

2012

Three-place predicates in English: Towards a unification-based computationally adequate approach to Role and Reference Grammar

Judith Gottschalk

BBDO Services GmbH, gottschalk.judith@googlemail.com

Follow this and additional works at: <https://arrow.tudublin.ie/itbj>



Part of the [Anthropological Linguistics and Sociolinguistics Commons](#)

Recommended Citation

Gottschalk, Judith (2012) "Three-place predicates in English: Towards a unification-based computationally adequate approach to Role and Reference Grammar," *The ITB Journal*: Vol. 13: Iss. 1, Article 6.

doi:10.21427/D7RR0M

Available at: <https://arrow.tudublin.ie/itbj/vol13/iss1/6>

This Article is brought to you for free and open access by the Journals Published Through Arrow at ARROW@TU Dublin. It has been accepted for inclusion in The ITB Journal by an authorized administrator of ARROW@TU Dublin. For more information, please contact yvonne.desmond@tudublin.ie, arrow.admin@tudublin.ie, brian.widdis@tudublin.ie.



This work is licensed under a [Creative Commons Attribution-Noncommercial-Share Alike 3.0 License](#)

Three-place predicates in English: Towards a unification-based computationally adequate approach to Role and Reference Grammar

Judith Gottschalk
BBDO Services GmbH
gottschalk.judith@googlemail.com

Abstract²⁵

This paper is concerned with a computational linguistics analysis of Role and Reference Grammar [RRG] (cf. Van Valin and LaPolla 1997; Van Valin (2005) and introduces research work in progress aiming to analyze the computability of RRG. The concept of computational adequacy is introduced as an important external principle from a philosophy of science perspective to sharpen the scientific principles of the area of functional computational linguistics. In addition, a pseudo-code-based meta-language is developed in order to semi-formalize the linking algorithm from semantics to syntax. This paper will show that RRG in its current fashion is not executable on an abstract machine model - called Random Access Machine - and is therefore not computationally adequate. It is highlighted that the semantics to syntax linking algorithm as proposed in Van Valin (2005) is in fact too coarsely grained to account for the variable undergoer linking in English three-place predicates. Also, the concept of intelligent software agents is introduced in order to account for the functional linguistic approach used in RRG. It will be shown that it is possible to account for variable undergoer linking in three-place predicates using constructional schemas as developed in Nolan (2011).

Based on the development of typed feature structures of thematic relations it is possible to show that semantic macroroles as developed in Van Valin (2005) are epiphenomenal. They are an unnecessary concept set on top of thematic relations, which is in conflict with the principle of economy as discussed in Van Valin and LaPolla (1997). It is shown that thematic relations are stored in inheritance networks in the mental lexicon and that they interact with constructional schemas for transfer verbs as they are developed in this paper. The concept of discourse representation structures is also of crucial importance in this paper. It will be shown that variable undergoer linking in English is based on information structure considerations. In order to develop a computationally adequate version of RRG, a revised version of the semantics to syntax linking algorithm is developed.

1. Introduction

From a computational linguistics point of view, Role and Reference Grammar [RRG] (cf. Van Valin and LaPolla 1997; Van Valin 2005) is a rather informally described linguistic theory. For the purpose of computational processing of RRG in diverse computational linguistic applications it would be necessary to specify the formal meaning of the theoretical framework of RRG. If this was the case it would be possible for RRG to be interpreted by an ordinary computer program (cf. Richter 2000: 5).

The present study introduces research work in progress which aims to formalize RRG and to analyze the computational complexity of RRG. It asks whether RRG is computationally tractable and thereby analyzes RRG's generative capacity. As a

²⁵ I would like to thank Elke Diedrichsen for interesting discussions about the analysis of three-place predicates within RRG, Kim Hülsewede for proof reading this paper and Lars Inderelst for some interesting philosophical discussions about the concept of computability. I also would like to thank my family for their never-ending support. Without it I would not be able to do my research. Especially I would like to thank Hagen Langer for several pleasant and interesting discussions about formalization and computability of linguistic theories which gave the main impetus for the development of the concept of computational adequacy. Also Brian Nolan receives my special thanks for his support with respect to the development of this approach to a new computationally adequate version of RRG, his encouragement with respect to developing this new account to RRG, many great discussions about unification-based approaches to RRG and his help in ways too numerous to mention.

starting point I will introduce a semi-formalization of the RRG linking algorithm from semantics to syntax in terms of a pseudo-code meta-language with which it will be possible to pinpoint problems occurring in the semantics to syntax linking algorithm. These give evidence that the linking algorithm in RRG should mainly be understood as a coarsely grained guidance principle rather than a formal linguistic theory which is applicable for computational linguistic applications.

In fact, RRG has already been used in some computational linguistic applications. Guest (2008) developed an RRG-based parser, which uses extensions of a chart parser to parse languages with different degrees of word order. In this approach, parsing is executed via syntactic templates as used in RRG instead of rules. Winther-Nielsen (2009) and Wilson (2009) describe a software tool called Role-Lexicon Module which is database driven and can be used for parsing of Biblical Hebrew. This system uses an EMDROS database and contains active chart parser which generates the layered structure of the clause. Nolan and Salem (2009) and Salem (2009) on the other hand have developed a machine translation program called *UniArab*, which uses a rule-based lexical framework to process Arabic based on RRG. In Murtagh (2011), a linguistically motivated Irish Sign Language conversational agent is introduced which uses RRG as the linguistic engine in the development of a conversational agent for sign languages. Also, within FunGramKB aspects of RRG, especially the semantic representation of RRG, are used to create a knowledge base for natural language processing. All these implementations focus on the software developed, but they do not focus on the specific computational problems caused by the architecture of RRG.

As Langer points out in personal communication, the principle with respect to the application of RRG within computational devices as described above is mainly the same as with programming languages: It is possible to write programs which terminate, however it is also possible to write programs which do not terminate. The previous applications of RRG in software implementations have shown that applicable RRG-fragments do exist. Otherwise RRG would not have any justification. However, the important question with respect to the computability of RRG is to test whether RRG allows the formulation of fragments for which it is not possible to develop an algorithm answering the fundamental question in a finite number of steps. This in turn leads to the question whether a string XY belongs to a fragment of a defined language or not. In fact programming languages are tools. However, they are not theories and therefore it is reasonable to give them maximal expressiveness. With respect to grammar formalisms such as RRG the situation is rather different: these formalisms ideally should be as expressive as necessary for the computation of natural languages. In this regard questions like ‘Is the natural language XY context-free?’ or ‘Are all natural languages maximally mildly-context-sensitive?’ are of high importance. The long term aim of my research will focus on this question based on a formalized version of RRG. However, the crucial point is: Grammar formalisms can only contribute relevant insights to this question if they are not able to generate a specific string of signs. The question which should be answered in the course of my research, for which this paper is the initial starting point, is whether RRG is Turing complete, hence, whether it is computationally adequate or not.

These questions also result in a revision of the levels of linguistic adequacy as mentioned in Van Valin and LaPolla (1997) since, as pointed out above, a linguistic theory can only contribute to the analysis of natural languages if it has computational

adequacy and thus reasonable explanatory power. In Van Valin and LaPolla (1997), the general goals of a linguistic theory are described as well as a number of levels of adequacy a linguistic theory should meet. As Van Valin and LaPolla (1997: 2) point out, the majority of linguists would agree that the first goal of a linguistic theory is to describe linguistic phenomena, while the second is to explain these phenomena and the third is to understand the cognitive basis of language. As pointed out in Van Valin and LaPolla (1997: 3), describing linguistic phenomena is one of the central goals in linguistics. For many linguists it is the primary goal of linguistics. This goal may include the description of individual languages, describing what is common to all languages - seeking for language universals - or, in how far languages differ from each other, which is the endeavor of language typology (cf. Van Valin and LaPolla 1997: 3). With respect to explanatory linguistics theories, Van Valin and LaPolla (1997: 3) explain the following:

The main impetus to the postulation of explanatory theories of linguistic phenomena came from Chomsky's early work in generative grammar. Chomsky (1957) argued that the proper role of linguistic theory is to provide criteria for selecting the most explanatory grammar from among a group of competing grammars.

The important question is: what should a linguistic theory explain? Van Valin and LaPolla (1997: 3) list a number of candidates for what a linguistic theory should explain. These candidates are given in (1):

- (1) Candidates for what a linguistic theory should explain
- a. how speakers use language in different social situations;
 - b. why human languages have the structure they do;
 - c. what is common to all human languages;
 - d. why human languages vary structurally the way they do;
 - e. how human languages change over time;
 - f. how speakers produce and understand language in real time;
 - g. the nature of native speakers' knowledge of their language;
 - h. how children learn language.

(Van Valin and LaPolla 1997: 4)

As pointed out in Van Valin and LaPolla (1997: 15), RRG is directly concerned with all these goals except (a) and (e). The last three topics in (1) explicitly deal with psychological questions about language. Many linguists, following Chomsky, maintain that cognitive issues are in fact the most important issues for a linguistic theory to deal with (cf. Van Valin and LaPolla 1997: 4). However, as pointed out by Van Valin and LaPolla (1997: 4), not all linguistic theories agree on which questions regarding psychology are the most important. Van Valin and LaPolla (1997: 4) list a number of three major facets of the psychology of language:

- (2) *Processing*: Which cognitive processes are involved when human beings produce and understand language on line in real time? How specialized to language are these processes?
Knowledge: What constitutes knowledge of language? How is it organized? How is it represented? How is it employed in language processing? How does knowledge of language relate to knowledge in other cognitive domains?
Acquisition: How do human beings come to have knowledge of language? What is the nature of the acquisition process? Is coming to know language similar to or different from acquiring knowledge in other cognitive domains? Does it involve knowledge from other cognitive domains?

(Van Valin and LaPolla 1997: 4)

In this paper I will focus on language processing and knowledge of language from a computational linguistics perspective. Here the basic idea is that computational linguistics is used as a means and a test bed to testify assumptions made in theoretical linguistics focusing on the facets of the psychology of language. Van Valin and LaPolla (1997: 5) point out that philosophers of science typically divide theories into two basic types: The first type is inductive while the second type is deductive. In inductive theories, generalizations are derived from the observation of many examples of the phenomenon under investigation. Van Valin and LaPolla (1997: 5) explain this idea as follows:

If one for example, examined a large number of birds of various species and concluded ‘all birds have wings’, this would be an inductive generalization describing as property of birds. The generalizations of structural linguistics are inductive in nature, as are the language universal proposed in the work of Greenberg (e.g. Greenberg 1966). The relationship between data and theory with respect to inductive theories is **data hypothesis**.

(Van Valin and LaPolla 1997: 5)

Deductive theories however work in a different way. Here, hypotheses are formulated and then tested against data in order to ascertain their validity. In this case, the hypotheses typically grow out of observations of phenomena but not directly as in inductive theories. In a deductive theory, hypotheses are formulated which are intended to explain the observed facts and to predict what has been observed before. This means deductive theories are explanatory theories, and the relationship between data and theory is **hypothesis data** (Van Valin and LaPolla 1997: 5). In this context Van Valin and LaPolla (1997: 5) note that often one set of hypotheses is proposed to account for a given observation or set of observations. However, the important question is how it is possible to chose the best one among a number of alternatives. In fact there are two types of criteria, empirical and theory-internal criteria. The empirical criteria ask whether a theory is in accordance with the known facts or experimental results. If this is not the case, it should be eliminated from consideration. If however two or more theories are empirically adequate, theory internal criteria come into play (cf. Van Valin and LaPolla 1997: 5). These theory internal criteria are given in (3):

(3) *Theory-internal explanatory criteria*

a. *Economy (Occam’s Razor)*: Is it the simplest theory?

b. *Motivation*: Are crucial explanatory constructs independently motivated or are they *ad hoc*?

c. *Predictiveness*: Do the hypotheses predict phenomena beyond those for which they were formulated?

(Van Valin and LaPolla 1997: 5)

It is not always easy to come up with explicit criteria for simplicity in a particular theoretical domain. However, the intuition behind the criteria in (3a) is straightforward: all other criteria being equal, the simplest theory is to be the preferred one (cf. Van Valin and LaPolla 1997: 5). Van Valin and LaPolla (1997: 5) describe the other two criteria as follows:

The second criterion, motivation, refers to the extent to which the hypotheses follow in a natural way from the preexisting theory and the extent to which the constructs invoked in the explanation are also required elsewhere in the theory. An account in which the explanatory constructs have no other function beyond dealing with the problem at hand is less highly valued than one in which they play a role in the explanation of other phenomena; in this case the constructs are said to be *independently motivated*, because they are required by the theory for phenomena other than the problem at hand.

(Van Valin and LaPolla 1997: 6)

The basic idea with respect to the third criterion is that hypotheses which make empirically testable predictions about other observed phenomena or phenomena which have not yet been observed are more highly valued than those which do not (cf. Van Valin and LaPolla 1997: 6). With respect to theory-internal criteria Van Valin and LaPolla (1997: 7) explain the following:

[...] the theory-internal criteria [...] play a central role in theoretical argumentation in linguistics. By referring to these criteria as ‘theory-internal’, we do not mean to imply that they are internal to any specific theory; rather, they are assumed by all linguistic theories. It is also possible to appeal to external phenomena in explanation, and this is a point of controversy among linguistic theories. An example of an external explanation would be an account of some syntactic pattern which makes crucial reference to semantics (i.e. the meaning of the pattern) and/or pragmatics (i.e. the context in which it occurs or the communicative function which it serves). A semantic explanation for a syntactic pattern would be an external explanation, on the standard (but not universally held) assumption that syntax and semantics are distinct from each other. In this instance we are dealing with external but language-internal explanations. It is also logically possible to appeal to language-external facts or principles in an explanation. For example, one could argue that some syntactic pattern holds in human languages because of the nature of human cognition or perception; such an appeal to non-linguistic aspects of cognition or perception would be an external explanation as well. (Van Valin and LaPolla 1997: 7)

As a functional linguistic theory RRG accepts external criteria in explanation which are given in table 1 below:

Table 1: Types of explanatory criteria

<i>Domain to be explained</i>	<i>Theory-internal criteria</i>	<i>External criteria</i>	
		<i>Language-internal</i>	<i>Language-external</i>
SYNTAX	Economy Motivation Predictiveness	Phonology Semantics Pragmatics Processing	Reasoning Categorization Perception ...

(Van Valin and LaPolla 1997: 7)

The fact that RRG as a functionalist linguistic theory accepts external criteria as given in table 1 is of great importance with respect to a computational linguistics approach to RRG in the vein of what is accounted for as functional computational linguistics by Nolan (2011). A functional computational linguistics account accepts both language-internal and language-external explanatory criteria. This fact will become obvious if the concept of a learning software agent is employed in order to account for a semi-formalization of RRG.

With respect to the level of adequacy in a linguistic theory, Van Valin and LaPolla (1997: 7) explain the following:

One of the most important arguments Chomsky made in *Syntactic structures* (1957), the monograph which introduced generative grammar to the field, was that linguistics should be considered a deductive, rather than an inductive, enterprise. Bloomfield had stated explicitly in his 1933 book, *Language*, that ‘the only valid linguistic generalizations are inductive generalizations’ (21) and one of Chomsky’s main goals was to make linguistic theory explanatory and not solely descriptive. (Van Valin and LaPolla 1997: 7)

In Chomsky (1965), levels of adequacy which grammar must meet are proposed. These levels of adequacy are listed in Van Valin and LaPolla (1997: 7f) and given in (4)

(4)

- (a) **observational adequacy** which means the grammar correctly predicts which sentences are well formed in a language and therefore grammatical and which are not;
- (b) **descriptive adequacy** which means the grammar is observationally adequate and assigns structural descriptions to the sentences in the language that captures native speaker intuitions about the structure and meaning of the sentences;
- (c) **explanatory adequacy** which means the grammar is both descriptively adequate and is part of a theory which provides an account of 'how these facts arise in the mind of the speaker-hearer' as pointed out in Chomsky (1994: 386)

(cf. Van Valin 1997: 7f)

With respect to these three levels of adequacy proposed in Chomsky (1965), Van Valin and LaPolla (1997: 8) note the following:

For Chomsky, 'the fundamental empirical problem of linguistics is to explain how a person can acquire knowledge of language' (1977: 81). The last two levels of adequacy are explicitly cognitive in nature, as they refer to native speaker intuitions and to language acquisition.

(Van Valin and LaPolla 1997: 8)

If one takes the criteria introduced in the paragraphs above into consideration, observational adequacy is the criterion of empirical adequacy, which applies to the sentences of a language, while descriptive adequacy is also based on empirical accuracy. In this case it applies to native speaker intuitions about sentences (cf. Van Valin and LaPolla 1997: 8). The theory-internal criteria given in table 1 come into play with respect to explanatory adequacy. This is an important point of disagreement among linguistic theories with respect to the application of external criteria in linguistic theories, since functional linguistic theories like RRG accept external criteria for explanatory adequacy, while formalist theories in the Chomskian tradition only accept theory-internal criteria (cf. Van Valin and LaPolla 1997: 8).

However, additional types of adequacy have been proposed, too (cf. Van Valin and LaPolla 1997: 8). These are described by Van Valin and LaPolla (1997: 8) as follows:

Dik (1978, 1991) proposes a broad notion of **psychological adequacy**, which states that a theory should be 'compatible with the results of psycholinguistic research on the acquisition, processing, production, interpretation and memorization of linguistic expressions' (1991: 248). This subsumes the criterion put forth in Kaplan and Bresnan (1982) that theories linguistic structure should be directly relatable to testable theories of language production and comprehension. Dik also proposes two additional types of adequacy: **pragmatic adequacy**, i.e. 'the theory and the language descriptions based on it should be interpretable within a wider pragmatic theory of verbal communication' (1991: 247), and **typological adequacy**, i.e. the theory should 'formulate such rules and principles as can be applied to any type of language without 'forcing', i.e. without adapting the language described to the theory already developed' (248).

(Van Valin and LaPolla 1997: 8)

RRG is a functional linguistic theory which is both monostratal and lexicalist (cf. Van Valin 1991, cf. Van Valin and LaPolla 1997). It was developed to answer the following questions, as described in Van Valin (2005: 1):

[...] (1) what would a linguistic theory look like if it were based on the analysis of languages with diverse structures such as Lakhota, Tagalog and Dyirbal, rather than on the analysis of English?,

and (2) how can the interaction of syntax, semantics and pragmatics in different grammatical systems be captured and explained?

(Van Valin 2005: 1)

This means the main focus of RRG is to be typologically adequate, as pointed out in Van Valin and LaPolla (1997). However in this paper I will show that a functional linguistic theory which also accepts the external criteria described in table 1 should also meet a level of adequacy which I will refer to as *computational adequacy*.

By computational adequacy I mean the fact that a theory should refer to formal systems which are computationally tractable in order to support the level of explanatory adequacy within a Chomskyan framework. This is because a theory which is not tractable and has a Turing complete generative power, as pointed out in Carpenter (1991) with respect to HPSG, is less explanatory than a theory which is tractable and has less generative power in the sense of being not Turing complete (Langer personal communication). As can be inferred from Van Valin (2006), RRG is mainly concerned with psycholinguistic adequacy rather than computational adequacy. As pointed out by Langer (personal communication), psycholinguistic adequacy, with which RRG is concerned, can be described as follows: Humans usually make specific linguistic errors. A psycholinguistically adequate model should account for the same mistakes. This means a psycholinguistically adequate model of language should have the same limitations, such as humans being able to accept only one single linguistic input. This means one is not able to read three books simultaneously or listen to five radio shows. Also, garden-path effects occur. This means psycholinguistic adequacy regards adequacy on the performance level rather than on the competence level linguistic theories are usually concerned with. Computational adequacy on the other hand means to process language with low storage demands without mistakes. This means in a Jeopardy competition the psycholinguistically adequate model would try to be as close as possible to the human competitors while a computational adequate model wants to win (Langer personal communication).

One crucial assumption with respect to computational adequacy is that it is based on the Church-Turing-thesis which assumes that everything which is computable on a machine is intuitively computable. The Church-Turing thesis is bidirectional. This means since natural language is intuitively computable, based on the Church-Turing-thesis it should also be computable on a machine (cf. Blass and Gorevitch 2001). This way a computer can be used as a test bed for linguistic theories in order to show that a linguistic theory actually works. Based on my definition of computational adequacy and on the Church-Turing thesis, my research aim is to test whether RRG, which until now has only been rather informally described is computable on a machine and by this intuitively computable. This way computational adequacy is in fact more basic than psycholinguistic adequacy and is necessary if a linguistic theory should be psycholinguistically adequate. This means computational adequacy has a different quality from psycholinguistic adequacy, since psycholinguistic adequacy operates on the performance level and presupposes psycholinguistic adequacy. Kaplan and Bresnan (1982) explain that a linguistic theory which should be a psycholinguistically real model of competence should be tied to a testable theory of performance in the sense of being psycholinguistically or computationally implementable. However, the concept of computational adequacy proposed in this paper differs strikingly from Kaplan and Bresnan's (1982) idea, since it operates on the competence level as laid out above. The aim of this paper is to support the level of computational adequacy in more detail by the

attempt to semi-formalize RRG in terms of a while-program which can be executed on a Random Access Machine.

In RRG, a single syntactic description which is semantically motivated is used to describe the constituent structure of natural languages. RRG does not assume any abstract underlying levels of syntactic representation as used in Government and Binding theory or Relational Grammar (cf. Gottschalk 2011: 31; cf. Van Valin 1991: 154; Van Valin 2005: 1). RRG employs a semantic representation based on Aktionsarten as developed by Vendler (1969). The formal representation of the RRG semantic structure is based on Dowty (1979). For the correspondence of the syntactic and semantic representation RRG employs a linking algorithm which is strictly procedural and uses a set of instructions to link the two representations to each other. This procedural algorithm is called the linking algorithm.

Van Valin (2005: 129) describes the basic idea of this algorithm as follows:

A distinctive feature of the RRG linking algorithm is that it is bidirectional; this is, it links the semantic representation to the semantic representation. Viewed in terms of a language processing model, the semantics-to-syntax linking is an aspect of the production process. In the comprehension process, the parser would take the input and produce a structured syntactic representation of adpositions and other grammatically relevant elements in the sentence. It is then the task of the grammar to map this structure into a semantic representation, as the first step in interpreting it, and this is where the syntax-to-semantics linking algorithm is required. The same syntactic representations are used in both the linking algorithms.

The use of macroroles will be of interest in this paper since I will show that they are, in fact, epiphenomenal and are mainly used as attributes in typed feature structures. Following Langer (personal communication) with respect to the computational adequacy of RRG the important questions to be asked are: (1) If RRG was more formal would it be tractable and computable? (2) Is RRG implementable? (3) Can RRG be used for the most important computational processes like parsing, generation, translation, learning etc.? (4) How much reservoir demands does RRG have and which runtime would RRG have (theoretical worst case, practical average case)? (5) What does the anytime capacity of an implementation of RRG look like? (6) Is a formalized and implemented RRG-linking algorithm robust? Questions (2) and (3) have been answered in the computational implementations of RRG mentioned above. In my research project I aim to find answers to questions (1), (4), (5) and (6).

This paper seeks to semi-formalize the semantics-to-syntax linking algorithm in terms of a pseudo-code meta-language in order to be able to base a formal analysis on these findings. Since RRG models cognitive processes, it will be necessary to somehow model these concepts in a computational framework. I will use the notion of an intelligent software agent for this purpose. A software agent can be executed on an abstract machine model, in this case a random access machine [RAM]. In this study it will be shown that a semi-formalized version of RRG is not executable if no revision of the linking algorithm takes place. In computer science it is a common technique to test whether an algorithm is computable in the sense of being executable by the attempt to execute them on an abstract machine model. Therefore the aim of this paper is twofold: The linking algorithm in the semantics to syntax linking will be semi-formalized in pseudo-code and the linking algorithm will be revised. Special attention will be put on the assignment of actors and undergoers in the place predicate constructions in English, since here specific problems in the linking process from semantics to syntax occur

which need to be solved in order to be able to have an algorithm which can be executed on a RAM.

