The Sequence Notation

Bos, Johan

Published in:
Proceedings of the 15th International Conference on Computational Semantics

The Sequence Notation: Catching Complex Meanings in Simple Graphs

Johan Bos
Center for Language and Cognition
University of Groningen
johan.bos@rug.nl

Abstract

Current symbolic semantic representations proposed to capture the semantics of human language have served well to give us insight in how meaning is expressed. But they are either too complicated for large-scale annotation tasks or lack expressive power to play a role in inference tasks. What I propose is a meaning representation system that it is interlingual, model-theoretic (by translation to first-order logic), and variable-free. It divides the labour involved in representing meaning along three levels: concept, roles, and contexts. As natural languages are expressed as sequences of phonemes or words, the meaning representations that I propose are likewise sequential. However, the resulting meaning representations can also be visualised as directed acyclic graphs.

1 Introduction

There are many proposals for representing meaning of natural language expressions in a formal way. These originate from various disciplines, including formal semantics (Thomason, 1974; Dowty et al., 1981; Heim, 1982; Kamp, 1984; Groenendijk and Stokhof, 1990; Chierchia, 1992), artificial intelligence (Schubert, 1976; Sowa, 1984, 1995; Schubert, 2015), and computational linguistics (Copestake et al., 2005; Banarescu et al., 2013; Abzianidze et al., 2017; Martínez Lorenzo et al., 2022). Although most of these do a tremendous job in analysing meaning, I think none of them offers a meaning representation that is the ideal candidate for large-scale annotation tasks in computational semantics requiring supervised machine learning: some of them lack expressive power, some of them are only partially interpretable, some of them are tailored to specific natural languages, and yet others are featured with a complex syntax that makes them unsuitable for human annotation tasks.

What nearly all of these semantic formalisms have in common is that they all share the property of using *variables* ranging over (first-order or higher-order) entities. Representations without variables have potential advantages and benefits when we think of human annotation efforts, machine learning approaches, and meaning visualisations techniques. Hence, the question I take at heart is whether it is possible to eliminate variables from formal meaning representations without losing expressive power required to interpret linguistic expressions.

The goal of this paper is to propose a meaning representation that is a healthy mixture of interlinguality, simplicity, and expressiveness. With interlinguality I mean a meaning representation that is not designed to support a single language. With simplicity I mean a kind of semantic representation that supports an intuitive way of drawing a graphical representation of the meaning that it is supposed to represent. With expressiveness I mean at least the expressive power of first-order logic (i.e., quantification, negation, and conjunction) and support for discourse phenomena such as co-reference and discourse structure.

Current graph-based meaning representations such as AMR, Abstract Meaning Representation (Banarescu et al., 2013) lack expressive power. Current logic-based meaning representations such as DRS, Disourse Representation Structure (Kamp and Reyle, 1993) are unattractive to represent as graphs as they require substantial reification (Abzianidze et al., 2020). What I propose is a formalism that combines AMR with DRS while removing notational redundancies such as variables and punctuation symbols. It takes the attractive and simple graph-based visualisation of AMR but adds the "boxes" of DRS, arriving at a formalism that includes negation and quantification as in predicate logic. The formalism accommodates two ways

Proceedings of the 15th International Conference on Computational Semantics pages 195–208
June 21–23, 2023.

of represent meaning: the variable-free sequential notation, and directed acyclic graphs. The variable-free sequence notation is expected to be advantageous for human annotation efforts and language technology applications that require machine learning (e.g., applying neural networks for the tasks of semantic parsing or natural language generation). This is because it doesn't require the process of using variables nor explicit indication of scope for logical operators like negation. The graph representation is convenient for human readers and for software designed to work with graphs.

2 Simplifying Meaning Representations

In this section I will present a new meaning representation system. Using this formalism, annotation can be done with a simple text editor. There are no logical variables but there is still support for negation and scope. The primary encoding of meaning is done in sequence notation. But the meanings can be visualised as directed acyclic graphs. The sequence notation can be applied to various meaning representation formalisms including AMR and DRS. In this paper I focus on the latter.

2.1 The Sequence Notation

I will introduce the sequence notation by first explaining what the elementary building block are. Then I explain how sequences can be constructed, visualised, and interpreted. The sequence notation has the following ingredients (with examples in brackets):

- Concepts (`cat.n.01`, `see.v.03`, ...)
- Constants (`"Mary"`, `speaker`, `20`, π, ...)
- Roles (`Agent`, `Theme`, `Patient`, ...)
- Operators ($=$, $\neq$, $\approx$, $<$, $\leq$, $\prec$, ...)
- Indices (..., `−2`, `−1`, `+1`, `+2`, ...)
- Contexts
- Separators (`NEGATION`, `CONJUNCTION`, `EXPLANATION`, `NARRATION`, ...)
- Connectors (..., `<2`, `<1`, `>1`, ...)

Concepts identify an entity or event as belonging to a certain class within a domain ontology. Concepts are always written in lower case and are represented as interlingual WordNet synsets as triplets comprising of a lemma, part of speech (n, v, a, or r) and a sense number, e.g., `cat.n.01` represents the first sense of the noun cat. I view a WordNet synset as language-neutral, even though in this paper I will use the synsets as defined in Princeton's American English WordNet 3.0 (Fellbaum, 1998) because of its common use in the NLP community. Adoption of a multi-lingual wordnet (Navigli and Ponzetto, 2012; Bond and Foster, 2013) would eventually be the target in a large-scale multilingual implementation.

Constants comprise proper names (of people, animals, organisations, locations, artifacts), numerical values (integers and reals), times and dates, literal mentions. They also include deictic references: the speaker of the utterance (`speaker`), the addressee (`hearer`), the utterance time (`now`) and location (`here`).

Roles connect an event to an entity (or relate two entities to each other). Roles always start with an uppercase character followed by lowercase to distinguish them from concepts. The roles used in this paper are by and large based on thematic role inventory provided by VerbNet and LIRICS (Kipper et al., 2008; Bonial et al., 2011). The connections between events and entities are established with *indices* (see § 3.3). The operators are used to express comparisons between entities and are written in mathematical notation or with three uppercase letters (EQU, NEQ, SIM, LES, LEQ, TPR, and so on).

All concepts are introduced in a *context*. Contexts are not explicit in sequence notation. A *separator* introduces a new context connecting it to a previously introduced context as indicated by its *connector* (see § 3.4). Separators are always written in all uppercase to distinguish them from roles and concepts.

