

# ACL Lifetime Achievement Award

## Word Play

Lauri Karttunen\*  
Palo Alto Research Center  
Stanford University

*This article is a perspective on some important developments in semantics and in computational linguistics over the past forty years. It reviews two lines of research that lie at the opposite ends on the field: semantics and morphology. The semantic part deals with issues from the 1970s such as **discourse referents**, **implicative verbs**, **presuppositions**, and **questions**. The second part presents a brief history of the application of **finite-state transducers** to linguistic analysis starting with the advent of **two-level morphology** in the early 1980s and culminating in successful commercial applications in the 1990s. It offers some commentary on the relationship, or the lack thereof, between computational and paper-and-pencil linguistics. The final section returns to the semantic issues and their application to currently popular tasks such as textual inference and question answering.*

### 1. Prologue

Thirty-eight years ago, in the summer of 1969 at the second meeting of COLING in Sångåsa-Såby in Sweden, I stood for the first time in front of a computational audience and started my talk on *Discourse Referents* by reading the following passage (Karttunen 1969a, 1976):

Consider a device designed to read a text in some natural language, interpret it, and store the content in some manner, say, for the purpose of being able to answer questions about it. To accomplish this task, the machine will have to fulfill at least the following basic requirement. It has to be able to build a file that consists of records of all individuals, that is, events, objects, etc., mentioned in the text and, for each individual, record whatever is said about it. Of course, for the time being at least, it seems that such a text interpreter is not a practical idea, but this should not discourage us from studying in the abstract what kind of capabilities the machine would have to possess, provided that our study provides us with some insight into natural language in general.

The paper went on to discuss the circumstances that allow a pronoun or a definite description to refer to an object introduced by an indefinite noun phrase. For example, in (1a), the pronoun *It* can refer to Bill's car, but in (1b) it cannot.

- (1) a. Bill has a car<sub>*i*</sub>.                      It<sub>*i*</sub>/The car<sub>*i*</sub> is black.  
    b. Bill doesn't have a car<sub>*i*</sub>.    \*It<sub>*i*</sub>/\*The car<sub>*i*</sub> is black. <sup>1</sup>

---

\* Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94062, USA. E-mail: karttunen@parc.com. This paper is the text of the talk given on receipt of the ACL's Lifetime Achievement Award in 2007.

<sup>1</sup> The subscripts <sub>*i*</sub> and <sub>*j*</sub> are referential indices. Two noun phrases with the same referential index are supposed to be coreferential, that is, they should refer to the same object.

A year later in 1970, I gave my first ACL presentation at the 8th annual meeting in Columbus, Ohio. The title of the invited talk was *The Logic of English Predicate Complement Constructions*. It started off with the following declaration (Karttunen 1971b):

It is evident that logical relations between main sentences and their complements are of great significance in any system of automatic data processing that depends on natural language. For this reason a systematic study of such relations, of which this paper is an example, will certainly have a great practical value, in addition to what it may contribute to the theory of the semantics of natural languages.

The paper presented a classification of verbs and constructions that take sentential complements, *that*-clauses and infinitival complements, based on whether the sentence commits the author to the truth or falsity of the complement clause. For example, all the sentences in (2) imply, for different reasons, that the complement is true while all the sentences in (3) imply that the complement is false.<sup>2</sup>

- (2) a. John forgot that Mary was sick.       $\rightsquigarrow$  Mary was sick.  
      b. Bill managed to solve the problem.     $\rightsquigarrow$  Bill solved the problem.  
      c. Harry forced Ed to leave.             $\rightsquigarrow$  Ed left.
- (3) a. John pretended that Mary was sick.     $\rightsquigarrow$  Mary was not sick.  
      b. Bill failed to solve the problem.        $\rightsquigarrow$  Bill did not solve the problem.  
      c. Harry prevented Ed from leaving.        $\rightsquigarrow$  Ed did not leave.

Neither one of these two papers would have been accepted at this 2007 ACL conference. There was no implementation, no evaluation, and very little discussion of related work. In the happy childhood of computational linguistics even the most junior person in the field, like myself, was allowed—even invited—to give a talk at the main COLING/ACL session about uncharted linguistic phenomena. It was a small field then.

A future historian of the field might be puzzled by the 1969 and 1970 papers. They were written by a postdoctoral research associate at the University of Texas at Austin, who had arrived from Finland in 1964 by way of the University of Indiana at Bloomington where he had just received a Ph.D. in Linguistics. Where did the young man acquire, and why was he spouting that kind of computational rhetoric, when the record shows that for the next ten years he never laid his hands on a computer?

The fact is that I did have a brush with computational linguistics before settling down to do pure semantics in the 1970s. I wanted to do linguistics because of *Syntactic Structures* (Chomsky 1957) and when the Uralic and Altaic Studies Department in Bloomington offered me a job in 1964 as a “native informant” in Finnish I accepted and managed to get into the Linguistics department as a graduate student. My job title turned out not to be accurate. During my first two years in Bloomington I was teaching Finnish on my own nine hours per week. Luckily, I signed up for a course on computational linguistics taught by an excellent teacher and mentor, Robert E. Wall. Bob Wall had participated in an early MT project at Harvard and in a project on automatic summarization at IBM. In his course, we learned formal language theory from notes that eventually became a book (Wall 1972), a bit of Fortran and COMIT, a language developed by Victor Yngve at MIT. I wrote a program on punched cards to randomly generate sentences from a small grammar of Finnish. Thanks to Bob, I was rescued from

<sup>2</sup> I am using the word *imply* as a generic cover term for *entail*, *presuppose*, and *conventionally implicate*. More about this in sections 2.2 and 2.3

the indentured servitude in the Uralic and Altaic Studies. In my third and last year in Bloomington, I worked as a research assistant in the Computer Center with no specific duties other than to be a liaison to the Linguistics Department. My only accomplishment in that role was to save piles of anthropological data from obsolescence by writing a program to transform rolls of 5-channel paper tape to 6-channel magnetic tapes. By doing that, I became one of the few linguists who could explain the joke *There are 10 kinds of linguists: those who know binary and those who don't*. I suspect that the data on my tapes for the Control Data 3600 computer have now been lost. We still have the data on manuscripts hundreds of years old but much of the content created in the first decades of the computer age is gone forever.