The paper is organized as follows: In section 2 I will introduce the RAM as abstract machine model and describe the pseudo-code based meta-language which will be used to semi-formalize the linking from semantics to syntax. Also in this section I will introduce the idea of intelligent software agents as a reasonable tool to implement cognitive processes in software implementations. In Section 3 I will introduce Van Valin's (2007) approach to the analysis of three-place predicates and will show some weak points in Van Valin's solution which is mainly based on macroroles as proposed in Van Valin (2005). I will also discuss Van Valin's arguments against the assumption of a third macrorole. In section 4 Van Valin's (2007) analysis of three-place predicates is introduced and is critically discussed. Section 5 is concerned with pinpointing the shortcomings in the linking from semantics to syntax. This will be achieved by the development of a pseudo-code-based semi-formalization of the linking algorithm from semantics to syntax with special focus on variable undergoer linking in three-place predicate constructions. It will be shown that the linking algorithm in its current state cannot account for these constructions from a computational linguistics perspective. In section 6 I will introduce the concept of constructional schemas as developed in Nolan (2011). Section 7 is concerned with a revision of the theory of the mental lexicon as proposed in Gottschalk (2010a, 2011b). Typed feature structures for lexical semantic relations are developed which are stored in inheritance networks in order to store the argument structure of verbs. Also, an inheritance network of thematic relations will be developed and it will be shown that based in this unification based approach the assumption of semantic macroroles as developed in Van Valin (2005) are epiphenomenal. Also lexical entries for verbs of transfer with which this paper is concerned are introduced.

The paper closes with section 8 in which an information structure-based analysis of three-place predicates in English based on discourse representation structures follows which shows that information structure governs variable undergoer linking in English. Also in this section a constructional schema for the give-relation in English is developed and the linking algorithm from semantics to syntax is revised to show what a computationally adequate version of RRG should look like. This is also done in terms of the pseudo-code meta-language developed in section 2.

2. Random Access Machines and the pseudo-code meta-language

A RAM is an abstract mathematical machine model (cf. Güting and Dieker 2004: 6) which has two memories, a program memory and a data memory. The program memory is able to hold a sequence of commands which are stored in a small set of instructions. The data memory on the other hand is an infinite sequence of memory cells or registers of the kind r_0, r_1, r_2, \dots , which are able to collect a natural number. In this case register r_0 is an accumulator. This means that it represents an implicitly used operator used for arithmetic operations, comparisons, etc. (cf. Güting and Dieker 2004: 7). There is also a program counter. This program counter points to the first command at the beginning of an operation and later it points to the command which is executed presently (cf. Güting and Dieker 2004: 7).

The set of instructions in a RAM contains both load- and store instructions of the accumulator, for arithmetic operations, comparisons and branch instructions. For all these commands the effects on the data memory and the command counter is precisely defined. From this perspective the set of instructions used in a RAM is a minimal excerpt of a machine- or assembler language which is used on a real computer (cf. Güting and Dieker 2007: 7). Formally, a RAM is equivalent to a Turing machine, but it is possible to execute while-programs on a RAM, while a Turing machine would need a special Turing-program. The advantage of using a RAM in this case is that it will be possible to formulate a while-program which is very similar to modern computer programs.

To analyze the problems occurring with the procedural approach to the linking algorithm as used in RRG which will be executed on a RAM and which are crucial for both the formalization and the implementation of RRG I will use the following pseudo-code-based meta-language to semi-formalize the semantics to syntax linking in three-place predicate construction in the RRG-linking algorithm to pinpoint these problems and to revise this part of RRG in later sections. This meta-language is given in (5)

- (5) a. **if** <condition> **then** <instruction> **end if**
 b. **if** <condition> **then** <instruction> **else** <instruction> **end if**
 c. **for** <loop-control> **do** <instruction> **end for**
 d. **while** <loop-control> **do** <instruction> **end while**
 (cf. Güting and Dieker 2004: 4)

The meta-language given in (5) uses some standard-notions from computer science and programming languages to make it possible to express the linking algorithm used in RRG in a more formal way. The basic idea is that the linking algorithms found in RRG are formulated in this meta-language which is the kind of language which can be understood by the RAM. In (5a) an **if-then** construction is used, which is common across many programming languages such as Java, Python, C and C++. In this example, **if** is a keyword which signals that a condition, which is usually boolean like 'x > 17', should be evaluated. If this condition is *true* then the instruction after the keyword **then** is executed. The construct in (5b) is an **if-then-else** construct which mainly works the same way as the **if** construct in (5a) with one exception: While the construct in (5a) is only executed if the condition after the keyword **if** is *true* and otherwise the program continues after the keyword **end if**, the construct in (5b) is always executed. Here the main idea is that the condition is checked. If this condition is true the instruction is executed. If, however, the condition is *false*, the instruction after the keyword **else** is executed. After this execution, the algorithm continues with the instruction which might follow the keyword **end if**. The construct in (5c) is a **for**-loop. In this construct, after the keyword **for** a loop-control like 'x = 1 to 5' follows. A loop which has a loop-control like 'x = 1 to 5' is mainly characterized by counting. In this case the variable x will take the values from 1 to 10 and iterates within the loop until the upper bond in this case 10 is reached. For each iteration an instruction is executed, e.g. 'x + 1'. This would mean that based on the range of the loop-condition, 1 is added to the variable x which can have a range from 1 to 10. The loop terminated after 10 is reached and then a program continues after the keywords **end for**. In (5d) also a loop is described. In this case it is a **while**-loop which works as follows: after the keyword **while** a loop-control follows which might have the following form 'x < 10' and it means that while the loop-control 'x < 10' is *true* the loop is executed iteratively and an instruction like 'x + 2'

follows until 'x < 10' is false. Also, this meta-language uses assignments of the form 'Actor = x' which means that x is assigned Actor. Equations are formulated with the use of = and the Boolean data types true and false are used to evaluate the Boolean content of an expression.

The aim of the employment of this meta-language is to semi-formalize the two RRG-linking algorithms in order to be able to pinpoint some problems within the procedural approach for the linking as described in Van Valin (2005), Van Valin and LaPolla (1997) and other works in RRG. However, since RRG is a linguistic framework which models cognitive processes, the attempt to use a classical approach to formalize and later to implement it would fail, since in a standard model no account is made for the fact that cognitive processes beyond the scope of an ordinary computer program are at work. Therefore it is necessary to introduce the concept of an intelligent software agent at this point of the discussion in order to account for the cognitive processes RRG attempts to model. The idea of an agent in the broad sense refers to anything that can be viewed as perceiving its environment through sensors and to act on the environment through actuators (cf. Russell and Norvig 2003: 32). Russell and Norvig explain the idea of an agent as follows:

A human agent has eyes, ears, and other organs for sensors and hand, legs, mouth, and other body parts for actuators. A robotic agent might have cameras and infrared range finders for sensors and various motors for actuators. A software agent receives keystrokes, file contents, and network packets as sensory inputs and acts on the environment by displaying on the screen, writing files, and sending network packets. We will make the general assumption that every agent can perceive its own actions (but not always the effects).

The term percept is used to refer to the perceptual inputs an agent has at any given instant while the term percept sequence is used to refer to the complete history of everything the agent has ever perceived (cf. Russell and Norvig 2003: 32). This means the choice an agent has at any given instant can depend on the entire percept sequence which it has observed to date (cf. Russell and Norvig 2003: 32). In mathematical terms, the agent's behavior is described by the agent function mapping of any given percept sequence to an action. The idea behind this function is described by tabulating the agent function that constitutes any given agent. Such a table is in fact an external characterization of the agent. Internally however the agent function is implemented by the use of an agent program (cf. Russell and Norvig 2003: 33). As Russell and Norvig (2003: 33) note it is important to keep these ideas separated since the agent function is an abstract mathematical description while the agent program is a concrete implementation running on the agent architecture. In the long term for the formalization of RRG in terms of its complexity and generative power it will be an important task to specify the mathematical function of the agent. However, for the purpose of this paper, which seeks to be a semi-formalized starting point, in order to have a basis for the later formalization of RRG it is sufficient to use a pseudo-code agent program. The basic idea of such an agent program is that it is executed on some sort of computing device, in the case of this paper on a RAM, which will be called *architecture* (cf. Russell and Norvig 2003: 44). What constitutes the agent is given in (6)

(6) agent = architecture + program

(Russell and Norvig 2003: 44)

The agent program I will use in this paper has the following skeleton: It takes the current percepts as input from the sensors and returns an action to the actuators. In this context it is important to take into account that the agent program takes the current percept as input while the agent function takes the entire percept history as input (cf. Russell and Norvig 2003: 44). In fact the agent program just takes the current percept as input. This is because nothing more is available from the environment. In cases where the agent's actions depend on the percept sequence, the agent will have to remember the percepts (cf. Russell and Norvig 2003: 45).

With respect to the table-driven approach for software agents Russell and Norvig (2003: 45) note the following:

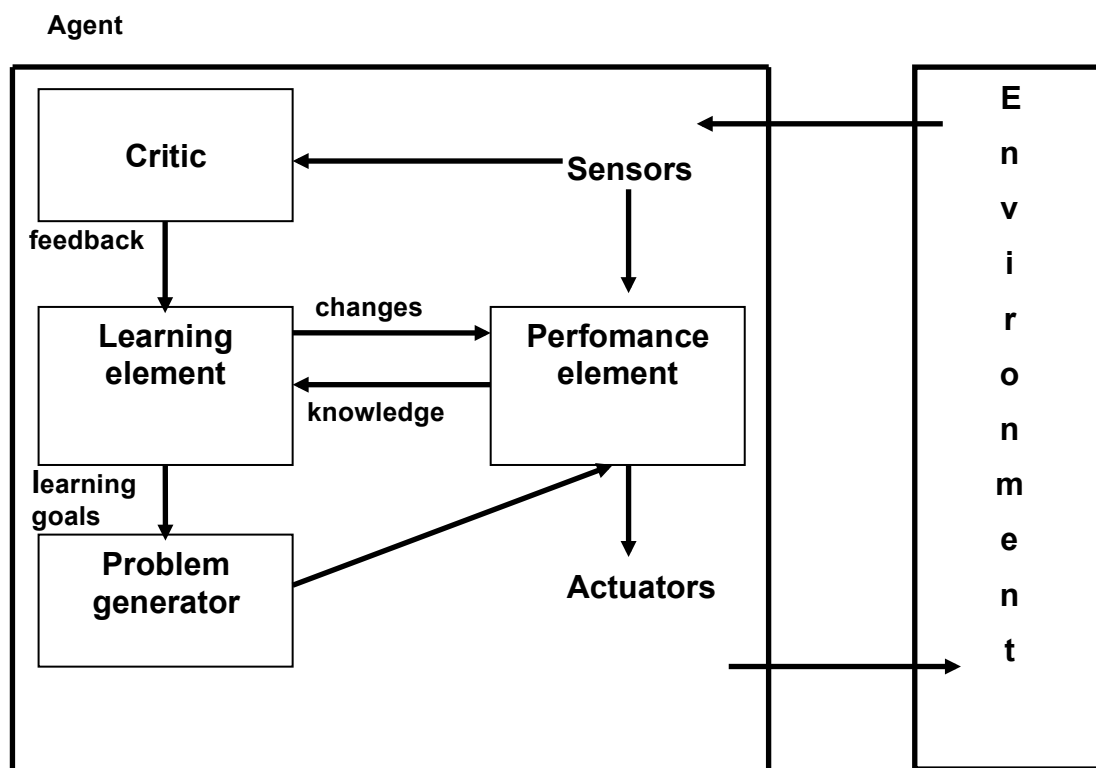
It is instructive to consider why the table-driven approach to agent construction is doomed to failure. Let P be the set of possible percepts and let T be the lifetime of the agent (the total number of percepts it will receive). The lookup table will contain $|P|^T$ entries. Consider the automated taxi: the visual input from a single camera comes in at the rate of roughly 27 megabytes per second (30 frames per second 640×480 pixels with 24 bits color information). This gives a lookup table with over $10^{250,000,000,000}$ entries for an hour driving. Even the lookup table for chess, a tiny, well-behaved fragment of the real world – would have at least 10^{150} entries. The daunting size of these tables (the number of atoms in the observable universe is less than 10^{80}) means that (a) no physical agent in this universe will have the space to store the table, (b) the designer would have no time to create the table, (c) no agent could ever learn all the right table entries from its experience, and (d) even if the environment is simple enough to yield a feasible table size, the designer still has no guidance about how to fill in the table entries.

Despite all this, table-driven-agent does what we want: it implements the desired function. The key challenge for AI is to find out how to write programs that, to the extent possible, produce rational behavior from a small amount of code rather than from a large number of table entries. We have many examples showing that this can be done successfully in other areas: for example, the huge tables of square roots used by engineers and schoolchildren prior to 1970s have now been replaced by a five-line program for Newton's method running on electronic calculators.

If an agent acts in an environment and interacts with other agents it needs to know both how the world evolves in the sense of having knowledge about the world and it needs to know what its actions with respect to other agents, too. For example, the agent needs to know what the result of questions understood as an action accomplishes. However, the agent also needs to learn. Learning affects the agent to operate in an initially unknown environment, becoming more competent than its initial knowledge alone might allow (cf. Russell and Norvig 2003: 51f). With respect to language and communication, the agent needs to learn more about communicative processes. However, in an interaction where information is transferred, the agent also needs to learn about circumstances in the environment and it needs to be able to construct and revise presuppositions which are crucial in communication. Also, an important part of learning with respect to language is learning new words and adding them to the lexicon and to understand constructions in which for example intransitive words, are used transitively or vice versa. Therefore, an agent program is needed which constitutes a learning agent. Also, language acquisition can be modeled on an intelligent agent. Thereby it can be used as a test bed for the application of a linguistic theory.

A learning agent can be divided into four conceptual components. The most important component is the distinction between the learning element, which is responsible for making improvements, and the performance element, which is responsible for selecting

actions. In this case the performance element contains a model-based agent which has static knowledge about the world, a set of rules according to which it acts, and actions it can perform. The state the agent holds can be updated, and it can have access to the rules (cf. Russell and Norvig 2003). In the case of a communication agent these rules would contain knowledge about the language in questions and by this it contains the grammar. The performance element of the agent uses feedback from the critic on how the agent is doing and determines how the performance element should be modified to be better in the future (cf. Russell and Norvig: 2003: 52). The last component of the learning agent is the problem generator, which is responsible for suggesting actions that will lead to new informative experiences. Here, the point is that if the performance element had its way, it would keep doing the actions that are best, given what the agent knows (cf. Russell and Norvig 2003: 52). What is important here is that the learning element can make changes to any of the knowledge components of the agent. The design of an intelligent software agent as used in this paper is described in figure 1:



(Russel and Norvig 2001: 53)

Figure 1: Architecture of Learning Agent

3. Macroroles in RRG

The application of semantic roles in the RRG framework is of importance for the analysis of three-place predicates in RRG. In this section I will describe the RRG account of semantics. Later in this section I will introduce a new account of the treatment of semantic roles in the RRG framework in order to account for a computational analysis of three-place predicates in RRG. RRG uses seven Aktionsarten which additionally all have causative counterparts (cf. Gottschalk 2010). Aktionsarten are assigned on the basis of tests which are language specific. Following Van Valin, these tests are the basis of the RRG approach to argument realization and syntactic

organizations (cf. Diedrichsen 2012: 2). The Aktionsarten are represented in terms of logical structures. They are given in 7.

(7)	State	predicate '(x) or (x, y)
	Activity	do '(x, [predicate '(x) or (x, y)])
	Achievement	INGR predicate '(x) or (x, y) or INGR do '(x, [predicate '(x) or (x, y)])
	Semelfactive	SEML predicate '(x) or (x, y) or SEML do '(x, [predicate '(x) or (x, y)])
	Process	PROC predicate '(x) or (x, y)]
	Accomplishment	PROC predicate '(x, (y)) & INGR predicate '((z), y)
	Causative	α CAUSE β where α and β are LSs of any type

There are three distinct levels of generality which apply to semantic roles: the first level is 'verb-specific' semantic roles like *killer*, *hearer*, *broken* etc. The second level in which semantic roles can apply are thematic relations. These are generalizations across verb-specific roles like *agent*, *instrument*, *experience*, *theme* or *patient*. The third level is generalized semantic macroroles which are a generalization across thematic relations (cf. Van Valin 2005: 53). These semantic macroroles are actor and undergoer. Actor is a generalization across agent, experience, instrument and other roles. Undergoer on the other hand is a generalization across agent, experience, instrument and other roles (cf. Van Valin 2005: 53). In RRG, two types of semantic roles are posited: thematic relations and semantic macroroles (cf. Van Valin 2005: 53). RRG uses a configurational approach to the assignment of semantic roles. The idea is that logical structures form the heart of a verb's lexical entry and correspond to thematic relations or θ -role lists assumed in other theories associated with a verb in its lexical entry (cf. Van Valin 2005: 53). In this context Van Valin (2005: 53) points out the following:

There is, however, no listing of thematic relations in a verb's lexical entry in RRG; rather, thematic relations are defined in terms of the argument positions in the decompositional logical structure representations, following Jackendoff (1976)

The continuum from verb-specific semantic roles to grammatical relations are given in figure 2. Definitions of thematic relation in terms of logical structures argument positions are given in table 2.

This table shows that RRG uses a configurational approach to the assignment and definition of thematic relations. However what is crucial is that only logical structures are associated with a verb's lexical entry rather than thematic relations. With respect to a new account to three-place predicates in RRG I will however show that it is reasonable to associate lexical entries of verbs directly with thematic relations in terms of typed feature structures stored in a lexical semantics relation hierarchy in the mental lexicon.

With respect to activity verbs, there are at least ten subclasses in RRG. The first argument of a non-motion activity verb is an effector. This is the participant which does some action which is not marked for volition and control. All other thematic relations which are associated with the first argument of activity verbs are in fact subtypes of effector (cf. Van Valin 2005: 56). What is important about activity verbs is they usually are single-argument verbs, but there are also some which have two arguments. This is the case with verbs like *eat*, *drink* and *play* (cf. Van Valin 2005: 56).

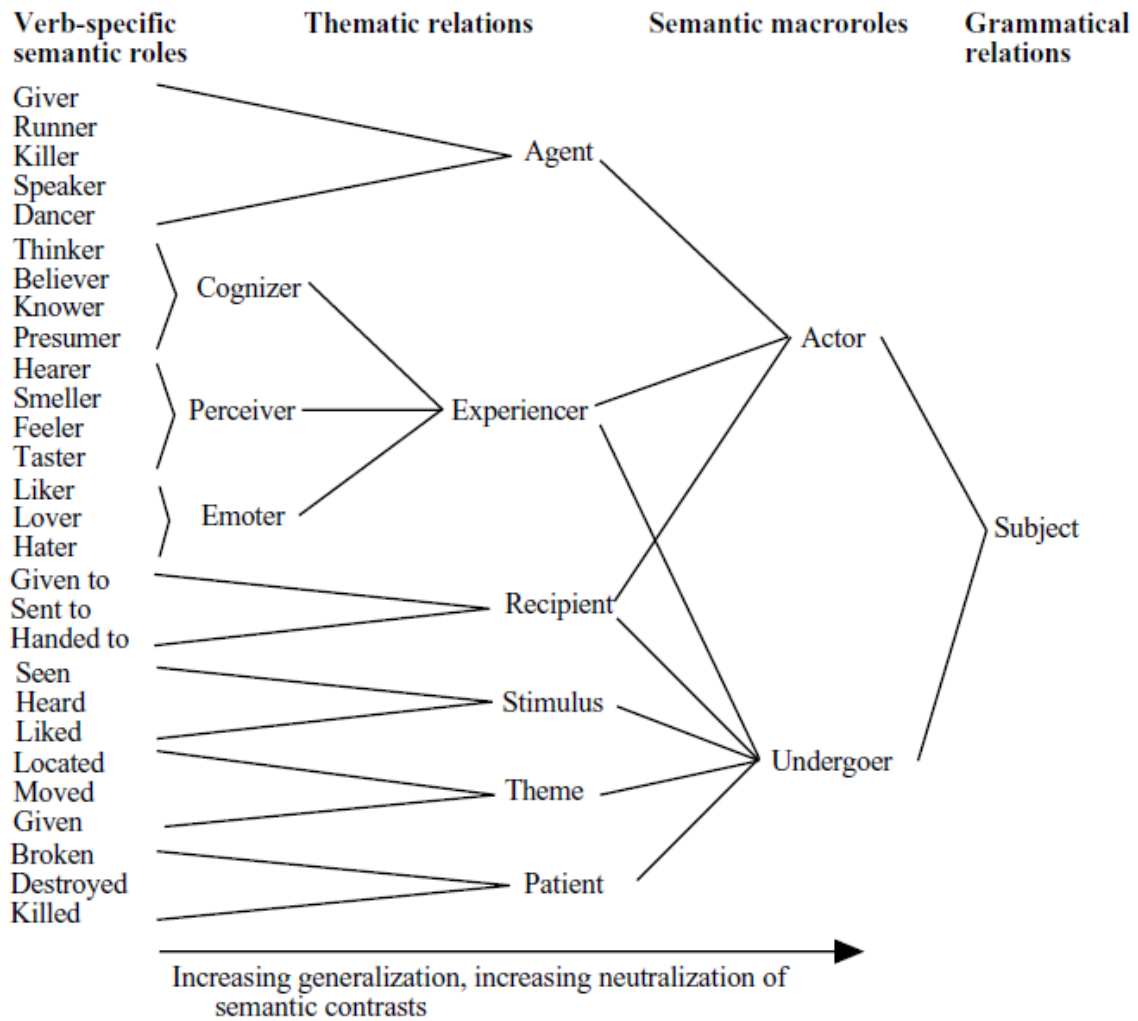


Figure 2: Continuum from verb-specific semantic roles to grammatical relations (Van Valin 2005: 54)

In table 2 one thematic relation is missing: that of agent. In comparison with many other approaches, RRG does not take agent to be a basic relation. The reason is that if agent is taken to be the intentional, volitional and controlling participant in an event, many verbs would appear which take agents in some sentences but not in others (cf. Van Valin 2005: 56). Rather, agentive thematic relations in table 2 are decomposed in terms of more finely grained thematic relations like effector, mover etc. which are more verb specific. In cases where agent applies, it requires an agentive DO-predicate in its logical structure, not even in a complex active accomplishment structure (cf. Diedrichsen 2012). As pointed out by Diedrichsen neither agent nor patient appear naturally without stipulation as parts of a simple logical structure. For Diedrichsen, the important question is why they are listed among the ‘relevant distinctions’ with thematic relations, while recipient does not occur. All arguments which are contained in the logical structure are realized. In the default situation all arguments in the logical structure of the predicate must appear in the core of the clause (cf. Van Valin 2005: 57). In passive constructions the situation is different, since the effector, if overtly expressed, occurs as oblique constituent in the periphery. (cf. Van Valin 2005: 57). RRG seems to posit a great many of thematic relations since they appear in table 2. However, as pointed out in Van Valin (2005: 57) only five distinctions are relevant which are shown in figure 3 (cf. Van Valin 2005: 57).

