Dynamic Semantics for a Controlled Natural Language

Rolf Schwitter
Centre for Language Technology
Macquarie University
Sydney NSW 2109 Australia
schwitt@ics.mq.edu.au

Abstract

In this paper I present a dynamic approach for constructing an unambiguous semantic representation for a text written in a controlled natural language called PENG. The semantic representation is built up incrementally – in left-to-right order – while the user of the PENG system writes the text. For each word form that the user types, the language processor creates a partial logical structure that is analysed in the context of a given information state and then either updates that information state or delays this process if necessary. After each processing step, the language processor reports what kind of syntactic categories can follow the current input. These predictive grammatical hints guide the writing process of the user and enforce the rules of the controlled language. The result is an unambiguous text in controlled natural language that has first-order equivalent properties and that can be further processed by a computer.

1. Introduction

In contrast to traditional semantics which sees semantic theory as primarily concerned with reference and truth, dynamic semantics adopts the view that the meaning of a sentence does not lie in its truth conditions but rather in its "information change potential" – its capacity for modifying given information states into new ones [2] [5].

At first glance, this view seems ideally suited to process and interpret a text incrementally by a computer as each word form comes along in left-to-right order while the text is written. This processing strategy sounds intuitive and works well for very simple sentences such as

Rona buys a house.

where the predicates of the logical representation can easily be aligned from left to right:

```
∃ X,E,Y
(named(X,rona) ∧ buy(E,X,Y) ∧ house(Y))
```

However, this view is deceptive, since the order in which states are updated in dynamic semantics is determined by the semantic structure of the underlying representation language and not by the left-to-right order of the linguistic elements at the sentence level. For example, the sentence

Rona buys a house in every city.

that contains the prepositional modifier *in every city* does not allow for the same naive processing strategy, since the predicate for the verbal event and the predicate derived from the preposition occur on the right hand side of the implication in the final semantic representation:

```
... ∀ C
(city(C) → ∃ E (buy(E,X,Y) ∧ loc(E,C)))
```

Technically, it is possible to thread information states through grammar rules but sometimes the updating of these information states needs to be delayed, since subsequent constituents with scope-bearing elements might enforce a specific form of embedding in the semantic representation (as the example above shows).

Moreover, there exist constructions in full natural language that do not allow for direct threading from left-to-right (see also [6]). For example, the use of any kind of ordered threading will tend to fail for Bach-Peters sentences [10], such as

The pilot who shot at it hit the Mig that chased him.

The paradoxical fact here is that this sentence is acceptable in the cross-referential interpretation (i.e. *it = the Mig that chased him* and *him = the pilot who shot at it*), although it conflicts with a universal constraint on interpretative circularity in natural language.

In this paper, I will not deal with full natural language, but with PENG, a controlled natural language [8], that has been restricted with respect to its grammar and its lexicon, and that can be unambiguously translated into a flattened notation for Discourse Representation Theory (DRT) [5]. Because this controlled language has well-defined formal properties, it allows for ordered threading. A number of computer implementations for DRT have been suggested over the years (for an overview see [1]). I will use and extend a declarative reformulation of DRT [4] that takes the idea of dynamic semantics seriously and that does not use any higher-order constructs.

The reminder of this paper is organised in the following way: In Section 2, I will give a brief overview of the PENG system and speak about the look-ahead feedback mechanism that guides the writing process. In Section 3, I will discuss a couple of characteristics and restrictions of the PENG grammar. In Section 4, I will introduce a flattened notation for DRT and motivate the advantages of this notation. In Section 5, I will show in detail how discourse representation structures can be constructed dynamically for a number of constituents in a systematic way. In Section 6, I will discuss briefly how DRSs for questions can be built using the same approach. Finally, in Section 7, I will summarize the advantages of the presented approach and give some hints for practical applications of the controlled natural language PENG.

2. The PENG System

The language processor of the PENG system consists of an unification-based phrase structure grammar and a top-down chart parser [8]. For each word form that the user enters into the text editor of the system, the parser produces a partial discourse representation structure and feedback information for the user. This feedback information consists of a set of look-ahead categories and additional information that contributes to the construction of an explicit paraphrase for the processed text [9].

