Constraint-based RMRS Construction from Shallow Grammars

Anette Frank
Language Technology Lab
German Research Center for Artificial Intelligence, DFKI GmbH
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
Anette.Frank@dfki.de

Abstract

We present a constraint-based syntax-semantics interface for the construction of RMRS (Robust Minimal Recursion Semantics) representations from shallow grammars. The architecture is designed to allow modular interfaces to existing shallow grammars of various depth – ranging from chunk grammars to context-free stochastic grammars. We define modular semantics construction principles in a typed feature structure formalism that allow flexible adaptation to alternative grammars and different languages.[1]

1 Introduction

Semantic formalisms such as MRS (Copestake et al., 2003) provide elegant solutions for the treatment of semantic ambiguities in terms of underspecification – most prominently scope. In recent work Copestake (2003) has investigated a novel aspect of underspecification in the design of semantic formalisms, which is concerned with the representation of *partial* semantic information, as it might be obtained from shallow, i.e. incomplete syntactic analysis. The main rationale for this type of underspecification is to ensure monotonicity, and thus upwards compatibility of the output of shallow parsing with semantic representations obtained from full syntactic parsing. Thus, Copestake's design of RMRS – Robust Minimal Recursion Semantics – provides an important contribution to a novel line of research towards integration of shallow and deep NLP. While previous accounts (Daum et al., 2003; Frank et al., 2003) focus on shallow-deep integration at the syntactic level, Copestake aims at integration of shallow and deep NLP at the level of semantics.

In this paper we review the RMRS formalism designed by Copestake (2003) and present an architecture for a principle-based syntax-semantics interface for RMRS construction from shallow grammars. We argue for a unification-based approach, to account for (underspecified) argument binding in languages with case-marking as opposed to structural argument identification. The architecture we propose is especially designed to support flexible adaptation to different types of shallow to intermediate-level syntactic grammars that may serve as a basis for RMRS construction. A challenge for principle-based semantics construction from shallow grammars is the flat and sometimes non-compositional nature of the structures they typically produce. We present RMRS semantics construction principles that can be applied to flat syntactic structures with various degrees of partiality.

[1]The research reported here was conducted in the project QUETAL, funded by the German Ministry for Education and Research, BMBF, under grant no. 01 IW C02.

2 RMRS – For Partial Semantic Representation

Copestake (2003) presents a formalism for partial semantic representation that is derived from MRS semantics (Copestake et al., 2003). Robust Minimal Recursion Semantics is designed to support novel forms of integrated shallow and deep NLP, by accommodating semantic representations produced by NLP components of various degrees of partiality and depth of analysis – ranging from PoS taggers and NE recognisers over chunk and (non-)lexicalised context-free grammars to deep grammars like HPSG with MRS output structures.

The potential of a variable-depth semantic analysis is most evident for applications with conflicting requirements of robustness and accuracy. Given a range of NLP components of different depths of analysis that deliver compatible semantic representations, we can apply flexible integration methods: apply voting techniques, or combine partial results from shallow and deep systems (Copestake, 2003).

To allow intersection and monotonic enrichment of the output representations from shallow systems on one extreme of the scale with complete representations of deep analysis on the other, the missing specifications of the weakest system must be factored out from the most comprehensive deep representations. In the RMRS formalism, this concerns the following main aspects of semantic information:

Argument encoding. A 'Parsons style' notation accommodates for partiality of shallow systems wrt. argument identification. Instead of predicates with fixed arity, e.g. *l4:on(e′,e,y)*, predicates and arguments are represented as independent elementary predications: *on(l4,e′)*, *ARG1(l4,e)*, *ARG2(l4,y)*. This accounts for uncertainty of argument identification in shallow grammars. Underspecification wrt. the type of argument is modeled in terms of a hierarchy over disjunctive argument types: $ARG1 \sqsubset ARG12$, $ARG2 \sqsubset ARG12$, $ARG12 \sqsubset \ldots \sqsubset ARGn$.

Variable naming and equalities. Constraints for equality of variables in elementary predications are to be added incrementally, to accommodate for knowledge-poor systems like PoS taggers, where the identity of referential variables of, e.g., adjectives and nouns in potential NPs cannot be established, or else chunkers, where the binding of arguments to predicates is only partially established.

An example of corresponding MRS (1.a) and RMRS (1.b) representations illustrate these differences, cf. Copestake (2003).