Table 2: Definitions of thematic relations in terms of logical structure argument positions (Van Valin 2005: 55)

I. STATE VERBS		
A. Single argument		
1. State or condition	broken' (x)	x = PATIENT
2. Existence	exist' (x)	x = ENTITY
B. Two arguments		
1. Pure location	be-LOC' (x, y)	x = LOCATION, y = THEME
2. Perception	hear' (x, y)	x = PERCEIVER, y = STIMULUS
3. Cognition	know' (x, y)	x = COGNIZER, y = CONTENT
4. Desire	want' (x, y)	x = WANTER, y = DESIRE
5. Propositional Attitude	consider' (x, y)	x = JUDGER, y = JUDGMENT
6. Possession	have' (x, y)	x = POSSESSOR, y = POSSESSED
7. Internal Experience	feel' (x, y)	x = EXPERIENCER, y = SENSATION
8. Emotion	love' (x, y)	x = EMOTER, y = TARGET
9. Attributive	be' (x, [pred'])	x = ATTRIBUTANT, y = ATTRIBUTE
10. Identificational	be' (x, [pred'])	x = IDENTIFIED, y = IDENTITY
11. Specificational	be' (x, y)	x = VARIABLE, y = VALUE
12. Equational	equate' (x, y)	x, y = REFERENT
II. ACTIVITY VERBS		
A. Single argument		
1. Unspecified action	do' (x, Ø)	x = EFFECTOR
2. Motion	do' (x, [walk' (x)])	x = MOVER
3. Static motion	do' (x, [spin' (x)])	x = ST-MOVER
4. Light emission	do' (x, [shine' (x)])	x = L-EMITTER
5. Sound emission	do' (x, [gurgle' (x)])	x = S-EMITTER
B. One or two arguments		
1. Performance	do' (x, [sing' (x, (y))])	x = PERFORMER, y = PERFORMANCE
2. Consumption	do' (x, [eat' (x, (y))])	x = CONSUMER, y = CONSUMED
3. Creation	do' (x, [write' (x, (y))])	x = CREATOR, y = CREATION
4. Directed perception	do' (x, [hear' (x, (y))])	x = OBSERVER, y = STIMULUS
5. Use	do' (x, [use' (x, y)])	x = USER, y = IMPLEMENT

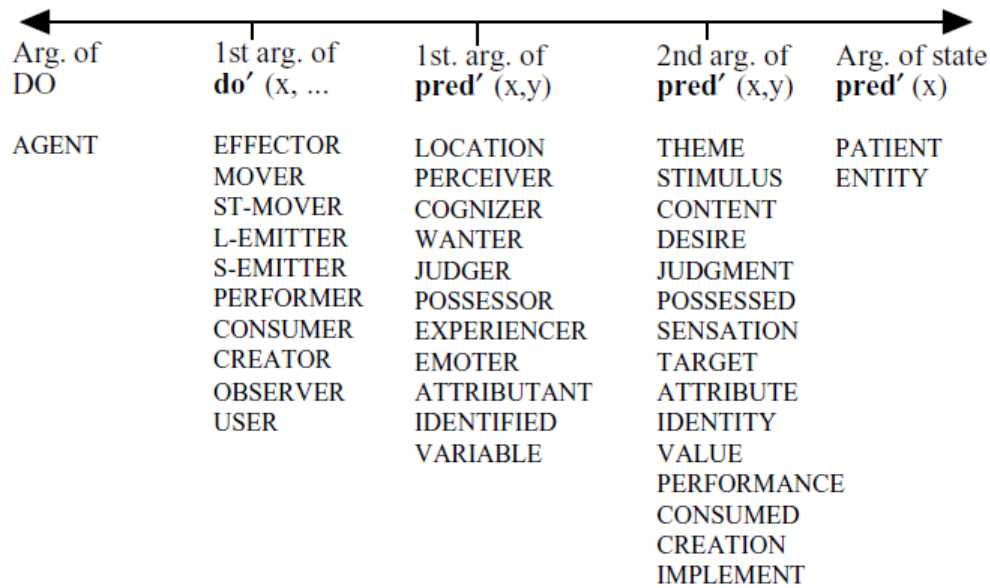


Figure 3: Thematic relations continuum in terms of logical structure argument positions (Van Valin 2005: 58)

Following Van Valin (2005: 57), agents are willful, controlling, instigating participants in states of affairs. Patients on the other hand are strongly affected participants (cf. Van Valin 2005: 57). Van Valin (2005: 57) explains the idea of the continuum of thematic relations used in RRG as follows:

Taking these as endpoints on the continuum makes it possible to place the other role-types with respect to them. The DO of lexicalized agency always co-occurs with the do' (x, ... which defines effector and its subtypes, and accordingly the first two columns are closely related to each other; all of them express participants which do something. At the other end of the continuum fall patient and theme, etc. The single argument of state $predicate'$ (x) includes those participants which are crushed, killed, smashed, shattered, broken, destroyed, etc., while the second argument of $predicate'$ (x,y) includes those participants which are placed, moved, thrown, given, possessed, transferred, seen, heard, loved etc. In terms of affectedness, the former type of participant is much more affected than the latter, hence the placement of the single argument of state $predicate'$ (x) at the end of the hierarchy.

There is also a middle continuum in which the first element of $predicate'$ (x, y) falls. If this is contrasted with the first argument of do' , what becomes obvious is that seeing, thinking, believing, possessing, etc. are less agent-like than speaking, doing, moving, performing and consuming are. This means their placement is to the right of effector, etc. In cases where the contrast is on the second argument of $predicate'$ (x, y), then the reverse conclusion follows (cf. Van Valin 2005: 58). Van Valin (2005: 58) explains the following with respect to the continuum of thematic relations:

Seeing, thinking, liking, believing, etc. involve some kind of internal activity (mental, emotional or perceptual) on the part of the participant, whereas being seen, being thought about, being liked or being believed does not require any action or effort of any kind on the part of the participant. Hence the participant denoted by the first argument is more active and hence more agent-like than the participant referred to by the second argument, and, accordingly, the first argument is closer to the agent end of the hierarchy than the second argument. (Van Valin 2005: 58)

From this, Van Valin (2005: 58) concludes that positioning of the different argument positions in the continuum in figure 3 is able to reflect the semantic contrasts among them. Van Valin (2005: 59) points out the following:

The theoretical implications of this system for deriving thematic relations from logical structures are very important. If it is the case that the thematic relations which a verb takes are a function of the argument positions in its logical structure, and there is a system of lexical representation in which there are independent criteria for assigning logical structures to verbs, then there are independent criteria for assigning thematic relations to verbs. This is the case because the thematic relations are a function of the logical structure of a verb, and there are independent criteria for attributing a logical structure to a verb. Thematic relations cannot be assigned arbitrarily. (Van Valin 2005: 59)

Logical structures in RRG are determined on the basis of tests given in Van Valin (2005: 59). As pointed out by Van Valin (2005: 59), the great advantage of this system of lexical representation is that there are tests providing independent criteria for the assignment of a particular logical structure. Hence a particular argument structure is assigned to a given verb (cf. Van Valin 2005: 59).

It is important to emphasize that in the system presented here, thematic relations play no direct role in the lexical representation; the relevant semantic properties of the verbs are expressed by the decompositional logical structure representations, not by thematic relations. Thus even though a large number of role labels like agent, cognizer, theme and patient have been used in this discussion, they are merely mnemonics for argument positions in logical structure. They have no independent status. Since there is as yet no adequate decompositional representation for the primitive state and activity predicates which are the argument-bearing components of the system and which carry the substantive semantic load, these labels are useful in that they indicate the subclass of the predicate; hence cognizer means 'second argument of a two-place state predicate of cognition', judgment means 'second argument of a two-place predicate of propositional attitude' and theme means 'second argument of a two-place predicate of location', for example. These labels will be used in this way, and it must be kept clearly in mind that these labels do not refer to independently meaningful relations but rather to argument positions in the logical structure or predicates of a certain type. (Van Valin 2005: 60).

In the model I will develop in this paper, the treatment of thematic relations will be rather different from the account proposed in Van Valin (2005). Instead of the thematic relations playing an underpart I will show that thematic relations are an essential part of the construction's signatures for three-place predicate constructions, which are specified in the lexical entry of the verb. I will also show that macroroles need not occur in the framework proposed in this paper. Macroroles are the second type of semantic roles used in the RRG framework and they are generalized semantic roles (cf. Van Valin 2005: 60). The two arguments are 'actor' and 'undergoer'. They are the primary arguments of transitive predications, either one of which may be the single argument of an intransitive verb. Generally actor is the most agent-like argument and undergoer is the most patient-like argument. The name macroroles refer to the fact that they subsume a number of specific thematic relations and they are motivated by the fact that in grammatical constructions groups of thematic relations are treated alike (cf. Van Valin 2005: 60). In this context, Van Valin (2005: 60) explains the following:

For example, themes and patients function alike for certain purposes in the grammar. It is necessary to distinguish them on semantic and other grounds. But nevertheless, the grammar, for certain purposes, treats these roles as essentially the same, e.g. they can be both the direct object in an active and the subject in a passive. In fact, active and passive in English can be described in terms of lists of thematic relations. Agent, effector, experience, perceiver, possessor, judger, etc. can be the subject of an active verb, while patient, theme, stimulus, possessed, location, etc., can be

direct object. In the English passive, patient, theme, stimulus, possessed, location, etc., can be subject, while agent, effector, experience, perceiver, possessor, judge, etc., can be the object of the preposition *by*. It appears that a significant generalization is being missed here, since there are long disjunctive lists of roles in these statements. But in fact, it is not an accident that they seem to group together the way they do, and the obvious generalization can be captured in terms of semantic macroroles: in an active clause the actor is subject and the undergoer is direct object, while in a passive the undergoer is subject and the actor is a peripheral PP: (Van Valin 2005: 61)

In RRG, the relation between macroroles and logical structure argument positions is captured in the actor-undergoer hierarchy [AUH] given in figure 4:

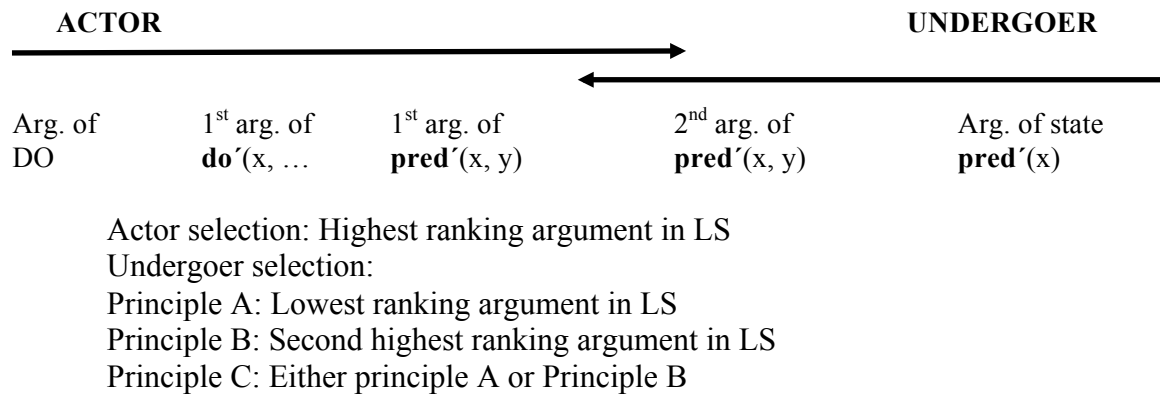


Figure 4: The Actor-Undergoer Hierarchy (Van Valin 2007: 61)

This revised version of the AUH is based on Van Valin (2007) rather than on Van Valin (2005). In the hierarchy, the leftmost argument is the actor and the rightmost argument is the undergoer. This is referred to as Principle A, which applies via default. Cross-linguistically, undergoer assignment is not as stable as actor assignment, since in three-place predicates undergoer assignment may be variable. Examples for variable undergoer linking are found in English dative shift constructions and ditransitives. Also German has variable undergoer linking. These English constructions with variable undergoer linking will be discussed in greater length in sections 4 and 8. Later in this paper a new account to variable undergoer linking in RRG will be developed.

In the AUH in figure 4, thematic relations along the continuum in figure 3 are represented. Prototypically actor is an agent, while the prototypical undergoer is patient. This depends on the logical structure of a particular verb, given in (7) (cf. Van Valin 2005: 61). In this context, Van Valin (2005: 61) explains:

It must be emphasized that the label 'undergoer' should not be taken literally, just as 'actor' should not. The actor of *see* does not do anything but is nevertheless an actor and the sense intended here, i.e. the logical subject; one could say that the actor is the participant which is responsible for the state of affairs, in the sense that it is impossible to have an action without an entity doing the action, a perceptual situation without a perceiving entity, or a cognitive or emotional situation without a participant experiencing the cognitive or emotional state. Similarly, the undergoer of *see* does not undergo anything, unlike the undergoer of e.g. *kill*, but it is still the undergoer of the verb, i.e. the logical object. In general, the undergoer represents the non-insignificant, affected participant in a state of affairs. The specific semantic content of the macrorole with a particular verb is supplied by the position of the argument in the logical structure, not by its macrorole status, although the two are clearly related. (Van Valin 2005: 62)

There is a ranking of argument positions or thematic relations in the AUH with respect to the selection of actor and undergoer selection which is supported by cross-linguistic

evidence. If a verb has an agent argument, it will always be actor and patient will always be undergoer. In cases where a verb has both a potential agent and an inanimate effector (i.e. instrument), the potential agent must be the actor and never the effector (cf. Van Valin 2005: 62).

In fact the AUH always applies to the assignment of actors and undergoers in RRG. It is a configurational approach and the assignment of macroroles takes place based on this approach. In cases where the arguments in the logical structure are not prototypical agents and patients the thematic relations continuum applies. If in a transitive construction the argument is in the leftmost position in the logical structure and it is semantically closer to the agent end of the thematic relations continuum then it is assigned actor. If, however, the argument in the logical structure is in the rightmost position and semantically closer to the patient end of the thematic relations continuum, it is undergoer. However, generally the assignment of actor and undergoer is based on the AUH rather than on the thematic relations continuum this in fact means, as pointed out in Van Valin (2005), the thematic relations continuum is better to be understood as a mnemonic.

Generally, the number of macroroles a verbs takes is predictable from its logical structure. There are only three possibilities: 0, 1, 2. In cases where a verb has two or more arguments in its logical structure as in [**do**'(x,)] CAUSE [PROC & INGR **beat**'(y, z)] or in **hear**'(x, y), the unmarked situation is for it to have two macroroles. For verbs having only one single argument in their logical structure as in **do**'(x, [**walk**'(x)]) or in PROC & INRG **open**'(y), the unmarked situation is to have only one macrorole. It is also possible for verbs to have no arguments and therefore no macroroles as in **do**'([**snow**']) (cf. Van Valin 2005: 63). In this context, Van Valin (2005: 63) explains:

The nature of macroroles is also a function of the verb's logical structure. If a verb has two, then they must be actor and undergoer. For verbs which have a single macrorole, the default choice follows directly from the logical structure, the macrorole will be actor; otherwise it will be undergoer.

The way to determine the number of macroroles is determined by the macrorole assignment principles given in (8):

(8) Default Macrorole Assignment Principles

- a. Number: the number of macroroles a verb takes is less than or equal to the number or arguments in its logical structure.
 1. If a verb has two or more arguments in its logical structure, it will take two macroroles;
 2. If a verb has one argument in its logical structure, it will take one macrorole.
- b. Nature: for verbs which take one macrorole,
 1. If the verb has an activity predicate in its logical structure, the macrorole is actor.
 2. If the verb has no activity predicate in its logical structure, the macrorole is undergoer.

(Van Valin 2005: 63)

If verbs are irregular and have exceptional transitivity, this is indicated in the lexical entry of the verb by '[MR α]. The variable α is replaced by the number of macroroles in the specific lexical entry (cf. Van Valin 2007: 39). In the framework developed in this

paper, I will develop specific lexical entries of verbs different from the approach developed in Van Valin (2005) and Van Valin (2007). One important point discussed in Van Valin (2005: 64f) and Haspelmath (2008) is if it is reasonable to assume more than two macroroles in RRG.

Van Valin (2005: 64) sees two possible justifications for the assumption of a third macrorole: (1) with a third macrorole it is possible to label the third argument of a ditransitive verb, (2) it would be possible to account for dative case assignment (cf. Van Valin 2005: 64). Following Van Valin (2005: 64), these justifications have no force in RRG, since the third argument of a ditransitive verb is a non-macrorole core argument. In German, the third argument would be a non-macrorole direct core argument, since it is not adpositionally marked. In English on the other hand, it would be a non-macrorole oblique core argument in a dative shift construction or a non-macrorole direct core argument in a double object ditransitive construction (cf. Van Valin 2005: 64f). I will discuss Van Valin's specific interpretation of these constructions in section 4. Van Valin (2005: 65) describes his arguments against the assumption of a third macrorole in RRG as follows:

There are strong empirical and theoretical reasons for rejection the postulation of a third macrorole. First and foremost, it is highly likely that it would not be universal like actor and undergoer. While all languages have cores with two core arguments, some languages have strongly disprefer and perhaps even do not permit three core arguments in a single core. Some serializing languages, e.g. Yoruba, Yatye (Stahlke 1970), fall into this category. In such languages, clauses with more than two arguments require complex expressions in which the additional argument is a core argument of a second nucleus in a second core. So, for example, in expressing a transfer verb meaning 'give' would be serialized with the transfer verb in order to express the recipient, or with a verb like 'break' or 'kill' a verb meaning 'take' or 'use' would be serialized in order to express the instrument.

The second argument against the assumption of a third macrorole is that there is no consistent morphosyntactic treatment of the third argument (cf. Van Valin 2005: 65). However, as pointed out in Van Valin (2005: 65), actor and undergoer have consistent coding properties cross-linguistically. In active voice constructions, actor and undergoer are always direct arguments of the verb. However, the concept of 'direct' varies morphosyntactically across languages. In English a direct argument is not marked by a preposition, while in German and Russian it is marked by a direct case rather than by a preposition (cf. Van Valin 2005: 65). In head-marking languages like Crow or Lakhota which are both Siouan languages 'direct' means being coded on the verb. In languages with case marking, actor and undergoer either have nominative and accusative or ergative and absolutive case. However, with respect to the third core argument, the treatment is not consistent since it may be a direct argument in the dative case, as in German, Russian or Dyrbal, or it may be an oblique argument marked by an adposition as in English or Jakaltek. Actor and undergoer on the other hand are never oblique arguments in the core (cf. Van Valin 2005: 65). For Van Valin (2005: 65), the important issue about the third macrorole is the following:

This raises a further issue: what exactly would count as a third macrorole? In an language like German or Russian, for example, it could be restricted to the third direct core argument of ditransitive verbs. But in a language like English, this would imply that only the *to*-PP with certain verbs would count as being the third macrorole. Why should this particular argument be so analyzed and not other oblique core arguments?

What Van Valin (2005: 65) refers to is that in a sentence like *Abby gives the blood test to Ducky* the third macrorole would be *to Ducky*. However, the question is why *on the litter* as in the *Palmer loaded the injured person on the litter* should not be given the same analysis? In fact, both are omissible PPs and both can occur as ‘direct object’ in an alternative clause pattern as in *Abby gives Ducky the blood test* or as in *Palmer loaded the litter with the injured person*. The next question arising is whether, if *on the litter* has this status in the first sentence, then should *with the injured person* also be analyzed the same way in the second example (cf. Van Valin 2005: 65).

From these findings Van Valin (2005: 65) concludes that it is difficult to justify why some oblique arguments should be analyzed as instantiating a third macrorole but not others. If, however, all oblique core arguments are analyzed this way, then whatever function and semantic content, it would be very different from that which is hypothesized in form of a third macrorole in German and Russian (cf. Van Valin 2005: 66). Van Valin’s third argument against the assumption of a third argument is as follows:

Third, a third macrorole would be markedly less important for the syntax than actor and undergoer and hence is difficult to justify on syntactic grounds. It would play little or no role in subject selection with intransitive verbs. The single argument of an intransitive verb is either an actor or an undergoer in the vast majority of cases, and in those cases where the single argument is non-actor or non-undergoer it does not correspond semantically to the third argument of three-argument verbs. It also plays no role in the major typology of syntactic systems: ergative vs. accusative vs. split-intransitive (e.g. Acehnese) [...]. These differences revolve around the treatment of actor and undergoer; the third argument of ditransitive is not a factor.

From the facts given above, Van Valin (2005) concludes that a third macrorole would be a qualitatively different concept from the two semantic macroroles posited in RRG. The point is it would not be universal, it would not receive consistent morphosyntactic treatment, and it would be relatively unimportant to the syntax (cf. Van Valin 2005: 66). What follows from Van Valin’s perspective is that there is no justification for positing a third macrorole (cf. Van Valin 2005: 66). Diedrichsen (2012) notes that in RRG, the concept of macroroles resembles the idea of ‘logical subject’ and ‘logical object’ semantically. In RRG, actor is the semantic counterpart of what is referred to as ‘subject’ in traditional approaches, since subject is the most agent-like argument. Undergoer on the other hand is the most patient-like and the semantic counterpart of the direct object (cf. Diedrichsen 2012). Diedrichsen (2012) explains the following:

While the traditional labels for grammatical relations, subject and object, are not used in RRG, the theory establishes the macroroles, which refer to semantic relations. Note that here with the description of macroroles, the semantic relations list comes back into play, which have been rejected before, in favor of the logical structures. Macroroles are generalizations across thematic relations. Actor is the subject of active transitive constructions, and undergoer is the subject of passive constructions. Thus, the macroroles are not merely semantic; rather they bridge the gap between semantic and grammatical relations. (Diedrichsen 2012).

Van Valin (2005: 60) points out that macroroles are motivated by the fact that in specific grammatical constructions, thematic relations are treated alike. Therefore, macroroles can be considered constituting the link from semantics to syntax within the interface between syntax and semantics (cf. Diedrichsen 2012). The advantage of this is that the approach used in RRG is basically functional, since arguments in a syntactic construction are characterized based on the verb semantics. Also, this approach is cross-linguistically applicable, because in all languages arguments in a transitive predication

can be distinguished in terms of an ‘agent-like’ and a ‘patient-like’ argument. However, Diedrichsen (2012) points out that in the formulation of the mapping between semantics and syntax, some generalizations are carried out which call into question the semantic motivation of syntactic facts, which include argument structure relations. Diedrichsen argues that facts of meaning are extracted from verb meanings and are selected in virtue of their contribution to argument structure properties. From this she concludes that the argument structure is decisive for the classification of verbs rather than the verbs semantics itself. This means the definition of argument structure properties based on verb classes becomes circular. Also Diedrichsen argues that argument structure positions coincide with the semantics of the verb itself. In logical structure representations the positions are argument positions in transitive and intransitive constructions. In fact the logical structures in the AUH are designed to form a continuum, its endpoints being agent and patient. A further important point Diedrichsen (2012) mentions is that the linking from semantics to syntax is based on the requirements of transitive constructions, as pointed out earlier in this section. The basic definition of actor and undergoer does not give any semantic motivation. Rather the construction-based characterization, that they are the two primary arguments in a transitive construction, gives the semantic representation (cf. Diedrichsen 2012). From these findings Diedrichsen (2012) concludes the following:

Thus the definition of the ‘generalized semantic roles’ and also the classification of verbs with respect to ‘grammatically relevant facets of meaning’ are based on the features of argument structure constructions. The positions in the logical structure are argument structure positions. Accordingly, it is questionable whether a theory with an elaborated account of logical structures, that explicitly denies the theoretical importance of thematic relations (see above), would necessarily need the concept of macroroles.

Diedrichsen (2012) argues that argument positions are in fact defined with respect to their role in argument structure constructions. From her perspective, macroroles abstract away from particular verb meanings that makes them applicable for argument structure patterns found in transitive constructions. As pointed out in Van Valin (2005), transitive constructions are the basic construction pattern rather than ditransitive constructions. This is obvious because of the assumption of two instead of three macroroles. However, in this paper I will show that in a computational linguistics approach, which is based on constructional schemas as developed in Nolan (2011) and a revised version of the theory of the mental lexicon as proposed in Gottschalk (2010a, 2011b), the assumption of macroroles as proposed in Van Valin (2005), is not necessary. Diedrichsen (2012) points out that if the construction was considered to be responsible for argument realization, this would yield the abandonment of the highly problematic concept of macroroles.

As will be shown later in this paper, syntactic processes such as argument alternations in three-place predicates can be described without the help of macroroles as used in Van Valin (2005). Rather, macroroles will be treated as attributes in typed feature structures which are part of a lexical semantics relations hierarchy containing the typed feature structures. The lexical semantics relations hierarchy is a hierarchical network of thematic relations used to semantically define them. They are connected with the lexical fingerprints of verbs via unification. Diedrichsen (2012) also explains the following:

The definition of the macroroles is based on argument positions in logical structures and their position with respect to each other. The correlation of argument positions and semantic relations is

carried out on the basis of the thematic relations continuum. Thus, ‘1st arg and 2nd arg of’ and ‘leftmost’ and ‘rightmost’ suffice to identify the arguments in the logical structure. The thematic relations continuum is necessary to give a semantic reference to the argument positions in the logical structure (see also Michaelis and Ruppenhofer 2001). The number of arguments and their syntactic realizations are provided by the construction. Macroroles are not necessary.