2.2 Forming Sequences

A role followed by a constant is an *anchor*. So, `Name "Mary"` is an anchor. A role followed by an index is a *hook*. Hence, `Owner +1` is a hook. A *simple sequence* is a sequence of one or more concepts, where a concept can be followed by zero or more anchors or hooks. Therefore, `dog.n.01` is a simple sequence, and so are `cat.n.01 dog.n.01`, and `cat.n.01 Owner +1 person.n.01 Name "Mary"`. A simple sequence represents a single *context*. A context is similar to a box in Discourse Representation Theory (Kamp and Reyle, 1993). They set the stage for the entities that play a part of the context.

A *complex sequence* is formed by combining two (simple or complex) sequences using a separator and connector. For instance, `person.n.01 NEGATION <1 smile.v.01 Theme -1` is a complex sequence, constructed from the simple sequences `person.n.01` and `smile.v.01 Theme -1` using the separator `NEGATION` and connector `<1` as glue. A complex sequence represents two or more contexts.

2.3 Graph Visualisation

A meaning in sequence notation can be visualised as a directed acyclic graph, where the vertices denote concepts, contexts or constants, and the edges are decorated by roles or comparison operators. Concept nodes are drawn as ovals and context nodes as boxes. Figure 1 shows how the sequence `male.n.02 Name "Tom" time.n.08 TPR now cry.v.02 Agent -2 Time -1` is visualised as a graph.

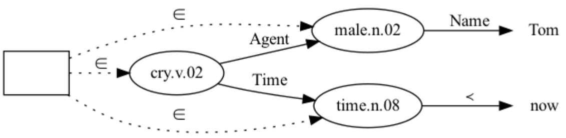

Figure 1: Graph for "Tom was crying."

Although contexts are implicit in the sequence notation, drawn as a graph the contexts become explicit as boxes. Each concept is related to a context with a membership edge connected to its context, as Figure 1 shows.

Note that the sequence notation corresponds to a topological ordering of its graph. As a directed acyclic graph can give rise to one or more topologic orderings, the preferred ordering is one that resembles the linguistic realisation. As a consequence, a meaning-preserving translation from a sentence in one language to another language could result in a single meaning representation that would show different orders in sequence notation for the two languages. This is exemplified for a simple English sentence (1) and its translation in Dutch (2) with a different word order.

(1) a. (that) a boy bought a book.
    b. `boy.n.01 buy.v.01 Agent -1 Theme +1 book.n.02`

(2) a. (dat) een jongen een boek kocht.
    b. `boy.n.01 book.n.01 buy.01 Agent -2 Theme -1`

2.4 Role Inversion

A role connects two entities, but can only be hooked to one. This could cause unwanted side-effects such as cycles in the corresponding graph (see previous section) or imperfect linguistic alignment (see next section). The mechanism of *role inversion*, as introduced in description logics, AI approaches of knowledge representation and AMR, is therefore a useful one to have at one's disposal because of the added flexibility in creating meanings.

Role inversion is defined as follows: $\forall R \forall x \forall y (R(x,y) \leftrightarrow \overleftarrow{R}(y,x))$, where $\overleftarrow{R}$ is the inversion of R. In words: every role, a binary relation, has a dual, and if you want to swap the arguments of a role, you can do so using the dual without changing the overall meaning. Following the convention in AMR, I use the `Of` suffix to indicate inverted roles. Consider (3) with an inverted role and compare it to the earlier (1).

(3) a. A boy bought a book.
    b. `boy.n.01 buy.v.01 Agent -1 book.n.02 ThemeOf -1`

Role inversion does not affect the truth-conditional meaning, and for checking syntactic equivalence of graphs inverted roles are normalised (Cai and Knight, 2013). Role inversion gives us flexibility in the sequence notation, which is useful in semantic annotation tasks where linguistic alignment is important.

2.5 Linguistic Alignment

For practical purposes (human annotation and verification and natural language processing technologies using machine learning) it is convenient to get a close alignment between the meaning representation and the natural language expression that it forms the interpretation of. It is hard to align meaning graphs with text, which is linear by nature (Pourdamghani et al., 2014; Liu et al., 2018; Anchiêta and Pardo, 2020; Blodgett and Schneider, 2021). I show how a reasonably fine-grained alignment can be provided using the sequence notation. (Appendix B shows an elaborated example.)

Because the sequence notation is simply a succession of hooked or anchored concepts, possibly divided by context separators, it gives us a lot of freedom in the way it can be encoded. As most writing systems in western cultures possess a left-to-right direction, it is convenient to follow this convention when describing languages following this direction, as I have done in the examples above. However, for annotation purposes a top-to-bottom organisation is handy and perhaps also the most neutral seen from the perspective of the various writing systems used for natural languages. It is also used in computational linguistics to annotate text with labels classifying word tokens in categories for tasks such as part-of-speech or named entity tagging, known as the column-based format (Buchholz and Marsi, 2006). Figure 2 gives us the idea.

```
boy.n.01                                % A boy
bought.v.01 Agent -1 Theme +3 % bought
quantity.n.01 2 QuantityOf +1 % two
box.n.03 MeasureOf +1              % boxes of
bonbon.n.01                           % bonbons.
```

Figure 2: Aligning a sequence meaning with text.

Even though there is no one-to-one mapping between words and elements of the meaning representation, the alignment is reasonably executed, with all concepts in line with a noun, adjective, or verb. Prepositions, determiners, and particles aren't always directly alignable, and nor are discontinuous expressions. The alignment could be further improved using the machinery introduced by Blodgett and Schneider (2021).

2.6 Evaluation

Evaluation of meaning representation becomes important and interesting when one wants to compare two meanings that are independently produced for the same input. This could be a comparison between computer output and gold standard annotation (curated by a semanticist), or a comparison between two human-created meanings in order to calculate inter-annotator agreement. A simple proposal using existing software is put forward in Poelman et al. (2022) who convert sequential meanings to PENMAN format (Kasper, 1989) and then use SMATCH to compute overlap of triples (Cai and Knight, 2013). Therefore no new machinery is required to evaluate meanings in sequence notation, and improved evaluation metrics such as SEMBLEU can also be adopted easily (Song and Gildea, 2019).

3 Interpreting Sequences

In the previous section I showed how sequential meanings can be constructed. In this section I explain how they are interpreted. Appendix A illustrates how sequential meanings can be converted to Discourse Representation Structures from DRT.

3.1 Concepts

A concept in a sequence has a dual purpose: it (a) introduces an entity within its context, and (b) classifies it to a particular concept. Hence, every entity has a corresponding one-place predicate, a "guard", that classifies it within some background knowledge ontology.[1] Roughly speaking, a simple sequence of concepts $[\![C_1 \ldots C_n]\!]$ corresponds to the first-order formula $\exists \mathrm{x}_1 \ldots \exists \mathrm{x}_n(C(\mathrm{x}_1)\ldots C(\mathrm{x}_n))$. In the terminology of Discourse Representation Theory (Kamp and Reyle, 1993), a concept C that is part of a context B introduces a fresh discourse referent x in the domain of B and a basic condition $C(x)$ in the set of conditions of B.