(1) *Every fat cat sat on a mat*

a. l0:every(x,h1,h2), l1:fat(x), l2:cat1(x), l3:CONJ, l4:sit1(e_{spast},x), l14:on2(e′,e,y), l9:CONJ, l5:some(y,h6,h7), l6:table1(y), qeq(h1,l3), qeq(h6,l6), in-g(l3,l1), in-g(l3,l2), in-g(l9,l4), in-g(l9,l14)

b. l0:every(x0), RSTR(l0,h1), BODY(l0,h2), l1:fat(x1), l2:cat1(x2), l3:CONJ, l4:sit1($e3_{spast}$), ARG1(l4,x2), l14:on2(e4), ARG1(l14,e3), ARG2(l14,x5), l9:CONJ, l5:some(x5), RSTR(l5,h6), BODY(l5,h7), l6:table1(x6), qeq(h1,l1), qeq(h6,l6), in-g(l3,l1), in-g(l3,l2), in-g(l9,l4), in-g(l9,l14), x0 = x1, x1 = x2, x5 = x6

3 RMRS from Shallow Grammars

We aim at a modular interface for RMRS construction that can be adapted to a wide range of *existing* shallow grammars such as off-the-shelf chunk parsers or probabilistic (non-)lexicalised PCFGs. Moreover, we aim at the construction of underspecified, but *maximally constrained (i.e., resolved)* RMRS representations from shallow grammars.

A unification-based account. Chunk-parsers and PCFG parsers for sentential structure do in general not provide functional information that can be used for argument identification. While in languages like English argument identification is to a large extent structurally determined, in other languages arguments are (partially) identified by case marking. In case-marking languages, morphological agreement constraints can yield a high degree of completely disambiguated constituents. Morphological disambiguation can thus achieve maximally constrained argument identification for shallow analyses. We therefore propose a *unification-based approach* for RMRS construction, where agreement constraints can perform morphological disambiguation for partial (i.e. underspecified) argument identification. Moreover, by interfacing shallow analysis with morphological processing we can infer important semantic features for referential and event variables, such as PNG and TENSE information. Thus, morphological processing is also beneficial for languages with structural argument identification.

A reparsing architecture. In order to realise a *modular* interface to existing parsing systems, we follow a reparsing approach: RMRS construction takes as input the output structure of a shallow parser. We index the nodes of the parse tree and extract a set of rules and lexicon entries with corresponding node indices. Reparsing of the original input string according to this set of rules deterministically replays the original parse. In the reparsing process we apply RMRS construction principles.

Constraint-based RMRS construction. We define constraint-based principles for RMRS construction in a typed feature structure formalism. These constraints are applied to the input syntactic structures. In the reparsing step the constraints are resolved, to yield maximally specified RMRS representations.

The RMRS construction principles are defined and processed in the SProUT processing platform (Drozdzynski et al., 2004). The SProUT system combines finite-state technology with unification-based processing. It allows the definition of finite state transduction rules that apply to (sequences of) typed feature structures (TFS), as opposed to atomic symbols. The left-hand side of a transduction rule specifies a regular expression over TFS as a recognition pattern; the right-hand side specifies the output in terms of a typed feature structure. The system has been extended to cascaded processing, such that the output of a set of rule applications can provide the input for another set of rewrite rules. The system allows several distinct rules to apply to the same input substring, as long as the same (maximal) sequence of structures is matched by these different rules. The output structures defined by these individual rules can be unified, by way of flexible interpreter settings. These advanced configurations allows us to state RMRS construction principles in a modular way.

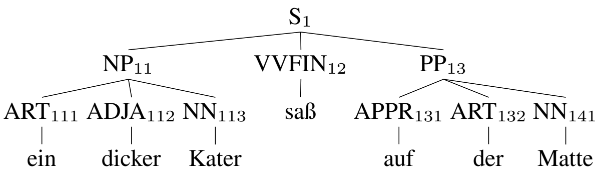

Figure 1: Input syntactic tree: *Ein dicker Kater saß auf der Matte* – A fat cat sat on the mat

```
phrase & [ID ”11”, CAT ”NP”, M-ID ”1”, M-CAT ”S”]
lex & [ID ”12”, CAT ”VVFIN”, M-ID ”1”, M-CAT ”S”]
phrase & [ID ”13”, CAT ”PP”, M-ID ”1”, M-CAT ”S”]
lex & [ID ”111”, CAT ”ART”, M-ID ”11”, M-CAT ”NP”]
lex & [ID ”112”, CAT ”ADJA”, M-ID ”11”, M-CAT ”NP”]
lex & [ID ”113”, CAT ”NN”, M-ID ”11”, M-CAT ”NP”]
lex & [ID ”131”, CAT ”APPR”, M-ID ”13”, M-CAT ”PP”]
lex & [ID ”132”, CAT ”ART”, M-ID ”13”, M-CAT ”PP”]
lex & [ID ”133”, CAT ”NN”, M-ID ”13”, M-CAT ”PP”]
```