Also, Diedrichsen (2012) explains that the signification of argument hierarchies means mapping thematic relations with syntactic relations appearing in the sentence. Syntactic relations however are defined as argument positions in monotransitive constructions. A further problem in Van Valin (2005) is that recipients, though being very important in the syntax, do not appear in the AUH (cf. Diedrichsen 2012). If, however, macroroles are not assumed but are rather treated as attributes in typed feature structures, as will be proposed in this paper, such a problem will not occur. Diedrichsen describes the advantages of using constructions in favor of macroroles as follows:

1. The constructional schemas are there already, they do not have to be introduced into the theory. They are very important part of RRG (Van Valin 2005: 131 – 135). Bulter (2009: 28) points out that RRG is a ‘constructional model’ to a certain degree. What would be necessary, though, is to formulate constructional schemas for argument structure constructions.
2. With constructions as main contributors of argument structure, it would be possible to describe the PSA, for example, with respect to the construction.
3. The macroroles have been one source for the identification of the PSA. With a constructional account, the macroroles would be dispensable. As the previous discussion has shown, macroroles are in deficit for many reasons. They don’t suffice to describe syntactic processes and phenomena, in particular with respect to ditransitive constructions.
4. It would be possible to treat constructions equally. Emerging constructions or spontaneous formations could be treated as constructions, not as mistakes or irregularities. This is especially important for the description of language change and variation. The fact that some constructions are more frequent than others would not be principally relevant for this description. (cf. Diedrichsen 2012: 9)

4. Van Valin’s (2007) approach to three-place predicates

Van Valin (2007) accounts for three-place predicates with the use of a modified version of the Actor-Undergoer-Hierarchy [AUH] - developed in Van Valin (2005) - in terms of identifying three different patterns for variable undergoer assignment in verbs with three argument positions. In the following I will discuss patterns of variable undergoer linking occurring in German and English. In Van Valin (2007) it is noted that abstract predicates in the lexical decomposition system employed in RRG can only have zero, one or two arguments. From these findings, Van Valin (2007: 43) concludes that three-place predicates must have complex LSs which are composited of at least two abstract predicates. Examples for the semantic representation of English three-place predicates are given in (9):

- (9) [do’(x,)] CAUSE [PROC & INGR **predicate**’(y, z)]
 e.g. *give, present* [do’(x,)] CAUSE [PROC & INGR **have**’(y, z)]
show [do’(x,)] CAUSE [PROC & INGR **see**’(y, z)]
teach [do’(x,)] CAUSE [PROC & INGR **know**’(y, z)]
load [do’(x,)] CAUSE [PROC & INGR **be-on**’(y, z)]
put [do’(x,)] CAUSE [PROC & INGR **be-LOC**’(y, z)]
 (cf. Van Valin 2007: 43)

Van Valin (2007: 44) interprets the logical structures in (9) based on Larson (1988) and others that within the embedded predication in sentences involving verbs like *give*, *present*, *show*, *teach*, *load* and *put*. In RRG the rightmost argument in LSs as given in (9) is the default choice for underoer. This means that in principle it is possible for the *y* argument to be selected as underoer. For English this is illustrated in (10) and (11), which may be termed ‘transfer alternation’ (cf. Van Valin 2007: 44).

(10) (cf. Van Valin 2007: 44)

- a. [do'(Abby,)] CAUSE [PROC & INGR **have'**(McGee, bloodtest)]
- b. Abby [Actor] gave the bloodtest [Undergoer] to McGee. Unmarked choice
- c. Abby [Actor] gave McGee [Undergoer] the bloodtest. Marked choice

(11) (cf. Van Valin 2007: 44)

- a. [do'(Abby,)] CAUSE [PROC & INGR **have'**(McGee, computer)]
- b. Abby [Actor] presented the computer [Undergoer] to McGee. Unmarked choice
- c. Abby [Actor] presented McGee [Undergoer] with the computer. Marked choice

In (10b), the leftmost argument in the LS (*Abby*) is selected as actor, while the rightmost argument is selected as undergoer (*the bloodtest*). Because the sentence is in active voice, the actor appears as core-initial PSA, which is similar to ‘subject’ in traditional grammatical theories, while the undergoer occurs in the immediate post-nuclear position, which in traditional grammatical theories is the ‘direct object’. In this sentence, the third argument *McGee* is a non-macrorole and therefore a preposition assignment rule applies, which assigns *to* to it (cf. Van Valin 2007: 44). The same analysis pertains for the default linking with *present* in (11b).

In the English examples (10c) and (11c), the actor selection is the same; however, undergoer selection is different. In these examples, *McGee*, which is the *y* argument (recipient), is assigned undergoer, leaving *the bloodtest* (10c) and *the computer* (11b) as non-macrorole arguments (cf. Van Valin 2007: 45). Both examples are active voice and the actor occupies the core-initial PSA (‘subject’) position. The interesting question in this context is what happens to the non-macrorole arguments *the bloodtest* and *the computer*? Based on the AUH it should be the default choice for undergoer, but it has been ‘passed over’ in favor of a lower ranking argument (cf. Van Valin 2007: 45). In (11c) this is the situation where the following rule applies:

(12) Preposition assignment rules for English

- a. Assign *to* to non-MR *x* argument in LS segment:
PROC & INGR / INGR **pred'**(*x*, *y*)
- b. Assign *from* to non-MR *x* argument in LS segment:
... PROC & INGR / INGR NOT **pred'**(*x*, *y*)
- c. Assign *with* to non-MR *y* argument if, given two arguments, *x* and *y*, in a logical structure, with *x* lower or equal to *y* on the AUH, *y* is not selected a macrorole.
(cf. Van Valin 2007: 42)

In (5c), the rule in (8c) applies and as shown in (5c) *the computer* is marked by the *with* rule, since here the verb is *present*. However, with a small class of verbs the *with* rule does not apply and the result is a sentence as given in (4c) (cf. Van Valin 2007: 45). Examples like these are often referred to as ‘ditransitive’.

What is crucial with respect to the analysis of three-place predicates in Van Valin (2007) is that several problems occur: RRG uses a configurational approach to the assignment of semantic macroroles based on the AUH, as proposed in Van Valin (2007: 61). As pointed out above, the basic idea of the AUH is that the left-most argument in the LS is assigned Actor and the right-most argument is assigned Undergoer. However, as pointed out in Haspelmath (2008), in Van Valin (2005) and Van Valin (2007) it is assumed that the situations (4b) and (5b) are the unmarked choice for undergoer assignment, since this construction is in accordance with the AUH. In fact this 'default' choice of English three-place predicates is not in accordance with native speaker intuitions. Also, with respect to English Van Valin's (2007) analysis runs into difficulties, since, as pointed out in Erteschik-Shir (1979) and Diedrichsen (submitted), the unmarked word order pattern in English is as in (10c) and (11c) rather than in (10b) and (11b). Further evidence for this comes from several interviews with native speakers of English, and from Haspelmath (2008). Since as pointed out in Haspelmath (2008: 85) variable undergoer linking is found in 10% of the languages and as shown in Siewierska (1998) and Haspelmath (2005a) it is not the case that the patterns given in (10b), (11b), are the more frequent patterns.

As pointed out in Haspelmath (2008: 86), this means that the unequal treatment of the patterns given in (10) and (11) in Van Valin's (2007) analysis and in RRG in general seems to be a feature inherited from transformational approaches where one alternating pattern is regarded as the underlying pattern from which the pattern is derived. However, RRG is a monostratal linguistic theory. No intrinsic reason can be found why one alternation pattern has a privileged status over another pattern (cf. Haspelmath 2008: 86). Therefore, in this paper I will provide a different analysis for these patterns based on the inclusion of information structure considerations and a detailed computational linguistics analysis of the linking algorithm from semantics to syntax as developed in Van Valin (2005).

In the next section I will show that Van Valin's (2007) analysis of three-place predicates is not only not in accordance with native speaker's intuitions and cross-linguistic data but is also problematic from a computational linguistics point of view, which renders the linking algorithm developed in Van Valin (2005) as not executable on a RAM.

5. The semantics to syntax linking algorithm in RRG and its shortcomings

As explained in section 1, the linking algorithm in RRG links the semantic representation of the clause in terms of logical structures [LSs] with the syntactic representation of the clause which is called the layered structure of the clause (cf. Van Valin 2005). The linking between semantic and syntactic representations is governed by the completeness constraint which is a very general constraint (cf. Van Valin 2005: 129). This constraint is stated in (13):

(13) Completeness constraint

All of the arguments explicitly specified in the semantic representation of a sentence must be realized syntactically in the sentence, and all of the referring expressions in the syntactic representation of a sentence must be linked to an argument in the logical structure in the semantic representation of the sentence.

The completeness constraint guarantees that there will be a match between the number of arguments in the clause and in the LS of the verb (cf. Van Valin 2005: 130). A crucial assumption in RRG is that the semantic representation of the sentence is built around the LS of the predicator. This predicator is usually the verb and the LS is put together in the lexicon (cf. Van Valin 2005: 130). As Van Valin (2005: 130) notes, for the semantics-to-syntax linking it is crucial for the selection of the syntactic template(s) which constitute the syntactic representation. In RRG syntactic representations are conceived as 'syntactic templates' which are stored in what is called the 'syntactic inventory'. In Van Valin and LaPolla (1997: 69ff) it is shown that syntactic templates are formally equivalent to immediate dominance [ID] rules. This might raise the question whether the use of syntactic templates from a computational point of view is more complex than using ID rules. As shown in Guest (2008), RRG templates are more suitable than rules, since many of the words in which errors are made are removed from the core parsing. This can be explained by the fact that functional words, which are denoted as operators in RRG, are not part of the syntactic representation, but rather are part of the operator projection (cf. Van Valin 2005). The idea of using syntactic templates rather than ID rules proposed in Van Valin and LaPolla (1997) however results in problems with respect to a formal analysis of RRG for which the present study is the initial starting points. This approach also results interesting problems from a computational point of view.

As pointed out by Langer (personal communication), the essential question with respect to the complexity of tree generation is the following: Is there a finite set of trees or is there a recursive set of rules generating a non-finite number of trees? If there is a finite set of trees, no problem occurs with respect to computational complexity. If, however, a recursive set of rules generating a non-finite number of trees, the situation is more difficult. If a generic tree is an instance of a graph the situation becomes even more difficult. In these situations, typically three possible variants are found (a) practically computational (b) belonging to context free grammars and related kinds of grammar with cubic computing time (c) computability is not possible. With respect to situation (c) this means that the generation of trees is approachable. However, it is not manageable with respect to huge amounts of data. This is the case with respect to unification grammars, Turing machines, DATR and programming languages. Also with respect to NULL or zero copular constructions in RRG, the situation becomes difficult with respect the computability of RRG. In these constructions, the nucleus is usually not occupied, which results in an ambiguity of this node which means that these trees may belong to category (b). However, this situation is close to the computational worst case (Langer personal communication). This means while the approach to syntactic templates proposed in Van Valin and LaPolla (1997) might be reasonable with respect to developing a robust algorithm, as pointed out in Guest (2008) it is not clear yet whether this position is reasonable from a complexity theory point of view. To be able to answer questions of the computability as well as the complexity of RRG, this study is a starting point which can be used in further studies of the computability of RRG. The principles governing the selection of appropriate core templates used in RRG are given in (14):

- (14) Syntactic template selection principle
- a. The number of syntactic slots for arguments and argument-positions within the core is equal to the number of distinct specified argument positions in the semantic representation of the core.
 - b. Language specific qualifications of the principle in (a):
 1. All cores in the language have a minimum syntactic valence of 1.
 2. Argument-modulation voice constructions reduce the number of core slots by 1.
 3. The occurrence of a syntactic argument in the pre/postcore slot reduces the number of syntactic slots by 1 (may override (1) above).
- (Van Valin 2005: 130)

The general idea of the principles in (14) is: if a verb has n arguments, there need to be n positions in the core for the arguments appearing in it. This is necessary in order to satisfy the completeness constraint in (13). The principles given in (b) are language specific and apply to English, which requires dummy subjects for argumentless verbs like *rain*, which has a passive and in which WH-words occur in the precore slot, while none of them apply to a language like Lakhota (cf. Van Valin 2005: 130). The linking procedure from the semantic structure and this way from LSs to the syntactic representation (layered structure of the clause) is given in (15).

- (15) Linking algorithm: semantics \rightarrow syntax (Van Valin 2005: 136)
1. Construct the semantic representation of the sentence, based on the logical structure of the predicator.
 2. Determine the actor and undergoer assignments, following the actor-undergoer hierarchy [...]
 3. Determine the morphosyntactic coding of the arguments
 - a. Select the privileged syntactic argument, based on the privileged syntactic argument selection hierarchy and principles [...]
 - b. Assign the arguments the appropriate case markers and/ or adpositions.
 - c. Assign the agreement marking to the main or auxiliary verb, as appropriate.
 4. Select the syntactic template(s) for the sentence following the syntactic template selection principle.
 5. Assign arguments to positions in the syntactic representation of the sentence.
 - a. Assign the [-WH] argument(s) to the appropriate positions in the clause.
 - b. If there is a [+WH] argument of a logical structure,
 1. assign it to the normal position of a non-WH-argument with the same function, or
 2. assign it to the precore or postcore slot, or
 3. assign it to a position within the potential focus domain of the clause (default = the unmarked focus position).
 - c. A non-WH argument may be assigned to the precore or postcore slot, subject to focus structure restrictions (optional).
 - d. Assign the [-WH] arguments(s) of a logical structure(s) other than that of the predicator in the nucleus to
 1. a periphery (default), or
 2. the precore or postcore slot, or
 3. the left- or right-detached position

What is of importance here is that all steps in the linking from semantics to syntax are subject to cross-linguistic variation. Languages like German and English for example show variable undergoer selection in three-place predicate constructions (cf. Van Valin 2005: 136). This is of crucial importance for step 2 in the linking algorithm in (15). Also, the privileged syntactic argument selection principles vary along two major parameters. In this case the accusative vs. ergative privileged syntactic argument selection and whether privileged syntactic argument selection is restricted to macrorole arguments or not (cf. Van Valin 2005: 136). This is of importance in step 3a. Also Van Valin (2005: 137) explains:

[...] information from constructional schemas can play a crucial role at this point. Case and agreement show substantial cross-linguistic variation (step 3b, c). The positions to which XPs are assigned in sentences varies not only within languages but across languages (step 5a), and the possibilities under step 5b cover the range of WH-question types found in human languages.

This in fact means example (4) lays out the general linking algorithm, which would have to be specialized for each individual language. The task for of this research work in progress is to formalize the linking algorithm in terms of a pseudo-code meta-language as introduced in section 1 in order to be able to investigate the computational adequacy of RRG as a long term aim. I will show that already steps 1 and 2 result in computational difficulties for the linking algorithm to be executed on a RAM which results in RRG not being computationally adequate at this point in the study. In this context Van Valin (2005: 137) explains:

The system in (15) [Van Valin's (5.5)] presumes that a speaker is realizing a specific communicative intention, and consequently whether the sentence will be, e.g. active or passive, declarative or interrogative, figures into the formulation of the semantic representation and concomitant syntactic template selection. Moreover, the discourse status (activation level) of the referents of the NPs is also represented [...].

The first question with respect to the computability of RRG in terms of a pseudo-code meta-language on a RAM occurs with respect to step 1 in the linking algorithm. Here, the task is to construct a semantic representation of the sentence, based on the logical structure of the predicator. In fact this is not only a lexical process where the LS is constructed in the lexicon and the variables in the LS are filled with referring expressions based from input of either mental states, external states and a knowledge base containing world knowledge. This means a mechanism is needed explaining how via a unification approach, as proposed in Gottschalk (2010a, 2011), the logical structure can be constructed.

To be able to account for this problem in a computational linguistic framework it is necessary to use the concept of a learning agent as introduced in section 2. A semi-formalization in terms of a pseudo-code meta-language looks as follows in (16).

(16)

```

algorithm step1
function Learning-Agent-with-State(percept) returns an action
  static: state, a description of the current world state;
           rules, a set of condition-action rules;
           action, the most recent action, initially none;
           learning, the state of being able to learn
           critic, the module criticizing the learning process
           problem generation, the module generating problems

state          Udate-States(state, action, percept);
rules          Rule-Match(state, rules);
learning       Udate-States (state, action, percept, critic);
critic        Udate-Learning (state, action, percept);
problem generation  Update-ProblemGeneration (state, action, percept, learning);
action
lexical_template = (state of internal && external affairs)
//instantiation of internal and external states of affairs with a referring lexical
template in the mental lexicon
array referring_expressions[ ] from lexicon && database
for number_argument_slots in lexical_template do
  if number_argument_slots == 1 do
    x = referring_expression;
  end if.
  if number_argument_slots == 2 do
    if referring_expressions[i] == animate and referring_expressions[j] =
    = inanimate do
      x = referring_expression(animate);
      y = referring_expression(inanimate);
    else
      choice
    end if.
  end if.
  if number_argument_slots == 3 do
    choice
  end if.
end for.
semantic_representation = lexical_template;

```

This algorithm instantiates a learning-agent which is in a specific state, has a learning element, a critic and a problem generator, as well as access to a number of rules, which determines its actions and also has access to specific actions it can execute. Neither the states nor the rules are further defined. However, the action is defined. In this case the action to execute is the instantiation of the semantic representation. What first happens is that a lexical template is instantiated with an internal or external state of affairs and then the referring expressions are stored in an array and the current state of the world is called. For the number of argument slots in this lexical template a number of routines is executed. What is checked first is if the number of argument slots in this array is 1, then

the only element of the array is assigned to *x*. If the number of referring expressions in this array is two, then it is checked whether there is an *animate* and an *inanimate* referring expression in the array. If this is the case, the *x* argument is assigned to the referring expression which has the attribute *animate* and the *y* variable is assigned to the referring expression which has the attribute *inanimate*. If there is none such ideal situation, the assignment of referring expressions depends on *choice*, a function which is not further described in this situation. In principle, *function* should be based on the static set of rules and mainly on perception which are not further described in this context. If, on the other hand, the number of argument slots is 3 then there is only *choice* since otherwise it is not possible to determine which referring expression should be assigned to the *x* variable and which referring expression should be assigned to the *y* and *z* variables in the lexical template. The last step in this algorithm assigns the now filled lexical template to the variable semantic representation, which will be used as expression in the remainder of this semi-formalization of the linking algorithm.

What is obvious from this algorithm developed in pseudo-code is that a full-fledged language processing account is difficult to formulate within the mechanisms described in RRG. This is also explained in Van Valin (2006), where he describes how the RRG-framework could be included to a processing model as suggested in Levelt (1989). This in fact means that to develop a language-processing model it is crucial for RRG to account for cognitive processes not taken into account in RRG. In fact, the attempt to formalize the first step in the RRG linking algorithm from semantics to syntax means to formalize RRG as a processing model, in this case a production model rather than a formal grammar. For reasons pointed out above, it is necessary to formalize cognitive processes, which requires a full-fledged processing model with RRG as ‘linguistic engine’. The questions immediately arising are: How can RRG as functional linguistic theory be formalized if the first step in the algorithm already requires a cognitive model as backbone? What are the consequences of this for a formalization of RRG? It will be a task for future research on the formalization of RRG to account for this. The next step in the linking from semantics to syntax in RRG is concerned with the assignment of Actor and Undergoer based on the Actor-Undergoer-Hierarchy as given in (15). A semi-formalization of this part of the algorithm is given in (17).

(17)

```

algorithm step2
if number_argument_slots == 1 in logical structure do
    if lexical_entry_verb == takes_undergoer do
        undergoer = referring_expression_x;
    else
        actor = referring_expression_x;
    end if.
if number_argument_slots == 2 in logical structure do
    actor = leftmost_argument;
    undergoer = rightmost_argument;
end if.
if number_argument_slots == 3 in logical structure do
    actor = leftmost_argument;
    undergoer = new.choice( );
    non_macrorole = new.choice( );
end if.

```


What is shown in the algorithm in (17) is that if the number of argument slots in the logical structure equals 1 it is necessary to have access to the lexicon in order to determine whether the lexical entry of the verb suggests that it can only be satisfied by an undergoer or whether it can be satisfied by an actor. However, the situation in which the verb takes an actor is the default situation. What is crucial in this part of the linking algorithm is that it is not possible to determine the macroroles of the sentence solely on the basis of the AUH, since in cases where the algorithm cannot determine which macrorole should be assigned, it needs to use the function choice, which is not described in more detail. Rather, and this is not stated in the linking algorithm as described in Van Valin (2005: 136), access to the lexicon is of crucial importance. However, even access to the lexicon cannot account for three-place predicates.

As shown in (17), the only situation where the AUH can apply as only basis for the determination of macroroles is a situation in which two argument slots occur in the logical structure. This suggests that from an RRG-perspective, being transitive is the default situation for verbs. However, this results from theory internal considerations in which the application of the AUH is an essential part of the theory, since as pointed out in Van Valin (2005) the assignment of macroroles in the lexicon is the marked situation (cf. Van Valin 2005: 66).

If three argument slots in the LS occur, RRG runs into a difficult problem, since as shown in the algorithm in (17), RRG cannot account for three-place predicates solely based on its procedural approach and on the AUH. As pointed out in section 4, Van Valin (2007) deals with this situation and describes how it is possible to account for three-place predicates in the semantics to syntax linking. However, the preposition assignment rules as proposed in Van Valin (2007) cannot apply to this part of the algorithm, since the assignment of prepositions takes place in step 3 of the linking algorithm in Van Valin (2005: 136). In addition, the AUH developed in Van Valin (2007) leaves a choice with respect to the assignment of three-place predicate. However, as will be shown in section 8 variable undergoer linking in English is governed by information structure considerations.

The question is: What basis can a variable undergoer be assigned on and how it is possible not to use the function choice? As will be shown later in this paper it is possible to structurally account for the assignment of undergoers in English. This can be done with the help of focus structure analysis, since, in fact it is focus structure which governs the assignment of variable undergoers in English. For this, discourse representation structures can be used. This will be described in section 8. Therefore, in this paper, I will revise the linking algorithm proposed in Van Valin (2005: 136) and use an approach in which discourse representation structures [DRS] are used in order to account for information structure considerations in three-place predicates.

The next step in the linking algorithm, as described in Van Valin (2005: 136), given in (15) is the determination of the morphosyntactic coding of arguments. A semi-formalization of this step looks as given in (18):

(18)

```

algorithm step3a
if number_argument_slots == 1 and actor = true do
    PSA = actor;
else
    PSA = undergoer;
end if.
if number_argument_slots == 2 and actor = direct_core_argument do
    PSA = actor
else
    PSA = undergoer
if number_argument_slots == 3 and actor = direct_core_argument do
    PSA = actor
else
    PSA = undergoer
end if.

```

The semi-formalized algorithm given in (18) accounts for step 3a in a language where the actor is a direct core argument as given in German. In situations where the number of argument slots equals 1, the situation is straightforward, since if ‘actor = true’ applies, actor is assigned PSA. If, however, ‘actor = true’ is not true, undergoer is assigned PSA. The situation becomes difficult, however, in situations where the number of argument slots equals 2. In this situation, the intelligent software agent and thereby the algorithm needs to have a mechanism to test in advance whether the actor is a direct core argument or not. However, in the linking algorithm as provided in Van Valin (2005: 136), no account is made for such a test, since in fact Van Valin (2005) seems to claim that sentences are constructed on the fly, which makes the application of some ‘pre-testing’ difficult. The same difficult situation applies in three-place predicates where the number of argument slots equals 3. Again, some pre-testing in order to account for the fact that only direct core arguments can be actors in languages like German, for which this algorithm is developed, would be necessary. Just like in two-place predicates, this algorithm also applies for passive constructions, since if actor is not a direct core argument then the passive applies which results in the undergoer being the PSA. The next substep in the semantics to syntax linking algorithm as developed in Van Valin (2005: 136) and specifically described for English will be described in (19) below.

The algorithm in (19), developed based on Van Valin and Diedrichsen (2006) for English, is straightforward in some respects, if the number of argument slots equals 1. In these situations, the PSA is assigned nominative case via default. In general, case assignment is also clear, if the number of argument slots is two, since in situations like these the PSA is assigned nominative and the accusative is assigned to the direct core argument. If a three-place predicate construction occurs, the algorithm checks whether undergoer is assigned to the y-Argument and then it assigns the PSA nominative case. The non-macrorole is accusative, and the dative undergoer. If, however, the y-Argument is under the PSA, it is assigned nominative case. The accusative is assigned non-macrorole and the dative is assigned undergoer. In cases where the first argument of **be-LOC**(x, y) is true then this argument is assigned dative case. If, however, the

situation ‘first_argument_of_be_LOC’(x, y) == true’ is true, then the first argument in this construction receives accusative case. The last substep within step 3 in the linking algorithm given in (15) is detailed in (20).