Just as I was starting to work on my dissertation in 1967, I had the good fortune of getting a one-year fellowship at the RAND corporation in Santa Monica, CA, in the group headed by David G. Hays, the author of the first textbook in our field (Hays 1967), and the founder of the Association for Machine Translation and Computational Linguistics AMTCL, the predecessor of our ACL. The main focus of Hays' team was Russian-to-English machine translation. Remarkably, Hays was also one of the authors of the infamous 1966 ALPAC report that inexorably caused the shutdown of all governmentally funded MT projects, including the one at RAND. Because the term *machine translation* had acquired a bad odor, the 1968 meeting of AMTCL dropped MT from its name and became ACL. At the 1970 meeting, the first one that I attended, people were still bitterly arguing about the matter.

In Hays' group at RAND I again met Martin Kay, who had been my teacher in a course on parsing at the 1966 Linguistic Institute at UCLA. My term paper for Martin's course had been on the computational analysis of Finnish morphology, a topic to which I would eventually return some fifteen years later. Happy to become Martin's student again, I learned Algol, an elegant new programming language, and got an understanding of the beauty of recursive algorithms. Martin was running an exciting weekly colloquium series. I teamed up with an intern by the name of Ronald Kaplan for a small study project and we gave a joint presentation about our findings. Ron and I agree that we did this together but neither one remembers what we said. It probably was about the similarities and differences between pronouns and logical variables, the topic of my first published paper (Karttunen 1969c).

At the time the prevailing assumption was that symbolic logic provides an appropriate system for semantic representation within transformational grammar (McCawley 1970). But as I showed in the 1969 CSL paper, even cases as simple as (4a) and (4b) could not be treated adequately within the proposed framework.

- (4) a. The man<sub>i</sub> who loved his<sub>i</sub> wife<sub>j</sub> kissed her<sub>j</sub>.  
 b. I gave each student a cookie<sub>i</sub>. Some of them ate it<sub>i</sub> right away.  
 c. The pilot<sub>i</sub> who shot at it<sub>j</sub> hit the Mig<sub>j</sub> that chased him<sub>i</sub>.

The problem with (4a) is that the phrase *his<sub>i</sub> wife* has to be treated *in situ* as it cannot be replaced by another coreferential noun phrase, say *Mary*, without changing the meaning. (4a) implies that only one man in some group of men loved his wife, which is not the same as there being just one man who loved Mary even if the two noun phrases pick out the same individual. For this reason there was no way in a system such as McCawley's to link *his<sub>i</sub> wife<sub>j</sub>* to *her<sub>j</sub>*. In the case of (4b), the problem is that there is no unique cookie to serve as the referent of *it*. Being an anaphoric pronoun linked to an antecedent does not necessarily mean that the two are coreferential, at least not in any naïve sense of coreference.

Even worse problem cases were known. (4c) is a typical “Bach-and-Peters sentence,” named after its inventors, Emmon Bach and Stanley Peters. As I was going to devote a chapter of my dissertation to this topic, I was lucky to run into them at some conference in San Diego. I found the two very intimidating in their suits and crew cuts. They looked like Haldeman and Ehrlichman, a pair of Nixon’s aides. But at least I found out that the problem had not been solved.

As the year at RAND when on, I spent less and less time at the computer and a lot of time walking up and down the Santa Monica pier just down the cliff from my office thinking about pronouns, variables, reference, and definiteness. I went up to UCLA a few times to discuss these issues with Barbara Partee and gave a talk in her seminar. The topic of my dissertation was *Problems of Reference in Syntax*, in principle due before I left RAND but finished half-a-year later (Karttunen 1969b).

By the summer of 1968 I had two job offers. David Hays was leaving RAND for SUNY in Buffalo and he offered me a job there. But I chose to become a Faculty Associate in the Linguistics Department at the University of Texas at Austin. Climate was one consideration, but, more importantly, Austin was where Emmon Bach and Stanley Peters were. My Indiana mentor, Bob Wall, had just moved into the same department. For the next ten years I had very little contact with Martin Kay and Ronald Kaplan but they became very important people in Act II of my life.

## 2. Act I: Framing Problems

I started my career in Austin in the fall of 1968 and became a regular faculty member in 1970. My work on discourse referents was largely done when I arrived in Austin. I went on to study so-called **implicative verbs** such as *manage* and *fail*, a subtopic in the discourse referents paper, and branched to other types of verbs that take sentential complements. One important semantic class of verbs with sentential complements, called **factives**, had already been identified and discussed by Zeno Vendler (1967) and Paul and Carol Kiparsky (Kiparsky and Kiparsky 1971) at MIT. Factive verbs were said to **presuppose** that the complement clause is true.

As it happened, I started to think about factives and presuppositions at MIT in the Fall of 1972. In the Spring before I had a surprise phone call from Paul Kiparsky who said that the MIT Department was still looking for a one-year replacement for David Perlmutter who was going on a sabbatical. Would I be interested? Of course I was. I had come to the U.S. seven years earlier to study linguistics because of Noam Chomsky and now I had an office just across from hall from his. But during the year I was there, I lost interest in transformational syntax. I found Chomsky’s Thursday lectures of that year, on themes later published as *Conditions on Transformations* (Chomsky 1973), un compelling.