Figure 2: TFS representations for lexical and phrasal nodes (here for tree of Figure 1)

```
phrase :> synsem & [M-ID #1, M-CAT #mcat]+
       –> phrase & [ID #1, CAT #mcat].
```

Figure 3: Reparsing rule

Cascaded Reparsing. We extract information about phrase composition from the indexed input parse trees. For each local subtree, we extract the sequence of daughter nodes as TFS, recording for each node its node identifier (ID) together with the identifier (M-ID) and category (M-CAT) of its mother node (cf. Figure 2). This implicitly encodes instructions for phrase composition that are employed in the cascaded system to guide phrase composition and concurrent semantics construction.

A general reparsing rule (cf. Figure 3) is applied to an input sequence of TFS for lexical or phrasal nodes and produces as output a TFS for the implicitly defined mother node. The rule specifies that for all nodes in the matched input sequence, their mother node identifier and category features (M-ID, M-CAT) must be identical, and defines the output (mother) node's local identifier and category feature (ID, CAT) by use of variable co-references (#var). Since the system obeys a longest-match strategy, the regular expression is constrained to apply to the same constituents as in the original parse tree.

Cascaded reparsing first applies to the sequence of leaf nodes. The output node sequence is enriched with the phrase-building information from the original parse tree, and is again input to the phrase building and semantics construction rules. Thus, we define a cyclic cascade, where the output of a cascade is fed in as input to the same rules. The cycle terminates when no phrase building rule could be applied to the input, i.e. the root category has been derived.

```
agr :> lex & [M-ID #1]*
      ( lex & [M-ID #1, CAT ”NN”, MSYN [AGR #agr]]+
      | lex & [M-ID #1, CAT ”ADJA”, MSYN [AGR #agr]]+
      | lex & [M-ID #1, CAT ”ART”, MSYN [AGR #agr]]+ )
        lex & [M-ID #1]*
   –> phrase & [ID #1, MSYN [AGR #agr]].
```

Figure 4: Modular agreement projection rules

Morpho-syntactic disambiguation. Before rule application, the SProUT system performs morphological lookup on the input words (Krieger and Xu, 2003). Morphological information is modeled in a TFS hierarchy with disjunctive types to underspecify ambiguities of inflectional features, e.g. case.

We define very general principles for morphosyntactic agreement, defining agreement between daughter and mother constituents individually for categories like determiner, adjective or noun (Figure 4). Since in our reparsing approach the constituents are pre-defined, the agreement projection principles can be stated independently for possible mother-daughter relations, instead of specifying complex precedence patterns for NPs. Defining morphological agreement independently for possibly occurring daughter constituents yields few and very general (disjunctive) projection principles that can apply to "unseen" constituent sequences.

The rule in Figure 4 again exploits the longest-match strategy to constrain application to the pre-defined constituents, by specifying coreferent M-ID features for all nodes in the rule's input sequence.

In reparsing, the (possibly disjunctive) morphological types in the output structure of the individual rule applications are unified, yielding partially resolved inflectional features for the mother node. For NP_{11}, e.g., we obtain CASE nom by unification of nom (from ART and ADJA) and nom-acc-dat (from NN). The resolved case value of the NP can be used for (underspecified) argument binding in RMRS construction.

4 Semantics Projection Principles for Shallow Grammars

Lexical RMRS conditions. Lexical entries for RMRS construction are constrained by types for PoS classes, with class-specific elementary predications (EP) in RMRS.RELS, cf. Figure 5. RELS and CONS are defined as set-valued features instead of lists. This allows for modular content projection principles (see below). We distinguish different types of EPs: ep-rel, defining relation and label, ep-rstr and ep-body for quantifiers, with LB and RSTR/BODY features. Arguments are encoded as a type ep-arg, which expands to disjunctive subtypes ep-arg-1, ep-arg-12, ep-arg-23, . . . , ep-arg-n.

```
rmrs-nn & [CAT "NN", MSYN [AGR #agr],STEM <#stem>,
           RMRS [KEY #1, BIND-ARG [AGR #agr ],
             RELS {ep-rel &[LB #lb, REL #stem] ,
                   ep-arg0 & #1 & [LB #lb, ARG0 var]},
             CONS { }]].
```

