Logic Programming Infrastructure for Inferences on FrameNet

Peter Baumgartner[1] and Aljoscha Burchardt[2]

[1] MPI Saarbrücken, baumgart@mpi-sb.mpg.de
[2] Saarland University, albu@coli.uni-sb.de

Abstract. The growing size of electronically available text corpora like companies' intranets or the WWW has made *information access* a hot topic within Computational Linguistics. Despite the success of statistical or keyword based methods, deeper Knowledge Representation (KR) techniques along with "inference" are often mentioned as mandatory, e.g. within the Semantic Web context, to enable e.g. better query answering based on "semantical" information. In this paper we try to contribute to the open question how to operationalize semantic information on a larger scale. As a basis we take the *frame* structures of the Berkeley FrameNet II project, which is a structured dictionary to explain the meaning of words from a lexicographic perspective. Our main contribution is a transformation of the FrameNet II frames into the *answer set programming paradigm* of logic programming.
Because a number of different reasoning tasks are subsumed under "inference" in the context of natural language processing, we emphasize the flexibility of our transformation. Together with methods for automatic annotation of text documents with frame semantics which are currently developed at various sites, we arrive at an infrastructure that supports experimentation with semantic information access as is currently demanded for.

1 Introduction

The growing size of electronically available text corpora like companies' intranets or the WWW has made *information access* a hot topic within Computational Linguistics. Without powerful search engines like Google the WWW would be of much lesser use. But there are obvious limitations of the current pure word-based methods. If one is e.g. searching for information about BMW buying Rover one might enter a query like (1) into a search engine.

(1) *BMW buy Rover.*

On inspection of some of the thousands of hits returned, two problems become visible:

Problem 1. Low Precision: Numerous irrelevant pages are returned like car sellers' pages that mention the query words in unintended contexts.

Problem 2. Low Recall: Even if the search engine does some linguistic processing to include results that are formulated e.g. in finite form (*BMW buys Rover*), relevant pages using semantically similar words like *purchase* instead of *buy* are missing.

There exist approaches that address problem 2. For example, some IR systems use the WordNet [Fel98] lexicon to try synonymous words in queries, while other approaches use learning techniques to detect and utilize similarities between documents. But apart from the fact that these systems rely heavily on redundancy among the text collections to be searched, they mostly do not address problem 1 at all.

A principled method is to analyze documents (and queries) in terms of *semantic* predicates and role relations. As we will explain in more detail in Section 2, the linguistically motivated *frame* structures of the Berkeley FrameNet II project [BFL98] are a suitable theoretical means for such an analysis. How to operationalize such information on a large scale is still an open research question. In order to avoid the pitfalls of the 1970's KR attempts, we do neither propose a new representation language nor a new formalization of "the world". Instead, we take the frame structures of the Berkeley FrameNet II project as a base and transform them into "logic". This idea of proposing logic for representing and reasoning on information stemming from natural language texts is by no means new and has in fact been heavily investigated in computational linguistics (CL) [HSAM93,BK00,dNBBK01,KK03,Bos04, e.g.]. In contrast to the mainstream, which relies on monotonic logic KR languages, we put forward the (nonmonotonic) *answer set programming paradigm* of logic programming. We exploit "nonmonotonicity" primarily as a tool to realize *default values* for role fillers. They allow to reason with defeasible information, which can be retracted when additional contextual information is provided, e.g. in incremental semantic interpretation.

Our main contribution is a transformation of the FrameNet II frames into normal logic programs to be interpreted under the stable model semantics. We have chosen this framework because of its good compromise between expressive power and computational complexity, its declarative nature and the availability of efficient interpreters for large programs [NS96,EFLP00,Wer03].

Our approach goes beyond the formalization of FrameNet I in a description logic in [NFBP02], as we are more concrete about "useful" inference based *reasoning services*. Together with methods for automatic annotation of text documents with frame semantics that are currently developed at various sites, we arrive at an infrastructure that addresses both problems mentioned above in a principled way. We emphasize the modularity and flexibility of our approach, which is needed to support experiments in our "soft" domain of reasoning on information from natural languages sources.

The rest of this paper is structured as follows. After recapitulating some notions from logic programming, we summarize in Section 2 FrameNet, thereby focussing on aspects relevant here. Section 3 is the main part: the translation of FrameNet frames to normal logic programs. Section 4 contains some conclusions and points to future work.

Preliminaries from Logic Programming. We assume the reader familiar with basic concepts of logic programming, in particular with the stable model semantics of normal logic programs [GL88]. See [Bar03] for a recent textbook.