My sense of what was interesting had changed. The “Linguistic Wars” (Harris 1995) between generative (George Lakoff, John Ross, James D. McCawley, Paul Postal, and others) and interpretive semantics (Ray Jackendoff and others) had been won by Chomsky for the interpretivists, although Chomsky himself was, and still is, skeptical of any kind of formal theory of meaning. My sympathies were with the losing side. But I sensed that both camps were essentially doing syntax, albeit in an abstract or a surfacy way. Barbara Partee had convinced me in our discussions about pronouns and variables at UCLA that model theory and intensional logic was the right approach to semantics. But it was going to take a while before I could do anything original within that emerging paradigm. At MIT I gave a “formal methods” course for a few linguistic students starting with Bob Wall’s text book (Wall 1972) and finishing with Montague Grammar that I was

just learning about myself (Montague 1970a, 1970b, 1973; Partee 1995) and a seminar on my own topics: discourse referents, implicative verbs, and presuppositions. I had one star student in the seminar by the name of Mark Liberman, who wrote a Master's Thesis poking holes in my emerging ideas about presuppositions.

In the 1970s, the Linguistics Department in Austin was an excellent place for a young semanticist. I learned tremendously from my colleagues there: Emmon Bach, Lee Baker, Stanley Peters, Carlota Smith, and Robert Wall. We had some excellent semantics and syntax students. David Dowty, Per-Kristian Halvorsen, Roland Hausser, Orvokki Heinämäki, Jim McCloskey, and Hans Uszkoreit got their degrees from UT Austin while I was there. Orvokki was my first Ph.D. student. She wrote an insightful thesis on the meaning of *before* and other temporal connectives (Heinämäki 1974).<sup>3</sup>

In the 1970s I worked on four general topics: discourse referents, implicative verbs, presuppositions, and questions. This is not an occasion to deep-end into any of these topics but I will discuss each of them briefly in the following sections to give a general idea of what I think my contributions were.

## 2.1 Discourse Referents

The obvious difference between definite and indefinite noun phrases is that garden-variety definite NPs such as *the car* in simple main clauses imply the existence of an individual or an object but indefinite noun phrases such as *a car* often do not. In that respect, definite NPs are similar to definite pronouns such as *it* in contexts where the pronoun does not play the role of a bound variable. The reason for the incoherence of (1b) is that it tells us explicitly that there is no such car.

Given an indefinite noun phrase, when is there supposed to be a corresponding individual that it describes? This was the question I tried to answer in the 1969 paper on discourse referents, excerpted from the first chapter of my Indiana dissertation. The conclusion I came to was that a pronoun or a definite description could refer back to (or be an anaphor for) an indefinite NP (the antecedent) just in case the existence of the individual was semantically implied by the text. Put in this simple way, the answer seems obvious but it gave rise to many problems some of which remain unsolved to this day. The novelty of the approach was that it rephrased the problem of **pronominalization** in semantic terms. Up to that point in transformational grammar, all the discussion about anaphors and antecedents had been about the constraints on their syntactic configurations.

Simple affirmative sentences such as *Bill has a car* in (1a) obviously imply existence and simple negative sentences such as *Bill doesn't have a car* in (1b) imply the opposite. The type of the verb matters, as seen in (5).

- (5) a. The director is *looking at* an innocent blonde<sub>*i*</sub>. She<sub>*i*</sub> is from Bean Blossom.  
 b. The director is *looking for* an innocent blonde<sub>*i*</sub>. #She<sub>*i*</sub> is from Bean Blossom.

So-called **intensional verbs** such as *look for*, *need* and *want* introduce an ambiguity. In (5b), the phrase *an innocent blonde* may be understood in two ways. In the specific sense it describes a particular individual that we can refer to as *she*. But (5b) can also be interpreted non-specifically, describing the type of girl the director is looking for. In that sense, the continuation is incoherent because there is not yet any individual to refer

<sup>3</sup> As I was preparing this talk, I heard the sad news from Finland that Orvokki Heinämäki had died.

to. (5a) has no such ambiguity; it entails the existence of an innocent blond in the actual world and we can talk about her.

But matters are more complicated. Although (5b) under a nonspecific reading of *an innocent blonde* does not establish a discourse referent in the actual world, we can nevertheless have one in a modal or hypothetical context, as in (6).

- (6) The director is looking for an innocent blonde<sub>*i*</sub>. She<sub>*i*</sub> must be 17 years old.

There is another problem here. If we interpret *an innocent blonde* non-specifically in (6), then *must* has a deontic reading. It is a requirement that she be 17 years old. However, on the specific reading *must* gets an epistemic interpretation. That is, we have made an inference about the age of the girl in question from her looks or other evidence.

My work on discourse referents was a harbinger of the vast literature yet to come on this topic including Bonnie Webber's 1978 dissertation (Webber 1978), Irene Heim's **file change semantics** (Heim 1982) and the theory of **discourse representation structures** (DR(S) Theory) proposed by Hans Kamp (1981) and Uwe Reyle (Kamp and Reyle 1993). Looking back at my old paper, I am amused by the youthful innocence with which it approached the topic but I am also impressed by the fact some of the problems it uncovered, such as the deontic/epistemic contrast in (6), apparently remain unsolved.

## 2.2 Semantics of Complementation

An indefinite noun phrase creates a stable discourse referent just in case the clause it is bound to is implied to be true by the context in which it appears.<sup>4</sup> That was the main idea in the 1969 paper. In the course of seeking evidence for this thesis, I came across an interesting class of verbs and constructions that give rise to such implications (Karttunen 1971a, 1971b). For example, the contrast between (7a) and (7b) is explained by the semantic properties of the two verbs, *manage* and *fail*.

- (7) a. John *managed* to get a sabbatical<sub>*i*</sub>. It<sub>*i*</sub> starts in September.  
 b. John *failed* to get a sabbatical<sub>*i*</sub>. \*It<sub>*i*</sub> starts in September.

(7a) **entails** that John got a sabbatical, (7b) entails that he didn't. The interesting fact about these verbs is that when we change the polarity from positive to negative we still get an entailment, but of the opposite polarity as seen in (8).