3.2 Anchors

Anchors connect a concept in a meaning representation with an external entity. It can be seen as a means of grounding or anchoring abstract units of meaning with concrete entities present in the real world. The denotation of an anchored concept is defined as follows: $[\![C\ Rc]\!] = \exists x(C(x) \wedge R(x, c))$.

3.3 Hooks

A hook connects ("hooks") a concept to another concept by a two-place relation. Recall that a hook is always (a) attached to a concept and (b) ends with an index. The indices replace the variables found in traditional meaning representation, inspired by work of Nicolaas Govert de Bruijn (1972). There are negative and positive indices. As concepts are strictly ordered in the sequential notation, we can refer to a concept by refering to the relative position the relation is situated: the index 0 refers to the current concept, -1 to the concept introduced before the current concept, -2 to the concept before that, and so on. Negative indices refer to entities introduced before, and positive indices refer to entities that are available later in the sequence: $+1$ refers to a concept that is introduced after the current index. This mechanism is crucial to understand how hooks work, and bears also resemblance with how co-reference is implemented in Lexical Functional

[1] This is reminiscent of guarded quantifiers (Andréka et al., 1998), and it 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, I do this by adding a one-place predicate (a concept).

Grammar (Kaplan and Bresnan, 1982). The first-order logic interpretation of a concept with hooks is thus roughly defined as follows: $[\![C\ H_1 \cdots H_n]\!] = \exists x(C(x) \wedge H_1(x, y_1) \wedge \cdots \wedge H_n(x, y_n))$. Indices without an antecedent concept correspond to free variables in first-order logic.

3.4 Separators

A separator divides a sequential meaning representation into two contexts: the context before, and the context after the separator. Hence, a sequential meaning representation with n separators has exactly $n + 1$ contexts. There are various kinds of separators. The type of separator tells us what logical or rhetorical relationship exists between the two contexts. A key application of separators is the treatment of negation, disjunction and universal quantification, but separators also find use in assigning discourse structure and rhetorical relations in text.

A separator decorated with a connector <1 means that the separator connects two local contexts. A connector <2 means that the context following the separator is attached to an earlier introduced context: not the previous context but the one just before that. Newly introduced contexts always connect to a previously introduced context. A new context cannot be linked to more than one context. Usually, a separator connects two adjacent contexts. But it is possible that a separator connects two contexts that are not adjacent. This happens with wide-scope interpretations, presuppositional accommodation, non-local discourse relations, and disjunction.

4 Semantic Phenomena

4.1 Negation and Disjunction

Negation has impact on the structure of meaning: it doesn't introduce a new conceptual entity or hook, but rather packages the information in what is asserted as positive information and what is negative. In sequence notation, negation introduces the separator `NEGATION`, stating that the negated information following the separator is attached to the context just before the separator (Figure 3). Its first-order equivalent is ∃x(person.n.01(x) ∧¬∃y∃z(book.n.02(z) ∧ buy.v.01(y) ∧ Agent(y,x) ∧ Theme(y,z))). In DRT parlance, the corresponding DRS would have a nested box with a unary negation operator (see Figure 9 in Appendix A).

```
person.n.01                        % Somebody
  NEGATION <1                      %
buy.v.01 Agent -1 Theme +1         % bought
book.n.02                          % no book.
```

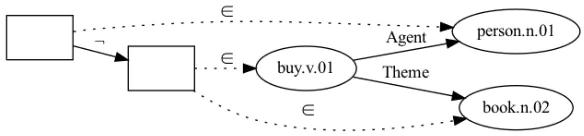

Figure 3: Graph for "Somebody bought no book."

Another example with negation is given in Figure 4, displaying a sequential meaning with three contexts, where the contextual index <2 ensures that the second negation is correctly attached to the main context, rather than the first negated context.

```
female.n.02                        % She
  NEGATION <1                      % is neither
rich.a.01 AttributeOf -1           % rich
  NEGATION <2                      % nor
famous.a.01 AttributeOf -2         % famous.
```

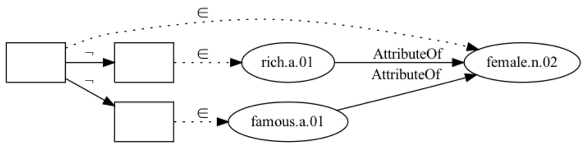

Figure 4: Graph for "She is neither rich nor famous".

Disjunction is represented in sequential meanings using the equivalence $(p_1 \vee p_2 \vee ... \vee p_n) \equiv \neg(\neg p_1 \wedge \neg p_2 \wedge ... \wedge \neg p_n)$. This representation has the advantage that no new separators are required, and that there is no limit to the number of disjuncts, as shown in Figure 5.

```
person.n.01 EQU speaker            % I
 NEGATION <1                       %
 NEGATION <1                       %
bake.v.02 Agent -1 Patient +1      % bake
bread.n.01                         % bread,
 NEGATION <2                       %
listen.v.01 Agent -3 Theme +1      % listen
music.n.01                         % to music,
 NEGATION <3                       % or
read.v.01 Agent -5 Source +1       % read comic
comic-book.n.01                    % books.
```

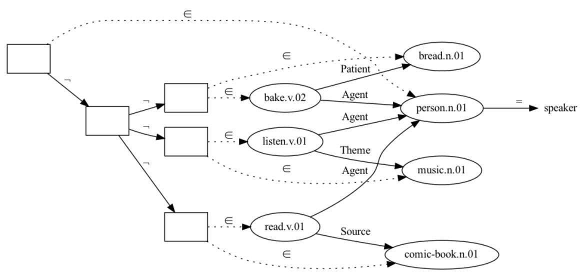

Figure 5: Graph exemplifying disjunction.

4.2 Universal Quantification

Universal quantification is encoded in sequential meanings by making use of the logical equivalence $\forall x(P(x) \rightarrow Q(x)) \equiv \neg\exists x(P(x) \wedge \neg Q(x))$. For instance, the sentence "Everyone smoked." is analysed as: it is not the case that there is a person that is not smoking. In sequence notation this would be `NEGATION <1 person.n.01 NEGATION <1 smoke.v.01 Agent -1`. The reason to use nested negation rather than a conditional is because this way there is no need to add two new separator relations—that would need to be coordinated as well, because unlike negation, a unary operator, implication and disjunction are binary operators—to the vocabulary.

