Bilingual Hebrew-English Generation of Possessives and Partitives: Raising the Input Abstraction Level

Yael Dahan Netzer and **Michael Elhadad**
Ben Gurion University
Department of Mathematics and Computer Science, Beer Sheva, 84105, Israel
(yaeln|elhadad)@cs.bgu.ac.il

Abstract

Syntactic realization grammars have traditionally attempted to accept inputs with the highest possible level of abstraction, in order to facilitate the work of the components (sentence planner) preparing the input. Recently, the search for higher abstraction has been, however, challenged (Elhadad and Robin, 1996)(Lavoie and Rambow, 1997)(Busemann and Horacek, 1998). In this paper, we contribute to the issue of selecting the "ideal" abstraction level in the input to syntactic realization grammar by considering the case of partitives and possessives in a bilingual Hebrew-English generation grammar. In the case of bilingual generation, the ultimate goal is to provide a single input structure, where only the open-class lexical entries are specific to the language. In that case, the minimal abstraction required must cover the different syntactic constraints of the two languages.

We present a contrastive analysis of the syntactic realizations of possessives and partitives in Hebrew and English and conclude by presenting an input specification for complex NPs which is slightly more abstract than the one used in SURGE. We define two main features – *possessor* and *ref-set* and discuss how the grammar handles complex syntactic co-occurrence phenomena based on this input. We conclude by evaluating how the resulting input specification language is appropriate for both languages.

1 Introduction

One of the first issues to address when selecting a syntactic realization component is whether its input specification language fits the desired application. Traditionally, syntactic realization components have attempted to raise the abstraction level of input specifications for two reasons: (1) to preserve the possibility of paraphrasing and (2) to make it easy for the sentence planner to map from semantic data to syntactic input

As new applications appear, that cannot start generation from a semantic input because such an input is not available (for example re-generation of sentences from syntactic fragments to produce summaries (Barzilay et al., 1999) or generation of complex NPs in a hybrid template system for business letters (Gedalia, 1996)), this motivation has lost some of its strength. Consequently, "shallow surface generators" have recently appeared (Lavoie and Rambow, 1997) (Busemann and Horacek, 1998) that require an input considerably less abstract than those required by more traditional realization components such as SURGE (Elhadad and Robin, 1996) or KPML (Bateman, 1997).

In this paper, we contribute to the debate on selecting an appropriate level of abstraction by considering the case of bilingual generation. We present results obtained while developing the HUGG syntactic realization component for Hebrew (Dahan-Netzer, 1997). One of the goals of this system is to design a generator with an input specification language as similar as possible to that of an English generator, SURGE in our case.

The ideal scenario for bilingual generation is illustrated in Figure 1. It consists of the

John gave a book to Mary
John natan sefer le-Mary

$$\left[\begin{array}{ll} \text{cat} & \text{clause}\\ \text{proc} & \left[\begin{array}{ll}\text{type} & \text{composite}\\ \text{relation-type} & \text{possessive}\end{array}\right]\\ \text{partic} & \left[\begin{array}{ll} \text{agent} & \left[\begin{array}{ll}\text{cat} & \text{proper}\\ \text{lex} & \text{'John'}\\ \text{gender} & \text{masculine}\end{array}\right]\\ \text{affected} & [1]\left[\begin{array}{ll}\text{cat} & \text{proper}\\ \text{lex} & \text{'Mary'}\\ \text{gender} & \text{feminine}\end{array}\right]\\ \text{possessor} & [1]\\ \text{possessed} & \left[\begin{array}{ll}\text{cat} & \text{common}\\ \text{lex} & \text{'book/sefer'}\end{array}\right] \end{array}\right]\end{array}\right]$$

Figure 1: Ideal scenario for bilingual generation

following steps:

1. Prepare an input specification in one language
2. Translate all the lexical entries (function words do not appear)
3. Generate with any grammar

In the example, the same input structure is used and the generator can produce sentences in both languages if only the lexical items are translated.

Consider the following paraphrase in English for the same input: *John gave Mary a book*.

The Hebrew grammar does not produce such a paraphrase, as there is no equivalent in Hebrew to the dative move alternation. In this case, we conclude that the input abstraction level is appropriate. In contrast, if the input had specified a structure such as indirect-object(prep=to/le, np=Mary), then it would not have been abstract enough to serve as a bilingual input structure.