- (8) a. John *didn't manage* to get a sabbatical<sub>*i*</sub>. \*It<sub>*i*</sub> starts in September.  
 b. John *didn't fail* to get a sabbatical<sub>*i*</sub>. It<sub>*i*</sub> starts in September.

There exists quite a number of such **two-way implicatives** that yield an entailment in both positive and negative contexts. Verbs like *manage* yield a positive entailment in positive contexts (++) and a negative entailment in negative contexts (--). Let us call them **++/--implicatives**. Two-way implicatives like *fail* flip the polarity, so we call them **+/-+ implicatives**. Table (1) gives a few examples of both types of verbs and constructions.

Table (2) contains examples of verbs and constructions that certainly yield an entailment in one direction but not necessarily the other way.

<sup>4</sup> The term "stable" is in contrast with "short-term" for referents that have a limited life-span. For example, we can talk about a non-existing car as in *I wish Mary had a car<sub>*i*</sub>. She could take me to work in it<sub>*i*</sub>. I could drive the car<sub>*i*</sub> too.* as long as we are elaborating a hypothetical situation.

++ / -- implicatives	+ - / - + implicatives
manage (to)	fail(to)
succeed (in)	neglect (to)
remember (to)	forget (to)
happen (to)	avoid ... (ing)
see fit (to)	refrain (from)
take the time ... (to)	shy away (from)
have the foresight (to)	stop NP (from)

**Table 1**  
Some Two-way Implicative Verbs and Constructions

++ implicatives	+ - implicatives	-- implicatives	- + implicatives
cause NP (to)	prevent NP (from)	can	hesitate (to)
force NP (to)	preclude NP (from)	be able (to)	

**Table 2**  
Some One-way Implicative Verbs and Constructions

For example, it is tempting to conclude from (9a) that the president attended the meeting—and if there is no reason to think otherwise, the reader is entitled to that conclusion. Nevertheless, the author may take away that “invited inference” (Geis and Zwicky 1971) without contradicting himself as in (9b).

- (9) a. The president was able to attend the meeting.  
b. The president was able to attend the meeting but decided not to.

The entailments of constructions involving more than one implicative verb have to be computed from “top-down.” The two examples in (10) establish a stable discourse referent because they both entail that a picture was taken.

- (10) a. John managed not to forget to take a picture.  
b. Bill failed to prevent John from taking a picture.

The early version of Kamp’s Discourse Representation Theory did not include any mechanism for computing lexical entailments about existence. I found the DRS boxes disappointingly static at the time.

The semantics of complementation that I proposed was picked up by some computational linguists. Among the early adopters were Joshi and Weischedel (1973). Ralph Weischedel’s Ph.D. dissertation (Weischedel 1975) showed that useful inferences can be computed directly by the parser, in contrast to the then prevailing view of the AI community that all inferences have to come from some giant inference engine. This was the starting point of Jerrold Kaplan’s work on “cooperative responses” in database systems (Kaplan 1977).

### 2.3 Presuppositions — Conventional Implicatures

The semantics of two-way implicatives puzzled me greatly when I first discovered them (Karttunen 1971a). If the entailments in (11) both hold, in standard logic it would follow

that the construction *manage to* is empty of meaning. In general, if  $p$  entails  $q$  and  $\neg p$  entails  $\neg q$ , then it logically follows that  $p$  and  $q$  are equivalent:  $p \equiv q$ .

- (11) a. John managed to speak.            $\models$  John spoke.  
       b. John did not manage to speak.    $\models$  John did not speak.

But this is of course wrong as far as (11) is concerned. Choosing the construction *manage to* commits the speaker to the view that there was some difficulty involved. All the verbs in Table 1 bring in some additional commitment over and beyond what is entailed although it is difficult in some cases to pin down exactly what it is. Furthermore, the commitment remains the same regardless of whether the sentence is affirmative or negative. It is also present in questions and conditionals as shown in (12).

- (12) a. Did John manage to speak?  
       b. If John managed to speak, it is a good sign.

The extra bits of meaning attached to the two-way implicatives were yet another instance of a phenomenon that had already been discussed for some time under the term **presupposition**. The term came from philosophers who had been debating heatedly and for a long time whether *The present king of France* is false, meaningless or lacking a truth value (Frege 1892; Russell 1905; Strawson 1950; Russell 1957; Strawson 1964). When linguists got into the act in the late 1960s, being more systematic observers of language, within a span of just a few years they collected a large zoo of other types of constructions besides definite descriptions that seem to involve presuppositions (Kiparsky and Kiparsky 1971; Keenan 1971; Fillmore 1971). Unfortunately, they did not sort them into different habitats. Here are some examples:

- Factive verbs: *Mary forgot/didn't forget that John had left.*
- Factive adjectives: *It is/isn't odd that the room is closed.*
- Change-of-state verbs: *John stopped/hasn't stopped smoking.*
- Verbs of judging: *John criticized/didn't criticize Harry for writing the letter.*
- Wh-questions: *Who is coming for dinner?*
- Headless relatives: *Chicago is/isn't where Fred met Sally.*
- Cleft sentences: *It was/wasn't John who caught the thief.*
- Pseudo-clefts: *What she wants/doesn't want to talk about is herself.*
- Temporal subordinate clauses: *John left/didn't leave after Mary called.*
- Iteratives: *Fred called/didn't call again. Fred ate/didn't eat another turnip.*

In addition to vastly enlarging the presupposition population, the linguistic community also came up with a problem that had been ignored in the philosophical literature up to that point:

**Projection problem:** How are the presuppositions of a complex sentence derived from the presuppositions of the component clauses?

This question was first posed by Langendoen and Savin (1971). Their answer was (Langendoen and Savin 1971, page 57):



The projection principle for presuppositions, therefore, is as follows: presuppositions of a subordinate clause do not amalgamate either with presuppositions or assertions of higher clauses; rather they stand as presuppositions of the complex sentence in which they occur.