Universal quantifiers in object position pose a challenge to meaning-text alignment in the sequence notation because of the scope they take over the transitive verb. An example is given in Figure 6, where the `CONJUNCTION` separator performs a merge of semantic information akin to merging of Discourse Representation Structures (Zeevat, 1991). This representational technique effectively gives the object wider scope, and is similar to presuppositional accommodation (Van der Sandt, 1992).

```
female.n.02                    % She
  NEGATION <1
  NEGATION <1
buy.v.01 Agent -1 Theme +1 % bought
  CONJUNCTION <2               % every
book.n.02                      % book.
```

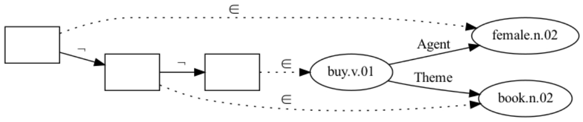

Figure 6: Graph displaying universal quantification.

4.3 Discourse Relations

Rhetorical relations are also encoded in sequential meanings by separators. Here I adopt the inventory of discourse relations as proposed in SDRT (Asher, 1993). Figure 7 shows an example where the rhetorical relation `EXPLANATION` connects two contexts. In sequential meanings discourse relations are always between single contexts. In SDRT, however, this is not necessarily the case because of the recursive nature of the segmented discourse representation structures. Yet sequential meanings can still capture rhetorical structure (Figure 8).

```
person.n.01                    % Someone
smile.v.01                     % smiles.
 EXPLANATION <1                %
male.n.01 EQU -2               % He
happy.a.01 Experiencer -1 % is happy.
```

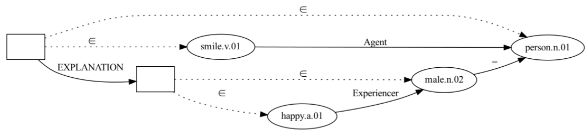

Figure 7: Graph visualisation for a short text.

As Asher and Lascarides (2003) have shown, anaphoric reference to compound discourse units is possible. The sequence notation would require additional machinery to catch this phenomenon. This could be something like a summation operation similar to handling split antecedents of plural pronouns in Discourse Representation Theory (Kamp and Reyle, 1993). This is probably also needed to cover the `CONTRAST` and `PARALLEL` discourse relations of SDRT.

```
person.n.01 Name "Max"          % Max
have.v.01   Pivot -1 Theme +2 % had
lovely.a.01 AttributeOf +1      % a lovely
evening.n.01                    % evening.
   ELABORATION <1
male.n.02   EQU -4              % He
have.v.01   Pivot -1 Theme +2 % had
great.a.01  AtttributeOf +1     % a great
meal.n.01                       % meal.
   ELABORATION <1
male.n.02   EQU -4              % He
eat.v.01  Agent -1 Patient +1 % ate
salmon.n.01                     % salmon.
   NARRATION <1
male.n.02     EQU -3              % He de-
devour.v.01 Agent -1 Patient +2 % voured
quantity.n.01 EQU +               % lots of
cheese.n.01   Quantity -1         % cheese.
   NARRATION <3
male.n.02   EU -11              % He
win.v.01    Agent -1 Theme +2 % won
dancing.n.01                    % a dancing
competition.n.01 Theme -1       % competition.
```

Figure 8: Sequential meaning for Asher and Lascarides (2003)'s celebrated example.

In SDRT, a `NARRATION` of a discourse unit U$''$ of U$'$, where U$'$ is an `ELABORATION` of U, would automatically invoke an `ELABORATION` relation of U$''$ to U, given the way SDRSs are constructed. This is not the case in sequence notation for the reason mentioned above. To capture such indirect discourse relations, some background inference rules would be needed.

5 Related Formalisms

The development of the sequence notation found inspiration from a wide spectrum of semantic representation systems, ranging from the classic semantic networks, Discourse Representation Theory, and Abstract Meaning Representations. In this section we will discuss how they are related: what do they have in common and how do they differ?

5.1 Semantic Networks

Semantic networks were introduced in the early 1970s to represent meaning (Simmons, 1973). Typically in these networks, a distinction is made between entity types and tokens (for instance, "a dog" would introduce two nodes in the network, one describing the set of dogs, and the other a particular member of that set, whereas in AMR just one node would be introduced in the semantic graph). The need for a richer network formalism was already recognised back then by Gary Hendrix, to cover linguistic phenomena such as universal quantification, hypothethical and imaginary situations. Hendrix (1975) introduced a method for partioning a semantic network into *spaces*. His use of spaces in semantic nets is strongly reminiscent to the way we employ contexts in the sequence notation, and is also similar to the Scoped Semantic Networks proposed by Power (1999).

A yet even more elaborative proposal was made around the same time by Len Schubert, who extended the expressive power of semantic nets with negation, disjunction and lambda expressions (Schubert, 1976). The resulting networks became rather cumbersome, and even Schubert himself remarks "I hasten to add that I am not urging universal adoption of this notation." These bunglesome additions might have been the reason why the extended networks never became mainstream in later years of AI and NLP, with the exception of the Conceptual Graphs proposed by Sowa (1984).

5.2 Discourse Representation Structures

One of the most elaborated semantic formalisms is probably Discourse Representation Theory (Kamp, 1984). Proposed in the early 1980s, it has seen many improvements, extensions, modifications, and reincarnations (Klein, 1987; Roberts, 1989; Zeevat, 1991; Van der Sandt, 1992; Kamp and Reyle, 1993; Asher, 1993; Reyle, 1993; Bos et al., 1994; Muskens, 1996; Van Eijck and Kamp, 1997; Frank and Kamp, 1997; Piwek, 2000; Kadmon, 2001; Beaver, 2002; Asher and Lascarides, 2003; Bos, 2003; Geurts and Maier, 2013; Kamp et al., 2011; Geurts et al., 2020). A wide range of linguistic phenomena are covered by DRT, among them conditionals, negation, modals, disjunction, presupposition, plurals, tense, aspect, and quantifier scope.

The contexts in sequence notation can be compared directly to the DRSs in Discourse Representation Theory. But sequential meanings discard representational redundancies: discourse referents are implicitly introduced by concepts. DRT has separate types of DRS conditions to model conditionals and disjunction, whereas the sequence notation only uses negation to cover these.

Standard DRT (Kamp and Reyle, 1993) follows a Davidsonian event semantics, whereas in this paper a neo-Davidsonian semantics is adopted that gives us the binary relations that enables simple graphical visualusation. Several features of DRT can be transferred to sequential meanings: blocking of anaphoric links by inaccessibility, merging of DRSs (Zeevat, 1991), and presuppositional accommodation (Van der Sandt, 1992).

