
Learning morphological phenomena

of modern Greek: an exploratory approach

Y. Kotsanis, *A. Kokkinos Dimitrios, *A.G. Manousopoulou and
*G. Papakonstantinou

*National Technical University of Athens

This paper presents a computational model for the description of concatenative morphological phenomena of modern Greek (such as inflection, derivation and compounding) to allow learners, trainers and developers to explore linguistic processes through their own constructions in an interactive open-ended multimedia environment. The proposed model introduces a new language metaphor, the 'puzzle-metaphor' (similar to the existing 'turtle-metaphor' for concepts from mathematics and physics), based on a visualized unification-like mechanism for pattern matching. The computational implementation of the model can be used for creating environments for learning through design and learning by teaching.

Introduction

Educational technology is influenced by and closely related to the fields of generative epistemology, Artificial Intelligence, and the learning sciences. Relevant research literature refers to the term *constructionism* (Papert, 1993) and *exploratory learning* (diSessa *et al*, 1995). Constructionism and exploratory learning are a synthesis of the constructivist theory of Piaget and the opportunities offered by technology to education on thinking concretely, on learning while constructing intelligible entities, and on interacting with multimedia objects, rather than the direct acquisition of knowledge and facts. These views are based on the approach that learners can take substantial control of their own learning in an appropriately designed physical and cultural environment (Harel, 1991). In parallel, most of the studies of the Vygotskian framework focus on the role of language in the learning procedure, considering conceptual thought to be impossible outside an articulated verbal thinking. Moreover, the specific use of words is considered to be the most relevant cause for childhood and adolescent differentiation (Vygotsky, 1962).

These approaches offer important pedagogical ideas for the creation of powerful computational principles, such as procedures and interactive objects (Pea, 1992). They

can be used in a flexible, reusable and modular way, and are usually referred to as the LISP-derived Logo-like environments (Hoyles *et al*, 1992; Georgiadis *et al*, 1993; diSessa *et al*, 1995). Moreover, they concretize the argument about how different languages (spoken and programming) can influence cultures that grow up around them (Papert, 1980).

Independent from educational technology, in the area of language technology natural-language processing systems have attempted to encode linguistic information and to endow computers with human language capability. A unification-based grammar formalism has been widely used in a variety of these systems as a pattern-matching technique for different purposes. Definite-clause grammars from the logic programming field, generalized phrase-structure grammars and typed-feature structures from the computational linguistics field are examples of formalisms based on unification (Shieber, 1986; Carpenter, 1992). Specific morphological processing systems are used in a fairly broad range of technological applications such as word processing, speech and language applications, and machine translation. These systems usually include a morphological processor that uses one or both of the following operations:

- an analyser, to recognize the combination of morphemes that form a word and/or the morphosyntactic features associated to the word;
- a synthesizer, to generate a well-formed word from its morphemes and/or the morphosyntactic features.

Many systems and applications which take this direction have been presented (Sproat, 1992), and specifically for the Greek language (Ralli, 1987; Kotsanis 1991; Markopoulos, 1994). The 'two-level morphology' theory, with its KIMMO implementation, has been by far the most successful and best-known general model of computational morphology (Koskenniemi, 1983; Antworth 1990). However, few references exist in the development of learning tools for modelling educational linguistic processes in an exploratory way (Sharples, 1986; Ohmaye 1992).

The objective of our approach is to provide learners with a powerful and natural learning environment to study morphology and words more generally. Learners are provided with tools to develop educational environments for language. This approach presents a computational model for the description of concatenative morphological phenomena of modern Greek (such as inflection, derivation, compounding – it can be extended to other languages as well) by allowing learners, trainers and developers to explore linguistic processes through their own constructions in an open-ended interactive multimedia environment. The proposed model introduces a new language metaphor, the 'puzzle metaphor' (similar to the existing 'turtle-metaphor' for concepts from mathematics and physics), based on a visualized unification-like mechanism for pattern matching. The computational implementation of the model can be used for creating environments for learning through design and learning by teaching, based on the experience that learners prefer to choose the role of the developer rather than the role of the user.