They were badly mistaken. Although the consequent clause of (13) by itself presupposes the existence of a unique king of France, (13) as a whole obviously does not.

(13) If France has a king, I bet the king of France speaks only French.

In a conditional sentence, a presupposition of the consequent clause can be “filtered” or “cancelled” away if it is entailed by the antecedent and general background knowledge. If a presupposition is not filtered locally and is not part of the context of the discourse, the reader or hearer must in some way adjust his state of knowledge to incorporate the new information. The idea is in Karttunen (1974, page 191):

If the current conversational context does not suffice, the listener is entitled and expected to extend it as required. He must determine for himself what context he is supposed to be in on the basis of what is said and, if he is willing to go along with it, make the same tacit extension that his interlocutor appears to have made.

Lewis (1979) called this process **accommodation**. There is a huge literature on the projection problem and accommodation. Among the papers often cited are Karttunen (1973), Stalnaker (1973), Karttunen (1974), Karttunen and Peters (1979), Gazdar (1979), Lewis (1979), Soames (1982), Heim (1983), van der Sandt and Geurts (1991), van der Sandt (1992), Zeevat (1992), Beaver (1995), Geurts (1999), Kamp (2001). Geurts (1999, page 5) sums up the early developments as follows:

An especially stark illustration of the disparity of the field, at least in its early days, is the work of a Karttunen, who within the span of six years published three theories that were mutually inconsistent, technically as well as conceptually.

I don't disagree with that assessment.<sup>5</sup> It seems to me that by now the notions of presupposition projection and accommodation have outlived their usefulness. It is evident that no uniform theory can account for all the phenomena that historically have been lumped together under the label *presupposition*.

But at least one good insight has emerged from this line of research. The accommodation strategy for definite descriptions is closely linked to anaphora resolution (van der Sandt 1992). One motivation for Kamp's DRS theory was to be able to handle “donkey anaphora” in sentences such as (14a).

- (14) a. If John has a donkey<sub>*i*</sub>, he beats it<sub>*i*</sub>.  
 b. If John has children, all of John's children are bald.

What van der Sandt observed was that the treatment of the anaphor in (14a) could be used in (14b) to eliminate the presupposition that John has children.

Assimilating presupposition projection into anaphora resolution is probably the right approach for definite descriptions and for iterative presuppositions triggered by prefixes such as *re-* in verbs like *recalculate* and particles such as *too* and *again*. However,

<sup>5</sup> Except for the dismissive Gricean implications triggered by the indefinite article in a *Karttunen*.

it does not seem applicable to the kinds of presuppositions triggered by implicative verbs or factives.

But the whole idea of accommodation is inappropriate for implicative verbs. Examples such as (11) and (12) commit the speaker to the view that it was difficult for John to speak. The audience may take note of that piece of information but it does not need to be accepted or accommodated for the discourse to proceed. Another phenomenon that does not call for any accommodation is *it*-clefts. As Ellen Prince (1978) showed, a sentence such as (15) does not covertly slip into the discourse a piece of new information disguised as being old. On the contrary, the rhetorical force of the *it*-cleft is to tell you something that presumably you did not know before in a manner that makes the new piece of information incontestable.

- (15) It was/wasn't Barbara Partee who in a private conversation around 1980 suggested to me that anaphora resolution and the satisfaction of the presuppositions of definite descriptions was the same problem.

In my joint last paper on presuppositions (Karttunen and Peters 1979), Stanley Peters and I proposed to do the sensible thing, namely to divide up the heterogeneous collection of phenomena that had been lumped together under this misbegotten label. We suggested that many cases that had been called presupposition are best seen as instances of what (Grice 1975) had called **conventional implicature**. Conventional implicatures are propositions that the speaker or the author of the sentence is committed to by virtue of choosing particular words or constructions to express himself. However, whether those implicatures are true or not does not have any bearing on whether the sentence is true or false. For example, because of the word *even*, (16) commits the author to the view that Bill is an unlikely person to agree with Mary.

- (16) Even Bill agrees with Mary.

But the meaning contributed by *even* plays no role in determining the truth conditions of the sentence. (16) is true if Bill agrees with Mary and false otherwise.

Our good advice went unheeded for a long time but in recent work by Christopher Potts (2004) we see an attempt to build the sort of two-dimensional semantics Stanley and I sketched out that separates conventional implicatures from truth-conditional aspects of meaning.

#### 2.4 Syntax and Semantics of Questions

My paper on questions (Karttunen 1977) was an ambitious effort to give a unified account in the framework of Montague Grammar of the meaning of all types of interrogative phrases including direct questions such as the examples in (17) and embedded interrogatives illustrated in (18).

- (17) a. Is it raining?  
 b. Do you want to go or do you want to stay?  
 c. Which book did Mary read?  
 d. Which girls date which boys?
- (18) a. John knows whether Bill smokes.  
 b. Mary is thinking about whether to stay home or go to the movies.

- c. Bill remembers to whom John gave the book?
- d. She doesn't care about who did what to whom.

Examples (17a) and (18a) are **yes/no questions**. (17b) and (18b) are **alternative questions** that pose two or more choices. As (17c,d) and (18c,d) illustrate, *wh-questions* may contain any number of interrogative quantifiers.

By and large, embedded *yes/no* and alternative questions have the same syntactic distribution as embedded *wh-questions*.<sup>6</sup> For that reason a syntactician would prefer to have just a single category of embedded questions. In the framework of Montague Grammar this is possible only if all types of embedded interrogatives have the same type of meaning. From a semantic point of view, it would be desirable to assign a single type of meaning to both direct and embedded questions. For example, *which girls date which boys* should have the same meaning as a direct question that it has when embedded under a verb such as *find out*. Finally, whatever meaning we assign to interrogatives, it should help us to elucidate the meaning of question-embedding verbs including the examples in (18) and the one in (19) that sets up a relation between two questions.