(19)

```

algorithm step 3b
algorithm step3b
if number_argument_slots == 1 do
    nominative = PSA
end if.
if number_argument_slots == 2 do
    nominative = PSA;
    accusative = direct_core_argument;
end if.
if number_argument_slots == 3 and undergoer = z-Argument do
    nominative = PSA;
    accusative = undergoer
    dative = non-marcorole
else if number_argument_slots == 3 and undergoer = y-Argument do
    nominative = PSA
    accusative = non-macrorole
    dative = undergoer
if first_argument_of_be_LOC'(x, y) == true do
    dative = first_argument_of_be_LOC'(x, y);
end if
if first_argument_of_BECOME/INGR_be_LOC'(x, y) == true do
    accusative_case = first_argument_of_BECOME/INGR_be_LOC'(x, y);
end if

```

(20)

```

algorithm step3c1
access state of the world && database
if simple_present == true or past_tense == true do
    new.agreement_nucleus();
end if.
if complex_tense or passive or copula_construction do
    new.agreement_auxiliary();
end if.

algorithm step3c2
new.nominal_agreement;

```

The algorithm in (20) realizes verb agreement and nominal agreement. The crucial point with respect to agreement in verbs is how both the nucleus and the auxiliary are determined. This is difficult, since, as pointed out in Van Valin (2005), in these steps apply to logical structures rather than to fully populated sentences. If, however, an LS is

populated with verb agreement, there is no way to determine whether an auxiliary occurs or not. In English, auxiliaries are used in complex tense forms, passive or copula constructions, while nucleus agreement occurs in sentences which are simple present, or past tense. The use of different tense forms heavily depends on language external considerations or on internal states coded by database access. This means the intelligent agent used in this framework needs to have access to internal and external states of the world and a database with world knowledge in order to determine the assignment of tenses, since this assignment is not governed by language internal considerations. In a full-fledged framework in which an intelligent software agent is used it is necessary to account for the interaction of language external considerations, which are realized in the grammar. A semi-formal representation of the algorithm in step 4 for English is given in (21).

(21)

```

algorithm 4
access state of the world && database && DRS
if topicalization == true or W_question == true do
    new.PrCS_template();
end if.
if embedded_clause == true do
    new.subordinate_clause_template;
end if.
if core_template == true do
    syn_arg_slots_core == num_dist_spec_arg_pos_sem_rep_core;
    min_core_valence = 1;
if passive == true do
    syn_arg_core_slots_passive = syn_arg_core_slots - 1;
end if
if PrCS == true or PoCS == true do
    syn_arg_slots_core_prepost = syn_arg_slots_core - 1;
end if
if nucleus_template == true do
    branching_template = non_finite_nuclear_auxiliary,
else
    new.non_branching_template(),
end if.
if RP == true and pronoun do
    new.pronoun_template();
end if.
if RP == true and common_noun == true do
    new.common_noun_template();
end if.
if RP == true and proper_noun == true do
    new.proper_noun_template();
end if.
if adjunct_modifier == true do
    new.periphery_template();
end if.

```

In the semi-formalized algorithm in (21), the referring syntactic templates are assigned to the referring expressions in the LS. Thereby, a syntactic structure is constructed from the LS. First, it is checked whether a topicalized clause or a W-question is used and the PrCS template is assigned. As pointed out in Van Valin (2005), sentence types are determined based on the operator projection. However, what is important in this part of the algorithm is that the sentence type is basically determined by external consideration. The intelligent software agent needs to take mental states and external states derived from the environment and a database with world knowledge into consideration. If these cognitive mechanisms were modeled within a computational framework, it would be possible to account for the determination of sentence types in this step. Also, the algorithm in (21) tests whether an embedded clause is found, so that a subordinate clause template is activated.

If a core template occurs, the number of distinct argument positions in the logical structure is assigned to the number of syntactic argument slots. The minimal core valance is 1. Also, in the algorithm in (21) it is checked whether a passive sentence occurs. In fact the occurrence of a passive can be determined by the application of auxiliaries. However, as pointed out above the question is how the application of passive is determined. Basically, the application of passive can be regulated by mental states or by information from the environment, which is not accounted for in the algorithm developed in Van Valin (2005), but which I have accounted for by the access from states in the world and the database. If, however, a mechanism is found to determine whether passive occurs, the number of syntactic argument slots is reduced by 1 and assigned the number of syntactic argument slots in the passive.

Since the PrCS basically occurs in cases of topicalization in English, information structure plays an important role in its application. This means that a mechanism for the determination of topic and focus needs to be applied in the linking algorithm in order to fit for cases of topicalization. In fact, RRG provides a mechanism to account for the determination of topic and focus, which however is highly underestimated in the framework developed in Van Valin (2005) and Van Valin and LaPolla (1997). This mechanism employs discourse representation structures that can be used in order to determine topic and focus. These structures will be employed in the course of this paper in order to account for three-place predicates, which this paper mainly focuses on. I have already accounted for this by accessing DRSs in the first step of this algorithm.

Also in the semi-formalized version of the semantic to syntax linking it is tested if a nucleus template occurs, since if this is the case the non-finite nuclear auxiliary is assigned a branching template and otherwise a new non-branching template is called and by this constructed. What is crucial with respect to the occurrence of these templates is that these templates can only occur if it is already determined whether a nucleus template occurs.

If it is true that an RP or a pronoun occurs, a pronoun template is constructed. If, however, an RP and a common noun occur, a new common noun template is constructed. The same mechanism occurs with proper nouns, since a proper noun template is constructed if an RP and a proper noun occur. These mechanisms indeed work in a semi-formalized version of the linking algorithm, since based on the LS it is possible to determine RPs as well as pronouns, common nouns and proper nouns.

Also, the intelligent software agent tests whether an adjunct modifier occurs, which can be determined based on the LS. If such an adjunct modifier occurs, a new periphery template is constructed.

In (22), the pseudo-code semi-formalization of step 5 in the semantics to syntax linking is given:

(22)

```

algorithm step5
nucleus = predicate;
complete_nucleus = nucleus + new.operator_projection_template( );
complete_clause = nucleus + clause;
if nucleus == finite and topicalization == false do
    complete_nucleus = nucleus + second_position_core;
else
    complete_nucleus = nucleus + second_position_core
end if.
if focus_structure_restrictions == true do
    core = +1.argument == true
    new.place_arguments_before_first_argument_in_Core;
end if.
if remaining_elements == true do
    if pronoun == true do
        remaining_positions = pronoun
    end if.
    if RP > PP == true do
        remaining_positions = RP
    else
        remaining_positions = PP
    if accusative == pronoun do
        remaining_positions = accusative;
    else
        remaining_positions = dative
    end if.

```

The algorithm in (22) assigns arguments to the positions in the syntactic representation of the sentence based on the selection of syntactic templates in (22). The operator '+' is used to denote the concatenation of syntactic templates. The first step in this algorithm is the assignment of the predicate in the LS to the nucleus in the syntactic template. What is important is that the operator projection is generated and assigned to the nucleus template in order to generate the complete nucleus. The clause on the other hand is a combination of the nucleus and the clause. These conjugations are indicated by the use of the operator '+' in order to account for concatenation of syntactic templates. In the next step, the nucleus is assigned to the second position of the core. This step takes place if the nucleus is finite and if no case of topicalization occurs. If, however, this statement is false and therefore a case of topicalization occurs, the nucleus is assigned to the second position, too. This statement seems strange, however if a case of topicalization occurs then an RP occurs in the first position of the clause,

while the topicalized element is in the PrCS. As shown in the algorithm in (22), the remaining positions in the clause are assigned after the nucleus is occupied. In these situations it is first tested whether a pronoun exists. If this is the case, the pronoun is assigned to the remaining positions in the clause. If, however, the statement 'RP > PP' is true, then RP is assigned to the remaining positions. Otherwise the PP is assigned to the remaining positions. In cases where an accusative pronoun occurs, the accusative pronoun is assigned to the last remaining positions. If however this is not true, dative is assigned to the remain positions.

What has been shown in this section is that the linking algorithm as developed in Van Valin (2005) faces numerous problems when it is semi-formalized and executed on a RAM, since in various situations the algorithm developed based on Van Valin (2005) is way to coarsely grained and should mainly be understood as a guiding principle rather than as a formal grammar. This in fact means that RRG in its current state is not computationally adequate, since it is not possible to execute a semi-formal version of the algorithm on an abstract machine model. Based on the Church-Turing thesis described in section 1, it results in an algorithm which is neither executable on a machine nor is intuitively executable, if one accepts Turing's strict perspective on cognition as it is done in this framework with respect to application and executability of algorithms. In the following sections I will introduce constructional schemas as developed in Nolan (2011).

6. Constructional schemas

As shown in section 5, RRG cannot account for variable undergoer linking, since either a high degree of cognitive indeterminism needs to be applied in terms of the function *choice*, in which a coarsely grained fuzzy algorithm leads to the necessity of mapping cognitive images of internal mental states or images of the external world in order to model the outer world in terms of logical structures. As pointed out in section 1, RRG is developed in order to be a psycholinguistically adequate theory. This is also shown in Van Valin (2007). In this paper, Van Valin shows that RRG can easily be plugged in into a psycholinguistic model of language processing. However as shown in sections 4 and 5, RRG is neither empirically adequate with respect to three-place predicates, since considerations of the architecture of RRG are not properly accounted for, as the analysis of three-place predicates in Van Valin (2007) is not in accordance with native speaker intuitions. Nor is RRG computationally adequate, since the algorithm cannot account for three-place predicates in its coarsely grained and fuzzy form. Therefore, in this section I will introduce a new way of analyzing three-place predicates in RRG which is both empirically adequate and computational adequate.

The point is RRG wants to be a cognitive functional linguistics theory, as proposed in Van Valin (2006) as well as in Van Valin and LaPolla (1997). This paper wants to develop a theory of RRG which is computationally adequate and therefore implementable in applications based on artificial intelligence and psycholinguistically adequate. However, as pointed out by Langer (personal communication), from a philosophy of science point of view the fuzziness proposed in the RRG linking algorithm, which employs the application of a cognitive endowment - not further developed - as well as the metaphorical use of the term *algorithm* in Van Valin and LaPolla (1997) and Van Valin (2005) and more recent work on RRG, as well as the omission of details, its reduction to small amounts of data, is at least potentially

immunized against falsification. However, this means that RRG as a cognitively functional linguistic model is more unscientific as it could potentially be. To account for three-place predicates in RRG, I will use constructions as grammatical models, as developed in Nolan (2011). Nolan (2011: 2) notes that there is current linguistic evidence, suggesting that to a certain degree, grammatical knowledge is organized in constructions and that these constructions may include both information about form and function/meaning. In RRG, constructions are used to capture idiosyncratic linguistic behavior and are stored within the syntactic inventory of constructions within the grammar (cf. Van Valin 2005). As pointed out by Nolan (2011: 3), this position in RRG is different from the one in other linguistic models that take constructions as more mainstream and central in form of a mapping function between both form and meaning. Van Valin (2005: 131f) characterizes construction as follows:

RRG recognizes the importance of grammatical constructions, and they are represented in terms of constructional schemas. Cross-constructional and cross-linguistic generalizations are captured in terms of the general principles and constraints that constitute the linking algorithm, e.g. the actor-undergoer hierarchy, the layered structure of the clause, the privileged syntactic argument selection hierarchy. Only the idiosyncratic, language features of constructions are represented in constructional schemas. Hence constructional schemas, by virtue of their reference to general principles, permit the capturing of cross-linguistic generalizations, while at the same time expressing language particular properties of grammars. (Van Valin 2005: 131f)

Nolan's (2011: 3) critique basically is that RRG seems to have a robust position of constructions, while it still retains the centrality of the lexicon within its lexicalist projectionist framework. However, the point is, the idea of the RRG notion of constructions in a syntactic inventory in which constructions only contain idiosyncratic information is oddly reminiscent of Chomsky's (1957) early idea of the lexicon as a depository of information on verbs with odd behavior in which these are listed. With respect to Van Valin's (2005) idea of a syntactic inventory Nolan (2011: 3f) explains the following:

Indeed, the very term 'inventory' suggests a list. One can make an inventory, a list, of one's books. But, to find an item on this inventory, as it is a list, one would need to search it sequentially. This traversal of a list / inventory would therefore proceed to be searched in sequence, for example almost like searching a phone book page by page / entry-by-entry from the beginning to find 'N' for 'Nolan'. (Nolan 2011: 3f)

Following Nolan (2011: 4), this notion of an inventory is not cognitively plausible, since this way a significant processing overhead, retrieval latency lag and a retrieval delay is created. However, this is not what happens cognitively, since construction retrieval as well as construction processing is immediately and blindingly fast (cf. Nolan 2011: 4). This is the reason why Nolan (2011: 4) proposed a different model for the storage of information on constructions which he describes as follows:

On the other hand, a 'repository' is a place where objects are stored, such that one can find them again. A repository is construed as a database of objects, indexed according to some criteria. In the real world, we can easily understand that a library is a repository of book indexed according to the Dewey numerical categorization system that is searchable; you can find your library item again and again, according to some search criteria. Under this view, we posit that instead of a syntactic inventory, a construction schema repository is therefore a database of constructional schemas. My hypothesis is that a construction is a unique grammatical object identified by a constructional SIGNATURE, which uniquely identifies the correct constructional schema in the repository. The construction can be directly retrieved using the constructional signature, which acts as its identifying retrieval key. (Nolan 2011: 4)

In his approach Nolan (2011: 4) assumes an isomorphic (1:1) relationship between the constructional signature and the schema stored in the repository facilitating the immediate discovery of the relevant and correct schema. Nolan (2011: 5) describes his idea as follows:

Once the schema is retrieved from the construction repository, an instance of it is activated in the mind to process the syntactic input string (the input clause in the syntax semantics direction, for example) to produce an output logical structure. The internal linking within the construction is then activated as an executable process to map between syntax-morphology-pragmatics-semantic etc., according to the internal specification of the particular construction.

In fact, the idea of the construction repository as developed in Nolan (2011) is that it has an internal structure as rich and complex as the lexicon. This repository also contains a structured relationship with the lexicon in which the constructions within draw on information, which is stored in the lexicon. This means the syntactic inventory is a 'database' of constructional schemas which are not stored as a simple list, but has a means to enable an appropriate constructional schema, which can be retrieved as required to certain criteria, and activated (cf. Nolan 2011: 5). The constructional schema proposed in Nolan's (2011: 6) approach has a constructional signature based on syntactic information, in this case a syntactic pattern of occurrence. With this signature it is possible to find the construction stored in the construction repository. The assumption is that in the same way each constructional schema is unique, each constructional signature associated with the schema body is unique. If an input sentence meets the pattern of occurrence with the construction signature, a constructional schema is selected from the construction repository (cf. Nolan 2011: 6). Nolan (2011: 7) describes the application of the construction signature as follows:

We must also recognize that a construction has an input. For example, from syntax, a clause is received for processing when the construction instance is activated following a schema retrieval based on the uniquely identifying signature match. Once the construction instance is activated and the various criteria at the syntax-semantic-pragmatic interfaces are applied within the construction is generated. This will deliver, assuming a construction executing in the syntax-semantics direction, a rich populated logical structure.

If the direction is from semantics to syntax, the idea is that the construction instance, which is once activated, will generate a well-formed clause as output in the target language. This works according to the principles of the particular construction. What is crucial is that it is possible to consider that the construction has a unique signature and an input string, which is processed in the activated schema, as well as an output of a particular kind (cf. Nolan 2011: 7). The exact nature of these depends on the direction of the execution of the schema, which has been activated before, either from syntax to semantics or from semantics to syntax. Also of importance is the construction body, which encodes the relationship between morphosyntax, semantics and pragmatics appropriately (cf. Nolan 2011: 7). Based on this, Nolan (2011: 7) concludes the following:

Within this perspective, we can usefully consider the construction as a type of grammatical object that can be uniquely identified, has internal structure, accepts an input and produces an output. The execution of the construction schema instance is sensitive to the direction of application, as we mentioned.

Based on Nolan (2011: 7) a construction has the following architecture as a structured grammatical object as shown in figure 5 below:

Signature: some pattern of [... x ₁ , y ₂ , z ₃ ...]
Input: clause (token ₁ , token ₂ , ... token _n)
WORKSPACE: input [1], [2], [3] and output [1]
Construction body Syntax: PSA Semantics: Linking Morphology: Prosody: Pragmatics:
Output: [LS]

Figure 5: Architecture of constructional schemas

The question to be asked with respect to the application of grammatical schemas and the linking roles that apply is, according to Nolan (2011: 7): *Where do lexical and/or grammatical rules reside?* The question is whether they are associated with constructions or whether there is some part of the model where they need to be stored for their activation as required. Also an important question is if the rules reside within the lexicon in a principled way. In Van Valin (2005: 161), a ‘workshop’ is proposed which according to Nolan (2011: 7) is some kind of cognitive factory for linguistics in the mind. In his approach to constructional schemas Nolan (2011: 7) however locates this concept as a motivated processing space, which he calls ‘workspace’. The workspace is located in each construction with a robust computational capability (cf. Nolan 2011: 7). The idea is that the relationship between the lexicon and the syntactic inventory of constructions and the application of various rules are applied within the workspace. In the workspace particular linguistically significant output is produced by the application of rules, which are construed as a generalized way for RRG to differentiate between information stored in the lexicon, versus what is actually computed on the fly in real-time speech act production (cf. Nolan 2011: 8). In this context, Nolan (2011: 8) notes the following:

Therefore, if one takes this, the online computation of the various constructions in real-time at speech act production, as something that needs to be accounted for, the question is where do the processing rules that are applied in real-time online computation reside. Obviously, of course, these reside in human memory and have real-time access to a processing workspace. This leads one to consider that an account of real time online computation of the speech act could be motivated as residing within the construction instance that is retrieved from the construction repository and activated each time as a ‘live’ grammatical object, for each construction.

In this approach to constructional schemas, each construction has its own workspace, which stores the internal linking processes of the construction store in the construction itself. This means the construction linking processes both address and manipulate this workspace in a principled way (cf. Nolan 2011: 8). What is crucial in this account is that the workspace is local with respect to the construction in question. This way it is an intrinsic part of the internal structure of a particular construction. It is partitionated

according to the needs of the construction (cf. Nolan 2011: 8). With respect to ditransitive constructions, Nolan (2011: 8) explains the following:

For example, a ditransitive construction will have a workspace with capacity to store the abstract information requirements of the three arguments and all language specific relevant features, for example, such that these can be accessed and processed within the construction as part of the mapping between form and meaning for the computation of meaning. This would allow slight clausal differences across constructions to be computed, for example, in the way construction of English, ‘he danced his way ...’, ‘the car made it’s way ...’, ‘musicians went on their merry way’, etc., within the ‘x:NP₁ V-ed PN way’ construction.

In the next section, I will introduce typed feature structures as means for the storage of linguistic knowledge and will also show how thematic relations are stored in terms of typed feature structures in an inheritance network of lexical semantic relations and a well defined number of thematic relations. They account for the fact that macroroles as used in Van Valin (2005) are epiphenomenal in fact. Since constructional schemas developed in Nolan (2011) are mainly concentrate on the syntax to semantics linking I will change Nolan’s account to constructional schemas in some way in order to fully account for Nolan’s idea of the construction as grammatical object. This will be done in section 8.

7. Typed feature structures and the storage of thematic relations in RRG

In this section, I will develop a revised version of the theory of the mental lexicon developed in Gottschalk (2010a, 2011b) in order to show that macroroles as proposed in Van Valin (2005) are in fact epiphenomenal, since they are rather attributes in typed feature structure than independent constructs in the grammar. Evidence for macroroles being epiphenomenal comes from the constructional schema, developed in Nolan (2011), which does not refer to macroroles in any way. Instead, the constructional schema developed in this section uses thematic relations. The use of typed feature structure representations is reasonable within a computationally adequate version of RRG, since this structure uses two paradigms found in computer science: the first one is the object-oriented approach in which complex, recursive and - in the case of the typed feature structures developed in this section - nested objects are used, which are attribute-value restrictions and constraints as well as multiple inheritance. The second paradigm from computer science employed in typed feature structures is the relational programming approach, which is declaratively, uses logical variables and non-determinism, in which backtracking can take place and existential query evaluation is possible²⁶. This way, RRG becomes computationally adequate and both the aim of formalization and implementability can be achieved.

In Gottschalk (2010a, 2011b), an account of a unification-based theory of the mental lexicon within RRG is given (cf. Gottschalk 2011b). The architecture of the RRG-lexicon as proposed in Gottschalk (2010a, 2011b) is based on the concepts used in the programming language DATR, which was invented by Evans and Gazdar (1996) in the late eighties, used for the representation of lexical knowledge of various domains (mostly applied in morphology and phonology) (cf. Gottschalk 2010a 33). These inheritance hierarchies and inference rules apply within the lexicon. In this architecture, individual elements in the networks, as well as the various inheritance hierarchies, are

²⁶ <http://www.ims.uni-stuttgart.de/~emele/TFS.html>

connected to each other (cf. Gottschalk 2010b: 33). This concept is very similar to inference rules in DATR, which can be monotonic on the one hand and non-monotonic on the other. Based on this conception, the issue of how Aktionsarten in general and particularly verbs of motion, with their various alternations, are structured and how they are stored in an RRG-compatible lexicon is investigated. A time-line model for RRG-Aktionsarten, based on Reichenbach (1947), which was developed in Gottschalk (2010a, 2011b) to give a description of the internal structure of events assumed within RRG. It is assumed that human knowledge is often represented in terms of inheritance networks and, therefore a model of inheritance networks to modify the present account to the lexicon in RRG is used. For this purposed inheritance networks of the operators constituting the logical structures for Aktionsarten in RRG are developed, as well as an inheritance work of the RRG-Aktionsarten, which shows how they are stored in the mental lexicon. Also, an inheritance network of the specific selection properties of a domain of verbs is developed and it is argued that verbs are stored in so-called Neighborhood Clusters. These neighborhood clusters form a further inheritance network which is stored in the mental lexicon. In Gottschalk (2010a, 2011b) it is argued that basic Aktionsarten in the lexicon are not needed, if Aktionsarten are analyzed and decomposed and operators are stored in terms of inheritance networks. It is also shown in Gottschalk (2010a, 2011b) that multiple lexicon entries for verbs are not needed, even if they occur in a multitude of contexts with different Aktionsart readings, if the idea of inheritance network is accepted. In this case, the concept of a workshop module and lexical rules as suggested by Van Valin and LaPolla (1997) and Van Valin (2005) is not needed (cf. Gottschalk 2010: 20).

Lexical entries, or lexical fingerprints as they are called in Gottschalk (2010a, 2011b), constitute a lexical semantic structure which contains a linguistically relevant subset of inheritance relations among elements in a semantic neighborhood cluster, referring to a specific semantic domain of verbs. The semantic neighborhood cluster of a specific domain of motion verbs inherits selectional properties of verbs from an inheritance network of selection properties. These properties refer to the manner of the specific verb domain and it shows the semantic restrictions of arguments a verb can take. In the model proposed in this paper I will use typed feature structures to capture lexical semantic information in order to represent how lexical semantic relations, which in some way refer to thematic relations of classes of semantic roles, are stored and how they interact with the selection property network of a specific verb domain and the lexical fingerprints of a domain of verbs. These semantic relations are represented in terms of typed feature structures. In these typed feature structures, a set of thematic relations similar to macroroles as proposed in Van Valin (2005) is used, which are merely attributes depending on a configurational approach to the assignment of thematic relations in the constructional schema, introduced in section 6, rather than having an independent status in the grammar as proposed in Van Valin (2005) (cf. Davis 2001: 76).

The lexical semantic representations for both lexical entries for verbs and lexical semantic representations for semantic roles developed in this section are stored in multiple non-monotonic inheritance networks. Davis (2001: 76) describes this idea as follows:

This provides the basis for relating semantically defined classes of verbs to one another, with the members of a given class sharing a portion of their semantics and not needing to specify semantic commonalities in each individual lexical entry. The advantages of a hierarchy of semantic

relations with proto-role attributes are that the overriding effect of certain entailments, such as causation, can be readily modeled, and that they provide a level of representation distinct from surface argument structure and transitivity suitable for stating linking principles. (Davis 2001: 76)

In this new approach, lexical semantic relations which describe the argument structure of verbs belonging to a specific lexical domain containing thematic relations are stored in a separate inheritance network. I will assume six typed feature structures for these relations, which are associated with the configurational approach to the assignment of thematic relations. For this I will introduce a number of thematic relations used in this specific approach which are stored as attributes in typed feature structures. These thematic relations are defined based on a generalization of possible arguments a verb can take. Thereby, the thematic relations assumed in this new approach to RRG differ from semantic macroroles, as assumed in Van Valin (2005). The thematic relations I will use in this account of RRG are structured in terms of lexical semantic representations, in terms of relations as typed feature structures and they can be viewed as partially decompositional. In this case their decomposition can be confined to a few elements, such as the causal structure of the semantics of a predicating element (cf. Davis 2001: 76).