Similarly, the English possessive marker is very close to the Hebrew "construct state" (smixut):

The King's palace
Armon ha-melex
Palace-cs the-king

The following input structure seems, therefore, appropriate for both languages:

$$\left[\begin{array}{ll} \text{cat} & \text{common}\\ \text{lex} & \text{``palace''/``armon''}\\ \text{possessor} & \left[\begin{array}{ll}\text{lex} & \text{``king''/``melex''}\\ \text{definite} & \text{yes}\end{array}\right]\end{array}\right]$$

There are, however, divergences between the use of smixut in Hebrew and of the possessive marker in English:

Segovia's pupil	*The pupil of Segovia*
** talmyd segovyah*	*talmyd Sel segovyah*
? The house's windows	*The windows of the house*
Halonot ha-bayit	*ha-Halonot Sel ha-bayit*

Our goal, therefore, is to design an input structure that is abstract enough to let the grammar decide whether to use a possessive marker vs. an of-construct in English or a Sel-construct vs. a smixut-construction in Hebrew.

A similar approach has been adopted in generation (Bateman, 1997), (Bateman et al., 1991) and in machine translation most notably in (Dorr, 1994). Dorr focuses on divergences at the clause level as illustrated by the following example:

I like Mary
Maria me gusta a mi
Mary pleases me

Dorr selects a representation structure based on Jackendoff's Lexical Conceptual Structures (LCS) (Jackendoff, 1990).

In the KPML system, the proposed solution is based on the systemic notion of "delicacy" and the assumption is that low-delicacy input features (the most abstract ones) remain common to the two target languages and high-delicacy features would differ.

In this paper, we focus on the input specification for complex NPs. The main reason for this choice is that the input for NPs in SURGE has remained close to English syntax (low abstraction). It consists of the following main sub-constituents: head, classifier, describer, qualifier and determiner.

In previous work (Elhadad, 1996), we discuss how to map a more abstract domain-specific representation to the SURGE input structure within a sentence planner. When moving to a bilingual generator, we have found the need for a higher level of abstraction to avoid encoding language-specific knowledge in the sentence planners. We specifically discuss here the following decisions:

- How to realize a possessive relation: *John's shirt* vs. *the shirt of John*
- How to realize a partitive relation: *all the kids* vs. *all of the kids*

In the rest of the paper, we first present basic contrastive data and existing analyses about possessives and partitives in Hebrew and English. We then present the input features we have designed to cover possessives and partitives in both languages and discuss how these features are used to account for the main decisions required of the realizer. We conclude by an evaluation of the bilingual input structure on a set of 100 sample input structures for complex NPs in the two languages and of the divergences that remain in the generated NPs. In conclusion, this bilingual analysis has helped us identify important abstractions that lead to more fluent generation in both languages.

2 Possessives and Partitives in Hebrew and English

This section briefly presents data on possessives and partitives in English and Hebrew. These observations delimit the questions we address in the paper: when is a genitive construct used to express possessives and when is an explicit partitive used.

2.1 Possessives in English

Possessives can be realized in two basic structures: as part of the determiner sequence (Halliday, 1994) (as either a possessive pronoun or a full NP marked with apostrophe-s as a genitive marker) or as a construct *NP of NP*.

In addition to possessive, the genitive marker can realize several semantic relations (Quirk et al., 1985) (pp.192–203): subjective genitive (*the boy's application — the boy applied*) , genitive of origin (*the girl's story — the girl told a story*), objective genitive, descriptive genitive (*a women's college — a college for woman*).

As a consequence of this versatility, the general decision of apostrophe vs. *of* is not trivial: Quirk claims that the higher on the gender scale, *i.e.,* the more animate the noun, the more the possessor realization tends to be realized as an inflected genitive:

- Person's name: *Segovia's pupil*
- Person's nouns: *the boy's new shirt*
- Collective nouns: *the nation's social security*
- Higher Animals: *the horse's neck*
- Geographical names: *Europe's future*
- Locative nouns: *the school's history*
- Temporal nouns: *the decade's event*

This decision also interacts with other realization decisions: if several modifiers must be attached to the same head, they can compete for the same slot in the syntactic structure. In such cases, the decision is one of preference ranking: *The boy's application of last year* vs. *last year's application of the boy.*

2.2 Possessives in Hebrew

Possessives in Hebrew can be realized by three syntactic constructions:

construct state	**free genitive**	**double genitive**
cadur ha-tynok	*ha-cadur Sel ha-tynok*	*cadur-o Sel ha-tynok*
ball the-baby	*the ball of the baby*	*ball-his of the-baby*

The construct state (called *smixut*) is similar to the apostrophe marker in English: it involves a noun adjacent to another noun or noun phrase, without any marker (like a preposition) between them (Berman, 1978). The head noun in the construct form generally undergoes morphological changes: *yaldah - yaldat*. *Smixut* is, on the one hand, very productive in Hebrew and yet very constrained (Dahan-Netzer and Elhadad, 1998b).