The look-ahead categories inform the user how the input string can be continued and enforce thereby the rules of the controlled natural language. For example, if the user is in the process of typing the noun phrase *every city*, then the editor first displays the look-ahead categories `adjective | noun` for words that can follow the determiner *every* and then the look-ahead categories `verb | aux_verb:[does] | rel_pronoun | preposition:[of] | full_stop` for words that can follow the noun *city*. This means that the user of the PENG system does not need to learn the controlled natural language but can simply follow these predictive hints that are generated by the language processor while the text is written.

The language processor of the PENG system communicates via a server with reasoning services (a theorem prover and a model builder) so that the text can be checked for its consistency and informativeness after each new sentence. If the reasoning services detect a logical problem in the emerging text, then the user will be alerted [9].

3. The Grammar of PENG

The grammar of PENG consists of simple and complex sentences. The grammar is processed by a top-down chart parser that builds discourse representation structures (DRSs). The following is a simplified excerpt of the grammar rules. In reality, all nodes contain additional arguments with syntactic, semantic and pragmatic information that is not displayed here. As we will see later, some of these arguments will be used to dynamically construct DRSs during the parsing process.

```
1.   s0 ---> n3, v3.
2.   s2 ---> s1, s1.
3.   s1 ---> cj, s0.
4.   n3 ---> det, n2, {anaphora_res}.
5.   n2 ---> n1, rc.
6.   n1 ---> a2, n1.
7.   n1 ---> n0, pp
8.   n1 ---> n0.
9.   v3 ---> v3, cj, v3.
10.  v3 ---> neg, v2.
11.  v3 ---> v2.
12.  v2 ---> v1, adv.
13.  v2 ---> v1, pp.
14.  v1 ---> v0.
15.  v1 ---> v0, n3.
16.  v1 ---> v0, n3, p2.
17.  v1 ---> v0, a2.
```

A couple of things are worth mentioning here: The grammar of PENG processes simple *(1)* and complex *(2 + 3)* sentences. Anaphoric expressions are resolved on the fly in PENG, this is done after a complete noun phrase has been processed *(4)*. Relative pronouns always modify the immediately preceding nominal constituent *(5)*. Prenominal adjectives can be coordinated in PENG *(6)*. To exclude ambiguity, only the preposition *of* can be used for prepositional phrases in postnominal position *(7)*. The scope of a verb phrase negation extends to the end of a simple sentence *(10)*. Another restriction of PENG is that prepositional or adverbial modifiers always modify the verbal eventuality *(12 + 13)*. Note that all restrictions of the language are mediated by a paraphrase in controlled language that explains the user the interpretation of a sentence by the machine.

4. A Flattened Notation for DRT

In our implementation, a DRS is represented as a term of the form `drs(U,Con)` consisting of a list `(U)` of discourse referents $[I_1, I_2, \ldots I_n]$ denoting entities and a list `(Con)` of conditions $[C_1, C_2, \ldots C_n]$ that describe properties or relations that these discourse referents must satisfy. DRSs can occur as constituents of larger (complex) DRSs. Complex DRS conditions are those involving implication, disjunction, and negation.

In our flattened notation for DRS conditions concepts such as `city(I)` are treated – in a first approximation – as typed individuals `obj([city],I)`. They do not introduce predicate symbols anymore and can therefore be referred to by simple terms (see also [3]). This notation has three main advantages: First, quantification over complex terms that would require higher-order quantification can now be conducted via first-order quantification. Second, the flattened notation simplifies the formalization of additional logical axioms to express various forms of linguistic and non-linguistic knowledge. Third, this notation increases the efficiency of the inference processes.