5.3 Abstract Meaning Representations

The Abstract Meaning Representation formalism (Langkilde and Knight, 1998) represents meaning of natural language sentences as rooted, directed acyclic graphs. It took the clarity of the early semantic networks, and techniques introduced by AI researchers such as role inversion. Large semantically annotated corpora were developed based on AMR (Banarescu et al., 2013), encoded by using the PENMAN notation introduced by Kasper (1989). These corpora sparked a lot of interest in computational linguistics, and gave rise to many new approaches to semantic parsing and generating text from meaning representations.

Drawing a parallel with the semantic networks introduced in the 1970s, history repeats itself, when many scholars realized that AMR has incomplete inference capabilities for negation (and other logical devices such as universal quantification). Several proposals for extending AMR were published (Bos, 2016; Stabler, 2017; Pustejovsky et al., 2019; Bos, 2020; Lai et al., 2020; Stein and Donatelli, 2021). However, none of these proposals were widely adopted.

Several features of AMR are also present in the sequence notation: the binary relations that support attractive graphical visualisation, the use of role inversion, and being agnostic to grammar. But there are also notable differences: the sequence notation is closer to surface wording because there is not as much decomposition as in AMR. The sequence notation supports logical quantification and negation, which AMR lacks. And the sequence notation adopts WordNet (Fellbaum, 1998) and VerbNet (Kipper et al., 2008) to interpret the non-logical symbols, whereas AMR is based on PropBank (Palmer et al., 2005), but not all non-logical symbols are interpreted (verb-based symbols are, noun-based symbols aren't). This makes AMR partly specific to English, even though there have been AMR corpora constructed for other languages.

6 Discussion

6.1 No Overdose of Variables

Variables require some kind of naming convention, effectively an arbitrary way of blessing entities with a unique identifier. It is this resort to a naming system that makes variables unattractive for applications such as machine learning and human annotation. Usually, there are some informal conventions involved in naming variables, such as giving a variable an index that is increased by every new concept introduced in the meaning representation, or using the next letter of the alphabet. Alternatively, as is done in AMR, the variable name is based on the name of the concept that it names (Banarescu et al., 2013). This works well for short sentences, but as soon as longer texts need to be taken into account, the naming system gets cumbersome in practice.

The system of indices in sequential meaning does not suffer from these issues. Furthermore, the indices are *relative*—not absolute—capturing local "distances" between concepts. This enables a generalisation of catching argument structure, independent of sentence or text length. Even for short sentences meaning representations with indices yield better results in neural parsing than those resorting to variables (Van Noord et al., 2018). Hence, using indices rather than variables has the potential to offer advantages respect to human annotation and machine learning. And even though in this paper the sequence notation is used to encode DRS-based meanings, it can also be used to produce AMRs, as the AMR in (4) and its translation in sequence notation (5) show.

(4) `(w / want-01 :arg0 (b / boy)`
`:arg1 (g / go-01 :arg0 b))`

(5) `boy want-01 :arg0 -1 :arg1 +1`
`go-01 :arg0 -2`

The sequence notation results in shorter and compact meaning representations, because no space is wasted on brackets and variables.

6.2 Compositionality

I don't say much about *compositionality* from the perspective of the syntax-semantics interface. This is a deliberate choice. Compositionality—the study of how meanings of complex expressions are derived from meanings of their parts—is a fascinating problem in formal and computational semantics (Montague, 1973; Dowty et al., 1981) in which many attempts have been formulated and implemented, in particular within the Montagovian tradition (Bos et al., 1996; Bender et al., 2015).

The assumption in any implementation of compositionality is that there are atomic units of expressions carrying meaning that cannot be further decomposed. But what these atomic units are is unclear in general, and can range from simple inflectional markers to multi-word expressions. An extreme direction in this tradition, however never been explored in computational semantics, is Natural Semantic Metalanguage, defining a small set of semantic primes of which meanings can be composed (Wierzbicka, 1996).

A theory of syntax that supports semantic theory is therefore not sufficient to completely uncover compositionality, and moreover, makes the formalism language dependent. Arguably, large semantic annotation efforts have been shipwrecked exactly on the dependence of a computational grammar (Bos et al., 2017; Abzianidze et al., 2017).

Instead, sequential meanings do not require a lexical theory of meaning, such that one could, for instance, give an interpretation for a preposition, article or adverb in isolation. It assumes the expressions that it maps meanings to are complete utterances. Giving up strong compositionality is, from one perspective, certainly attractive, as it makes the formalism language-neutral and opens the door for multi-lingual computational semantics. Having said this, there are natural ways to break down sequantial meanings into smaller pieces (concepts, hooked/anchored concepts, contexts, and so on).

7 Conclusion

The meaning represention that I proposed has much in common with AMR (Banarescu et al., 2013) and DRS (Kamp and Reyle, 1993). But there are notable differences. Like AMR but unlike DRS, sequential meanings are agnostic to any method or theory of syntax. Like AMR, but unlike DRS, sequential meanings can be viewed as simple graphs. Like DRS, but unlike AMR, there is an explicit way of assigning scope to logical operators. Unlike AMR and DRS, there are no variables in sequential meanings.

The quote "make everything as simple as possible, but not simpler", often attributed to Albert Einstein, is perhaps what summarises the sequence notation. It provides a language that I think cannot be simpler than it is, at the same time making it possible to describe complex meaning representations (including negation, disjunction, quantification, and discourse structure) with a formal interpretation. As there are only binary relations, and the binary relations can be inverted, a sequential meaning can be visualised as a directed acyclic graph, resulting in graphs that are simpler than those previously proposed for Discourse Representation Theory (Basile and Bos, 2013; Abzianidze et al., 2020). The sequence notation therefore offers a visual aid for verification of meanings.

I think the sequence notation is also a convenient way of annotating text with meaning representations. The notation is simple, no logical variables are needed, meanings can be manually entered and corrected in a standard text editor. The sequence notation supports the alignment between meaning representations and corresponding linguistic realisation in an approximate manner, where at least the order of the concepts corresponds with the order as they are introduced in the text by nouns, verbs, adjectives and adverbs. Yet I understand that not everyone is convinced that annotation with the sequence notation would be simpler than say AMR or DRS. This paper has no evidence for this claim and is solely based on personal experience. Additionally, I have observed that researchers with logic background have become accustomed to the use of variables, making it considerably challenging for them to abandon the familiarity of such notation.

Currently the sequence meaning notation has been put in practice in the Parallel Meaning Bank (Abzianidze et al., 2017). In future work the idea is to take advantage of the sequence notation and annotate larger (multi-sentence) multi-lingual documents with meaning representations that include rhetorical structure.