Free genitive constructs use a prepositional phrase with the preposition *Sel*. Many studies treat *Sel* as a case marker only (cf. (Berman, 1978) (Yzhar, 1993) (Borer, 1988)).

The choice of one of the three forms seems to be stylistic and vary in spoken and written Hebrew (cf. (Berman, 1978), (Glinert, 1989), (Ornan, 1964), and discussion in (Seikevicz, 1979)). But, in addition to these pragmatic factors and as is the case for the English genitive, the construct state can realize a wide variety of semantic relations (Dahan-Netzer and Elhadad, 1998b), (Azar, 1985), (Levi, 1976). The selection is also a matter of preference ranking among competitors for the same syntactic slot. For example, we have shown in (Dahan-Netzer and Elhadad, 1998b) that the semantic relations that can be realized by a construct state are the ones defined as classifier in SURGE. Therefore, the co-occurrence of such a relation with another classifier leads to a competition for the syntactic slot of "classifier" and also contributes to the decision of how to realize a possessive.

Consider the following example:

$$\begin{bmatrix} \text{cat} & \text{common} \\ \text{head} & \begin{bmatrix} \text{lex} & \text{“Simlah”/“dress”} \end{bmatrix} \\ \text{classifier} & \begin{bmatrix} \text{lex} & \text{“Sabat”} \end{bmatrix} \\ \text{possessor} & \begin{bmatrix} \text{cat} & \text{common} \\ \text{lex} & \text{“yalda”/“girl”} \end{bmatrix} \end{bmatrix}$$

If only the possessor is provided in the following input, it can be mapped to a construct state:

Simlat ha-yaldah
dress-cs the-girl
the girl's dress

If a classifier is provided in addition, the construct-state slot is not available anymore[1], and the free genitive construct must be used:

Simlat ha-Sabat Sel ha-yaldah
dress-cs the-Shabat of the-girl
The Shabat dress of the girl

2.3 Partitives in English

The partitive relation denotes a subset of the thing to which the head of a noun phrase refers. A partitive relation can be realized in two main ways: as part of the pre-determiner sequence (Halliday, 1994), (Winograd, 1983) using quantifiers that have a partitive meaning (*e.g., some/most/many/one-third (of the) children*) or using a construction of the form *a measure/X of Y*.

There are three subtypes of the partitive construction ((Quirk et al., 1985)[p.130], (Halliday, 1994)): measure *a mile of cable*, typical partitives *a loaf of bread, a slice of cake*, and general partitives: *a piece/bit/of an item of X*.

In the syntactic structure of a partitive structure, the part is the head of the phrase (and determines agreement), but the Thing - is what is being measured. This creates an interesting difference between the logical and syntactic structure of the NP.

(Mel'cuk and Perstov, 1987) defines the *elective* surface syntactic relation which connects an *of*-phrase to superlative adjectives or numerals. An elective phrase is an elliptical structure: *the rightmost [string] of the strings*. It can be headed by an adjective in superlative form (*the poorest among the nation*), a numeral (*45 of these 256 sentences*), ordinal (*the second of three*) or a quantitative word having the feature *elect*: *all, most, some of...* The elective relation can be used recursively (*Many of the longest of the first 45 of these 256 sentences*).

In the case of quantifier-partitives, one must decide whether to use an explicitly partitive construct (*some of the children*) or not (*some children*). The structure that does not use *of* is used for generic NPs (when the head is non-definite: *most children*). For specific reference, the *of*-construction is optional with nouns and obligatory with pronouns:

all (of) the meat
all of it

2.4 Partitives in Hebrew

There are two possible ways to express partitivity in Hebrew: using a construction of

[1] If the classifier had been specified in the input as a semantic relation as discussed in (Dahan-Netzer and Elhadad, 1998b), an alternative realization (*The girl's dress for Shabat*) could have been obtained.

the form *X me-Y*, or using a partitive quantifier. In contrast to English, quantifiers that are marked as partitive, cannot be used in an explicitly partitive structure:

*rov ha-yeladym – * rov me-ha-yeladym – most of the children*
*Se'ar ha-yeladym – * Se'ar me-ha-yeladym – the rest of the children*
*col ha-yeladym – * col me-ha-yeladym – all of the children*

Conversely, a quantifier that is *not* marked as partitive can be used in an explicitly partitive structure:

harbeh yeladym – many children
harbeh me-hayeladym – many of the children
mewat ha-yeladym – few the-children
mewat me-ha-yeladym – few of the-children

There are complex restrictions in Hebrew on the co-occurrence of several determiners in the same NP and on their relative ordering within the NP. To explain them, Glinert (Glinert, 1989) adopts a functional perspective, quite appropriate to the needs of a generation system, and identifies a general pattern for the NP, that we use as a basis for the mapping rules in HUGG:

```
[partitive determiner amount head
classifiers describers
post-det/quant qualifiers]
```

Yzhar and Doron (Doron, 1991) (Yzhar, 1993) distinguish between two sets of determiners, that they call D and Q quantifiers. The distinction is based on syntactic features, such as position, ability to be modified, ability to participate in partitive structures and requirement to agree in number and gender with the head. This distinction is used to explain co-occurrence restrictions, the order of appearance of D vs Q quantifiers and the recursive structure of D determiners: D determiners can be layered on top of other D determiners. A single Q quantifier can occur in an NP and it remains attached closest to the head.

In (Dahan-Netzer, 1997) and (Dahan-Netzer and Elhadad, 1998a), we have refined the D/Q classification and preferred using functional criteria: we map the Q quantifiers to the "amount" category defined by Glinert, and the D set is split into the partitive and determiner categories – each with a different function. Of these, only partitives are recursive.

Given these observations, the following decisions must be left "open" in the input to the realizer: how to map a possessor to different realizations; in which order to place co-occurring quantifiers; and whether to use an explicit *of* construct for partitive quantifiers. The input specification language must also enforce that only acceptable recursive structures be expressible.

3 Defining an Abstract Input for NP Realization

3.1 Input Features

The input structure for NPs we adopt is split in four groups of features, which appear in Figure 3.1:

- Head or reference-set: defines the thing or set referred to by the NP
- Qualifying: adds information to the thing
- Identifying: identifies the thing among other possible referents
- Quantifying: determines the quantity or amount of the thing.

The main modifications from the existing SURGE input structure are the introduction of the *ref-set* feature and the update of the usage of the *possessor* feature.

For both of these features, the main requirement on the realizer is to properly handle cases of "competition" for the same restricted syntactic slot, as illustrated in the *Shabat dress* example above.

The possible realizations of *possessor* are controlled by the feature `realize-possessor-as` `free-genitive`, `bound` or `double-genitive`. Defaults (unmarked cases) vary between the two languages and the co-occurrence constraints also vary, because each form is mapped to different syntactic slots.

For example, a bound possessor is mapped to the determiner slot in English, while in Hebrew it is mapped to a classifier slot.

Qualifying features	**English Realization**	**Hebrew Realization**
classifier	*Leather shoe* *Electric chair*	*nawal wor* *cise' HaSmaly*
describer	*Pretty boy*	*yeled yafeh*
qualifier	*A story about a cat* *A story I read*	*sypur wal Hatul* *sypur S-kar'aty*
possessor	*The king's palace* *A palace of a king* *The book of his*	*Armon ha-melex* *Armon Sel melex* *Armono Selo*
Identifying features		
distance	*That boy*	*yeled zeh*
Ordinal	*The third child*	*ha-yeled ha-SlySy*
status (deictic2)	*The same child*	*Oto yeled*
Definite yes/no	*The/a book*	*(ha) sefer*
Selective yes/no	*Some/∅ children*	*camah yeladym, (Eyzeh) yeled EHad*
Total +/-/none	*All/No/∅ children*	*col/Af yeled*
Quantifying features		
Total +/-/none	*All/None/∅ (of) the children*	*col hayeladym, Af EHad me-ha-yeladym*
Cardinal	*The three children*	*SloSet ha-yeladym*
Fraction	*One-third of the children*	*SlyS me-ha-yeladym*
Multiplier	*Twice his weight*	*ciflaym miSkalo*
degree +	*(The) many cars*	*harbeh mexonyot, ha-mexonyot ha-rabot*
degree -	*A little butter*	*kZaT Hema'h*
degree none	*Some children*	*camah yeladym*
comparative yes	*More cars*	*yoter mexonyot*
superlative yes	*The most cars*	*rov ha-mexonyot*
evaluative yes	*Too many cars*	*yoter m-day mexonyot*
orientation -	*Few cars*	*mewaT mexonyot*