4.1. Nouns

Lexical entries for nouns contain two DRS conditions in PENG. The first DRS condition consists of three arguments. The first one of these arguments is a discourse referent (`C`) denoting a concept, the second one is a term (e.g. `[city]`) naming that concept and the third one is a discourse referent (`I`) denoting an object that falls under that concept. The second DRS condition specifies the structure of this object that can be either `atomic`, `group`, or `mass` reflecting a lattice-theoretic structure of the domain [7]:

```
[obj(C,[city],I),struc(I,atomic)]
[obj(C,[singing],I),struc(I,atomic)]
[obj(C,[men],I),struc(I,group)]
[obj(C,[coffee],I),struc(I,mass)]
```

The concept-denoting discourse referent `C` is mainly used for resolving anaphoric references between nominalized expressions and the preceding verbs (e.g. *Rona sings. The singing is loud*). The noun *coffee* has two entries, one for `mass` and one for `atomic`, these are selected depending on the syntactic structures under investigation, for example *a spoon of coffee* (= `mass`) or *a coffee* (=`atomic`).

4.2. Verbs

DRS conditions for verbs introduce event- or state-denoting discourse referents (`E` or `S`) in PENG:

```
[pred(E,[live],I),evtl(E,event)]
[pred(S,[have],I1,I2),evtl(S,state)]
[pred(E,[give],I1,I2,I3),evtl(E,event)]
```

Treating eventualities for verbs as discourse referents accommodates the fact that they can be modified by attributes, nominalized and even be referred to by anaphoric expressions.

4.3. Adjectives

Lexical entries for adjectives contain a single DRS condition. The form of this condition is depending on whether the adjective is relational or non-relational:

```
[prop(S,[next,to],I1,I2)]
[prop(S,[happy],I)]
```

Adjectives in copulative structures (e.g. *is happy*) pick out states but this is not the case for adjectives in prenominal position (e.g. *the happy man*) which only ascribe properties. In the latter case the discourse referent *S* will not appear in the representation.

4.4. Prepositions

DRS conditions for prepositions used in modifying prepositional phrases look similar to DRS conditions for adjectives but they have an additional condition that specifies their thematic role:

```
[prop(T,[in],I1,I2),role(T,time)]
[prop(L,[in],I1,I2),role(L,location)]
```

The thematic role is calculated from the type of the preposition and the type of the head noun to distinguish different kinds of semantic roles such as `time` for *in the morning* or `location` for *in the house*.

4.5. Adverbs

DRS conditions for most adverbs have a similar form as those for prepositions, since adverbs modify verbal eventualities in PENG:

```
[prop(M,quick,I),role(M,manner)]
```

In contrast to these modifiers, there exist modifiers that have a quantifying potential such as *daily* or *weekly*. These quantifying modifiers are treated in the same way as their corresponding noun phrases *every day* and *every week*.

5. Constructing DRSs

To process a sequence of sentences S_1, S_2, ... S_n in DRT [5], the *original* DRS construction algorithm starts with the syntactic analysis of the first sentence S_1 and then transforms it with the help of DRS construction rules into a DRS K_1 which serves as the context for processing the second sentence S_2. This approach is sequential and does not emphazise the dynamic aspect of transforming information states (DRSs) while syntactic constituents are parsed.

The dynamic properties of DRT can be reclaimed by a declarative reformulation of the DRT in an unification-based model of grammar [4]. Following this approach, each syntactic constituent can be related to an incoming and an outgoing DRS. The outgoing DRS is constructed from the incoming one plus information derived from that constituent. Thus, the meaning of any constituent is the change in the DRS, when it is processed. But as I will argue, the embedding of certain DRS conditions into the preceding DRS needs sometimes be delayed, especially if we want to deal with optional constituents such as prepositional and adverbial modifiers in an uniform way.