The puzzle metaphor

Metaphors are potentially important components of educational learning environments,

for providing dynamic models of systems that learners can explore and study. The turtle-metaphor, one of the most well-known, which uses the kinematic image of a curve as a moving point, is based on deeply rooted intuitions concerning body motion. It can be used to generate rich mathematical and science environments, offering a visually attractive and comprehensible introduction to programming (Papert, 1980; Hoyles *et al*, 1992). This mathematical-oriented metaphor is widely and internationally used for a broad range of ages and activities (Kynigos, 1992; Georgiadis *et al*, 1993; Blaho *et al*, 1994; diSessa *et al*, 1995). However, there are almost no references for a similar language metaphor (Tinsley *et al*, 1995; Vosniadou *et al*, 1995), with the exception of the phrase-books and boxes, a general oriented language microworld for linguistic explorations (Sharples, 1986).

By focusing our interest on morphological phenomena, the most adapted interpretation is based on the approach that words are built up from smaller meaningful units, namely morphemes. Morphological *analysis* (recognition) is concerned with retrieving the structure of morphemes that form a word. Morphological *generation* is concerned with producing an appropriate word-form from some set of morphosyntactic and semantic features (Sproat, 1992). Important for performing these tasks is that inflected, derived or compound words are built up via the successive application of word-formation rules, which are similar to syntactic rules for sentence formation. For example, we can define the following simplified rule for the combination of an infinitive verb with the nominalizing *er* morpheme (Bear, 1986) such as *read* – *reader*:

noun → verb nominal

<noun number> = singular

<verb form> = infinitive

<nominal string> = 'er'

General word-formation rules can be expressed as a context-free grammar of the form:

word → stem inflectional_ending

stem → stem derivational_suffix

Our linear puzzle-metaphor concept introduces a slightly modified view of a word-formation mechanism. The puzzle metaphor, underlying the context-free grammar, does not express the relation between the category of each morpheme (e.g. stem, ending) but the *position* (from now on called puzzle-type) that a morpheme has inside the word (**W**):

R: <i>complete</i>	morphemes that do not concatenate,
S: <i>beginning</i>	morphemes that are concatenated only from the right,
D: <i>intermediate</i>	morphemes that are concatenated from both right and left,
I: <i>ending</i>	morphemes that are concatenated only from the left.

Thus our context-free phrase structure of the word-grammar model (for word **W** recognition and generation) consists of the following three (and only these three) word-formation rules:

W → R
 W → SI
 S → SD

The above production rules generate only the following listed sequence of morphemes:

R, SI, SDI, SDDI, SDDDI, ...

For example, the Greek words *κάτω* (under, down), *παρακάτω* (further down), *γραφική* (graphic) and *περιγραφική* (descriptive) are analysed as:

R: [κατω]_R
 SI: [παρα]_S [κατω]_I
 SDÉ: [γραφ]_S [ικ]_D [η]_I
 SDDÉ: [περι]_S [γραφ]_D [ικ]_D [η]_I

Any morpheme can belong to more than one puzzle-type. Table 1 contains examples of various morphemes of the Greek language.

	Position	in the word	Category	Examples of morphemes
R	xyz		functional word non-inflected non-inflected	ο, η, (the – article), εμεις (we) περι (about, for) (metro)
S	xyz-		prefix stem prefix (particle) augmentation	περι-γραφ-ή (description) γραφ-ικ-ή (graphic/al) ἀ-γραφ-ος (unwritten) ἐ-γραφ-α (I was writing)
Symbol	-xyz-		prefix stem derivat. suffix synthetic vowel augmentation	αυτο-περι-γραφ-ή (autodescription) περι-γραφ-ή (description) γραφ-ικ-ότητ-α (picturesqueness) ακτιν-ο-γραφ-ι-α (x-ray) περι-ἐ-γραφ-α (I was describing)
I	-xyz		non-inflected inflect. ending	παρα-κάτω (further down) γραφ-ή (writing)

Table 1: Examples of puzzle-types and morphemes

The building blocks of the metaphor

The above mentioned context-free grammar describes which morphemes may combine with which. To determine the appropriate order of morphemes within words, we have to define for each morpheme a set of attribute-value pairs (which are user-defined and constitute a *feature structure*) containing several morphosyntactic pieces of information, for example word category (part of speech), gender and voice. These pieces of information can also have semantic characteristics, for example ‘consists of’. The determination of the attribute-value pairs does not restrict the underlying morpheme but

rather extends it to concatenate with the ‘valid’ next and/or previous morphemes (Pentheroudakis *et al*, 1993). This means that each of the three puzzle-types (S, D, or I) can be used as a previous or next puzzle-type (sketched with dotted-lines in the figures that follow).

