRS graph for "Every farmer who owns a donkey beats it."