A *rule* is an expression of the form $H \leftarrow B_1, \ldots, B_m, \mathsf{not}\ B_{m+1}, \ldots, \mathsf{not}\ B_n$, where $n \geq m \geq 0$ and H and B_i (for $i = 1, \ldots, n$) are atoms over a given (finite) signature with variables. We assume the signature contains no function symbol of arity greater

than 0 (i.e. the only function symbols are constants).[3] A rule is implicitly universally quantified and thus stands for all its (finitely many) ground instances. The operator "not" is the default negation operator. A *normal logic program* (or *program* for short) consists of finitely many rules. The programs derived below are *domain restricted*[4] or can be turned into domain restricted form easily, so that systems like KRHyper [Wer03] or smodels [NS96] can be applied to compute their stable models. Finally, the head H of a rule may be the special symbol $\bot$, which is intended to mean "false". We assume a rule $A \leftarrow \bot, \mathsf{not}\ A$, where A is a nullary predicate symbol not occurring elsewhere (with this rule, no stable model of any program can satisfy $\bot$).

2 FrameNet

The FrameNet project [BFL98] provides a collection of linguistically motivated so-called *frames* that describe typical situations and their participants[5] and link these to linguistic realizations. A word (or linguistic construction) that *evokes* a frame is called *frame-evoking element* (*FEE*). The participants (or roles) of the frame are called *frame elements* (*FEs*). These are local to particular frames.

Figure 1 shows two frames together with example sentences from the FrameNet data. The frame ACQUIRE is described as *A Recipient acquires a Theme.* [...] *The Source causes the Recipient to acquire the Theme.* [...]. This frame can be evoked by FEEs like *acquire.v, acquisition.n, get.v, obtain.v*.

The second example frame COMMERCE_GOODS-TRANSFER is described as the frame in which [...] *the Seller gives the Goods to the Buyer (in exchange for the Money).* [...]. The FEEs include *buy.v, purchase.v, purchaser.n, rent.v*.

Frame: ACQUIRE	
FE	**Example**
RECIPIENT	**Hornby** *obtained* his first patent in 1901.
SOURCE	You may *get* more money **from the basic pension**.
THEME	We *acquired* **a darts board**.

Frame: COMMERCE_GOODS-TRANSFER	
FE	**Example**
BUYER	**Jess** *bought* a coat.
GOODS	This young man *rented* **the old lady 's room**.
MONEY	Pat *paid* **14 dollars** for the ticket.
SELLER	**Kim** *sold* the sweater.

Fig. 1. Example Frame Descriptions.

Disregarding some details we will present below, all of the following sentences found on the WWW can be analyzed as instances of COMMERCE_GOODS-TRANSFER with BUYER *BMW* and GOODS *Rover*.

(2a) *BMW bought Rover from British Aerospace.*

(2b) *Rover was bought by BMW, which financed* [...] *the new Range Rover.*

[3] With this (common) restriction, all reasoning tasks relevant for us are decidable.

[4] Every variable occurring in a rule must also occur in some non-negated body atom (i.e. in one of $B_1, \ldots, B_m$ in the above notation).

[5] In linguistic terms one should speak of *predicates* and *semantic roles*.

(2c) *BMW's purchase of Rover for $1.2 billion was a good move.*
(2d) *BMW, which acquired Rover in 1994, is now dismantling the company.*

Note that such an analysis solves problem 1 and 2 (from Section 1). First, it generalizes over linguistic variations such as word class or active/passive voice: a query like (1) would match documents containing any of these sentences (given frame annotation). Second, the query would not match documents containing sentences like (3) because in this case the BUYER role is filled with *Ford*.

(3) *Ford's deal to buy Land Rover from BMW is completed.*

Aim of FrameNet. The term *frame* might remind the reader of early approaches in AI as well as CL that did not fully achieve their aim of modeling the world in terms of conceptual structures. Repeating any "ad hoc" modeling is not the aim of FrameNet. Instead, the aim of the FrameNet project is to provide a comprehensive frame-semantic description of the core lexicon of English. The current on-line version of the frame database contains almost 550 frames and 7,000 lexical entries with annotated examples from the British National Corpus.

Frame Relations. Frames can be subdivided into two classes: "small" frames that have a linguistic realization and "big" frames that are more abstract or script-like and serve for structuring the resource. The frame COMMERCIAL_TRANSACTION described above is in fact of the latter kind. It is (via another intermediate frame) related to two *perspectivized* frames COMMERCE_BUY and COMMERCE_SELL that do have realizations in words like *buy.v, purchase.v, purchaser.n* and *price.n, retail.v, sale.n, sell.v*, respectively. The latter two frames share only some FEs with COMMERCIAL_TRANSACTION. E.g. BUYER and MONEY in the case of COMMERCE_BUY. This modeling is based on linguistic theory: sentences like (4) or (5) are linguistically complete in contrast to e.g. (6).