In Figure 1 two morphemes have been determined. The first morpheme, the stem $\beta\alpha\sigma$ (of the word $\beta\acute{\alpha}\sigma\text{-}\eta$ – basis) is of puzzle-type S and has an associated feature structure. It can be concatenated with the associated feature structure of the inflectional ending η , which is a morpheme of puzzle-type I.

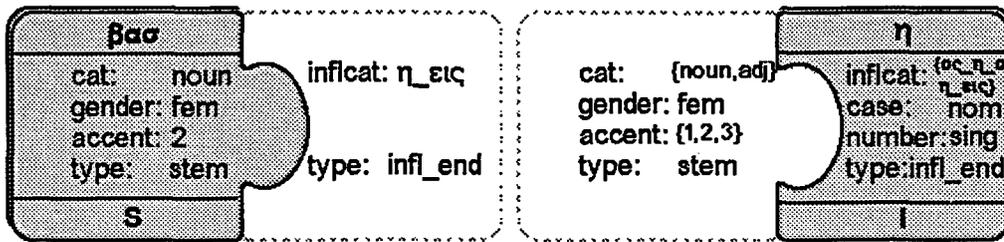


Figure 1: Definition of basic building blocks

Based on the above analysis, we have determined nine combinations of puzzle-types that constitute the basic building blocks of the model.

No	Symbol	Prev-Morpheme-Next	Examples
1	R		$\mu\acute{o}\nu\omicron$ (only)
2	\emptyset SI		$\emptyset \beta\acute{\alpha}\sigma\text{-}\eta$ (basis)
3	\emptyset SD		$\emptyset \beta\acute{\alpha}\sigma\text{-}\iota\kappa\text{-}\acute{\eta}$ (basic/al)
4	SDI		$\beta\acute{\alpha}\sigma\text{-}\iota\kappa\text{-}\acute{\eta}$ (basic/al)
5	SDD		$\beta\acute{\alpha}\sigma\text{-}\iota\kappa\text{-}\acute{\omicron}\tau\eta\tau\text{-}\alpha$ (basicity)
6	DDI		$\mu\omicron\nu\omicron\text{-}\beta\acute{\alpha}\sigma\text{-}\iota\kappa\text{-}\acute{\eta}$ (monobasic)
7	DDD		$\mu\omicron\nu\omicron\text{-}\beta\acute{\alpha}\sigma\text{-}\iota\kappa\text{-}\acute{\omicron}\tau\eta\tau\text{-}\alpha$ (monobasicity)
8	DI \emptyset		$\beta\acute{\alpha}\sigma\text{-}\iota\kappa\text{-}\acute{\eta} \emptyset$ (basic/al)
9	SI \emptyset		$\beta\acute{\alpha}\sigma\text{-}\epsilon\iota\varsigma \emptyset$ (bases)

Table 2: Examples of basic building blocks

Any morpheme can belong to one or more of the above building blocks. Figure 2 demonstrates how complex building blocks can be created. The morpheme $\beta\alpha\sigma$, the stem of the word $\beta\acute{\alpha}\sigma\text{-}\eta$ – basis, or $\beta\alpha\sigma\text{-}\iota\kappa\text{-}\acute{\eta}$ – basic, can be an S and D puzzle-type at the same time. The morpheme $\epsilon\iota\varsigma$ is the plural inflectional ending of the word $\beta\acute{\alpha}\sigma\epsilon\iota\varsigma$ – bases, and can be attached to an S or D puzzle-type.

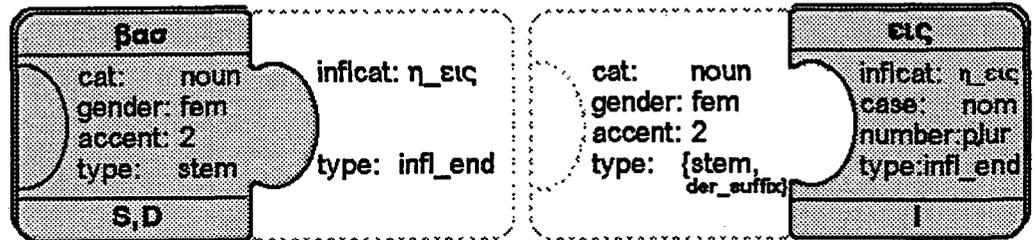


Figure 2: Definition of complex building blocks

The unification process and the associated feature structures

To establish a valid concatenation of two morphemes (beyond the rules of the context-free grammar), a pattern-matching mechanism is used. This unification mechanism is based on the notion of combining the information from two feature structures to get a feature structure with all of the information of both (Shieber, 1986).