Figure 5: Lexical types with RMRS EPs

```
cont_proj :>  [M-ID #1]*
              [M-ID #1, RMRS [RELS #rels, CONS #cons]]
              [M-ID #1]*
          ->  [ID #1, RMRS [RELS #rels, CONS #cons]].
```

Figure 6: Content projection

Content projection. The content projection rule (Figure 6) assembles the RMRS conditions in RELS and CONS features of the daughter constituents. In SProUT, the unification of output structures with set-valued features is defined as set union. While the classical list representation would require multiple content rules for different numbers of daughters, the set representation allows us to state a single content principle: it applies to each individual daughter, and yields the union of the projected set elements as the semantic value for the mother constituent.

Argument and variable binding. Management features (KEY, BIND-ARG) propagate values of labels and variables for argument binding. The maximally specific type ep-arg-x of the arguments to be bound is determined by special bind-arg principles that define morpho-syntactic constraints (case, passive). For languages with structural argument identification we can employ precedence constraints in the regular expression part of argument binding rules.

Content projection from flat structures. A challenge for principle-based RMRS construction from shallow grammars are their flat syntactic structures. They do not, in general, employ strictly binary structures as assumed in HPSG (Flickinger et al., 2003). Constituents may also contain multiple heads (cf. the PP in Fig. 1). Finally, chunk parsers do not resolve phrasal attachment, thus providing discontinuous constituents to be accounted for.

With flat, non-binary structures, we need to assemble EP (ep-arg-x) conditions for argument binding for each potential argument constituent of a phrase. In the SRroUT system, this can again be done without explicit list operations, by application of individual argument binding rules that project binding EP conditions for each potential argument to the RELS feature of the mother. Thus, similar to Figure 6, we can state general and modular mother-daughter principles for argument binding. For multiple-headed constituents, such as flat PPs, we use secondary KEY and BIND-ARG features. For argument binding with chunk parsers, where PP attachment is not resolved, we will generate in-group conditions that account for possible attachments.

5 Comparison to Related Work

Compared to the RMRS construction method Copestake (2003) applies to the English PCFG parser of Carroll and Briscoe (2002), the main features of our account are argument identification via morphological disambiguation and definition of modular semantics construction principles in a typed unification formalism. The architecture we propose can be applied to sentence- or chunk-parsing. The rule-based SProUT system allows the definition of modular projection rules that can be tailored to specific properties of an underlying shallow grammar (e.g. identification of active/passive voice, of syntactic NP/PP heads). In future work we will compare our semantics construction principles to the general model of Copestake et al. (2001).

Acknowledgements I am greatly indebted to my colleagues at DFKI, especially the SProUT team members Witold Drożdżyński, Hans-Ulrich Krieger, Jakub Piskorski and Ulrich Schäfer, for their technical support and advice. Special thanks go to Kathrin Spreyer for support in grammar development.

References

A. Copestake, A. Lascarides, and D. Flickinger. 2001. An Algebra for Semantic Construction in Constraint-based Grammars. In *Proceedings of the ACL 2001*, Toulouse, France.

A. Copestake, D. Flickinger, I. Sag, and C. Pollard. 2003. Minimal Recursion Semantics. Ms.

A. Copestake. 2003. Report on the Design of RMRS. Technical Report D1.1a, University of Cambridge, University of Cambridge, UK., October. 23 pages.

M. Daum, K.A. Foth, and W. Menzel. 2003. Constraint-based Integration of Deep and Shallow Parsing Techniques. In *Proceedings of EACL 2003*, Budapest, Hungary.

W. Drozdzynski, H.-U. Krieger, J. Piskorski, U. Schäfer, and F. Xu. 2004. Shallow processing with unification and typed feature structures — foundations and applications. *Künstliche Intelligenz*, 1:17–23.

D. Flickinger, E. M. Bender, and S. Oepen. 2003. MRS in the LinGO Grammar Matrix: A Practical User's Guide. Technical report, Deep Thought Project Deliverable 3.5.

A. Frank, M. Becker, B. Crysmann, B. Kiefer, and U. Schäfer. 2003. Integrated Shallow and Deep Parsing: ToPP meets HPSG. In *Proceedings of the ACL 2003*, pages 104–111, Sapporo, Japan.

H.-U. Krieger and F. Xu. 2003. A type-driven method for compacting mmorph resources. In *Proceedings of RANLP 2003*, pages 220–224.