Acknowledgments

Previous versions of this work have been presented on various occasions, including the online ILFC seminar on 18 January 2022, the Workshop in honor of the 80th birthday of Hans Kamp (which was held from 23–25 June 2022 in Stuttgart), the Workshop on Ten Years of Babelnet and Multilingual Neurosymbolic Natural Language Understanding (Rome, 4–5 July 2022), the kick-off workshop of the MWE-SEMPRE project in Düsseldorf on 8 September 2022, and the ALiAS-'23 workshop in Amsterdam on 6 January 2023. I would like to thank the participants of all these events for the valuable feedback they provided. In particular, I am grateful to Anette Frank for pointing out the connection between LFG and the relative indices in sequential meanings (see § 3.3); to Alex Lascarides and Nicholas Asher for demonstrating the differences between SDRT and the way discourse is structured in sequentual meanings (see § 4.3); and to Nathan Schneider for showing me the differences between AMR and sequential meanings with respect to meaning-text alignment (see § 2.5).

Additionally, the comments of the three anonymous IWCS-2023 reviewers helped me to further improve this article. Reviewer #1 with the encouraging words "an interesting proposal that might go someplace" suggested to look at approaches proposed in description logic, as it reminded them of sequential meanings, and yes thanks for the suggestion, I will explore this area of research in the future. And yes I don't present a model theory directly but instead do so by translation to DRT. And no, I am not bothered by the lack of support for generalised quantifiers (see Appendix 5), I am more worried about *kinds* and subsective adjectives. Reviewer #2 "a notation which has some attractive properties" made me rewrite several parts of the paper to make the presentation clearer. And indeed large language models might make all these efforts obsolete but who knows, perhaps explainability might become fashionable. Reviewer #3 was "unconvinced that annotating in sequence notation would be easier, faster and more reliable" and yep you might be right about this and perhaps it is just a matter of taste of what one is used to. However, the expressive capacity goes well beyond the state-of-the-art in large-scale semantic annotation efforts.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and 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*, pages 242–247, Valencia, Spain.

Lasha Abzianidze, Johan Bos, and 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*, pages 23–32, Online. Association for Computational Linguistics.

Rafael Anchiêta and Thiago Pardo. 2020. Semantically inspired AMR alignment for the Portuguese language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1595–1600, Online. Association for Computational Linguistics.

Hajnal Andréka, István Németi, and Johan Van Benthem. 1998. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274.

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

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Studies in natural language processing. Cambridge University Press.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria.

Valerio Basile and Johan Bos. 2013. Aligning formal meaning representations with surface strings for wide-coverage text generation. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 1–9, Sofia, Bulgaria.

David I. Beaver. 2002. Presupposition Projection in DRT: A Critical Assesment. In *The Construction of Meaning*, pages 23–43. Stanford University.

Emily M. Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 239–249, London, UK. Association for Computational Linguistics.

Austin Blodgett and Nathan Schneider. 2021. Probabilistic, structure-aware algorithms for improved variety, accuracy, and coverage of AMR alignments. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3310–3321, Online. Association for Computational Linguistics.

Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia, Bulgaria. Association for Computational Linguistics.

Claire Bonial, William Corvey, Martha Palmer, Volva V. Petukhova, and Harry Bunt. 2011. A hierarchical unification of lirics and verbnet semantic roles. In *2011 IEEE Fifth International Conference on Semantic Computing*, pages 483–489.

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

Johan Bos. 2016. Expressive power of abstract meaning representations. *Computational Linguistics*, 42(3):527–535.

Johan Bos. 2020. Separating argument structure from logical structure in AMR. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 13–20, Barcelona Spain (online). Association for Computational Linguistics.

Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. 2017. The Groningen Meaning Bank. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, volume 2, pages 463–496. Springer.

Johan Bos, Björn Gambäck, Christian Lieske, Yoshiki Mori, Manfred Pinkal, and Karsten Worm. 1996. Compositional Semantics in Verbmobil. In *The 16th International Conference on Computational Linguistics*, pages 131–136, Copenhagen, Denmark.

Johan Bos, Elsbeth Mastenbroek, Scott McGlashan, Sebastian Millies, and Manfred Pinkal. 1994. A Compositional DRS-Based Formalism for NLP-Applications. In *International Workshop on Computational Semantics*. University of Tilburg, The Netherlands.

Nicolaas Govert de Bruijn. 1972. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. In *Indagationes Mathematicae (Proceedings)*, volume 75, pages 381–392. Elsevier.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City. Association for Computational Linguistics.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Gennaro Chierchia. 1992. Anaphora and dynamic binding. *Linguistics & Philosophy*, 15:111–183.

Ann Copestake, Dan Flickinger, Ivan Sag, and Carl Pollard. 2005. Minimal recursion semantics: An introduction. *Journal of Research on Language and Computation*, 3(2–3):281–332.

David R. Dowty, Robert E. Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics*. Studies in Linguistics and Philosophy. D. Reidel Publishing Company.

Christiane Fellbaum, editor. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press.

Anette Frank and Hans Kamp. 1997. On context dependence in modal constructions. In *Proceedings of the 7th Semantics and Linguistic Theory Conference*, pages 151–168.

Bart Geurts, David I. Beaver, and Emar Maier. 2020. Discourse Representation Theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, spring 2020 edition. Metaphysics Research Lab, Stanford University.

Bart Geurts and Emar Maier. 2013. Layered discourse representation theory. In Alessandro Capone, Franco Lo Piparo, and Marco Carapezza, editors, *Perspectives on Linguistic Pragmatics*, Perspectives in Pragmatics, Philosophy & Psychology, pages 311–327. Springer.

Jeroen Groenendijk and Martin Stokhof. 1990. Dynamic Montague Grammar. In *Papers from the Second Symposium on Logic and Language*, pages 3–48.

Irene Heim. 1982. *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, University of Massachusetts.

Gary G. Hendrix. 1975. Expanding the utility of semantic networks through partitioning. In *Proceedings of IJCAI*, pages 115–121.

Nirit Kadmon. 2001. *Formal Pragmatics*. Blackwell.

Hans Kamp. 1984. A Theory of Truth and Semantic Representation. In Jeroen Groenendijk, Theo M.V. Janssen, and Martin Stokhof, editors, *Truth, Interpretation and Information*, pages 1–41. FORIS, Dordrecht – Holland/Cinnaminson – U.S.A.

Hans Kamp, Josef van Genabith, and Uwe Reyle. 2011. Discourse Representation Theory. In Dov M. Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, volume 15, pages 125–394. Elsevier, MIT.

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

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press, Cambridge, MA. Reprinted in Mary Dalrymple, Ronald M. Kaplan, John Maxwell, and Annie Zaenen, eds., *Formal Issues in Lexical-Functional Grammar*, 29–130. Stanford: Center for the Study of Language and Information. 1995.