Table 3 contains examples of how the unification mechanism actually works (note that a value of an attribute-value pair, which consists of a feature structure, can also be a feature structure).

Feature structure a	Feature structure b	Unification a U b
cat: noun	gender: fem	cat: noun gender: fem
cat: noun gender: fem	cat: noun	cat: noun gender: fem
cat: noun	cat: adj	-
cat: noun	cat: {noun, adj}	cat: noun
cat: NP	agreement: [pers: 3rd]	cat: NP agreement: [pers: 3rd]
agreement: [num: sing]	agreement: [pers: 3rd]	agreement: [num: sing pers: 3rd]

Table 3: Unification examples (curly braces {} denote disjunction)

The unification mechanism is performed only while parsing similar morpheme categories. For example, in a puzzle-type $X=S$ or D or I , unification will be applied only between the two morphemes where the first belongs to puzzle-type X and the previous or next of the second belongs also to puzzle-type X .

The morpheme of Figure 3, $\beta a\sigma$, can be concatenated with the inflectional ending η if the latter is an 'inflectional ending' (value of the 'type' attribute) and has an attribute 'inflectional category' with the value ' η_etc '. The derived inherits the values: noun, feminine, singular, nominative, and is stressed at the penult (value 2 of the attribute 'accent'). The morpheme $\beta a\sigma$ cannot be concatenated with a morpheme which has different values for the attributes 'type' and 'inflectional category'.

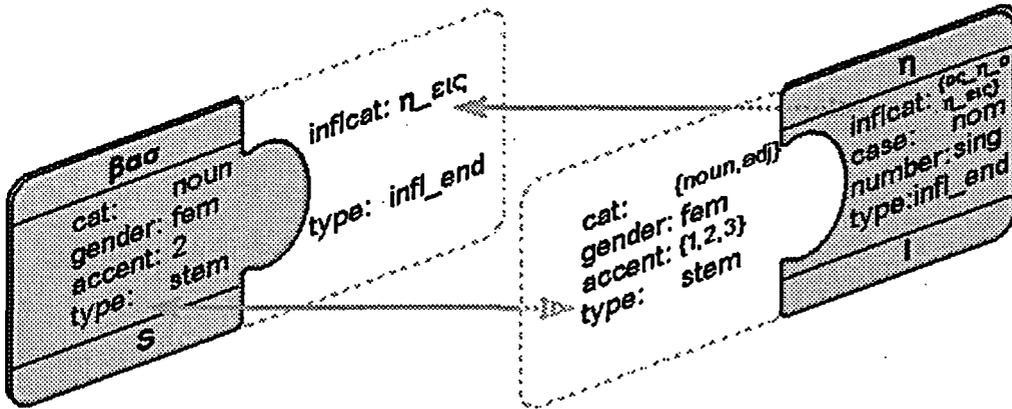


Figure 3: The unification process of two building blocks

Attribute-value pairs, depending on their function, can be classified in two groups with the following characteristics:

Inheritance or concatenation of values. As soon as the unification mechanism is started, the resulting allowed morpheme pairs contain a set of attribute-value pairs. In the word-formation process, the features of the word are inherited from the *head* of a morphological constituent. If, for example, the unification mechanism generates:

$[\beta a\sigma]_{\text{noun}}[\iota\kappa]_{\text{adjective}}[\eta]$ -

the word will inherit the attribute 'cat' (part of speech) with value 'adjective', since the right-hand head rule dominates and is frequently cited in morphology (Selkirk, 1982; Sproat 1992). Moreover, there are a number of cases which appear to have left-headed morphological constructions which have to be defined to the inheritance process (for example, the non-inflected prefix, *ava*, when attached to the word $\beta a\sigma\eta$ for the formation of the word $ava\beta a\sigma\eta$ – ascension, shifts the accent from the second to the third syllable from the ending of the word).