5.1. Simple Sentences

As explained, every constituent of a sentence has an incoming and an outgoing DRS. This relation can be modeled by using a difference list of the form *DrsIn-DrsOut* in Prolog. For example, if the noun *city* were handled by a single rule, that rule (simplified here) would be:

```
n0([cat:cn,
    arg:[ind:[C,I]|R],
    drs:[drs(U1,C1)|D]-
        [drs([C,I|U1],[C3,C2|C1])|D]])
    --->
    { lexicon([lex:Noun],
              [cat:cn,
               arg:[ind:[C,I]|R],
               con:[C3,C2]]) },
    Noun.
```

In this rule the two DRS conditions `obj(C,city,I)` and `struc(I,atomic)` that are available in the lexicon for the noun *city* were unified with the variables `C3` and `C2` and added in the outgoing DRS list to the conditions `C1` of the incoming DRS. The discourse referents `C` and `I` are added in the outgoing DRS to the discourse universe `U1` of the incoming DRS. The variable `D` stands for a superordinated DRS that might contain accessible discourse referents.

A simplified rule for the verb is shown below. This rule uses a similar mechanism as the rule for nouns:

```
v0([cat:tv,
    arg:A1,arg:A2,
    evtl:E,
    drs:[drs(U1,C1)|D]-
        [drs([E|U1],[C3,C2|C1])|D]])
    --->
    { lexicon([lex:Verb],
              [cat:tv,arg:A1,arg:A2,
               evtl:E,con:[C3,C2]]) },
    Verb.
```

The variables `C3` and `C2` unify with the DRS conditions of the lexical entry for the verb and the result is added to the outgoing DRS. The variable for the eventuality `E` is added to the discourse universe `U1` of the outgoing DRS.

The core of the DRS construction algorithm is located in the rules for the determiners. Determiners are the most important constituents for establishing the logical structure of a sentence, despite their minor syntactic role. Semantically, a determiner has two arguments: a restrictor and a scope. The restrictor consists of the remaining conditions within a noun phrase (= `n3-det`) and the scope is made up of the conditions outside the noun phrase.

Here is an example for the determiner `no`:

```
d0([cat:det,agr:G,spec:no,
    drs:D1-[drs(U1,[drs(U2,C2) ->
            drs([],[˜drs(U3,C3)])|C1]|D3],
    res:[drs([],[])|D1]-D2,
    sco:[drs([],[])|D2]-
        [drs(U3,C3),drs(U2,C2),
         drs(U1,C1)|D3]])
    --->
    { lexicon([lex:Determiner],
              [cat:det,agr:G,spec:no]) },
    Determiner.
```

The restrictor `res` pushes an empty DRS `drs([],[])` onto the incoming DRS list and makes this the active information state where all discourse referents and conditions for the remaining noun phrase are collected. The scope `sco` takes the restrictor's outgoing DRS and pushes a new empty DRS onto it and makes this again a new active information state where all discourse referents and conditions outside the noun phrase are collected. The DRSs for the restrictor and the scope are then embedded into a complex condition (consisting of an implication and a negation) representing the meaning of the negative determiner:

```
[drs(U1,[drs(U2,C2) ->
   drs([],[˜drs(U3,C3)])|C1]|D3]
```

Below is the phrase structure rule for a simple sentence, that shows that the scope of the noun phrase and eventually of its determiner is the semantics of the verb phrase:

```
s0([...,drs:D,...])
    --->
    n3([...,arg:A,drs:D,sco:S,...]),
    v3([...,arg:A,drs:S,...]).
```

5.2. VP Modification

The processing of optional constituents that modify a verb phrase such as in

> *Rona buys a house in every city.*
>
> *Rona works in no city and lives in no village.*
>
> *Rona sells a house to Aaron in a city.*

needs an additional threading mechanism for those DRS conditions that are derived from the verb, since we do not know – in our incremental approach – whether an optional constituent will follow the verb or not while it is processed.

```
v2([...,arg:A,drs:D1-D2,...])
    --->
    v1([...,arg:A,evtl:E,drs:D1-D2,
        sco:S2-S3,sco:hold:S1,...]),
    pp([...,evtl:E,drs:S2-S3,sco:S1,...]).
```

With the help of the additional argument `sco:hold`, the DRS conditions `S1` for the verb are threaded into the scope `sco` of the prepositional phrase so that the correct DRS for the sentence can be constructed. For example, the sentence

Rona buys a house in every city.