Figure 2: Input features

When possessives are realized as free genitives, they are mapped to the slot of qualifiers, usually in the front position. Borochovsky (Borochovsky, 1986) discusses exceptions to this ordering rule in Hebrew:

Vawadah l-wirwurym Sel ha-miSTarah
The commission for-appeals of the-police
** Vawadah Sel ha-MiSTarah l-wirwurym*

In this example, the purpose-modifier is "closer" semantically to the head than the possessor. The ordering decision must rely on semantic information (purpose) that is not available in our general input structure (cf. (Dahan-Netzer and Elhadad, 1998b) for an even more abstract proposal).

Realization rules in each language take into account the restrictions on possible mappings for the possessor by unifying the feature `realize-possessive-as` based on the lexical properties of both the head and the possessor:

Construct-state not ok for possessive relation with proper name:
? Simlat Hanah – ? dress-cs Hanah
Double possessive ok for person names and possessor:
Simlatah Sel Hanah – dress-cs-her of Hanah
Double possessive not ok for non-possessive relation:
** Simlatah Sel ha-Sabat*
** dress-cs-her of the-Shabat*

Similarly, the possible realizations of the partitive are controlled by the feature `realize-partitive-as`: `of` or `quantifier`.

Quantifiers are classified along the portion/amount dimension. This system can be realized either lexically by quantifiers marked as partitive, or by using an explicit partitive syntactic structure *X me-Y/X of Y*.

Because the realization grammar uses the knowledge of which word realizes which function, the distinction among partitive quantifiers, amount quantifiers and determiners predicts the order of the words in the Hebrew NP. The standard order is:
`[partitive determiner amount head]`

As noted above, only partitives can enter into recursive structures, in both Hebrew and English. Accordingly, our input specification language enforces the constraint that only a single amount and a single identification feature can be present simultaneously.

Whenever a partitive quantifier is desired, the input specification must include a ref-set feature instead of the head. This enforces the constraint that partitives yield recursive constructs, similarly to Mel'cuk's elective-relation. Such recursive structures are illustrated in the following example:

wasarah me-col ha-mafgynym
ten of-all the-demonstrators
Ten of all of the demonstrators

$$\begin{bmatrix} \text{cat} & \text{np} \\ \text{cardinal} & \begin{bmatrix} \text{value} & 10 \end{bmatrix} \\ \text{ref-set} & \begin{bmatrix} \text{total} & + \\ \text{ref-set} & \begin{bmatrix} \text{lex} & \text{"mafgyn"} \\ \text{definite} & \text{yes} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The input is abstract enough to let the realization grammar decide whether to build an explicitly partitive construction. This decision depends on the lexical features of the realizing quantifiers and is different in English and Hebrew, as discussed above.

Additional realization rules take into account additional co-occurrence restrictions. For example, in Hebrew, if the "portion" part is modified with adjectives, then an explicitly partitive construction must be used:

ha-rov ha-gadol mi-beyn ha-yeladym
the-most the-big of-from the-children
The vast majority of the children

In summary, we have presented a set of input features for complex NPs that include the abstract *possessor* and *ref-set* features. These two features can be mapped to different syntactic slots. Realization rules in the grammar control the mapping of these features based on complex co-occurrence restrictions. They also take into account the lexical properties of specific quantifiers and determiners when deciding whether to use explicitly partitive constructions. Finally, the input structure enforces that only partitive relations can enter into recursive structures. Both HUGG in Hebrew and SURGE in English have been adapted to support this modified input specification.

4 Conclusion

To evaluate whether the proposed input structure is appropriate as a bilingual specification, we have tested our generation system on a set of 100 sample inputs for complex NPs in English and Hebrew. In the experiment, we only translated open-class lexical items, thus following the "ideal scenario" discussed in the Introduction. Despite the divergences between their surface syntactic structure, the input structures produced valid complex NPs in both languages in all cases.

We identified the following open problems in the resulting sample: the selection of the unmarked realization option and the determination of the default value of the definite feature remain difficult and vary a lot between the two languages.

This case study has demonstrated that the methodology of contrastive analysis of similar semantic relations in two languages with dissimilar syntactic realizations is a fruitful way to define a well-founded input specification language for syntactic realization.

References

M. Azar. 1985. Classification of Hebrew compounds. In R. Nir, editor, *Academic Teaching of Contemporary Hebrew.* International Center for University Teaching of Jewish Civilization, Jerusalem. (in Hebrew).