If the unification generates (in the above example):

$[\beta a\sigma]_{\text{stem}}[\iota\kappa]_{\text{derivational_suffix}}[\eta]_{\text{inflectional ending}}$

then the attribute 'type' of the word (with the values 'stem', 'derivational_suffix' and 'inflectional_ending' for the three morphemes respectively) is derived by concatenating the values; in this case the value would be 'stem & derivational_suffix & inflectional_ending'.

Values: Every value of an attribute-value pair can be either a string (text), a sound, a bitmap (picture), an animation, a video or a user-defined procedure which will start a computation or will return a value.

The handling of multimedia objects and user-defined procedures as values in a feature structure constitutes a powerful environment which covers a broad range of requirements (for example, handling of unusual morphological phenomena or audio-visual word-formation from audio-visual morphemes).

Implementation

The suggested model has been implemented using the following environments:

1. Prototyping using Prolog

To test the expressive power of the proposed formalism and to determine if the models generate all possible words of the given grammar (completeness), and if the words are 'valid' in the underlying vocabulary (soundness) (Gazdar 1989), a pilot system has been developed using Prolog which represents morphemes as follows:

```
str( MorphemeName, ListOfLettersInMorpheme ).
morf( MorphemeName, TypesOfMorpheme, AttributesOfMorpheme ).
next( MorphemeName, TypesOfNextMorpheme, AttributesOfNextMorpheme ).
prev( MorphemeName, TypesOfPreviousMorpheme, AttributesOfPreviousMorpheme ).
```

The TypesOf... fields are lists consisting of symbols from [R, S, D, I], while AttributesOf... fields are lists of the form [[attr-1,[value-1-1,value1-2, . . .]], [attr-2,[value-2-1,value2-2, . . .]], . . .]. Multiple next() and prev() facts are allowed for one morpheme. The distinction between the name of the morpheme and the actual list of letters that constitute it allows us to encode morphemes with similar behaviour under a single name. An example database would be:

```
str( bas, ['β', 'α', 'σ'] ).
morf( bas, [s,d], [ [type,[stem]], [cat,[noun]], [gen,[fem]], [accent,[2]] ] ).
next( bas, [d], [ [type,[der_suffix]] ] ).
next( bas, [i], [ [type,[infl_end]], [inflcat,[ç_áéò]] ] ).

str( ik, ['ι', 'κ'] ).
morf( ik, [d], [ [type,[der_suffix]], [cat,[adj]], [accent,[1]] ] ).
prev( ik, [s,d], [ [type,[stem]], [cat,[noun,adj]] ] ).
next( ik, [d], [ [type,[der_suffix]], [cat,[noun]] ] ).
next( ik, [i], [ [type,[infl_end]], [inflcat,[os_η_σ]] ] ).

str( i, ['η'] ).
morf( i, [i], [ [type,[infl_end]], [inflcat,[os_η_οη_εις]], [numb,[sg]], [case,
[nom,acc,voc]] ] ).
```

```
prev(i, [s, d], [[type, [stem, der_suffix]], [cat, [noun, adj]], [gen, [fem]],
  [accent, [1, 2, 3]]].
```

The above database and its associated grammar accept (generate) the following two, and only two, valid Greek words:

```
βαση          [bas,i]
βασικη       [bas,ik,i]
```

The current version of the prototype contains a lexicon of 200 different morphemes which generate about 5,000 valid words (nouns and verbs). At this point we should mention that right recursion of the grammar terminates normally, without the need to define a value for the maximum allowed depth which is directly proportional to the maximum number of concatenated morphemes.

2. Educational Logo-like multimedia environment

After receiving the outcome of the previous mentioned prototype in Prolog, we started implementing the proposed model by developing a microworld using a Logo-like interpreter (based on Comenius Logo: Blaho *et al*, 1994). This interpreter is equipped with new powerful primitive procedures and data types. Moreover, it works under the familiar Windows environment and takes full advantage of today's multimedia capabilities.