- (19) Whether Mary comes to the party depends on who invites her.

With these desiderata in mind, I came to the conclusion that the best solution would be to adopt an approach proposed by Hamblin (1973) for direct questions and carry it further. Hamblin's idea was to let every direct question denote a set of propositions, namely, the set of propositions expressed by all the possible answers to the question. For example, under Hamblin's analysis *Is it raining?* denotes the set containing two propositions  $\{It\ is\ raining, It\ is\ not\ raining\}$ . My improvement of that idea was to make the meaning of a question be a function that in each possible world picks up the set of true answers to the question.

Under this new analysis it is possible to relate, for example, the meaning of *know* with a *that*-complement to the meaning of *know* with an embedded question as in (18a). If Bill in our actual world is a smoker, then in our world *whether Bill smokes* picks out the set consisting of the proposition that Bill smokes. In that case, what John knows is that Bill smokes. In examples such as (19) the meaning of *depend on* can be explicated as a function that in each world maps the true answers to *who invites Mary* onto the true answer(s) to *whether Mary comes to the party*. I worked out these ideas with all the rigor of a Montague grammarian. After years of apprenticeship I had finally become a competent formal semanticist.<sup>7</sup>

A less restless soul would have stopped right there. But I didn't. As others saw it, I fell from the pinnacle of semantics into the low life of finite-state automata. My semanticist friends kept asking, "What happened to you Lauri? You were such a good semanticist." The politely unstated premise was that I had fallen onto skid row.

<sup>6</sup> There are two yet unexplained exceptions to this generalization. So-called "emotive factives" such as *be surprised* take embedded *wh-questions* but no *whether-questions*: *You'd be surprised where you find us*, \**You'd be surprised whether you find us*. "Dubitative" verbs such as *doubt* have the opposite characteristic.

<sup>7</sup> Compared to the lively activity on the presupposition playground, the field of questions attracts few visitors. For later developments, see Hausser and Zaefferer (1979), Hausser (1983), Groenendijk and Stokhof (1984), Ginzburg (1992, 1996).

### 3. Interlude

Towards the end of the 1970s I began to think that I had stumbled on, and helped to create and frame, more problems in semantics than I could ever solve. It was time to move on and leave the mess for others to clean up. In a bold move I signed up to teach a course on computational linguistics. As every professor knows, teaching a course on something you know next to nothing about is a great learning opportunity. To get some idea of how the field had developed in the previous ten years I went to the 1979 ACL Annual Meeting in La Jolla, California, and immediately ran into two of my old colleagues from RAND, Martin Kay and Ron Kaplan, very surprised to see me. "What are you doing here?", they asked. I said I had picked up a new hobby and was planning to do some computational work on Finnish morphology, the topic of my term paper for Martin's course in 1966.

On my sabbatical year at the Center for Advanced Study in Behavioral Sciences (CASBS) at Stanford, 1981-1982, I often went to visit Martin at the Palo Alto Research Center (PARC) just a short drive from CASBS. I learned about **unification** and Interlisp. I got to compute on the Alto personal computer and even had my own personal 1 MB floppy for it, about the size of a large briefcase. On the floppy was the project Martin and I were collaborating on, a unification based parser/generator for Finnish. Finnish was a good test case for Martin's **functional unification grammar** (FUG) formalism. In FUG, constituents could be labeled by a syntactic category such as NP and could be assigned functional roles such as CONTRAST and TOPIC. We showed how the constraints on Finnish word order could be described and implemented in those terms (Karttunen and Kay 1985a).

I joined the Artificial Intelligence Center at SRI in 1984. It was a good time to make the move from academia to industrial research. SRI had an excellent mix of computational linguists and AI people: Barbara Grosz, Jerry Hobbs, David Israel, Robert Moore, Fernando Pereira, Ray Perrault, Stuart Shieber, and Hans Uszkoreit were there, among others. SRI's AI Center and Xerox PARC were cofounders of the new Center for the Study of Language and Information (CSLI) at Stanford, funded by a generous grant from the System Development Foundation, an offshoot of the RAND Corporation. At SRI, Stuart Shieber had designed and implemented his influential PATR II formalism for unification-based grammars (Shieber et al. 1983). I implemented it (Karttunen 1984; Karttunen and Kay 1985b; Karttunen 1986) at CSLI in Interlisp on a Xerox Dandelion, a wonderful machine with Interlisp as the language of the operating system.

In 1987 I joined my friends at Xerox PARC to concentrate on my other computational interest: **finite-state morphology**. In making the cross-town transit from Menlo Park to Palo Alto, I graduated from my lovely Dandelion to the top-of-the-line Xerox Dorado, still the best computing experience in my life. After all the years spent on theorizing and playing with formalisms, I wanted to do something practical that would have an impact on the real world.

### 4. Act II: Providing Solutions

In the early 1980s, morphological analysis of natural language was a challenge to computational linguists. Simple cut-and-paste programs could be written to analyze strings in particular languages, but there was no general language-independent method available. Furthermore, cut-and-paste programs for analysis were not reversible, they could not be used to generate words.

Generative phonologists of that time described morphological alternations by means of ordered **rewrite rules** introduced by Chomsky and Halle (1968). These rules are of the form  $\alpha \rightarrow \beta / \gamma \_ \delta$ , where  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  can be arbitrarily complex strings or feature matrices. It was not understood how such rules could be used for analysis.

In 1981 I had organized a conference on parsing in Austin. Present at the conference was a visitor from Finland, Kimmo Koskenniemi, who was looking for a dissertation topic. Martin Kay and Ronald Kaplan were also there and it turned out that all four of us were interested in morphology. I demoed a small system I had build with my students for Finnish (Karttunen, Uszkoreit, and Root 1981). Martin and Ron reported that they had recently made a breakthrough discovery in computing with rewrite rules. Kimmo went on to California to visit them at PARC to learn more. That was the beginning of our long collaboration.