In this section I will analyze a domain of transfer verbs in English. Many, but not all of these verbs are three-place predicates. Some of them are generic three-place predicates and some of them are verbs which can have a valence alternation. The domain of verbs I will investigate are the following: *put, give, load, present, lend, sell, receive, buy, borrow* and *transfer*. In a first step I will develop a neighborhood cluster for this domain of verbs to which I will refer to as transfer verbs. A neighborhood cluster consists of verbs belonging to a closed domain of very specific verbs. In the case of this study I will be concerned with transfer verbs. This class of verbs can be determined by a number of decomposition tests. The granularity of district clusters depends on the granularity of these tests. In this paper I will develop a rather coarsely grained neighborhood cluster of transfer verbs, which is mainly used for the purpose of representing how neighborhood clusters of verbs should interact with typed feature structures of lexical semantic relations. A decomposition of the domain of transfer verbs investigated in this paper is given in (23):

(23)

- a. Abby **gave** the file to McGee.
[do'(x,)] CAUSE [PROC & INGR have'(y, z)]
Transfer of a non-abstract entity realized as z argument from x argument in LS to y argument in LS. Neutral way of causing the y argument to have the z argument.
- b. Abby **presented** Gibbs with a coffee.
Abby **presented** a coffee to Gibbs.
Transfer a non-abstract entity realized as z from x argument in LS to y argument in LS. The agent gives the theme to the participant as a present for free.
- c. Abby **put** the piece of evidence on the desk.
[do'(x,)] CAUSE [PROC & be-LOC'(y, z)]
Transfer a non-abstract entity realized as z from x argument in LS to y argument in LS as change of location.
- d. Gibbs **loaded** the truck.
Gibbs **loaded** the gun into the truck.
Gibbs **loaded** the truck with the gun.

- [do'(x, [be-LOC'(x, y))]
 [do'(x,)] CAUSE [PROC & INGR be-LOC'(y, z)]
 Abstract entity realized as z from the x argument on the z argument.
- e. McGee **lent** Abby the computer game.
 McGee **lent** the computer game to Abby.
 [do'(x,)] CAUSE [PROC & INGR have'(y, z)]
 Transfer a non-abstract entity realized as z LS from the x argument to the y argument. Y-Argument receives z argument from x argument without transferring the ownership of z argument.
- f. Gibbs **sold** his gun.
 Gibbs **sold** Agent Franks his gun.
 Gibbs **sold** his gun to Agent Franks.
 [do'(x, [sell'(x, y))]
 [do'(x,)] CAUSE [PROC & INGR have'(y, z)]
 Transfer a non-abstract entity realized as y argument from x argument transfer of a non-abstract entity realized as z argument in LS from the x argument to the y argument. Describes a monetary transfer of a y z argument from x argument to y argument in cases of occurring as three-place predicates. Otherwise monetary transfer of y argument from x argument.
- g. Tony **received** the file.
 Tony **received** the file from Siva.
 [do'(x, [have'(x, y))]
 [do'(x,)] CAUSE [PROC & INGR not-have'(y, z)]
 Transfer of a non-abstract entity realized as y argument to x argument transfer a non-abstract entity realized as z from y argument to x argument. Describes a neutral way of transfer to x argument.
- h. Tony **bought** a playstation.
 Tony **bought** a playstation from McGee.
 [do'(x, [have'(x, y))]
 [do'(x,)] CAUSE [PROC & INGR not-have'(y, z)]
 Transfer of a non-abstract entity realized as y argument to x argument transfer of a non-abstract entity realized as z argument from y argument to x argument. Describes a monetary transfer of a y argument z argument to x argument from y argument in cases of occurring as three-place predicates.
- i. Siva **borrowed** a motorbike.
 Siva **borrowed** a motorbike from Tony.
 [do'(x, [have'(x, y))]
 [do'(x,)] CAUSE [PROC & INGR not-have'(y, z)]
 Transfer of a non-abstract y argument to x argument transfer of a non-abstract entity realized as z argument from y argument to x argument. Describes a transfer of a y argument z argument to x argument from y argument in cases of occurring as three-place predicates without transferring the ownership of y z.

Based on the decompositional analysis in (23), it is possible to conclude that in the neighborhood cluster of transfer verbs analyzed in this paper, a transfer from one destination to another destination takes place. In this way, it is possible to analyze transfer as some kind of abstract motion. This is also pointed out in Davis (2001: 113). The transfer direction is either $x \rightarrow y$ or $y \rightarrow x$. The transfer direction (23a – 23f) is $x \rightarrow y$. In (23g – 23i) on the other hand the transfer direction is $y \rightarrow x$. As can be seen in these examples, too, the theme of the transfer referred to as z argument in the logical

structure usually is a non-abstract entity like a book, or a computer or a file for example. Also, the x variable in these transfer verbs is usually animate, while the z variable, which is the theme in these examples, needs not necessarily be inanimate. This is the case in *Abby gave McGee the dog*. In this case *the dog* is animate, however in comparison to *Abby the dog* it is less volitional. This is also the case with respect to *McGee*. As can be seen in the examples in (23), all verbs of this domain share some features, however, they also differ in some respects. Each verb has some properties which differentiate it from other verbs in this domain. What will be of importance for the development of a neighborhood cluster of transfer verbs is that the differentiating feature of transfer verbs is [+ moving direction of transfer]. The verbs (23a – 23f) have the feature [+ moving direction $x \rightarrow y$] while the transfer verbs in (23g – 23i) have the feature [- moving direction $x \rightarrow y$]. I will refer to these features of transfer verbs as ontological feature nodes (cf. Gottschalk 2010a, 2011b).

In comparison to the motion verbs analyzed in Gottschalk (2010a, 2011b), there is no neutral way of transfer in this neighborhood cluster, since transfer is always bound to a specific direction. This means a neighborhood cluster for English transfer verbs as an abstract node, which is called transfer and needs to be differentiated from the verb *transfer*. This way, a neighborhood cluster of transfer verbs in terms of an inheritance network looks as follows:

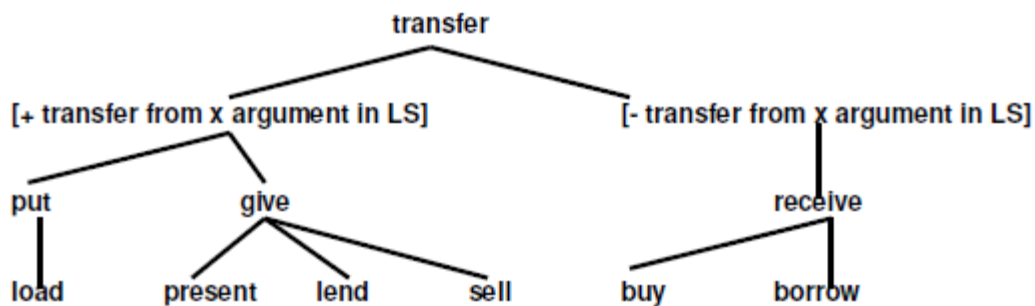


Figure 6: Neighborhood cluster of motion verbs

In this neighborhood cluster the ontological feature nodes, which in a full-fledged theory of the mental lexicon in RRG are semantically defined in a database, inherit everything from the abstract root node, which is connected to the selectional property network, which will be developed in the next step. The verbs *put* and *give* both inherit from the ontological feature node [+ transfer from x argument in LS]. The verbs *put* and *give* have a different status in this network, since *put* refers to a locational transfer of a non-abstract entity, in which in the logical structure the location is decomposed as **be-LOC'**, while *give* refers to the transfer, which is focused on a recipient, which is also a location, but not in the sense of being loaded on a truck or put on a shelf. The verbs *present*, *lend* and *sell* all inherit from *give*, which is more neutral with respect to the transfer. This is because in *present* the transfer is for free, as a present, which is semantically coded, while *lend* refers to a transfer which does not change the ownership of the entity. In *sell* on the other hand, ownership changes, therefore it inherits from *give* and differs from *lend*. The verb *receive* on the other hand inherits from the ontological feature node [- transfer from x argument in LS] and is a neutral way of transfer of the $y \rightarrow x$ direction, as pointed out in the decomposition in (23). *Buy* on the other hand inherits from *receive*. Also, it has some additional features of change with respect to ownership, which is in fact a monetary transfer. The verb *borrow* has the

same status as the verb *lend*; with respect to ownership, however, it describes the $y \rightarrow x$ direction, and therefore inherits from *receive*.

Before I will develop the lexical fingerprints of these transfer verbs, I will develop a selectional property network, which is based in the decompositions in (23). From the decompositions in (23) it is possible to derive a general logical structure [GLS] for the domain of transfer verbs under investigation, which contains internal variables, which mark the semantic differences in verbs of a specific neighborhood cluster. As pointed out in Van Valin and LaPolla (1997: 117), it is possible to unite all verbs of a specific lexical domain in a single general logical structure. The differences in verb meanings are derived from the way internal variables in the GLS are interpreted (cf. Van Valin and LaPolla 1997: 117). In Gottschalk (2010a, 2011), the model of GLSs developed in Van Valin and LaPolla (1997: 116-8) is adapted. Based on the decompositions in (23), it is possible to determine internal variables which have to be realized in the GLS. A possible GLS for two- and three-place transfer verbs is given in (24):

(24)

- a. [**do'**(x, [**movement.direction**(α).**in**(β).**manner'**(x, (y)))]
- b. [**do'**(x, \langle) **movement.direction**(α).**in**(β).**manner**] CAUSE[PROC & INGR **true.value of.have'**(x, z)]

As has been pointed out above transfer verbs code some kind of abstract motion. I have accounted for this by using **movement.direction**(α) in order to refer to either the movement direction $x \rightarrow y$ or to the movement direction $y \rightarrow x$. The term **in**(β).**manner** refers to the manner in which the transfer takes place. At a first glance this encoding seems to be confusing, but in fact *presenting* for example happens in the manner of giving something to someone for free, which is some kind of abstract manner which is encoded in the verb. (24a) refers to two-place predicates or transfer verbs, which can alternate between being transitive and ditransitive, like *sell* for example. As pointed out in Gottschalk (2010a, 2011b), alternations like these are lexically motivated. In fact, none of the verbs under investigation in this paper are lexically transitive or intransitive. Rather, based on an inheritance process within the lexicon, input from internal or external states and input from information structure send to the lexicon, it is possible to determine whether a verb should be transitive or ditransitive in language processing. I will refer to the use of information structure in this revised and computationally adequate version of the semantics to syntax linking algorithm in the next section.

What is also shown in the decompositions in (23) is that only non-abstract entities are transferred in the verbs in the neighborhood-cluster developed in this paper. However, it is also the case that in some verbs, like *teaching* and *learning*, a kind of transfer takes place. Of course these verbs also belong to the domain of cognitive verbs. However, since in these verbs a kind of transfer also takes place, verbs like *teach* and *learn* would inherit properties from two district clusters in some non-monotonic way. This is also pointed out in Davis (2001). Based on these findings, it is possible to develop a selectional property network of transfer verbs given in figure 7. It is important to note that the Selectional-Property-Network [SPN] is a superior network, which is connected to the specific Neighborhood cluster. The concept of SPNs refers to the notion of district clusters, which refer to verbs of a specific lexical category like transfer (cf. Gottschalk 2010a: 36). In fact there exists a big number of other district clusters, for example the district cluster of emotion verbs or of cognition verbs. As can be seen in

figure 7, the SPN for transfer verbs has an abstract node called *transfer*, which has two nodes *abstract entity* and *non-abstract entity*. These nodes also have some semantics which I will not describe in this paper. The α variable in the GLS is satisfied by some non-abstract entity, which is stored in a separate network not developed in this paper, and which is connected with the nodes in this SPN. Information about these variables is inherited from the selectional property network to the neighborhood cluster of transfer verbs, in which the semantic information of the verbs is stored.

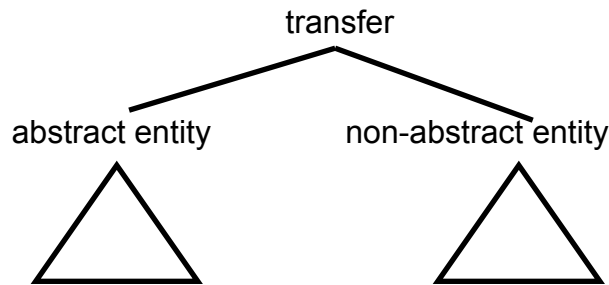


Figure 7: Selectional property network of transfer verbs

In a further network yet to be developed is a network of lexical semantic relations, which are typed feature structures containing information on the possible thematic relations. These lexical semantic relations contain information on the semantic argument structures of verbs, and semantically define the semantic relations realized in a specific verb (cf. Davis 2001). The approach to lexical semantic relations used in this paper is adapted from Davis (2001), but differs in some respects. The basic idea of the use of a network of lexical semantic relations is based on the observations about variable undergoer linking in English three-place predicates. As will be shown in the next section, variable undergoer linking in English is in fact a lexical process pragmatically motivated by information from the information structure, a fact Van Valin (2007) and Van Valin (2005) do not account for. In RRG, it is generally assumed that macroroles, which are a generalization of thematic relations, are part of the grammar. In fact however, semantic roles, which are items of lexical semantic structures, are stored as typed feature structures in the lexicon. The advantages of this account to semantic roles is that variable undergoer linking does not occur randomly based on the AUH, as proposed in Van Valin (2007), but rather based on some lexical and pragmatic processes, taking place within the syntax-semantics-pragmatics interface. However, for a lexically motivated process of variable undergoer assignment it is necessary to somehow place lexical semantic relations, which semantically define which thematic relations are used, in the linking process. Davis (2001) has developed an inheritance network of several lexical semantic relations, found cross-linguistically. This network is given in Figure 9. The inheritance network of lexical semantic relations shows how lexical semantic relations, which describe the argument structure of verbs, are stored lexically and how the different lexical semantic relations inherit information from each other. The root node of this network is *rel*, which is an abstract node referring which node the most basic lexical semantic relations *soa-rel*, *act-rel* and *und-rel* inherit from. A typical *act-rel* would refer to an intransitive activity verb like *sing*, while a typical *und-rel* would refer to a typical intransitive verb, coding a situation where some kind of what Van Valin (2005) would refer to as undergoer occurs, like in *die*. A *soa-rel* on the other hand refers to some intransitive state which is coded within a verb. The other lexical semantic relations referred to in this network are special kinds of the three lexical semantic relations *soa-rel*, *act-rel* and *und-rel*, which I will not discuss

in any detail in this paper. I will rather concentrate on a detailed description of the *receive-rel* and *give-rel*, which are of special interest for the analysis of transfer verbs in English developed in this paper. A representation of these lexical semantic relations in terms of typed feature structures looks as follows (25).

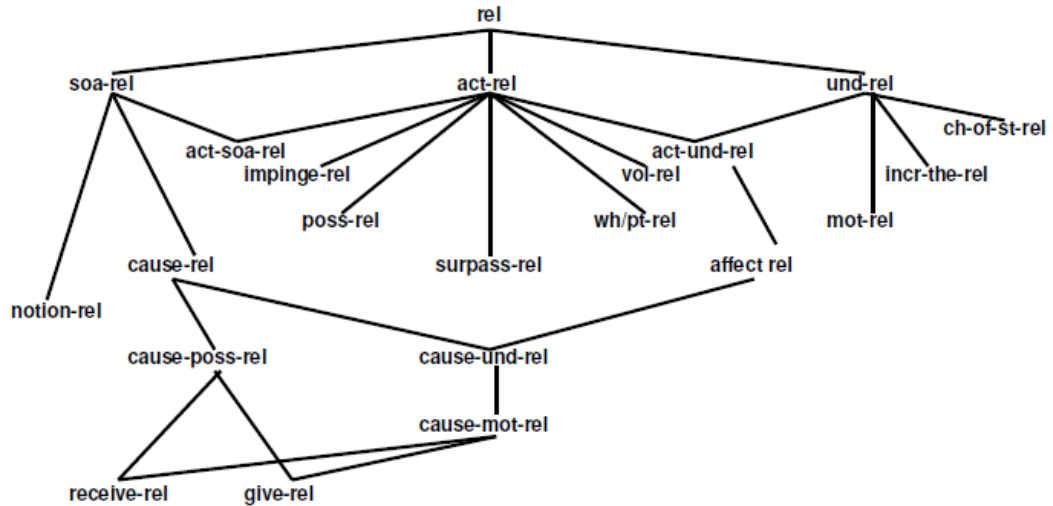
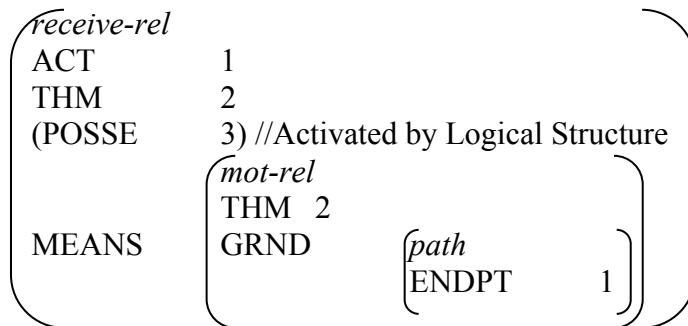


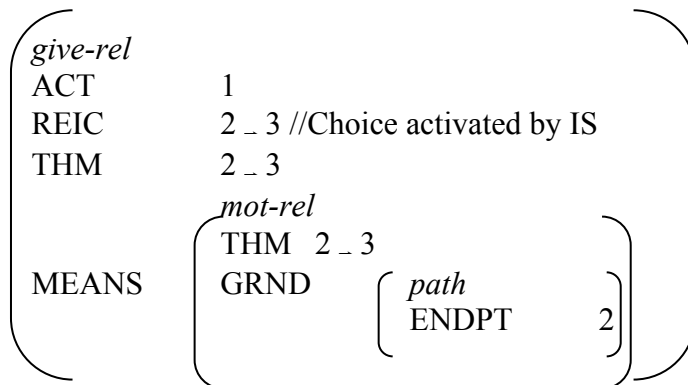
Figure 8: Inheritance network of lexical semantic relations (cf. Davis 2001: 131)

(25) (cf. Davis 2001)

a.



b.



The typed feature structures in (24) are organized as follows: In (24a) there is a constraint referred to by *receive-rel*, which marks the lexical semantic relation. This lexical semantic relation attributes ACT, referring to actor, which in difference to

macroroles, as used in Van Valin (2005), assigns to a thematic relations which has a specific semantic structure which will be described later in this section. The second attribute in this lexical semantic relation refers to theme, which is a thematic relation referred to in the thematic relations continuum in Van Valin (2005: 58). The third attribute in this typed feature structure is POSSE, which refers to possessor, also contained in the thematic relations hierarchy in Van Valin (2005: 58). This attribute is optional, since the same typed feature structure refers to transitive and ditransitive transfer verbs. All these attributes are coindexed and these coindices refer to tokens in the constructional schema for transfer verbs, which will be developed in the next section. These are the tokens which can be activated in the signatures developed in Nolan (2011) and it will be an important task to show how the typed feature structures for lexical semantic relations interact with the constructional schemas developed in Nolan (2011). In fact, tokens in the constructional schemas, coindices in the typed feature structure for lexical semantic relations and variables in logical structures for verbs unify in some way and finally result in an argument in the semantic representation of the clause. In the typed feature structure in (24a), the coindexation is determined, since no case of variable undergoer linking occurs in verbs which inherit from *receive*. The typed feature structure in (24a) has an embedded typed feature structure, which refers to the abstract motion event which takes place in transfer verbs in English indicated by the constraint *motion-rel*. Also, the *motion-rel* has attributes. These attributes are THM for theme and GRND, which refers to the attribute *ground* describing a path transversed by another participant in event (cf. Davis 2001). This thematic relation is not coindexed, since it is an internal thematic relation which is not overtly marked. The *motion-rel* hosts a further typed feature structure called *path*, which refers to the ENDPT referring to endpoint, which in the morphosyntactic realization of the clause refers to the endpoint of a transfer. This is coindexed by 1, which means that it refers to the argument which is coded as token 1 in the constructional schema. The *motion-rel* is in fact a feature of the internal thematic relation MEANS, which refers to the manner in which the transfer takes place and is similar to the manner concept in the GLS.

The architecture of the *give-rel* is nearly the same as the architecture of the *receive-rel*. It also hosts a *mot-rel* hosting a *path* which is a feature of the thematic relation MEANS. However, some of the thematic relations in the *give-rel* are different from the *receive-rel*. As the *receive-rel* the *give-rel* has the thematic relation ACT, which is coindexed with token 1 in the signature of the constructional schema. The *give-rel* has the thematic relation recipient, indicated by REIC and THM. The thematic relation REIC is not found in the thematic relations continuum in Van Valin (2005: 58), but as can be seen here and as already noted by Diedrichsen (2012) and shown in Nolan (2011), there are good reasons to assume that there is a need for the thematic relation theme in RRG. Both thematic relations REIC and THM can be either coindexed with the tokens 2 or 3 in the signature of the constructional schema to be developed in section 6. This indetermined coindexation is necessary, since in *give-rels* cases of variable undergoer linking occur. As will be shown in section 8, information structure plays an important role in variable undergoer linking. Therefore, the lexicon needs to account for this fact by some lexical indeterminism and underspecification, instead of hosting two different types of lexical semantic relations of the same kind. In this new account of semantic to syntax linking in three-place predicates in RRG, it is also necessary to semantically define the attributes referring to thematic relations in the linking algorithm. A semantic decomposition of these attributes is given in (26).

- (26) a. **Actor**
 [+ effecting influencing participants]
 [+ volitional]
 [+ notion perception of participnats]
 [+ forceful contact on
 [+ includes another participant in state or event]
 [+ possesses another participant in state or event]
- b. **Undergoer**
 [- effecting influencing participants]
 [± volitional]
 [+ undergoers a change of state in event]
 [+ incremental theme in event]
 [+ moves with respect to another participant]
- c. **SOA:**
 [+ conceived perceived by another participant in event or state]
 [+ resulting event state caused by event]
 [+ event state necessarily accompanying another event]
- d. **THM:**
 [+ entity which undergoes a change of state in terms of moving]
- e. **REIC:**
 [+ undergoes a transfer process as receiving entity]
 [+ acts as receiving entity in a transfer process]
- f. **MEANS:**
 [+ abstract instrument used to achieve a state or activity]
- g. **GRND**
 [+ path transversed by another participant in event]
- h. **IMP-ON**
 [+ forcefully impinged in an event]
- i. **PART:**
 [+ included in part of another participant in state event]
- j. **INF:**
 [+ inferior compared to another participant]
- k. **POSSD:**
 [+ possessed by another participant in state event]
- l. **ENDPT**
 [+ state of affairs coding the goal of an activity]
- m. **POSSE**
 [+ acts as possessing entity]
 [+ undergoes a transfer as possessing entity]

(cf. Davis 2001)

In (25a – 25l) a semantic decomposition of attributes referring to thematic relations used in the typed feature structures of the lexical semantic relations in (24) is given. In this decomposition, I mainly refer to the positive features defining the attributes in the typed feature structures. It is possible to represent these attributes in an entailment based inheritance network of attributes, which displays an inheritance hierarchy of these attributes. This inheritance network is given in Figure 9. The hierarchical network of attributes in lexical semantic relations describes an entailment based hierarchy of the attributes used in typed feature structures, which shows entailment relations between the different attributes. This network has an abstract node called *attributes of lexical semantic relations* which is the root. This root has four basic attributes *actor*, *undergoer*, *state of affairs* and *means*, referring to the attributes which form a superset

hierarchy of attributes lower in the hierarchy. The attributes GRND, PART, INF, REIC, POSS, POSSE all inherit features from Actor and Undergoer which are entailed in these attributes. The attribute PART inherits from actor, since they entail features of this attribute, while THM and IMP-ON only inherit from Undergoer, since in these two attributes only features of Undergoer are entailed and this way inherited. The network of lexical semantic relations inherits from the hierarchical inheritance network of attributes in lexical semantic relations, since this way attributes in the typed feature structures in the first network are semantically defined via inheritance.

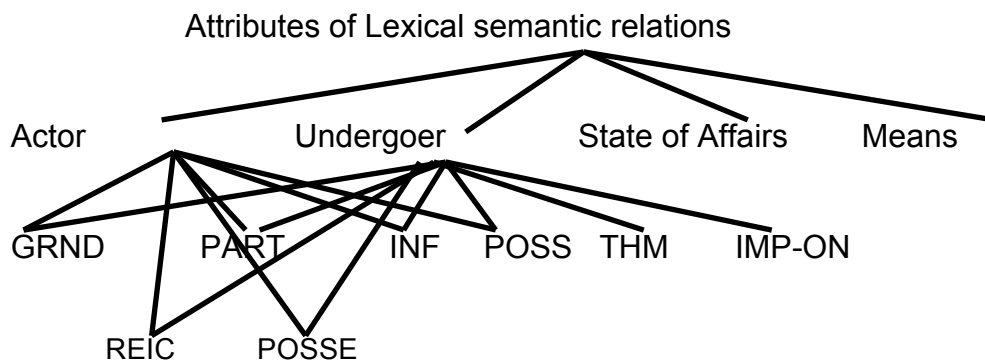


Figure 9: Hierarchical inheritance network of attributes in lexical semantic relations

In the last step of the development of a revised version of the mental lexicon which makes use of typed feature structures, I will develop lexical fingerprints, which in fact form the nodes in the neighborhood cluster of motion verbs, describing the inheritance relations within this neighborhood cluster and how the different networks in this section interact with each other. The neighborhood clusters contain lexical fingerprints as semantic descriptions of the nodes. They are also typed feature structures, since the key concepts of unification-based grammar formalisms are used.