(4) *BMW buys Rover.*
(5) *Daimler-Chrysler sold Mitsubishi.*
(6) * *BMW buys.*

In the latest FrameNet release, a number of relations between frames have been defined and already partly been annotated. Of particular interest for us are the following:

Relation	**Example**
Inherits From	COMMERCE_BY inherits from GETTING.
Uses	COMMERCE_BUY uses FEs BUYER and GOODS from COMMERCE_GOODS-TRANSFER (but not e.g. MONEY).
[Is] Subframe of	COMMERCIAL_TRANSACTION has subframes COMMERCE_GOODS-TRANSFER and COMMERCE_MONEY-TRANSFER.

Inherits From: All FEs of the parent frame are inherited by the child frame, e.g. the frame COMMERCE_BUY inherits the FEs RECIPIENT and THEME from GETTING (modulo a renaming into BUYER and GOODS, respectively[6]).

[6] To keep things simple, we ignore such renamings here. As will become obvious below, our approach includes a renaming facility for roles which can easily be generalized to cover cases like this.

Uses: The Uses relation links a frame to related "background" frames. In most cases it describes partial inheritance as in the example above where COMMERCE_BUY inherits only the FEs BUYER and MONEY from COMMERCIAL_TRANSACTION.

Subframe of: The Subframe relation holds between a frame that stands for a complex event and frames that describe (temporally ordered) sub-events.

These definitions are provided as glosses to human readers. Naturally, from a logical or machine perspective, these specifications are comparatively vague. Our current goal is not to provide once-and-for-all interpretations of these relations (and of roles). Instead, we want to come up with a formalization that supports further research by allowing to experiment with different interpretations. It is by no means obvious whether e.g. a frame instance of a frame *N* that is a subframe of another frame *M* automatically evokes an instance of *M* or e.g. *N*'s siblings. Neither is there a global answer as to which FEs may or must be filled given a concrete instance of a frame. Such decisions may well differ among applications (Section 3.2 discusses some usage scenarios).

The question how natural language sentence are mapped into frame structures is beyond the scope of this paper. It is the central issue of the SALSA project [EKPP03] we are affiliated with.

3 Transformation of FrameNet to Logic Programs

This section contains our main result, the transformation of FrameNet frames to logic programs. To initiate a running example, consider the COMMERCE_BUY frame. This is what FrameNet gives us about it:

Frame: COMMERCE_BUY	
Inherits From	GETTING
FEs	BUYER, GOODS
Subframe of	–
Uses	COMMERCE_GOODS-TRANSFER
FEEs	*buy.v, lease.v, purchase.v, purchase_act.n, purchaser.n, rent.v*

We find it easiest to describe our transformation by starting with a description logic (DL) view of frames (see [BCM+02] for a comprehensive textbook on DL). A natural DL definition – a TBox axiom – of the COMMERCE_BUY frame (neglecting the "Uses" relation) is as follows:

$$\begin{aligned}\text{COMMERCE\_BUY} \equiv\ & \text{GETTING} \\ & \sqcap\ \exists\ \text{COMMERCE\_BUY\_BUYER}.\top \sqcap \exists\ \text{COMMERCE\_BUY\_GOODS}.\top \\ & \sqcap\ \exists\ \text{FEE}.\{\textit{buy.v}, \textit{lease.v}, \textit{purchase.v}, \textit{purchase\_act.n}, \textit{purchaser.n}, \textit{rent.v}\}\end{aligned}$$

Some comments seem due: the role names, such as COMMERCE_BUY_BUYER are prefixed now by the frame name they belong to. This reflects the mentioned local namespace property, which implies that the same role name used in different frames may denote different relations.

Because of using the top concept $\top$, roles may be filled with arbitrary elements — FrameNet does not *yet* provide more specific typing information.[7]

[7] Recently, FrameNet has started to annotate semantic types to frames, FEs, and even FEEs. But this task is far from trivial and the set of types is still preliminary.

The range of FEE is an explicitly defined concept which consists precisely of the stated strings. Such set expressions are available in some DLs.

Our transformation can basically be seen to follow the standard predicate logic semantics of the indicated DL reading of frames. As a significant difference, however, the existential quantifier are treated as *integrity constraints*. Under this view, populating the COMMERCE_BUY class without, say, supplying a filler for the GOODS role will result in an inconsistency, and the currently considered model candidate will be retracted. In contrast, any DL reasoner would then fill the role with a Skolem term in order to satisfy the role restriction. However, with *default values* as introduced in Section 3.2, the effect of existentially quantified roles can be simulated to some degree.

The following description of our transformation is separated into three parts, each one treating a different aspect.