Robert T. Kasper. 1989. A flexible interface for linking applications to penman's sentence generator. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 153–158, Philadelphia.

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

Ewan Klein. 1987. VP Ellipsis in DR Theory. In Jeroen Groenendijk et al., editors, *Studies in Discourse Representation Theory and the Theory of Generalised Quantifiers*, volume 8, pages 161–187. FLORIS, Dordrecht.

Kenneth Lai, Lucia Donatelli, and James Pustejovsky. 2020. A continuation semantics for abstract meaning representation. In *The Second International Workshop on Designing Meaning Representations (DMR 2020)*, Barcelona, Spain.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, pages 704–710.

Yijia Liu, Wanxiang Che, Bo Zheng, Bing Qin, and Ting Liu. 2018. An AMR aligner tuned by transition-based parser. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2422–2430, Brussels, Belgium. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Abelardo Carlos Martínez Lorenzo, Marco Maru, and Roberto Navigli. 2022. Fully-Semantic Parsing and Generation: the BabelNet Meaning Representation. In *Proceedings of the 60th Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), pages 1727–1741, Dublin, Ireland. Association for Computational Linguistics.

Richard Montague. 1973. The proper treatment of quantification in ordinary English. In J. Hintikka, J. Moravcsik, and P. Suppes, editors, *Approaches to Natural Language*, pages 221–242. Reidel, Dordrecht.

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

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Paul Piwek. 2000. A formal semantics for generating and editing plurals. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

Wessel Poelman, Rik van Noord, and Johan Bos. 2022. Transparent semantic parsing with Universal Dependencies using graph transformations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4186–4192, Gyeongju, Republic of Korea.

Nima Pourdamghani, Yang Gao, Ulf Hermjakob, and Kevin Knight. 2014. Aligning English strings with Abstract Meaning Representation graphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 425–429, Doha, Qatar. Association for Computational Linguistics.

R. Power. 1999. Controlling logical scope in text generation. In *Proceedings of the 7th European Workshop on Natural Language Generation*, Toulouse, France.

James Pustejovsky, Nianwen Xue, and Kenneth Lai. 2019. Modeling quantification and scope in abstract meaning representations. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 28–33, Florence, Italy. Association for Computational Linguistics.

Uwe Reyle. 1993. Dealing with Ambiguities by Underspecification: Construction, Representation and Deduction. *Journal of Semantics*, 10:123–179.

Craige Roberts. 1989. Modal subordination and pronominal anaphora in discourse. *Linguistics and Philosophy*, 12(6):683–721.

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

Lenhart K. Schubert. 2015. Semantic representation. In *AAAI Conference on Artificial Intelligence*.

L.K. Schubert. 1976. Extending the expressive power of semantic networks. *Artificial Intelligence*, 7:163–198.

R.F. Simmons. 1973. Semantic networks: Their computation and use for understanding english sentences. In R. Schank and K. Colby, editors, *Computer Models of Thought and Language*, pages 63–113. W.H. Freeman & Co.

Linfeng Song and Daniel Gildea. 2019. SemBleu: A robust metric for AMR parsing evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4547–4552, Florence, Italy. Association for Computational Linguistics.

John F. Sowa. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.

John F. Sowa. 1995. Syntax, semantics, and pragmatics of contexts. In *ICCS*, pages 1–15.

Ed Stabler. 2017. Reforming AMR. In *Formal Grammar 2017. Lecture Notes in Computer Science*, volume 10686, pages 72–87. Springer.

Katharina Stein and Lucia Donatelli. 2021. Representing implicit positive meaning of negated statements in AMR. In *Proceedings of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 23–35, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Richmond Thomason. 1974. *Formal Philosophy. Selected Papers of Richard Montague*. Yale University Press, New Haven.

Jan Van Eijck and Hans Kamp. 1997. Representing discourse in context. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 179–237. Elsevier.

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

Anna Wierzbicka. 1996. *Semantics: Primes and Universals*. Oxford Universiy Press.

Hendrik Willem Zeevat. 1991. *Aspects of Discourse Semantics and Unification Grammar*. Ph.D. thesis, University of Amsterdam.

A Translation to DRS

Here I sketch a translation from sequential meaning notation to DRT's Discourse Representation Structure (DRS). Although the sequential meaning system presented here bears strong similarities with Discourse Representation Theory (Kamp and Reyle, 1993), it is significantly different from it:

1. Events are represented in a neo-Davidsonian way whereas in DRT a Davidsonian way is assumed (i.e., without adopting an inventory of thematic roles);

2. All non-logical synbols are interpreted using WordNet as supporting ontology, whereas in DRT these remain uninterpreted;

3. A single `NEGATION` relation is used to capture negation, disjunction and conditionals, whereas DRT has special complex conditions for them in the DRS language;

4. There is no syntactic check for free and bound variables, whereas the geometrical structure of DRS immediately shows accessibility of referents.

5. There is no support for generalised quantifiers unlike DRT that has duplex conditions to accommodate them. If one were to incorporate generalised quantifiers into sequential meanings one would likely resort to adding new separators to the inventory. For instance, for "A guitar has six strings", we would arrive at something like `GENERALISATION <1 guitar.n.01 MOST < have.v.02 Pivot -1 Theme +1 string.n.03 Quantity 6`. These two separators would need to be coordinated though: one cannot exist without the other.

6. There is no different in representation of singular and plural noun phrases—the model theory behind sequential meanings allows entities in the domain to range over plural noun phrases as well.

Despite these differences, the similarities with DRT become immediately clear when one sketches the translation from sequential meanings to DRS (Kamp and Reyle, 1993). Only *closed* sequential meanings can be translated to DRS, so each index needs to have an antecedent context, each connector needs to link to an existing context, and in the resulting DRS no free variables should occur.

The easiest way to explain the translation from sequential meanings to DRS is to take the corresponding rooted directed acyclic graphs as starting point. The root node is always a context. The translation to DRS starts with this context, initiated as an empty DRS. Recall that a DRS consists of a domain (a set of discourse referent) and a set of (basic and complex) DRS-conditions. All entities with concept C that are members of this context are added to the domain of the DRS with a fresh discourse referent. The concept is translated as unary predicate applied to this discourse referent and added to the conditions of the DRS. All hooks and anchors of this concept are added to the conditions as binary predicates, where the internal argument is the same as the discourse referent.