#### 4.1 Origins

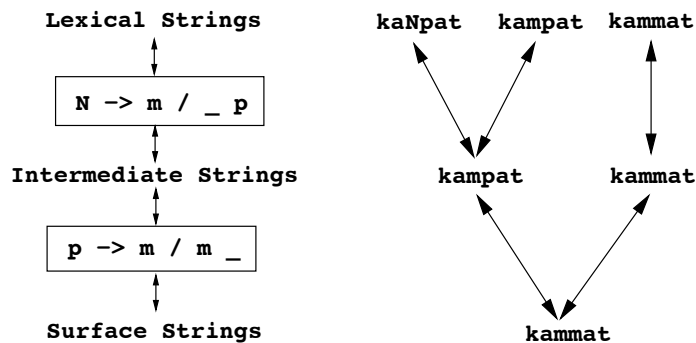
The discovery Kaplan and Kay had made was actually a rediscovery of a result that had been published a decade before in a book that none of us knew about at that time, a UC Berkeley dissertation by C. Douglas Johnson (1972). Johnson observed that while the same context-sensitive rule could be applied several times recursively to its own output, phonologists have always assumed implicitly that the site of application moves to the right or to the left in the string after each application. For example, if the rule  $\alpha \rightarrow \beta / \gamma \_ \delta$  is used to rewrite the string  $\gamma\alpha\delta$  as  $\gamma\beta\delta$ , any subsequent application of the same rule must leave the  $\beta$  part unchanged, affecting only  $\gamma$  or  $\delta$ . Johnson demonstrated that the effect of this constraint is that the pairs of inputs and outputs produced by a phonological rewrite rule can be modeled by a **finite-state transducer**.<sup>8</sup>

Johnson was already aware of an important mathematical property of finite-state transducers established by Schützenberger (1961): for any pair of transducers applied sequentially there exists an equivalent single transducer. Any cascade of rule transducers can in principle be composed into a single transducer that maps lexical forms directly into the corresponding surface forms, and vice versa, without any intermediate representations.

Koskenniemi was impressed by the theoretical result he learned from Kaplan and Kay, but not convinced about the practicality of the approach for morphological *analysis*. Traditional phonological rewrite rules describe the correspondence between lexical forms and surface forms as a one-directional, sequential mapping from lexical forms to surface forms. Even if it was possible to model the *generation* of surface forms efficiently by means of finite-state transducers, it was not evident that it would lead to an efficient analysis procedure going in the reverse direction, from surface forms to lexical forms.

Let us consider a simple illustration of the problem with two sequentially applied rewrite rules,  $N \rightarrow m / \_ p$  and  $p \rightarrow m / m \_$ . The corresponding transducers map the lexical form *kaNpat* unambiguously to *kammat*, with *kampat* as the intermediate representation. However if we apply the same transducers in the opposite direction to the input *kammat*, we get the three results shown in Figure 1. This asymmetry is an

<sup>8</sup> Johnson did a careful analysis of what at the time was one of the most comprehensive descriptions of phonological alternations described in the Chomsky–Halle paradigm. This was the unpublished MIT Qualifying Paper by James D. McCawley on Finnish. The data came from McCawley’s class mate, Paul Kiparsky, a native speaker of the language. One can claim that every advance in computational morphology in the last thirty years involves Finnish and people whose last name begins with *K*. See Section 4.3.



**Figure 1**  
Deterministic Generation, Nondeterministic Analysis

inherent property of the generative approach to phonological description. If all the rules are deterministic and obligatory and if the order of the rules is fixed, each lexical form generates only one surface form. But a surface form can typically be generated in more than one way, and the number of possible analyses grows with the number of rules that are involved.

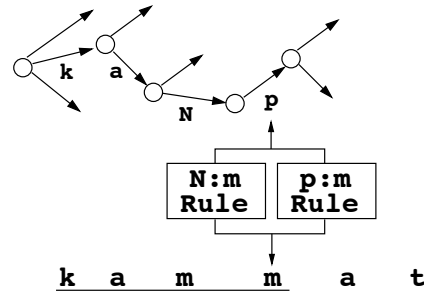
#### 4.2 Two-level Morphology

Back in Finland, Koskenniemi invented a new way to describe phonological alternations in finite-state terms. Instead of cascaded rules with intermediate stages and the computational problems they seemed to lead to, rules could be thought of as statements that directly constrain the surface realization of lexical strings. The rules would not be applied sequentially but in parallel. Each rule would constrain a certain lexical/surface correspondence and the environment in which the correspondence was allowed, required, or prohibited. For his 1983 dissertation, Koskenniemi (1983) constructed an ingenious implementation of his constraint-based model that did not depend on a rule compiler, composition or any other finite-state algorithm, and he called it **two-level morphology**. Two-level morphology is based on three ideas:

- Rules are symbol-to-symbol constraints that are applied in parallel, not sequentially like rewrite rules.
- The constraints can refer to the lexical context, to the surface context, or to both contexts at the same time.
- Lexical lookup and morphological analysis are performed in tandem.

Applying the rules in parallel does not in itself solve the overanalysis problem illustrated in Figure 1. The two constraints sketched above allow *kammat* to be analyzed as *kaNpat*, *kapat*, or *kammat*. However, the problem becomes manageable when there are no intermediate levels of analysis. In Koskenniemi's 1983 system, the lexicon was represented as a forest of **tries** (= letter trees), tied together by continuation-class links

from leaves of one tree to the root of another tree or trees.<sup>9</sup> Lexical lookup and the analysis of the surface form are performed in tandem. In order to arrive at the point shown in Figure 2, the analyzer has traversed a branch in the lexicon that contains



**Figure 2**  
Following a Path in the Lexicon

the lexical string *kaN*. At this point, it only considers symbol pairs whose lexical side matches one of the outgoing arcs of the current state. It does not pursue analyses that have no matching lexical path. All the rule networks must accept every lexical:surface pair. In the case at hand, the *p:m* pair is accepted by the *N:m Rule* that requires a *p* as the right context on the lexical side and by the *p:m Rule* that requires an *m* as the left context on the surface side. In two-level rules, zero (epsilon) is treated as an ordinary symbol. Because of this, a two-level rule represents an **equal-length relation**. Conceptually, the system in Figure 2 simulates the intersection of the rules and the composition of the lexicon.