3.1 Basic Frame Transformation

For the purpose of this paper, we describe a frame named N as the sets $IsA(N)$, $FE(N)$, $Uses(N)$, and $FEE(N)$, which are precisely the frame names listed as "Inherits From" at N, the role names listed as "FE", the frame names listed as "Uses", and the strings listed as "FEE", respectively. Each set may be empty. In particular, if $FEE(N)$ is empty, this indicates that N is a "big" frame without linguistic realization.

We also need the following (recursive) definition. For a given frame N, $FE^\star(N)$ consists of all roles of N, including the ones to be inherited. Formally, define

$$FE^\star(N) = FE(N) \cup \{FE^\star(M) \mid M \in IsA(N)\} \ .$$

To describe our main transformation "basic", we found it helpful to single out a certain aspect, viz., mapping of specific roles between two frames N and M[8]:

Transformation: partialRoleMapping(N, M, FEs)
Input: N, M: frames names; FEs: set of role names
Output: the following rules:

$(N \Rightarrow M)$	*$(M \Rightarrow N)$*
For each $FE \in FEs$ the rule	For each $FE \in FEs$ the rule
$M\_FE(x,y) \leftarrow N(x), N\_FE(x,y)$	$N\_FE(x,y) \leftarrow M(x), M\_FE(x,y)$

The partialRoleMapping transformation maps the fillers of roles of the frame N (if present) to fillers of roles of the frame M, and vice versa. Such a mapping is needed because of the local namespace property of roles of frames (as explained above). It "translates" roles with the same names between frames by following the convention to include in a role name the frame it belongs to. Based on this transformation, we can now introduce the announced basic transformation.

[8] Notational conventions: x, y, z denote object-level variables, *italic* font is used for schematic variables of transformations, and sans serif font is used for names to be taken literally.

Transformation: $\mathsf{basic}(N)$
Input: N: frame name
Output: the following rules:

($\Rightarrow$-direction (1))

For each $\mathsf{M} \in \mathit{IsA}(N)$ the rule

$$M(x) \leftarrow N(x)$$

($\Rightarrow$-direction (2))

For each $FE \in FE^{\star}(N)$ the rule

$$\bot \leftarrow N(x), \mathsf{not\ some\_}N\mathsf{\_}FE(x)$$

($\Rightarrow$-direction (3))

If $FEE(N) \neq \emptyset$, the rule

$$\bot \leftarrow N(x), \mathsf{not\ some\_}N\mathsf{\_FEE}(x)$$

($\Leftarrow$-direction)

Let $FE(N) = \{FE_1, \ldots, FE_k\}$, for some $k \geq 0$.
Let $\mathit{IsA}(N) = \{M_1, \ldots, M_n\}$, for some $n \geq 0$.

The rule

$$N(x) \leftarrow \mathsf{some\_}N\mathsf{\_}FE_1(x), \ldots, \mathsf{some\_}N\mathsf{\_}FE_k(x), M_1(x), \ldots, M_n(x), \mathsf{some\_}N\mathsf{\_FEE}(x)$$

(If $FEE(N) = \emptyset$, then the body atom $\mathsf{some\_}N\mathsf{\_FEE}(x)$ is omitted)

(Role inheritance)

For each $\mathsf{M} \in \mathit{IsA}(N)$, the result of $\mathsf{partialRoleMapping}(N, M, FE^{\star}(M))$

(Auxiliary definitions (1))

For each $FE \in FE^{\star}(N)$ the rule

$$\mathsf{some\_}N\mathsf{\_}FE(x) \leftarrow N\mathsf{\_}FE(x,y)$$

(Auxiliary definitions (2))

For each $FEE \in FEE(N)$ the rule

$$\mathsf{some\_}N\mathsf{\_FEE}(x) \leftarrow N\mathsf{\_FEE}(x, FEE)$$

Some comments: the *$\Rightarrow$-direction (1)* rule should be obvious. The *$\Rightarrow$-direction (2)* rules express the integrity constraint viewpoint of existentially quantified roles. The $\mathsf{some\_}N\mathsf{\_}FE$ predicate used there is defined under *Auxiliary definitions (1)*. There, in the body atom $N\mathsf{\_}FE(x,y)$, the variable x stands for a frame instance (token) and y stands for the role filler. The test for whether the roles are filled or not has to be done for all roles, including the inherited ones. This explains the use of $FE^{\star}(N)$ there. The *$\Rightarrow$-direction (3)* rules are similar to the rules under *$\Rightarrow$-direction (2)*, this time testing for the presence of a FEE filler (if N prescribes FEEs at all); it uses the rules under *Auxiliary definitions (2)*. There, in the body atom $N\mathsf{\_}FEE(x,y)$, the variable x again stands for a frame instance (token).