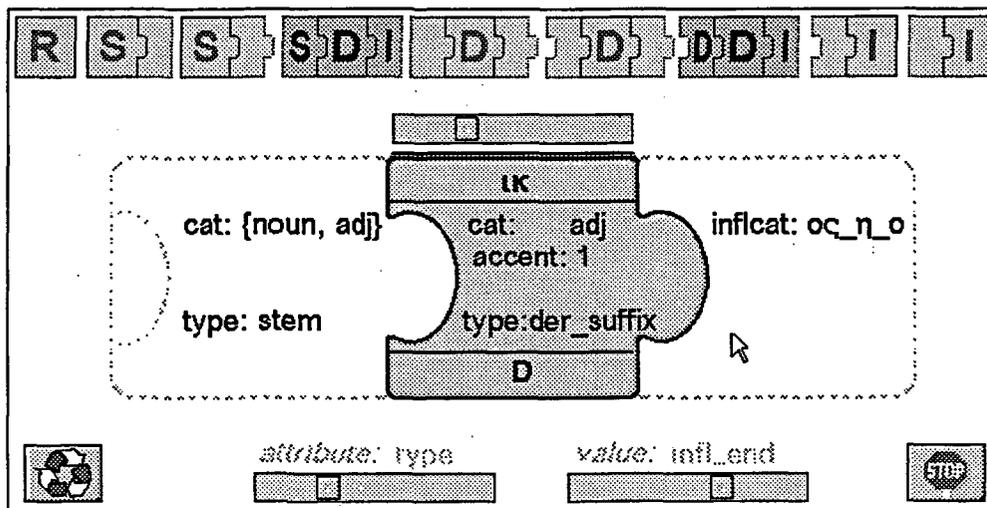


Figure 4: Working interactively with the 'makemorph' primitive.

Using the proposed microworld, the user will be able to work in a friendly Windows environment for handling the data (morpheme objects), and to recognize and generate valid morphemes and/or words. Further, the microworld allows or helps the user to group his or her data. The implemented microworld introduces the following six new primitives:

makemorph:	create basic or extended morphemes
getmorph:	search for and modify an existing morpheme
getword:	recognize and generate a valid concatenation of morphemes and build up a valid word
setpair:	define and/or modify attribute-value pairs and their characteristics
loadbase:	use an existing or user-defined knowledge base
savebase:	save a knowledge base

All primitives have two modes of operation: graphical and command-line. The graphical mode of operation is intended for the visual interaction with the various morpheme objects. The command-line mode is intended to provide users with primitives to implement their own or extend existing lexical application. The system also contains special-purpose attribute-value pairs for executing video, sound, image, music, and user-defined procedures.

In its present form, the system is being tested by a small group of educators and students. It will be enhanced with the data of an educational multimedia dictionary developed at Doukas School in Athens (Kotsanis *et al*, 1996) and will be tested in classroom activities of the same school.

Conclusions

The suggested model and its educational value constitute a learning-by-doing environment which simplifies functions and lexical representations without neglecting the expressive power of linguistic processes. Furthermore, it can introduce the idea of grammars and parsing to secondary-level students. The open-ended design of the environment gives the opportunity easily to develop audio-visual lexical applications for students by using different teaching architectures (Schank, 1994) and enhancing the kernels of authoring environments. In this unified environment (for use and development), learners, trainers and developers have access to and can reuse the same resources and data. For example, students will be able to design a spelling checker or express orthographic rules on their own.

This environment can be enhanced in many ways. First, it can be modified to allow more than one next or previous puzzle-type (S, D, or I) while defining morphemes. This results in the ability to describe long-distance dependencies where the existence of one morpheme is allowed by another morpheme which is not adjacent to it (for example, *joy*, **joy-able*, *en-joy-able*: Sproat, 1992). It can also be extended to allow the description of phonological phenomena (Antworth, 1990). Further, all lexical components of the model can be changed to graphical objects or even objects in space, so that it can be used by children at the primary educational level. Finally, in order to enhance the existing graphical representation of the puzzle-types beyond their two-dimensional movement, they can be moved and rotated in three-dimensions and their shapes changes (user-defined or selected from the existing library).

Acknowledgements

The authors would like to thank S. Piperidis, A. Giabris, Y. Maistros, C. Kynigos, P. Giakoumis, N. Filippaki, A. Triantafyllou, G. Markopoulos, G. Drivas and C. Doukas for their contribution and support in the above study.

