A Deep Computational Grammar of Hebrew

Livnat Herzig Sheinfux
Department of Computer Science
University of Haifa

Petter Haugereid
Department of Linguistic, Literary and Aesthetic Studies
University of Bergen

Nurit Melnik
Department of Literature, Language and the Arts
The Open University

Shuly Wintner
Department of Computer Science
University of Haifa

1 Background

We present the current state of HEGRAM, a deep linguistic computational grammar of Modern Hebrew. HEGRAM is implemented in the Linguistic Knowledge Builder (LKB) system and grounded in the theoretical framework of Head-driven Phrase Structure Grammar (HPSG).

HPSG (Pollard and Sag, 1994) is a constraint-based grammatical theory. In HPSG all linguistic objects (i.e., words, phrases, and clauses) are represented as typed feature structures. The basic mechanism by which linguistic objects are related to each other is structure-sharing. Structure-sharing occurs when two paths in a feature structure lead to the very same (token-identical) node. As a result, the information content associated with that node is the unification of the information provided by the various shared paths. A linguistic expression is said to be grammatical when the information contributed by components of the linguistic object is compatible and can accumulate to form a complete description of the expression.

HPSG has logical and mathematical foundations which make it amenable to computational implementation. Moreover, the declarative nature of HPSG grammars makes them non-directional and thus suitable for both parsing and generation.

2 About HEGRAM

HEGRAM is derived from the LinGO Grammar Matrix, which is an open-source starter-kit for the development of broad-coverage, precision HPSG grammars for diverse languages (Bender et al., 2002). The Matrix provides a skeleton of a grammar, which covers basic lexical and phrasal types, semantic composition, and the infrastructure for unbounded dependencies and coordination.

The current state of the grammar is significantly different from the Matrix-derived one. We have refined and extended the core grammar with the goal of ultimately achieving broad coverage. One notable modification is the adoption of the packed argument-frame approach proposed by Haugereid (2011) to account for multiple argument frames per verb. In addition we extended the coverage to account for the Hebrew copular construction, including the possibility of zero copula, as well as other language-specific features such as noun-adjective agreement, and accusative case marking. Currently, HEGRAM contains a toy lexicon of approximately 120 items, but we are actively working on interfacing the grammar with the MILA computational resources (Itai and Wintner, 2008) that will provide access to a large-scale lexicon and a morphological processor.

At this point our grammar covers "canonical" clauses with SVO word order, different complement types, verbs with multiple argument frames, long distance dependencies (wh-questions and non-subject topicalization), non-verbal predicates (aka "nominal clauses") including zero copula constructions, and control verbs. In what follows we will focus on control verbs to illustrate key features of our analysis.

3 Control

Control verbs take infinitival VPs with unexpressed subjects as complements. In subject control (1a), the unexpressed subject of the VP complement is identified with the subject of the control verb. With object control verbs (1b), the unexpressed subject of the VP is the object of the control verb.

(1) a. dani hivṭiax̲ la-yalda latet la-kelev 'oxel
Danny promised to.the-girl to.give to.the-dog food
'Danny promised the girl to give the dog food.'

b. dani hirša la-yalda latet la-kelev 'oxel
Danny allowed to.the-girl to.give to.the-dog food
'Danny allowed the girl to give the dog food.'

Control is a phenomenon which poses challenges to a computational analysis for two reasons. First, one syntactic argument in a sentence assumes two semantic roles. For example, in (1a) the subject, Danny, is both the 'promiser' and the 'giver'. In (1b) the girl is both the 'allowed' and the 'giver'. Moreover, the semantic relation between the verb and its implicit subject is not local. Deep linguistic processing is required in order to identify and represent these non-trivial relationships between arguments; existing parsers, whether phrase-structure or dependency-based (Goldberg, 2011), are unable to produce such representations.

4 Proof of Concept

As proof of concept, in what follows we present the HEGRAM analysis of an example sentence (2), which involves object control and a wh-question. In addition to providing the correct semantic linking between the matrix object and the unexpressed subject of the VP complement, the parser is required to recognize the syntactic and semantic role of the 'displaced' wh-element *ma* ('what').