The *$\Leftarrow$-direction* rule derives that an individual x is an instance of N if (i) all its roles $FE(N)$ are filled (and also $FEE(N)$ if present), and (ii) x belongs to all the frames that N inherits from. Notice it is not necessary in the rule body to test if *all* the roles $FE^{\star}(N)$ of N are filled for x, because the inherited ones *must* have been filled due to (ii) when the rule is applied. The *Role inheritance* rules map the role fillers of inherited roles to role fillers for instances of N, as explained. Notice that the $\mathsf{partialRoleMapping}$ transformation realizes this mapping in the converse direction, too. Indeed, because it is applied to the inherited roles only, this is what is expected.

3.2 Default Values

Depending on the frame and concrete application, it may be useful to *not* consider a "missing" role filler as an indication of inconsistency in the current model candidate. For instance, an utterance like *BMW bought* [*at high risk*]. might well be taken to fill a

COMMERCE_BUY frame. In order to achieve consistency, the GOODS role of the COMMERCE_BUY instance populated by a linguistic text analysis component in reaction to this sentence has to be filled, and a dummy value, say, unspecified_FE could be used as a substitute for a more specific, preferable one. This suggests to use *default values*. Fortunately, default values can be incorporated without effort in our setting by the following transformation.

Transformation: defaultValue(N, FE)
Input: N: frame name; FE: a role name
Output: the following rules, with the free predicate symbol default_N_FE:

(*Choice of fill with default value or not)*

$$N\_FE(x,y) \leftarrow \mathsf{not\ not}\_N\_FE(x,y), N(x), \mathsf{default}\_N\_FE(x,y)$$

$$\mathsf{not}\_N\_FE(x,y) \leftarrow \mathsf{not}\ N\_FE(x,y), N(x), \mathsf{default}\_N\_FE(x,y)$$

(Case of waiving default value)

$$\bot \leftarrow N(x), \mathsf{default}\_N\_FE(x,y), N\_FE(x,y), N\_FE(x,z), \mathsf{not\ equal}(y,z)$$

The left rules represent an even cycle through default negations, similar as in the propositional program $A \leftarrow \mathsf{not}\ B$, $B \leftarrow \mathsf{not}\ A$. For this program there are exactly two stable models: one where A is true and B is false, and one the other way round. Using such even cycles as a "choice" operator is a well-known programming technique. Here, it realizes two models, one where $N\_FE(x,y)$ is true, and one where it is false. The right rule expresses that there cannot be a default value as a role filler in presence of another, different (default) role filler[9].

The transformation for default values is completed by the following two rules. They express that there must be at least one role filler for the considered role:

$$\bot \leftarrow N(x), \mathsf{not\ some}\_N\_FE(x) \qquad \mathsf{some}\_N\_FE(x) \leftarrow N\_FE(x,y)$$

However, because these rules are readily obtained from the basic transformation when applied to N, they need not be generated and thus are excluded from the transformation.

Notice the resulting program does not include rules to define the default_N_FE predicate. Such rules are external to the transformation and should be supplied to provide default values as appropriate[10]. If none is supplied, then the rules obtained from the defaultValue transformation are vacuously true and hence are insignificant for the result (the stable models). This suggests to apply the defaultValue transformation along with the basic transformation, for all roles, and nothing will be lost.

The usefulness of our approach clearly depends on its flexibility to interact with other components of a larger system. As the following considerations show, we expect

[9] The "equal" predicate, which means syntactic equality, can be defined as $\mathsf{equal}(x,x) \leftarrow$.

[10] A designated constant like unspecified_FE can be taken to supply a uniform default value by adding facts default_N_$FE(x,$ unspecified_FE$)$ for certain frames N and roles FE.

the defaultValue facility to be particularly important in this regard. In general, it is flexible enough to express domain-independent, domain-dependent, frame-dependent, or situation-dependent default values.

(1) If the application is such that the, say, GOODS role must be filled in order to meaningfully process a COMMERCE_BUY frame, then no default value should be supplied.

(2) Specific settings might allow for plausible default values. For instance, in a stock market domain, a uniform default value could be supplied as

$$\mathsf{default\_COMMERCE\_BUY\_GOODS}(x, \mathsf{share}) \leftarrow \mathrm{COMMERCE\_BUY}(x) \ ,$$

where share is an instance of an appropriate frame representing shares.