with the universally quantified prepositional modifier will result in the following DRS:

```
[drs([A,B,C],
     [drs([D,E],
          [obj(D,city,E),
           struc(E,atomic)]) ->
           drs([F,G],
               [prop(F,in,G,E),
                role(F,location),
                pred(G,buy,C,B),
                evtl(G,event)]),
     obj(A,house,B),
     struc(B,atomic),
     named(['Rona'],C),
     struc(C,atomic)])]
```

6. Questions

Texts written in PENG, can be queried in PENG. For example, the following complex question with a coordinated verb phrase

Where does Rona work $_{[GAP]}$ *and live* $_{[GAP]}$*?*

is processed incrementally as a declarative PENG sentence and finally translated into the following DRS:

```
[whq
 drs([A,B,C,D,E],
      pred(A,[live],E),evtl(A,event),
      prop(B,[where],A),role(B,location),
      pred(C,[work],E),evtl(C,event),
      prop(D,[where],C),role(D,location),
      named(['Rona'],E),struc(E,atomic))]
```

The important thing here is that threading allows us to deal with questions (and coordination) in a natural way using a filler-gap mechanism for unbounded dependencies.

7. Conclusions

In this paper, I introduced a flattened notation for DRT and showed how discourse representation structures can be dynamically constructed for a controlled natural language in left-to-right order while the text is written. The writing process is guided by look-ahead categories that show the user of the PENG system after each word form that has been entered how to continue the text. The result is an unambiguous text in controlled natural language that has the same formal properties as the corresponding representation in DRT. The flattened representation simplifies the formalization and the processing of additional (linguistic and non-linguistic) background axioms. The controlled natural language PENG can be used in situations where precise texts (e.g. software specifications, legal documents, axioms for formal ontologies) need to be written. In a next step, I will try to find a subset of PENG that can be translated automatically into a variant of description logic.

Acknowledgments

The research reported in this paper was conducted while the author was on OSP working at the University of Zürich in Switzerland. I gratefully acknowledge financial support from Macquarie University and from the Australian Research Council (grant DP0449928).

References

[1] A. W. Black, Some Different Approaches to DRT, *Draft*, Centre for Cognitive Science, University of Edinburgh, June 27, 1997.

[2] J. Groenendijk, M. Stokhof, Dynamic Predicate Logic, *Linguistics and Philosophy*, Vol. 14, pp. 39-100, 1991.

[3] J. R. Hobbs, Discourse and Inference, *Draft*, USC Information Science Institute, Marina del Rey, California, November 3, 2003.

[4] M. Johnson, E. Klein, Discourse, anaphora and parsing, in *Proceedings of the 11th International Conference on Computational Linguistics*, Coling 86, Bonn, pp. 669-675, 1986.

[5] H. Kamp, U. Reyle, *From Discourse to Logic*, Dordrecht: Kluwer, 1993.

[6] D. Milward, R. Cooper, Incremental Interpretation: Applications, Theory, and Relationship to Dynamic Semantics, In *Proceedings of the 15th International Conference on Computational Linguistics, Coling 94*, Kyoto, Japan, pp. 748-754, 1994.

[7] U. Schwertel, Plural Semantics for Natural Language Understanding, A Computational Proof-Theoretic Approach, *PhD Thesis*, University of Zurich, 2003.

[8] R. Schwitter, English as a Formal Specification Language, *Proceedings of the Thirteenth International Workshop on Database and Expert Systems Applications (DEXA 2002)*, Aix-en-Provence, pp. 228-232, 2002.

[9] R. Schwitter, A. Ljungberg, D. Hood, ECOLE: A Look-ahead Editor for a Controlled Language, *Proceedings of EAMT-CLAW03*, Controlled Language Translation, May 15-17, Dublin City University, pp. 141-150, 2003.

[10] H. Zeevat, Static Semantics, In J. van Benthem (ed.), *Partial and Dynamic Semantics I*, DYANA Report 2.1.A, Centre for Cognitive Science, University of Edinburgh, 1990.