The lexical fingerprints of transfer verbs, which are nodes in the inheritance network of the neighborhood cluster developed in this section, are typed feature structures, too, which make use of unification and contain attributes and constraints. In Gottschalk (2010a, 2011b), lexical fingerprints were not explicitly referred to as typed feature structures, but already in this version of the mental lexicon in RRG they were in fact used as typed feature structures. This inheritance network has an abstract root node, which does not function as primitive. Rather, it is used as an assemblage point for information inherited from the SPN, to spread information to be inherited to the ontological feature nodes and via these nodes to the primitives of the two branches of the neighborhood cluster. Therefore, this assemblage point has no manner qualities. The assemblage point in this network has the following manner of motion qualities: $\langle \text{manner of transfer} \rangle = \text{assemblage point} _ \beta = \langle _ \rangle$. Nevertheless, the assemblage point in this neighborhood cluster inherits all relevant qualities from the non-abstract entity node in the SPN, which on the other hand inherits everything from its mother node of the SPN, which is the abstract primitive of the domain of transfer verbs (cf. Gottschalk 2010: 39).

If a verb inherits all qualities of its predecessor node this is represented by the expression $\langle \rangle = = predecessor^{27}$. In this example, predecessor is used as a variable which can be replaced by the name of the relevant predecessor (cf. Gottschalk 2010: 40). As the root node of a specific neighborhood cluster, the assemblage point contains in its fingerprint the basic information about the selectional properties [SPs] of the domain of verbs under investigation. In the lexical fingerprints, the SPs indicate the content of the variables determined by the GLS (cf. Gottschalk 2010: 40). In case of the verb domain examined here, the SPs look as follows $\langle \text{selectional properties} \rangle = \alpha _ \text{non-abstract entity} _ \beta = \langle \text{manner of motion} \rangle$. The SP is satisfied by the non-abstract entity which is always externally realized. The β -variable in the GLS is satisfied by the reference point $\langle \text{manner of transfer} \rangle$ of the particular transfer verb of the neighborhood cluster. Consequently, it is also a kind of variable, or, in this case, it forms a reference point within the particular fingerprints. It is possible to refer to this behavior as local inheritance within a node, or rather a fingerprint (cf. Gottschalk 2010: 40).

There is also a fourth attribute contained in lexical fingerprints for transfer verbs, which does not occur in the framework introduced in Gottschalk (2010a, 2011b). This node is $\langle \text{lexical semantic relation} \rangle$. Since the root node of the neighborhood cluster is an assemblage point, this attribute looks as follows: $\langle \text{lexical semantic relation} \rangle = = give-rel _ receive-rel$. This means the assemblage point inherits all properties from the typed feature structures of the give- and receive-relation and collects their properties in order to spread them to the other nodes in the network. Nodes lower in the inheritance network can either inherit *give-rel* or *receive-rel* from the assemblage point.

Verbs which are direct daughter nodes of the assemblage point in the neighborhood cluster are basic verbs. In general, basic verbs inherit from the superior ontological feature node, which is connected to a world ontology describing the features of this node in detail. Ontological feature nodes have semantic features, too, which are represented as typed feature structures. As the ontology is based on binary features and in case of transfer verbs the most relevant binary feature is [\pm transfer from x argument in LS], I only use one ontological feature node in this framework. However, if there was a more finely grained semantic description of these verbs, there could in principle be more of such nodes. However, the world ontology of the ontological description nodes is not part of the inheritance network developed in this paper (cf. Gottschalk 2010a: 40). Basic verbs usually inherit all qualities from these ontological feature nodes. In case of this network, ontological feature nodes are [+transfer from x argument in LS] and [-transfer from x argument in LS]. Furthermore, basic verbs have a second characteristic: They inherit the Selectional Properties from the root node, which is the assumable point of the particular domain of verbs. Since the semantic qualities of the assemblage point are not passed on to the ontological description node, the inheritance quality inside of this specific network is expressed by the reference point $\langle \text{selectional properties} \rangle = = transfer$. This can be understood as a global inheritance, which is able to skip nodes, in this case the ontological description nodes. Furthermore, basic verbs contain the reference point $\langle \text{manner of transfer} \rangle$, where their idiosyncratic qualities are determined (cf. Gottschalk 2010a: 40). As pointed out before, basic verbs can either inherit *give-rel* or *receive-rel* from the assemblage point. However, some lexical

²⁷ The structure of typed feature structure of the lexical fingerprints is adapted from DATR (cf. Evans and Gazdar 1996). The terms global inheritance and local inheritance is also adapted from DATR.

fingerprints in this framework have an additional attribute called <pointer to constructional schema> which operates as pointer to a signature in the construction repository, as developed in Nolan (2011). Since this is only pointed, in fact signatures are not stored in the lexical fingerprints; the pointed is rather used as post-it or bookmark, pointing to a signature in the construction repository. This attribute is also inherited from verbs higher in the hierarchy to verbs lower in the hierarchy via default, and therefore only the basic verbs contain this attribute.

The daughter nodes of basic verbs inherit all qualities from their mother nodes, but these sub verbs have their own idiosyncratic qualities as well. This is the reason why they require a <manner of transfer> - attribute, where the qualities of these references, inherited from the basic verb, can be over-written. At this point, the role of non-monotonic inheritance is made clear. The verbs with the most idiosyncratic qualities are farthest below in the hierarchy of this network. The resulting lexical fingerprints in terms of typed feature structures look as follows:

(27)

transfer:

<> == non-abstract entity
 <selectional properties> $y = \alpha _ \neg$ non-abstract entity $_ \beta =$ <manner of motion>
 <manner of transfer> == assemblage point $_ \beta =$ <.
 <lexical semantic relation> == *give-rel* $_$ *receive-rel*.

give:

<> == [+ transfer from x argument in LS]
 <selectional properties> == *transfer*
 <manner of transfer> == Transfer of a non-abstract entity realized as z argument from x argument in LS to y argument in LS. Neutral way of causing the y argument to have the z argument.
 <lexical semantic relation> == *give-rel*
 <pointer to constructional repository> == \wedge [RP_{Actor} V [RP_{Recipient} | PN_{Recipient}] RP_{Theme}]; \wedge [RP_{Actor} V RP_{Theme} [PREPPN | RP]_{Recipient}]; \wedge [RP_{Actor} V RP_{Theme} [PREP indef det N]_{Recipient}]

put:

<> == [+ transfer from x argument in LS]
 <selectional properties> == *transfer*
 <manner of transfer> == Transfer a non-abstract entity realized as z from x argument in LS to y argument in LS as change of location.
 <lexical semantic relation> == *give-rel*

present:<> == *give*

<manner of transfer> == Transfer a non-abstract entity realized as z from x argument in LS to y argument in LS. The agent gives the theme to the participant as a present for free.

lend:<> == *give*

<manner of transfer> == Transfer a non-abstract entity realized as z LS from the x argument to the y argument. Y-Argument receives z argument from x argument without transferring the ownership of z argument.

sell:<> == *give*

<manner of transfer> == Transfer a non-abstract entity realized as y argument from x argument - transfer of a non-abstract entity realized as z argument in LS from the x argument to the y argument. Describes a monetary transfer of a y - z argument from x argument to y argument in cases of occurring as three-place predicates. Otherwise monetary transfer of y argument from x argument.

load:<> == *put*

<manner of transfer> == Transfer a non-abstract entity realized as z from the x argument on the z argument.

receive:

<> == [- transfer from x argument in LS]

<selectional properties> == *transfer*

<manner of transfer> == Transfer of a non-abstract entity realized as y argument to x argument - transfer a non-abstract entity realized as z from y argument to x argument. Describes a neutral way of transfer to x argument.

<lexical semantic relation> == *give-rel*<pointer to constructional repository> == \wedge [RP_{Actor} V [RP_{Recipient} | PN_{Recipient}] RP_{Theme}]; \wedge [RP_{Actor} V RP_{Theme} [PREPPN | RP]_{Recipient}]; \wedge [RP_{Actor} V RP_{Theme} [PREP indef det N]_{Recipient}]

buy<> == *receive*

<manner of transfer> == Transfer of a non-abstract entity realized as y argument to x argument _ transfer of a non-abstract entity realized as z argument from y argument to x argument. Describes a monetary transfer of a y argument _ z argument to x argument from y argument in cases of occurring as three-place predicates.

borrow:<> == *receive*

<manner of transfer> == Transfer of a non-abstract y argument to x argument _ transfer of a non-abstract entity realized as z argument from y argument to x argument. Describes a transfer of a y argument _ z argument to x argument from y argument in cases of occurring as three-place predicates without transferring the ownership of y _ z.

With respect to the connection of the several inheritance networks developed in this section, the interaction of the networks works as follows: The SPN inherits information about Aktionsarten from the Aktionsart network, which is not described in this paper, but discussed in great length in Gottschalk (2010a, 2011b). Information for both Aktionsarten for two-place predicates and three-place predicates is inherited from this network via unification to the SPN, which inherits information about selectional properties to the neighborhood cluster, containing lexical fingerprints in terms of typed feature structures as nodes in the network. The neighborhood cluster on the other hand also inherits information about lexical semantic relations from the lexical semantic relations network via unification. The lexical semantic relations network on the other hand inherits information on the semantic description from the inheritance network of thematic relations.

What is shown in this section is that it is possible to account for both alternations in verb valence and argument structure with respect to semantic relations lexically. Since thematic relations are used in this framework, it is not necessary to use generalized macroroles as proposed in Foley and Van Valin (1984), Van Valin (1993), Van Valin (2005) and Van Valin (2007). In fact, macroroles as used in RRG are epiphenomenal, since if typed feature structures are used and thematic relations are assigned configurationally, no macroroles are needed. This is also in accordance with Diedrichsen (2012). In the next section I will show with respect to my constructional schemas, English three-place predicates for transfer verbs, that macroroles are not needed if constructional schemas, which are necessary in order to account for variable undergoer linking are used, since, as shown in section 5 the semantics to syntax linking algorithm cannot account for variable undergoer linking. Instead of the application of the AUH as proposed in Van Valin and LaPolla (1997) and Van Valin (2005), it is possible to account for the assignment of thematic relations in the constructional schemas if the information from the lexicon is accessed. If one assumes the existence and importance of signatures in constructional schemas, which are accounted for by the pointers in verbs with lexically three-argument positions in the lexicon, or the

possibility to exhibit a valence alternation which refers to constructions stored in the construction repository, macroroles are not needed in RRG and are epiphenomenal. As pointed out in Davis (2001), they are part of the hierarchical lexicon, since otherwise it would not be possible to explain why some arguments can be assigned to some verbs and some cannot. Therefore, thematic relations belong to the meaning of the verb and are in fact a lexical phenomenon. Based on these findings and on the following it is possible to conclude that macroroles are set on top of the idea of the configurational approach used in RRG, as already pointed out in Diedrichsen (2012) This is the reason why macroroles are epiphenomenal and why they are not needed in RRG. In fact there are no advantages in using macroroles, since even the assignments of PSAs do not need the concept of macroroles. The privileged syntactic argument selection hierarchy as described in Van Valin (2005) does not need macroroles, since it is based on a configurational approach operating on the logical structures of verbs. This means, in order to become more economic and therefore meet the internal principle of economy as referred to in section 1, macroroles are not needed. This idea results in a theory which should be unification-based, as this approach to the computability of RRG already shows, and which needs constructional schemas as pointed out in Nolan (2011). In the next section I will introduce an information structure-based analysis of RRG, which makes use of discourse representation structures and constructional schemas as developed in Nolan (2011) which are as pointed out before are revised to a certain degree in order to account for both the semantics to syntax linking and the syntax to semantics linking as discussed in Nolan (2011). This way I will account for variable undergoer linking in three-place predicates which is computationally adequate.

8. Constructional Schema for three-place predicates in English

For the development of a computationally adequate algorithm for an computationally adequate analysis of three-place predicates in the RRG linking from semantics to syntax, which makes use of constructional schemas, the concept of discourse representation structures [DRSs] is also of crucial importance. This is because in variable undergoer linking, information structure considerations play an important role, as pointed out in Van Valin and LaPolla (1997: 423), Diedrichsen (submitted) and Erteschik-Shir (1979). Therefore, in this section I will develop an information structure based analysis of three-place predicates in English. The data used for this analysis is given in (28):

- (28)
- | | | |
|-----|------------------------------|---|
| a. | Abby gave a file to Gibbs. | |
| a'. | Abby gave Gibbs a file. | √ |
| b. | Abby gave a file to the cop. | |
| b'. | Abby gave the cop a file. | √ |
| c. | Abby gave a file to her. | |
| c'. | Abby gave her a file. | √ |
| d. | Abby gave the file to Gibbs. | |
| d'. | Abby gave Gibbs the file. | √ |
| e. | Abby gave the file to her. | |
| e'. | Abby gave her the file. | √ |
| f. | Abby gave the file to a cop. | √ |
| f'. | Abby gave a cop the file. | |

(cf. Erteschik-Shir 1979)

The examples of three-place predicates, which are marked by a hook, are the unmarked cases based on native speaker interviews discussed in Erteschik-Shir (1979). This is not in accordance with Van Valin (2007), where the non-hooked examples are the unmarked cases, since they are in accordance with the AUH, as already discussed in section 4. In this section, I will show that information structure considerations govern the alternations found in the examples in (27) above. In what follows I will analyze the information structure considerations in these examples before developing constructional schemas in order to account for three-place predicate constructions in English. In the context of information structure, RRG differentiates between pragmatic presupposition and pragmatic assertion. Based on Lambrecht (1994), Van Valin (2005: 69) describes these two concepts as follows:

Pragmatic Presupposition: The set of propositions lexicographically evoked in an utterance which the speaker assumes the hearer already knows or believes or is ready to take for granted at the time of speech.

Pragmatic Assertion: The proposition expressed by a sentence which the hearer is expected to know or believe or take for granted as a result of hearing the sentence uttered.

In RRG, focus is defined as the semantic component of a proposition which is pragmatically structured. In this proposition, focus is what is contained in the assertion but not in the presupposition (cf. Van Valin 2005: 69). Topic on the other hand is information which is both contained in the presupposition and in the assertion (cf. Van Valin 2005). RRG differentiates three focus types (cf. Van Valin 2005: 70). The universally unmarked focus type, which is similar to the traditional notion of ‘topic-comment’, is predicate focus (cf. Van Valin 2005: 70). Following Van Valin (2005: 70), who bases his definition on Lambrecht (2000), predicate focus is defined as follows:

Predicate focus structure: Sentence construction expressing a pragmatically structured proposition in which the subject is a topic (hence with the presupposition) and in which the predicate expresses new information about this topic. The focus domain is the predicate focus phrase (or part of it).

Formally, predicate focus can be described as follows:

(29)

Sentence:	<i>McGee's computer crashed.</i>
Presupposition:	‘McGee's computer is available as a topic for comment x’
Assertion:	‘x = crashed’
Focus:	crashed
Focus domain:	Verb plus remaining post-verbal core constituents (cf. Van Valin 2005: 70)

Sentence focus differs strikingly from predicate focus, since it has no topical subject. Instead, the entire sentence is in the focus domain (cf. Van Valin 2005: 71). Van Valin (2005: 71) defines predicate focus based on Lambrecht (2000) as follows:

Sentence focus structure: Sentence construction formally marked as expressing a pragmatically structured proposition in which both the subject and the predicate are in focus. The focus domain is the sentence, minus any topical non-subject arguments.

In a more formal way, predicate focus can be represented as in (30).

- (30) Question: What happened?
 Sentence: *McGee's computer crashed.*
 Presupposition: None
 Assertion: 'McGee's computer crashed.'
 Focus: 'McGee's computer crashed.'
 Focus domain: Clause (cf. Van Valin 2005: 71)

In narrow focus, the focus domain is a single constituent. This constituent can either be subject, object, an oblique or a verb (cf. Van Valin 2005: 71). A more formal description of narrow focus looks as follows.

- (31) Question: I heard McGee's tablet pc crashed?
 Sentence: *McGee's computer crashed.*
 Presupposition: 'McGee's x crashed.'
 Assertion: x = computer
 Focus: 'computer'
 Focus domain: RP (cf. Van Valin 2005: 72)

As pointed out by Van Valin (2005: 72), Lambrecht (1986) distinguishes unmarked narrow focus and marked narrow focus. The difference is where narrow focus falls. In English, unmarked narrow focus is found if narrow focus falls on the final constituent. Marked narrow focus on the other hand is found when it falls to the left or right of the final constituent. WH-questions are a common example of narrow focus. In a sentence like *Whom did Gibbs shoot?* With the corresponding answer *He shoot* _i, the WH-word and the RP filling its slot in a reply are both marked as narrow foci (cf. Van Valin 2005: 72). Also, in yes-no questions narrow focus is found (cf. Van Valin 2005: 72). Formally, the interaction of focus as well as presupposition and assertion is captured by DRSs, which in RRG-terms are introduced in Van Valin (2005: 171). Van Valin (2005: 171) describes the idea of DRSs as follows:

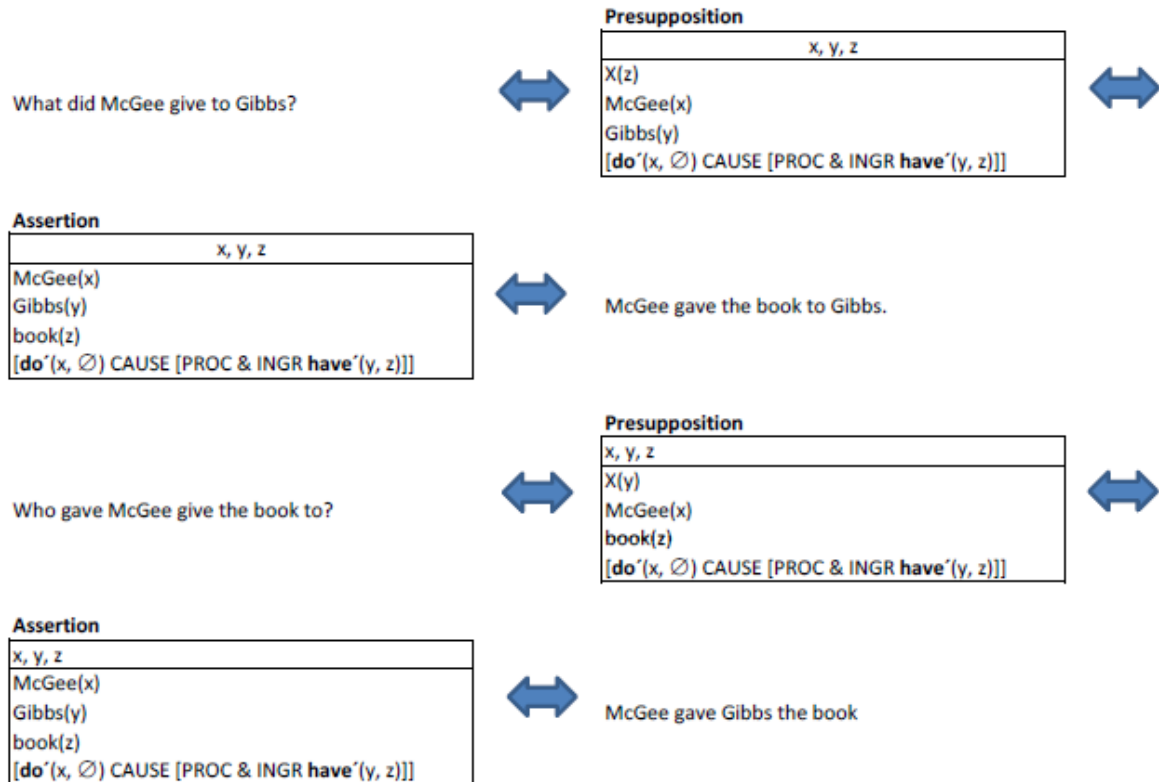
In section 3.1 above, Lambrecht's (1994) definition of 'focus' is 'the semantic component of pragmatically structured proposition whereby the assertion differs from the presupposition'. This yields the actual focus domain, and, in order to derive it, it is necessary to have representations of both the pragmatic assertion and the pragmatic presupposition. This can be done in terms of simplified version of Discourse Representation Theory (Kamp and Reyle 1993; Asher 1999; von Heusinger 1999). [...] These is a discourse representation structure for each sentence, and it contains the referent and the proposition expressed in the sentence. Coreference relations between pronominals and established referents are explicitly represented. In order to derive the different focus structures, it is necessary to have a representation of the presupposition and the assertion. According to von Heusinger (1999: 202), 'the background structure is constructed in the same way [as the foreground, i.e. asserted, structure] except for focused expressions, which are represented by designated variable of the appropriate type'.

An example of how DRSs interact with focus assignment in English three-place predicates is given in (32) below.

- (32) a. Speaker A: What did McGee give to Gibbs?
 Speaker B: McGee gave the book to Gibbs.
 b. Speaker A: Who did McGee give the book to?
 Speaker B: McGee gave Gibbs the book.

In speaker A’s utterance in (32a) *what* is the focus and in speaker A’s utterance in (32b) *who* is the focus; both utterances are examples of narrow focus. The interesting question now is which focus structure types are found in the responses from the B-speakers in (32). Since the B-speaker-responses are used in order to respond to a sentence, which is of the narrow focus type, also these sentences are narrow focus. This is because the focus domain is a single constituent in this case an RP (cf. Van Valin 2005: 72). It is possible to represent (16) in the following DRSs.

(33) DRS structures



The question is *What did McGee give to Gibbs?* and the presupposition of the speaker uttering this sentence is ‘x was given to Gibbs by McGee’. The assertion uttered in response is that *the book* was given to Gibbs by McGee. What is new in this assertion and this way a possible candidate for focus is *the book*. The second question in figure 1 is *Who gave McGee give the book to?* Here, the presupposition of the speaker is ‘X was given a book by McGee’. In the response *McGee gave Gibbs the book*, the RP *Gibbs* is not in the presupposition, however, it is in the assertion and this means it is focus. This can clearly be seen in the DRSs used. Based on the native speaker interviews introduced in Erteschik-Shir (1979) and also shown in the example in 31, it is also possible to conclude that in cases of predicate focus and sentence focus the order is *McGee gave Gibbs the book*. This conclusion is possible, since this is the unmarked situation. Unmarked situation in these examples does not mean that this word order is the basic word order, since in this approach both constructions the ditransitive and the dative-shift are treated alike. This is in contrast to Van Valin’s (2007) analysis.

What was shown thus far is that information structure as assumed in Erteschik-Shir (1979) governs variable undergoer linking in three-place predicates in English. It was

also shown that DRSs, as used in Van Valin (2005) can be used in order to determine the focus of a sentence. However, it is important to note that the postverbal element in three-place predicate constructions is not always focus, since, following Lambrecht (1994), focus usually is a pragmatically relation of an element to a proposition (cf. Lambrecht 1994: 217).

As pointed out by Nolan (personal communication) it is an empirical question whether there is one constructional schema which can handle the linking bi-directionally or whether there are two separated constructional schemas; if the latter was the case there would be two constructional schemas, one for the semantics to syntax linking and one for the syntax to semantics linking. In this paper I will use an approach to constructional schemas which is somewhere in between these two ideas. My idea is that a construction, like three-place predicates in English, is a grammatical object in the sense that the construction can be compared with an object in programming languages which can be manipulated by the programming language. This idea is similar to the concept of the construction as grammatical object developed in Nolan (2011). In Nolan's approach a construction as grammatical object is a structured grammatical object which has a unique constructional signature as was pointed out in section 6. In my approach to constructional schemas the idea is very similar and here a grammatical object within a computational adequate version of RRG is a data structure which has two methods, the syntax to semantics linking and the semantics to syntax linking. For illustrative purposes the method for the semantics to syntax linking in three-place predicates is also illustrated in the fashion of a constructional schema. In this approach, as in Nolan (2011), a constructional schema has signature which can be used to identify the constructional schema and then it contains these two methods, one for each linking direction. In some way these different methods have similar structures however they differ with respect to the explicit algorithm they contain which is to be used to model the appropriate linking direction. The advantage of this approach in which constructional schemas are developed along the lines of a concept from computer science is that on the one hand the bi-directionality of the linking algorithm is reflected more naturally and on the other hand this approach supports the concept of computational adequacy I have developed in this paper.