(3) Consider again the *BMW bought at high risk* example. The anaphora resolution component of a NLP inference system[11] might find out that either rover or chrysler would be a suitable role filler for the GOODS role. This disjunctive information can be represented by the following two facts (suppose e is an instance of the COMMERCE_BUY frame we consider):

$$\begin{array}{l} \mathsf{default\_COMMERCE\_BUY\_GOODS}(\mathsf{e}, \mathsf{rover}) \leftarrow \\ \mathsf{default\_COMMERCE\_BUY\_GOODS}(\mathsf{e}, \mathsf{chrysler}) \leftarrow \ . \end{array}$$

An analysis of the resulting program shows there are two stable models: one with rover as the only GOODS role filler for e, and chrysler in the other model. The existence of the two stable models thus represents the uncertainty about the role filler in question; it has the same effect as disjunctively assigning the two fillers to the GOODS role (if disjunction were available in the language). [12].

(4) It may make sense to supply default values for more than one role. A sentence like *The purchase was risky.* may give rise to populate a COMMERCE_BUY frame where both the BUYER and the GOODS role are filled with, say, a default value unspecified_FE.

(5) The assumption that the FEEs listed in a frame is a linguistically exhaustive listing might be too strong. For instance, instead of a FEE listed, some anaphoric expression might be present. A possible solution is to include in the FEEs an additional element, say, unspecified_FEE that acts as a default value. The realization is through the default value transformation applied to N and FEE, $\mathsf{defaultValue}(N, \mathit{FEE})$ (thus treating FEE as a role), and adding a rule $\mathsf{default\_}N\mathsf{\_}\mathit{FEE}(x, \mathsf{unspecified\_FEE}) \leftarrow N(x)$.

3.3 The Uses Relation

For a human reader, the Uses relation links a frame under consideration to other frames that are relevant to understand the situation it describes. For example, to understand a buying situation, one has to be aware of what a goods transfer situation is. From a

[11] Anaphora resolution based on deductive methods is investigated e.g. in [BK00,dNBBK01].

[12] This approach thus is in line with those logic-based approaches in computational linguistics that represent a dialogue by a collection of models, which can be further pruned as more information becomes available. See e.g. [KK03,Bos04] for recent work. As a difference, we are working with a *nonmonotonic* logic instead of classical logic.

formal perspective, it seems to mean a partial inheritance relation, where inheritance is restricted to the roles common to a frame and a frame it is in the Uses relation with. We propose the following transformation[13]:

Transformation: $\mathsf{uses}(N)$
Input: N: a frame name
Output: the following rules:

($N \Rightarrow Uses(N)$ (1))

For each $M \in Uses(N)$ the rule

$$M(x) \leftarrow N(x)$$

($N \Rightarrow Uses(N)$ (2))

For each $M \in Uses(N)$,
for each $FE \in FE^\star(M) \setminus FE(N)$ the rule

$$\mathsf{default\_}M\mathsf{\_}FE(x, \mathsf{unspecified\_FE}) \leftarrow N(x)$$

($Uses(N) \Rightarrow N$)

Let $FE(N) \setminus \{FE^\star(M) \mid M \in Uses(N)\}$
$= \{FE_1, \ldots, FE_k\}$, for some $k \geq 0$.
Let $Uses(N) = \{M_1, \ldots, M_n\}$, for some $n \geq 0$.

The rules

$$N(x) \leftarrow \mathsf{some\_}N\mathsf{\_}FE_1(x), \ldots, \mathsf{some\_}N\mathsf{\_}FE_k(x), M_1(x), \ldots, M_n(x), \mathsf{some\_}N\mathsf{\_FEE}(x)$$

(If $FEE(N) = \emptyset$, then the body atom $\mathsf{some\_}N\mathsf{\_FEE}(x)$ is omitted)

(Partial role inheritance)

For each $M \in Uses(N)$, the result of $\mathsf{partialRoleMapping}(N, M, FE^\star(M) \cap FE(N))$

This transformation treats the Uses relation in a similar way as the basic transformation treats the Inherits From relation. The Uses relation also defines an inheritance hierarchy of roles, in parallel to the Inherits From relation, however where only explicitly stated roles are inherited. These are precisely those roles in $FE(N)$ that are also roles of some concept M that N uses. The set $\{FE_1, \ldots, FE_k\}$ mentioned under $Uses(N) \Rightarrow N$ therefore is the complementary set of roles, those that are *not* inherited. Only those have to be tested for being filled, in analogy to what the rule under $\Leftarrow$*-direction* in the basic transformation does (see explanation there).

The rules under *Partial role inheritance* are mappings precisely for the inherited roles, individually for each frame M that N uses, i.e. the roles $FE^\star(M) \cap FE(N)$. By definition of partialRoleMapping, these roles are mapped also "upwards", from N to a frame M that N uses. The remaining roles of such a frame M are the roles $FE^\star(M) \setminus FE(N)$, and for these roles default values are supplied by the $N \Rightarrow Uses(N)$ *(2)* rules. Together, thus, and in conjunction with the rule under $N \Rightarrow Uses(N)$ *(1)*, this has the effect that M will be populated with all roles filled whenever N is populated. Finally, notice that some definitions for rules mentioned can be skipped, because they are part of the basic transformation.