References

- Antworth, E. (1990), *PC – KIMMO: A Two-Level Processor for Morphological Analysis*, Occasional Publications in Academic Computing 16, Summer Institute of Linguistics, Dallas TA.
- Bear, J. (1986), 'A morphological recognizer with syntactic and phonological rules', *COLING-86* (Association for Computational Linguistics).
- Blaho, A., Kalas, I. and Matusova, M. (1994), 'Environment for environments: new metaphor for Logo', in *Exploring a New Partnership: Children, Teachers and Technology*, Philadelphia PA: IFIP Transaction A-58.
- Carpenter, B. (1992), *The Logic of Typed Feature Structures: Inheritance, (In)Equations and Extensionability*, Cambridge: CUP.
- diSessa A., Hoyles, C. and Noss R. (1995), *Computers and Exploratory Learning*, NATO ASI Series F:146.
- Georgiadis, P., Gyftodimos, G., Kotsanis, Y. and Kynigos, C. (eds.) (1993), *Logo-like Learning Environments: Reflection and Prospects*, Proceedings of the Fourth European Logo Conference, University of Athens and Doukas School.
- Gazdar, G. and Mellish, C. (1989), *Natural Language Processing in Prolog*, Reading MA: Addison-Wesley.
- Harel, I. (1991), *Children Designers*, Cambridge MA: MIT Media Laboratory.
- Hoyles, C. and Noss R. (eds.) (1992), *Learning Mathematics and Logo*, Cambridge MA: MIT Press.
- Koskenniemi, K. (1983), *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. Ph.D. thesis, University of Helsinki.
- Kotsanis, Y. and Maistros, Y. (1991), 'Describing morphological phenomena of modern Greek using a unification grammar formalism', *Information Systems*, 16 (6).
- Kotsanis, Y., Raptis, K. et al (1996), *User Needs Analysis for Educational Multimedia Dictionary*, Tech. Report No 2.2.2, Doukas School, Athens (Research Programme DIALOGOS, Co-ordinator ILSP, supported by the grant EPET II, 715, GSRT).
- Kynigos, C. (1992), 'The turtle metaphor as a tool for children's geometry', in Hoyles C. and Noss R. (eds.), *Learning Mathematics and Logo*, Cambridge MA: MIT Press.
- Markopoulos, G. (1995), 'Two-level morphology in modern Greek', *Proceedings of the Second International Conference of Greek Linguistics* (September 1995), Salzburg.
- Ohmaye, E. (1992), *Simulation-Based Language Learning: An Architecture and a Multimedia Authoring Tool*, Ph.D. thesis, Northwestern University.

- Papert, S. (1980), *Mind-Storms, Children, Computers and Powerful Ideas*, New York: Basic Books.
- Papert, S. (1993), *The Children's Machine: Rethinking School in the Age of the Computer*, New York: Basic Books.
- Pea, R. D. (1992), 'Distributed multimedia learning environments: why and how', *Interactive Learning Environments*.
- Pentheroudakis, J. and Vanderwende L. (1993), 'Automatically identifying morphological relations in machine-readable dictionaries', *Proceedings of the Ninth Annual Conference of the UW Centre for the New OED and Text Research*.
- Ralli, A. and Galiotou, E. (1987), 'A morphological processor for modern Greek', *Proceedings of the Third European Chapter of the Association for Computational Linguistics* (April 1987).
- Selkirk, E. (1982), *The Syntax of Words*, Cambridge MA: MIT Press.
- Schank, R. C. (1994), 'Active learning through multimedia', *Multimedia IEEE*, Spring.
- Sharples M. (1985), *Cognition, Computers and Creative Writing*, Chichester: Ellis Horwood.
- Shieber, S. (1986), *An Introduction to Unification-Based Approaches to Grammar*, Chicago: University of Chicago Press.
- Sproat, R. (1992), *Morphology and Computation*, Cambridge MA: MIT Press.
- Tinsley, J. D. and Van Weert T. J. (eds.) (1995), *Liberating the Learner: Proceedings of the Sixth IFIP World Conference on Computers in Education (WCCE '95)*.
- Vosniadou, S., De Corte, E. and Mandl, H. (eds.) (1995), *Technology-Based Learning Environments*, NATO ASI Series F:137.
- Vygotsky, L. S. (1962), *Thought and Language*, Cambridge MA: MIT Press.