In (34) the architecture of the constructional schema as grammatical object for three-place predicates in English is shown. As can be seen the construction as grammatical object for three-place predicates in English which is represented by a constructional schema as developed in Nolan (2011) has a signature which uniquely identifies the construction. The object which is represented by the constructional schema in (34) refers to transfer situations in English and can be used to represent both the semantics to syntax linking and the syntax to semantics linking in RRG in a computationally adequate version, which can be executed as part of an intelligent software agent on a RAM. The constructional schema in (34) has three possible signatures, which can be used to identify the construction and activate it in the sense of Nolan (2011). English three-place predicates for transfer verbs have three different signatures. In the first one, the first RP is the actor, while the third token can either be an RP or a pronoun, which are both recipient. The last RP in the signature is the theme. However, this construction can also be activated by two further signatures. The second signature also has an RP as first token, which is actor. The third token in this signature is an RP, which is theme, while the fourth token is a pronoun or an RP, which is both marked by a preposition and is a recipient. In the third signature, an RP is actor and the third token is an RP, which

is theme. The last token can either be an indefinite RP, which is recipient, and marked by a preposition. In all examples, the verb is in V2-position. It is also noted that the signature consists of four tokens with an optional preposition. This constructional schema can either receive a clause consisting of tokens [1], [2] or [3] as input or it can receive an LS as input. Here, grammatical patterns for sentences are stored, which in an implementation of intelligent software agents are identified after parsing took place. In these grammatical patterns, the verbs in V2-position are inflected. Since this construction schema is activated by a signature, also semantic roles are assigned. The other possibility is that the constructional schema receives an LS as input which is generated within the lexicon in advance and sent to the constructional schema as grammatical object. This will be described later in this paper.

(34)

<p>Signature [1] = [RP_{Actor} V [RP_{Recipient} PN_{Recipient}] RP_{Theme}] OR [2] = [RP_{Actor} V RP_{Theme} [PREP PN RP]_{Recipient}] OR [3] = [RP_{Actor} V RP_{Theme} [PREP indef det N]_{Recipient}] as tokens [1 2 3 (PREP) 4]</p>
<p>Input RP_{Actor} [_] and V = pred. [TNS: _] and [RP_{Recipient} [_] PN_{Recipient} [_]] and RP_{Theme} [_] = [4] elseif RP_{Active} [_] and V = pred. [TNS: _] and RP_{Theme} [_] and [[PREP] PN_{Rec} [_] RP_{Rec} [_]] = [5] elseif RP_{Actor} [_] and V = pred. [TNS: _] and RP_{Theme} and [[PREP] PN_{Rec} [_] RP_{Rec} [_]] = [6] or LS = [do'(x, <) CAUSE [PROC & INGR have' (x, y)]]</p>
<p>Method body if semantics to syntax linking == true do syntactic pattern = semantics.to.syntax.linking(LS) else LS = syntax.to.semantics.linking([4] or [5] or [6], tokes[1, 2, 3, prep 4])</p>
<p>Output [do'(x, <) CAUSE [PROC & INGR have' (x, y)]] = LS or Well formed three-place predicate construction = LS for semantics to syntax linking or RP_{Actor} [_] and V = pred. [TNS: _] and [RP_{Recipient} [_] PN_{Recipient} [_]] and RP_{Theme} [_] = syntactic pattern elseif RP_{Active} [_] and V = pred. [TNS: _] and RP_{Theme} [_] and [[PREP] PN_{Rec} [_] RP_{Rec} [_]] = syntactic pattern elseif RP_{Actor} [_] and V = pred. [TNS: _] and RP_{Theme} and [[PREP] PN_{Rec} [_] RP_{Rec} [_]] = syntactic pattern</p>

The grammatical object for three-place predicates which is represented as constructional schema does not contain a workspace since as pointed out by Nolan the workspace is construction dependent since however this construction consists of two methods as shown in the method body of the grammatical object the workspace is actually part of the methods which describe the linking direction. What happens in the method body is that it is tested whether semantics to syntax linking takes place or whether syntax to semantics linking takes place. In the first case a semantics to syntax linking method is called which receives an LS as input and in the latter case the syntax to semantics linking takes place; in this case the method receives a clause as input. After a method is executed within a constructional schema which is represented as a sub constructional schema the result is stored in either a variable called LS or in a variable called syntactic patterns which in the output section of the constructional schema assign the proper output to this variable. The output of construction as grammatical object of the constructional schema which represents a grammatical object is the LS [do'(x,) CAUSE [PROC & INGR have' (x, y)]] or the information *Well formed three-place predicate construction*. The information which output should be realized by the assignment from the method for the syntax to semantics linking by the syntax to semantics linking method. Basically the representation in this case is a bit fuzzy from a computational point of view but clear for illustrative purposes since generally the output is realized by the variable LS and this can either be an LS or the information that the sentence is well-formed. The same is true for the three different possibilities for the output. In an implementation only the variable would be represented however for purposes of illustration it is shown that the variable can be realized by three different syntactic patterns. By realizing the construction as grammatical object represented as constructional schema containing methods it is possible to have a close set of language specific constructions which are treated as objects which can properly account for the bi-directional linking in RRG with the use of methods representing the two linking directions. With the use of constructional schemas which represent constructions as grammatical objects the linking algorithm developed in Van Valin (2005: 136) becomes computational adequate in the sense that it can be executed on a RAM. A possible method for the semantics to syntax linking in for three-place predicates in English whose , which exhibits variable undergoer linking is given in (35) below.

The semantics to syntax method which is part of the constructional schema which realizes three-place predicates as objects in this approach contains as workspace as exemplified in Nolan (2011) which contains the input which is sent to the method which is called in the method body of the constructional schema for the grammatical object. However most importantly this method which is part of the constructional schema contains a construction body in which the actual linking takes place. In the construction body of the semantics to syntax linking is described. First, the logical structure retrieved from the linking algorithm to which this information was sent from the lexicon in advance. Information of lexical semantic relations is inherited.

(35)

English three-place predicate construction ‘give-rel’ Semantics to Syntax Linking
Workspace: input [1], [2] or [3]
Construction Body [SEM → SYN] Retrieve the logical structure from linking algorithm Inherit information of lexical semantic relations from lexicon Assign LS(x) argument with actor according inherited information from lexical semantic relations from lexicon if predicate focus structure == true or sentence focus == true or token 3 == narrow focus do coindex token [3] with recipient in logical structure coindex token [4] with theme in logical structure generate syntactic pattern = [1] elseif signature [3] == true do coindex token [3] with theme in logical structure coindex token [4] with recipient in logical structure generate syntactic pattern = [2] elseif coindex token [3] with theme in logical structure coindex token [4] with recipient in logical structure generate syntactic pattern = [3] end if Morphology: none Pragmatics: topic = PSA

Arguments in the LS are assigned actor, which is defined based on the signature, which was activated based on typed feature structures stored in the lexicon is retrieved. If an example of predicate focus structure is found or sentence focus is true, or token 3 is in narrow focus, then token 2 is coindexed with recipient in the logical structure, and token 4 is coindexed with theme in the logical structure, and syntactic pattern [1] is generated and stored in the variable ‘syntactic pattern’. If, however, signature [3] is true because it is activated based on the interaction of the construction repository and the lexicon, then token [3] is coindexed with theme in the logical structure, and token [4] is coindexed with recipient, and syntactic pattern [2] is generated. In all other cases, token [3] is coindexed with theme in the logical structure and token [4] is coindexed with recipient in the logical structure, and syntactic pattern [3] is generated. As noted in the constructional schema, morphology does not play any role, while the PSA is always topic in this construction. What is important to note in this context is that of course the method cannot only retrieve values by the object of which it is part but it can also retrieve information from the general linking algorithm. This is what takes place in the construction body. After the constructional schema has applied as kind of method in the pseudo-code meta-language, the populated logical structure is sent to the algorithm again and the usual linking process can start. For a revision of the semantics to syntax linking algorithm, this means in the first step DRSs automatically filled in discourse situation are analyzed and focus structure is determined algorithmically. The determination of topic and focus is of importance for the new version of the RRG linking from semantics to syntax developed in this paper. The algorithm for the determination of the different focus types is given in (36) below:

(36)

```

access information from DRS for presupposition;
access information form DRS for assertion;
if RP in presupposition and assertion do
    topic = RP;
if (NUC and RP1 ... RPn in assertion == true) and (NUC and RP1 ... RPn in
presupposition == false) do
    predicate focus = (NUC and RP1 ... RPn);
if DRS for presupposition == false and DRS for assertion == true do
    focus = assertion;
if (RP or PP or NUC not in presupposition) and (RP or PP or NUC in assertion) do
    narrow focus = RP or PP or NUC;

```

In the pseudo-code representation of an algorithm for the determination of focus types as given in (34), what happens first is that information is accessed from both the DRS for presupposition and the DRS for the assertion. In a first step it is checked whether an RP is contained in both presupposition and assertion. If such an element exists, it is assigned topic. In case NUC and RP₁ ... RP_n are contained in assertion and not in the presupposition, these constituents are assigned predicate focus. In cases where no presupposition exists but the statement 'DRS for assertion == true' is true, then the whole assertion is assigned focus. In situations where RPs, PPs, or NUCs are not contained in the presupposition, but occur in the assertion, either the RP, PP or NUC is assigned narrow focus. In the second step of the new version of the semantics to syntax linking algorithm in RRG, the logical structure is accessed from the lexicon via unification of the logical structure, information from the lexicon and a database containing world knowledge. Afterwards a logical structure is accessed from the lexicon. In the next step of the new linking algorithm, a populated logical structure is accessed from the lexicon. In this paper I will not give an algorithmic description of the cognitive processes which construct a full-fledged logical structure. It will be a task for future research to describe how logical structures are constructed in the lexicon, since it would be necessary to develop lexical entries for all parts oil assume that logical structures are already constructed when they are accessed from the lexicon. The next step in the new version of the linking algorithm takes place after the fully populated logical structure is sent from the lexicon to the linking algorithm. In Van Valin (2005: 137) it is proposed that the output of speech as well as having a rough architecture of a database containing world knowledge. This task is way beyond the scope of this paper. Therefore, in the next step of the new linking algorithm, I the first step in the linking algorithm is a fully constructed logical structure with attached operators and a notion of the discourse status of the referents in the logical structures. Such a logical structure is given in (37) below:

(37)

a. McGee gave Siva the USB-stick.

b. <_{IF} DEC <_{TNS} PAST < [[do'(McGee_{ACT}, <)]
 CAUSE [PROC & INGR have'(Siva_{ACS}, USB-stick_{ACT})] >>>
 (cf. Van Valin 2005: 137)

Following Van Valin (2005: 79), it is noted that the activation level of referents of RPs filling argument positions is coded in logical structures. Van Valin (2005: 79) differentiates five levels of activation:

For simplicity's sake, only five levels of activation will be coded: active, i.e. actively under consideration in the discourse by means of direct mention; accessible, i.e. not actively under consideration but readily recognized by the addressee due either to knowledge of the world or to occurrence in the immediate environment of the speech situation; inactive, i.e. previously mentioned but not actively under consideration and not assumed by the speaker to be recognized by the addressee; brand new – anchored, i.e. not previously mentioned but related to something already mentioned or accessible; and brand new – unanchored; i.e. not previously mentioned or related to anything previously mentioned (Prince 1981b, Chafe 1987). Propositions may also have different levels of activation (Dryer 1996).

Since, however, in the new version of the linking algorithm topic and focus are determined in the first step rather than later in the algorithm, topic and focus are assigned in the lexicon within a logical structure construction processes. This is possible since the arguments in the logical structure are matched with information from DRSs. Also, the activation levels are assigned in this step of the linking algorithm. Besides information on the activation level - topic and focus as well as the attachment of operators to the logical structure - also pointers to possible signatures in the construction repository are contained in the logical structure, which is sent from the lexicon to the algorithm. These pointers to the construction repository occur in situations where the necessity of a construction is coded in the lexicon. This way the logical structure, which is sent to the linking algorithm, looks as given in (38).

- (38) a. Whom did McGee give the USB-stick?
 b. McGee gave Siva the USB-stick.
 c. <_{IF DEC} <_{TNS PAST} < [[**do**'(McGee_{ACT} --> Topic _agent, <)]
 CAUSE
 [PROC & INGR **have**'(Siva_{ACS} --> Focus _recipient, USB-stick_{ACT} _theme)] >>>>
and ^ [RP_{Actor} V [RP_{Recipient} | PN_{Recipient}] RP_{Theme}];
 ^ [RP_{Actor} V RP_{Theme} [PREP PN | RP]_{Recipient}]; ^ [RP_{Actor} V RP_{Theme} [PREP
 indef det N]_{Recipient}]

The next step in the linking algorithm therefore looks as follows (39).

(39)

access full-fledged logical structure from the lexicon
if pointer to signature in construction repository == **true do**
 new constructional schema **in** construction repository;
else determine morphosyntactic coding properties in one-place predicates
or determine morphosyntactic coding properties in two-place predicates.

This part of the algorithm accesses populated logical structures from the lexicon and checks whether a pointer to a signature in the construction repository exists. If this is the case, a new constructional schema is called. Otherwise, the morphosyntactic coding properties of the logical structures with assigned thematic relations are determined. In cases where a constructional schema is called, the assignment of thematic relations within the constructional schema takes place. Afterwards a logical structure with

assigned thematic relations is sent to the linking algorithm, where the other linking steps take place. Constructional schemas are flexible with respect to what part of the linking takes place within them. It is possible, as in the case of three-place predicates of the *give-rel* in this section, that they only take place to a certain degree. However, it is also possible that the whole linking takes place in the constructional schema. Since in this paper I focus on the generation of three-place predicates, I will not go on to describe how the linking should precede in more detail.

9. Conclusion

In this paper the concept of computational adequacy as an important theory external principle was developed and it was shown that this principle is of greatest importance if the Church-Turing-thesis is taken into account. The concept of intelligent software agents was introduced as important basis for the implementation of a functional linguistic theory, as RRG and a pseudo-code-based analysis of the semantics to syntax linking algorithm was developed, which was executed on a RAM. It was shown that the semantics to syntax linking algorithm, as developed in Van Valin (2005), is too coarsely grained and fuzzy to account for three-place predicates with variable undergoer linking. Based on a lexical approach to transfer verbs, typed feature structures for the give- and receive-relation were introduced and it was shown that thematic relations are stored in the mental lexicon within an inheritance network. As shown in section 7, this unification-based approach can be used in order to show that semantic macroroles, as developed in Van Valin (2005), are epiphenomenal and that the AUH is superfluous. Also, a constructional schema for the give-relation of transfer verbs in English was developed, which is connected with the lexicon via unification. It was shown that the analysis of DRSs should be the first step in the semantics to syntax linking and that it is possible to account for variable undergoer linking in English via information structure. In general, it was shown that in a new version of the linking algorithm from semantics to syntax it is necessary to first analyze DRSs by parsing and that this information is sent to the lexicon. Also, it was shown that topic and focus are assigned to the logical structure in the mental lexicon and that a pointer to the signatures in the constructional schemas stored in the construction repository, as developed in Nolan (2011), is used to connect logical structures in the mental lexicon. It was shown that the second step in the semantics to syntax linking algorithm is the access to the lexicon and that a test as to whether a constructional schema for the particular sentence exists. If this is the case, constructional schemas are called as functions. Since this paper is concerned with research work in progress, it will be a task of future research to answer the several questions left open in this paper.

References

- Asher, Nicholas. 1999. Discourse and Focus/background distinction. In P. Bosch and R. van der Sandt (eds.), *Focus: linguistic, cognitive and computational perspectives*, 247-67. Cambridge: CUP
- Blass, Andreas and Yuri Gurevitch. 2001. Algorithms: A Quest for Absolute Definitions. <http://www.math.lsa.umich.edu/~ablass/abs.pdf>
- Bloomfield, Leonard. *Language*. New York: Holt.
- Butler, Christopher S. 2009. The Lexical Constructional Model: genesis, strengths and Challenges. Available: http://www.lexicom.es/drupal/files/Butler_2009.pdf
- Carpenter, Bob. 1991. The Generative Power of Categorical Grammars and Head-Driven Phrase Structure Grammars with Lexical Rules. *Computational Linguistics* 17(3):301-314.
- Chafe, Wallace. 1987. Cognitive Constraints on information flow. In Tomlin (Ed.) *Coherence and grounding in discourse*, 21 - 51. Amsterdam: John Benjamins.
- Chomsky, Noam. 1957. *Syntactic Structures (Janua Linguarum no.4)* Gravenhage: Mouton.

- Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.
- Chomsky, Noam. 1977. *Essays on form and interpretation*. New York: North Holland.
- Chomsky, Noam. 1994. Bare phrase structure. In G. Webelhuth (ed.) *Government and binding theory and the minimalist program*, 383 – 439. Cambridge, MA: Blackwell.
- Davis, Anthony R. 2001. *Linking types in the hierarchical lexicon*. Stanford: CSLI Publications.
- Dik, Simon C. 1978. *Functional Grammar*. Amsterdam: North-Holland.
- Dik, Simon C. 1991. Functional Grammar. In Droste and Joseph, (eds.), 247 – 74.
- Diedrichsen, Elke. 2012: The theoretical importance of constructional schemas in RRG. In: Wataru Nakamura (ed.). *New perspectives in Role and Reference Grammar*. London: Cambridge Scholars Publishing
- Diedrichsen, Elke. Submitted: The influence of Person Hierarchies on preferences with word order in German ditransitive constructions. To appear in Siewierska, Anna (†) and Eva van Lier (eds.): “Referential Hierarchies in Three-participant constructions”, Online-Journal “Linguistic Discovery”
- Dowty, David. 1974. *Word meaning and Montague Grammar*. Dordrecht: Reidel.
- Erteschik-Shir, Nomi. 1979. Discourse constraints on dative movement. In Givón T. (ed.) *Syntax and Semantics*, vol. XII: *Discourse and syntax*. New York: Academic Press. 441-67
- Evans, Roger und Gerald Gazdar. 1996. DATR: *A language for lexical representation*. *Computational Linguistics* 22, 2. 167 - 216.
- Foley, William A., and Robert D. Van Valin, Jr. 1984. *Functional syntax and universal grammar*. Cambridge: Cambridge University Press.
- Gottschalk, Judith. 2010a. Storage of linguistic knowledge in the mental lexicon: An approach within Role and Reference Grammar. ITB Journal, Issue 19. 20 – 45.
<http://www.itb.ie/files/journal/issue-19.pdf>
- Gottschalk, Judith. 2011a: The precore slot in Icelandic: A topological analysis of V2-clause structure within Role and Reference Grammar. ITB-Journal Issue 21.
<http://www.itb.ie/files/journal/issue-21.pdf>
- Gottschalk, Judith. 2011b. Storage of linguistic knowledge in the mental lexicon: An approach within Role and Reference Grammar. Paper presented at the workshop on Functionally motivated computational approaches to models of language and grammar . Societas Linguistica Europaea, University of La Rioja, Logroño, Spain.
- Greenberg, Joseph H. 1966. Some universals of grammar with particular reference of the order of meaningful elements. In Joseph H. Greenberg(ed.), *Universals of language*, 73 – 113. Cambridge, MA: MIT Press.
- Guest, Elizabeth. 2008. Parsing for Role and Reference Grammar. In Van Valin, Robert D., Jr. (ed). *Investigations of the Syntax-Semantics-Pragmatics Interface*. Amsterdam: John Benjamins.
- Güting, Ralf H. and Stefan Dieker. 2004. Datenstrukturen I. Kurseinheit 1: Einführung Programmiersprachliche Konzepte für Datenstrukturen. In: FernUniversität Hagen, ms.
- Haspelmath, Martin. 2008. Ditransitive constructions – Towards a new Role and Reference Grammar account? In Van Valin, Robert D., Jr. (ed). *Investigations of the Syntax-Semantics-Pragmatics Interface*. Amsterdam: John Benjamins.
- Haspelmath, Martin. 2005a. Argument marking in ditransitive alignment types. *Linguistic Discovery* 3(1): 1 – 21. <http://linguistic-discovery.dartmouth.edu/>
- Heusinger, Klaus, von. 1999. Intonation and information structure. Habilitationsschrift, University of Konstanz.
- Kamp, Hans and Uwe Reyle. 1993. *From Discourse to Logic*. Hingham, MA: Kluwer.
- Kaplan, Ronald and Joan Bresnan. 1982. Lexical-Functional Grammar: a formal system for grammatical representation. In Bresnan (ed.), *The mental representation of grammatical relations*. 173 – 281. Cambridge, MA: MIT Press.
- Lambrecht, Knud. 1994. *Information Structure and Sentence Form*. Cambridge: CUP
- Larson, Richard. 1988. On double object construction. *Linguistic Inquiry* 19: 335-92
- Levelt, Willem. 1989. *Speaking*. Cambridge: MIT Press.
- Michaelis, Laura A. and Josef Ruppenhofer. 2001. *Beyond alternations: A construction based approach to the applicative pattern in German*. Stanford: CSLI Publications.
- Murtagh, Irene. Towards a Linguistically Motivated Irish Sign Language Conversational Avatar. ITB Journal, Issue 19. 72 – 101.
- Nolan, Brian. 2011. *Constructions as grammatical objects: A new perspective on constructions in RRG*. Paper presented at the International Conference on Role and Reference Grammar on "Functional Linguistics: Grammar, Communication & Cognition". Facultad de Letras, at the Pontificia Universidad Católica de Chile, in Santiago de Chile.

- Nolan, Brian and Yasser Salem. UNIARAB: A Universal Machine Translator System for Arabic Based on Role and Reference Grammar. <http://linguistics.buffalo.edu/people/faculty/vanvalin/rrg.html>
- Prince, Ellen. 1981b. Toward a taxonomy of given-new information. 223-256. In Cole (Ed.) *Radical pragmatics*. New York: Academic Press.
- Reichenbach, Hans. 1947. *Elements of Symbolic Logic*. New York: Radom House.
- Richter, Frank. 2000. *A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar*. <http://tobias-lib.uni-tuebingen.de/volltexte/2004/1203/>
- Russell, Stuart and Peter Norvig. *Artificial Intelligence: A Modern Approach Second Edition*. New Jersey: Prentice Hall
- Salem, Yasser. 2009. A generic framework for Arabic to English machine translation of Simplex sentences using the Role and Reference Grammar linguistic model. <http://linguistics.buffalo.edu/people/faculty/vanvalin/rrg.html>
- Siewierska, Anna. 1998. Languages without objects. *Languages in Contrast* 1(2): 173 –190.
- Stahlke, Herbert, F.W. 1970. Serial verbs. *Studies in African Linguistics*. 1.60-99.
- Van Valin, Robert D, Jr. 2005. *Exploring the Syntax-Semantics Interface*. Cambridge: CUP.
- Van Valin, Robert D, Jr. 2006. Semantic Macroroles and Language Processing. Ina Bornkessel, Matthias Schlesewsky, Bernard Comrie, and Angela Friederici (2006). 263-302. <http://linguistics.buffalo.edu/people/faculty/vanvalin/rrg.html>
- Van Valin, Robert D, Jr. 2007. The Role and Reference Grammar Analysis of Three-Place Predicates. <http://linguistics.buffalo.edu/people/faculty/vanvalin/rrg.html>
- Van Valin, Robert D., Jr., und Randy J. LaPolla. 1997. *Syntax: Structure, meaning and function*. Cambridge: CUP.
- Van Valin, Robert D., Jr. & Elke Diedrichsen (2006): *A Bonsai Grammar for German*. http://www.coli.uni-saarland.de/~tania/CMGD/RRG_BonsaiGrammarGerman.pdf
- Vendler, Zeno. 1957. *Linguistics in philosophy*. Ithaca: Cornell University Press.
- Wilson, Chris. 2009. Lex: a Software Project for Linguists. Technical Report SEE-J Hiphil 6 [<http://www.see-j.net/index.php/hiphil>] (2009). Available at <http://www.see-j.net/index.php/hiphil/article/view/60>
- Winther-Nielsen, Nicolai. .2009. Biblical Hebrew parsing on display: The Role-LexicalModule (RLM) as a tool for Role and Reference Grammar. SEE-J Hiphil 6 [<http://www.see-j.net/index.php/hiphil>], 1-51. Available at <http://www.see-j.net/index.php/hiphil/article/view/78>. PDF