We would like to point out that the transformation for the Uses relation is not meant to be conclusive. Experiments on real data may suggest a different treatment of the Uses relation. It will also be interesting to devise suitable transformations of the Subframe relation.

[13] There is no *precise* description of the Uses relation available yet. FrameNet is considering a subdivision of this relation.

3.4 Usage Scenarios

The transformations described so far are intended to be applied to the whole FrameNet. More precisely, if $\mathcal{N}$ is a set of frames, such as those of FrameNet II, then we consider the logic program $P(\mathcal{N}) = \bigcup_{N \in \mathcal{N}} \mathsf{basic}(N) \cup \{\mathsf{defaultValue}(N, FE) \mid FE \in FE^{\star}(N)\} \cup \mathsf{uses}(N)$, possibly additionally equipped with default values for specific roles as discussed in Section 3.2 and additional facts stemming from linguistic text analysis components. We have chosen to transform into normal logic programs, because its associated answer set programming paradigm provides a good compromise between expressive power and computational complexity, its declarative nature and the availability of suitable, state-of-the-art interpreter like KRHyper [Wer03] or smodels [NS96][14]. These systems are made to cope with programs far larger than the ones resulting in our case[15]. They are capable of enumerating the stable models of $P(\mathcal{N})$, which can be inspected by other system components to determine the result of the overall computation.

In the usage scenarios we have in mind, "small" frames, those that have a linguistic realization (i.e. those having a FEE property), shall be populated as a result of textual analysis. By the way the transformation is defined, the information in these frame instances is combined by transferring it up the Inherits from and Uses hierarchies, thereby providing default values as needed. This way, explicitly presented knowledge shall be completed to get more abstract views on it. To make this a little more concrete, probably the most basic class of inference covers systematic syntax-near cases a (linguistic) automatic frame assignment system cannot cover. Take e.g. the following two sentences.

(7a) *Mary promised to purchase a BMW.*

(7b) *Mary promised the purchase of a BMW.*

The second sentence might in many cases be equivalent in meaning to the first sentence. But most linguistic (syntactic) theories don't have the means to describe this equivalence. The problem is that the subject of this sentence *Mary* fills the subject position of the verb *promise*. But *Mary* is also understood as subject of the verb *purchase* and (probably) as the actor acting in the event described by the noun *purchase*. For the verb case, linguistic theory has an answer in terms of subject sharing of so-called *subject control verbs* like *promise, offer, deny*. Here, the subject of the control verb is know to be identical with the subject of the embedded verb. But in the second sentence, there is no embedded verb and nouns are not considered to have a subject.

In contrast, in a FrameNet analysis both, verb and noun evoke a COMMERCE_BUY frame. But as we argued only in the verb case, syntax based methods can fill the BUYER role with *Mary*. Here, a defeasible inference could fill the respective role in the noun case. This inference is defeasible because the sentence might continue as follows.

(8) *Mary promised the purchase of a BMW by her husband before the vacations start.*

---

[14] In particular, the resulting programs are domain-restricted, as required by these systems or can easily be made conforming.

[15] The number of rules in $P(\mathcal{N})$ is quadratic in $|\mathcal{N}|$. The quadratic component derives from the partialRoleMapping transformation, which, fortunately, results in very simple rules that can be worked off deterministically.

In such cases where an actual filler is available, the inference mechanism should fill the respective role with that.

A more ambitious kind of inference is involved in the following example. The phrase (9) is a real corpus example from a text talking about questions of possession in the former GDR.

(9) *Owner of two-family houses which have bought before 1989* [. . .].

For this example, automatic syntax-based methods would return two frames, POSSESSION with OWNER *Owner of two-family houses* and POSSESSION *two-family houses*, and COMMERCE_BUY with BUYER *which*. Depending on the depth of the linguistic analysis, *which* might have already been resolved to *Owner of two-family houses*. But in any case, the GOODS of the COMMERCE_BUY were empty. At this point an heuristic inference could infer that the GOODS are the houses. If additional knowledge was available about the relation of buying and possession (e.g. by FrameNet's Causative relation), this should be used here as well. Once again, the inference is defeasible as the context might tell us that the text is about owner of two-family houses which have bought, say a car, before 1989.

4 Conclusion and Outlook

In this paper we have argued that reasoning services on the basis of FrameNet's frames can satisfy the growing demand of integrating semantic information in order to improve large scale natural language processing, such as document searching.