Once this is completed for all members of a context, the process is recursively repeated for contexts that are connected to the current context. There are two main cases here: (1) `NEGATION` adds a complex unary condition $\neg B$ to the DRS, where B will be the result of the translation of the context associated to the negation; and (2) `CONJUNCTION` does not start a new DRS, but instead continues adding information to the current DRS. The other separators build up a structure as in SDRT (Asher, 1993). To illustrate the procedure, I show in Figure 9 the DRSs that are the result of translating two sequential meanings presented earlier in this paper.

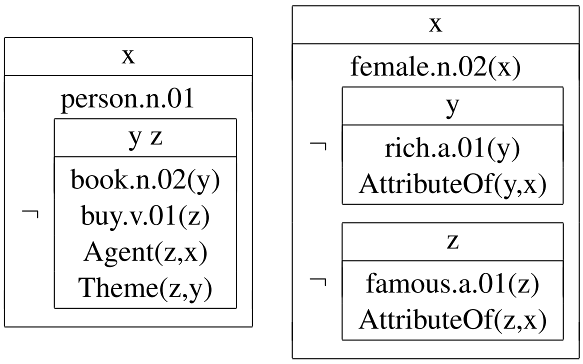

Figure 9: DRS equivalents of the sequential meanings shown in Figure 3 and Figure 4.

B Semantic Annotation Example

Figure 10 shows an elaborated example in sequence notation aligned with its textual input. Figure 11 visualises the corresponding graph.

```
male.n.02 Name "Pierre Vinken"                       % Pierre Vinken,
  APPOSITION <1
quantity.n.01 EQU 61                                 % 61
measure.n.02 Quantity -1 Unit "year"                 % years
old.a.01 AttributeOf -3 Value -2                     % old,
  CONJUNCTION <2
time.n.08 TSU now                                    % will
join.v.01 Theme -5 CoTheme +1 Role +3 Time -1        % join
board.n.01                                           % the board
nonexecutive.a.01                                    % as nonexecutive
director.n.02 Attribute -1                           % director
time.n.08 MonthOfYear 11 DayOfMonth 29 TOV -5        % Nov. 29.
  ELABORATION <1
male.n.02 Title "Mister" Name "Vinken" EQU -10       % Mr. Vinken
be.v.03 Theme -1 Co-Theme +2 Time +1                 % is
time.n.08 EQU now
chairman.n.01 Of +1                                  % chairman of
company.n.01 Name "Elsevier N.V."                    % Elsevier N.V.,
  APPOSITION <1
country.n.02 Name "The Netherlands"                  % the Dutch
publishing_group.n.01 Source -1 EQU -2               % publishing group.
```

Figure 10: Meaning in sequence notation aligned for the first text of the Wall Street Journal corpus (Marcus et al., 1993). The text is here included as comments on each line following a percentage sign, and is not part of the actual meaning representation. Three different comparison operators are used here: EQU (equality), TSU (temporally succeeds), and TOV (temporally overlaps). The resulting graph is shown in Figure 11.

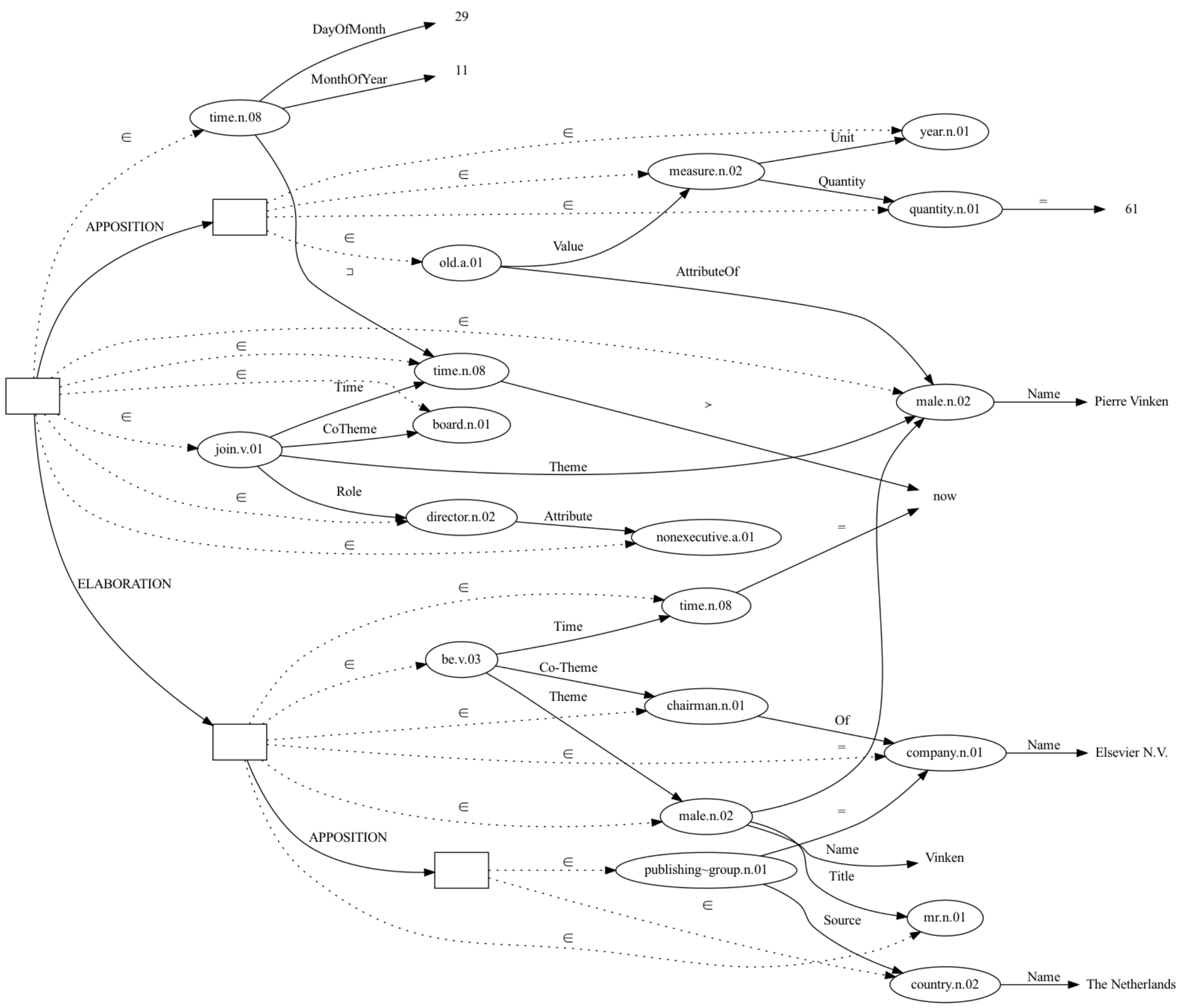

Figure 11: Graph visualisation of the WSJ corpus text "Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29. Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group."