R. Barzilay, K. McKeown, and M. Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceeding of ACL'99*, Maryland, June. ACL.

J.A. Bateman, C.M. Matthiessen, K. Nanri, and L. Zeng. 1991. The re-use of linguistic resources across languages in multilingual generation components. In *IJCAI 1991*, pages 966–971, Sydney, Australia. Morgan Kaufmann.

J.A. Bateman, 1997. *KPML Development Environment: multilingual linguistic resource development and sentence generation.* GMD, IPSI, Darmstadt, Germany, release 1.1 edition. www.darmstadt.gmd.de/publish/komet/kpml.html.

R. Aronson Berman. 1978. *Modern Hebrew Structure*. University Publishing Projects, Tel Aviv.

H. Borer. 1988. On morphological parallelism between compounds and constructs. In Geert Jooij and Jaap Van Marle, editors, *Yearbook of Morphology 1*, pages 45–65. Foris publications, Dordrecht, Holland.

E. Borochovsky. 1986. The hierarchy of modifiers after the noun. *Leshonenu*, 50. (in Hebrew).

S. Busemann and H. Horacek. 1998. A flexible shallow approach to text generation. In *INLG'98*, pages 238–247, Niagara-on-the-Lake, Canada, August.

Y. Dahan-Netzer and M. Elhadad. 1998a. Generating determiners and quantifiers in Hebrew. In *Proceeding of Workshop on Computational Approaches to Semitic Languages*, Montreal, Canada, August. ACL.

Y. Dahan-Netzer and M. Elhadad. 1998b. Generation of noun compounds in Hebrew: Can syntactic knowledge be fully encapsulated? In *INLG'98*, pages 168–177, Niagara-on-the-Lake, Canada, August.

Y. Dahan-Netzer. 1997. HUGG - Unification-based Grammar for the Generation of Hebrew noun phrases. Master's thesis, Ben Gurion University, Beer Sheva Israel. (in Hebrew).

E. Doron. 1991. The NP structure. In U. Ornan, E. Doron, and A. Ariely, editors, *Hebrew Computational Linguistics*. Ministry of Science. (in Hebrew).

B. Dorr. 1994. Machine translation divergences: A formal description and proposed solution. *Journal of Computational Linguistics*, 20(4):597–663.

M. Elhadad and J. Robin. 1996. An overview of SURGE: a re-usable comprehensive syntactic realization component. In *INLG'96*, Brighton, UK. (demonstration session).

M. Elhadad. 1996. Lexical choice for complex noun phrases: Structure, modifiers and determiners. *Machine Translation*, 11:159–184.

R. Gedalia. 1996. Automatic generation of business letters: Combining word-based and template-based nlg through the distinct handling of referring expressions. Master's thesis, Ben Gurion University, Beer Sheva Israel. (in Hebrew).

L. Glinert. 1989. *The Grammar of Modern Hebrew*. Cambridge University.

M. A. K. Halliday. 1994. *An Introduction to Functional Grammar*. Edward Arnold, London, second edition.

R.S. Jackendoff. 1990. *Semantic Structures*. MIT Press, Cambridge MA.

B. Lavoie and O. Rambow. 1997. A fast and portable realizer for text generation systems. In *ANLP'97*, Washington, DC. www.cogentex.com/systems/realpro.

J.N. Levi. 1976. A semantic analysis of Hebrew compound nominals. In Peter Cole, editor, *Studies in Modern Hebrew syntax and semantics*. North-Holland, Amsterdam.

I.A. Mel'cuk and N.V. Perstov. 1987. *Surface-syntax of English, a formal model in the Meaning Text Theory*. Benjamins, Amsterdam/Philadelphia.

U. Ornan. 1964. *The Nominal Phrase in Modern Hebrew*. Ph.D. thesis, Hebrew University, Jerusalem. (in Hebrew).

R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. 1985. *A comprehensive grammar of the English language*. Longman.

C. Seikevicz. 1979. *The Possessive Construction in Modern Hebrew: A Sociolinguistic Approach*. Ph.D. thesis, Georgetown University, Washington D.C.

T. Winograd. 1983. *Language as a Cognitive Process: Syntax*, volume I. Addison-Wesley, Reading, MA.

D. Yzhar. 1993. Computational grammar for noun phrases in Hebrew. Master's thesis, Hebrew University, Jerusalem. In Hebrew.