Our aim was to arrive at an infrastructure that supports testing different formalizations of the frame relations on a larger amount of corpus data. To this end, we gave transformations of the lexicographic frame and frame relation definitions into a logic programming setting, which we expect to be feasible also with respect to practical efficiency considerations. Although our translation of the frames and frame hierarchy are in the spirit of description logics, we have argued that both, the vague specification of the frame relations and the defeasible character of the kinds of inferences we are interested in do not lead naturally to characterization within description logics.

It has to be stressed that what we presented here is work in progress. The transformations proposed are not too difficult to implement, and we will conduct a number of pilot studies within different settings. Once e.g. the SALSA [EKPP03] project will supply methods for automatic frame assignment to natural text, we have a basic architecture for semantics-based natural language processing as described in the introduction of this paper. We are well aware that our system might undergo a number of changes underway not only because the FrameNet resource itself is still developing.

The kinds of inference we want to model on the basis of what we present here cannot be characterized by criteria such as soundness or completeness with respect to a readily defined semantics of FrameNet[16]. Their appropriateness or usefulness primarily

[16] In fact, our transformation *equips* FrameNet with a precise, declarative semantics by means of the transformations proposed. Nevertheless, some interesting properties of our transformation can be proven. For instance, that arguable reasonable properties of "inheritance" are realized. We did not do so here for lack of space.

depends on such factors as the application at hand and on additional linguistic or extra-linguistic evidence available.

Our long-term goals include a treatment of selectional *preferences* (rather than *restrictions*) which will enable a more fine-grained modeling of e.g. sortal information about the filler of particular roles. For example, in Fig 1 *from the basic pension* fills the role SOURCE of frame ACQUIRE which is perfectly acceptable for a human. This example shows that a formalization of sortal information will have to include mechanisms for dealing with preferences and type casting (e.g. to deal with typical linguistic patterns like metonymies as in *Washington announces a new drug policy*). Including preferences would also make it possible to formulate *heuristic inferences* beyond our current assignment of default values.

Acknowledgements. We thank the anonymous referees for their helpful comments. The detailed comments and suggestions we found very valuable to improve the paper.

References

[Bar03] C. Baral. *Knowledge representation, reasoning and declarative problem solving*. Cambridge University Press, 2003.

[BCM+02] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *Description Logic Handbook*. Cambridge University Press, 2002.

[BFL98] C. F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In *Proc. of COLING-ACL-98*, Montreal, Canada, 1998.

[BK00] P. Baumgartner and M. Kühn. Abducing Coreference by Model Construction. *Journal of Language and Computation*, 1(2):175–190, 2000.

[Bos04] J. Bos. Computational semantics in discourse: Underspecification, resolution, and inference. *Journal of Logic, Language and Information*, 13(2):139–157, 2004.

[dNBBK01] H. de Nivelle, P. Blackburn, J. Bos, and M. Kohlhase. Inference and computational semantics. *Studies in Linguistics and Philosophy, Computing Meaning*, 77(2):11–28, 2001.

[EFLP00] T. Eiter, W. Faber, N. Leone, and G. Pfeifer. Declarative problem-solving using the DLV system. In *Logic-based artificial intelligence*, pages 79–103. Kluwer, 2000.

[EKPP03] K. Erk, A. Kowalski, S. Pado, and M. Pinkal. Towards a resource for lexical semantics: A large German corpus with extensive semantic annotation. In *Proc. of ACL-03*, Sapporo, Japan, 2003.

[Fel98] C. Fellbaum, editor. *WordNet. An electronic lexical database*. MIT Press, Cambridge/Mass., 1998.

[GL88] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In Robert Kowalski and Kenneth Bowen, editors, *Proc. of 5th ICLP*, 1988.

[HSAM93] J. R. Hobbs, M. E. Stickel, D. E. Appelt, and P. Martin. Interpretation as abduction. *Artificial Intelligence*, 63(1-2):69–142, 1993.

[KK03] M. Kohlhase and A. Koller. Resource-adaptive model generation as a performance model. *Logic Journal of the IGPL*, 11(4):435–456, 2003.

[NFBP02] S. Narayanan, C. J. Fillmore, C. F. Baker, and M. R. L. Petruck. FrameNet Meets the Semantic Web: A DAML+OIL Frame Representation. In *Proc. of AAAI*, 2002.

[NS96] I. Niemelä and P. Simons. Efficient implementation of the well-founded and stable model semantics. In *Proc. of JICSLP*, Bonn, Germany, 1996. The MITPress.

[Wer03] C. Wernhard. System Description: KRHyper. Fachberichte Informatik 14–2003, Universität Koblenz-Landau, 2003.