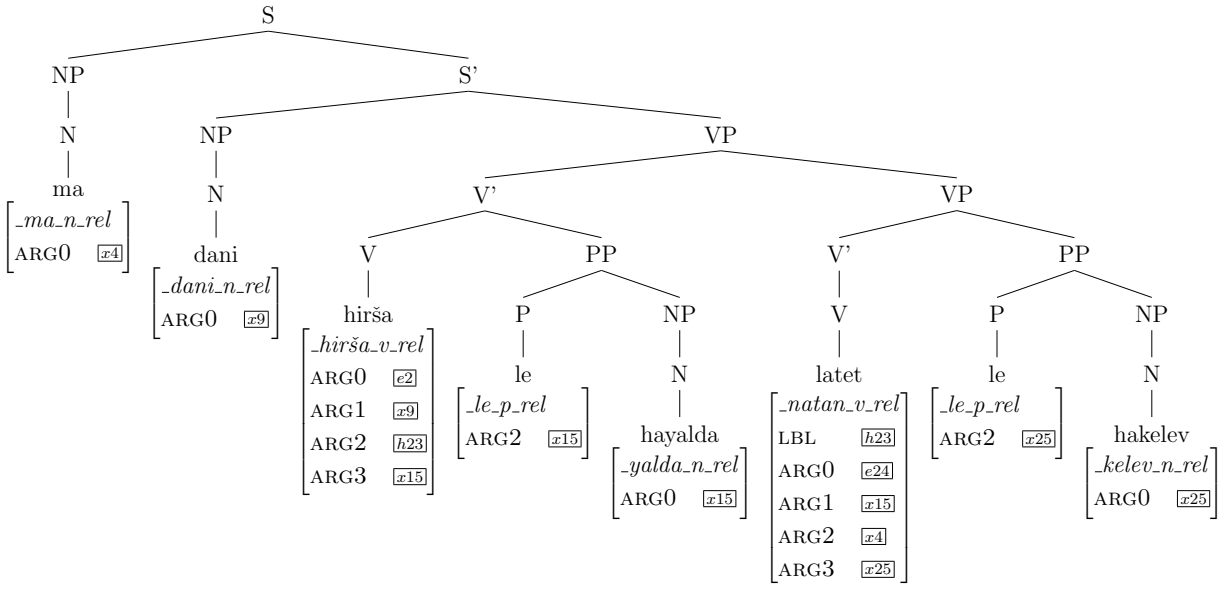
(2) ma dani hirša la-yalda latet la-kelev
what Danny allowed to.the-girl to.give to.the-dog
'What did Danny allow the girl to give the dog?'

The analysis produced by the LKB includes a syntactic phrase structure tree and a semantic representation (3). The semantic approach adopted by the LKB is Minimal Recursion Semantics (MRS; Copestake et al. (2005)). With MRS, linguistic expressions are assigned a syntactically flat semantic representation of linguistic expressions, which consists of a list of semantic relations and constraints on possible scope relations among them. Structure-sharing is expressed by way of co-indexation of arguments.

In the abbreviated MRS representation embedded in the tree below, the subject argument (ARG1) of *_hirša_v_rel*, the semantic relation denoted by the matrix verb *hirša* ('allow'), is indexed x9, and is structure-shared with the ARG0 of *_dani_n_rel*, thus indicating referential identity. Conversely, the subject of *_natan_v_rel*, the relation denoted by the VP, is indexed x15, and is structure-shared with the ARG0 of the relation denoted

(_yalda_n_rel). Additional structure-sharing relations indicate further semantic relations denoted by the sentence.

(3)



References

Emily M. Bender, Dan Flickinger, and Stephan Oepen. The grammar matrix: an open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *COLING-02 Workshop on Grammar engineering and evaluation*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1118783.1118785.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332, 2005. ISSN 1570-7075. doi: 10.1007/s11168-006-6327-9. URL http://dx.doi.org/10.1007/s11168-006-6327-9.

Yoav Goldberg. *Automatic Syntactic Processing of Modern Hebrew*. PhD thesis, Ben Gurion University of the Negev, Israel, 2011.

Petter Haugereid. A grammar design accommodating packed argument frame information on verbs. In Helena Hong Gao and Minghui Dong, editors, *PACLIC*, pages 31–40. Digital Enhancement of Cognitive Development, Waseda University, 2011. ISBN 978-4-905166-02-3.

Alon Itai and Shuly Wintner. Language resources for Hebrew. *Language Resources and Evaluation*, 42(1):75–98, March 2008.

Carl Pollard and Ivan A. Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press and CSLI Publications, 1994.