Koskenniemi's two-level morphology was the first practical general model in the history of computational linguistics for the analysis of morphologically complex languages. The language-specific components, the rules and the lexicon, were combined with a universal runtime engine applicable to all languages.

**4.2.1 The Texas KIMMO system.** I met Koskenniemi again in Finland around Christmas time in 1982. He had just finished the first implementation of a two-level system and gave me a printout of the program to take along, a thick stack of Pascal code. Back home I unfolded the long printout on the floor of a corridor and spent quite a bit of time crawling up and down the code trying to understand what it did, and learning Pascal along the way. I was going to teach computational linguistics again in the Spring. Having figured out Kimmo's program, it occurred to me that doing a Lisp implementation of the two-level model would be a good class project.

We completed the project and published a collection of papers on the topic, along with our Lisp code (Karttunen 1983; Gajek et al. 1983). To make sure that Koskenniemi got the credit for the invention, we called it the KIMMO system. The name stuck and inspired many other KIMMO implementations. The most popular of these is PC-KIMMO, a free C implementation from the Summer Institute of Linguistics (Antworth 1990).

In Europe, two-level morphological analyzers became a standard component in several large systems for natural language processing such as the British Alvey project (Black et al. 1987; Ritchie et al. 1987, 1992), SRI's CLE Core Language Engine (Carter

<sup>9</sup> The TEXFIN analyzer I had demoed to Koskenniemi on his 1981 visit to Austin had the same lexicon architecture (Karttunen, Uszkoreit, and Root 1981).

1995), the ALEP Natural Language Engineering Platform (Pulman 1991) and the MULTEXT project (Armstrong 1996).

**4.2.2 A compiler for two-level rules.** In his dissertation Koskenniemi (1983) introduced a formalism for two-level rules. The semantics of two-level rules was well-defined but there was no rule compiler available at the time. Koskenniemi and other early practitioners of two-level morphology constructed their rule automata *by hand*. This is tedious in the extreme and very difficult for all but very simple rules.

To address this problem Kaplan, Kay, and I pooled our CSLI funds and invited Koskenniemi to Stanford in the Summer of 1985. Although two-level rules are conceptually quite different from the rewrite rules studied by Kaplan and Kay, the methods that had been developed for compiling rewrite rules were applicable to two-level rules as well. In both formalisms, the most difficult case is a rule where the symbol that is replaced or constrained appears also in the context part of the rule. This problem Kaplan and Kay had already solved by an ingenious technique for introducing and then eliminating auxiliary symbols to mark context boundaries. Another fundamental insight they had was the encoding of context restrictions in terms of double negation. For example, a constraint such as “*p* must be followed by *q*” can be expressed as “it is not the case that something ending in *p* is not followed by something starting with *q*.” In Koskenniemi’s formalism,  $p \Rightarrow \_ q$ .

In the course of the summer, Kaplan and Koskenniemi worked out the basic compilation algorithm for two-level rules. The first two-level rule compiler was written in InterLisp by Koskenniemi and me in 1985-87 using Kaplan’s implementation of the finite-state calculus (Koskenniemi 1986; Karttunen, Koskenniemi, and Kaplan 1987). The current C version two-level compiler, called TWOLC, was created at PARC (Karttunen and Beesley 1992). It has extensive systems for helping the linguist to avoid and resolve rule conflicts, the bane of all large-scale two-level descriptions.

**4.2.3 Two-level descriptions.** Many languages have been described morphologically in the two-level framework. But in many cases the work has been done for companies such as Lingsoft and Inxight that are in the morphology business, and the descriptions have not been made public for obvious reasons. Here are some of the languages for which there is a large-scale two-level grammar and a publication describing it: Finnish (Koskenniemi 1983), Estonian (Uibo 2006), German (Schiller 1996), Northern Sámi (Moshagen, Sammallahti, and Trosterud 2006), Turkish (Oflazer 1994).

### 4.3 Lexical Transducers

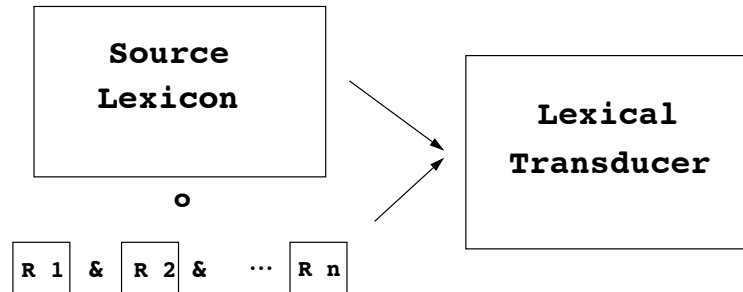
Soon after arriving at PARC I made a serendipitous discovery. At the time PARC was collaborating with Microlytics, a company that marketed spell-checkers, the first success story of finite-state morphology.<sup>10</sup> Microlytics had licensed from Koskenniemi’s company, Lingsoft, the rights to the Finnish analyzer. I was asked to extract from the Lingsoft two-level analyzer a network of surface forms that could be fed to Kaplan’s compression routine to make a Finnish spell-checker in the Microlytics format. For that task I designed an algorithm that simultaneously carried out the intersection of Koskenniemi’s 23 rule automata and the composition with the lexicon. I was surprised to see that not only did it work but the result was not significantly larger than the

---

<sup>10</sup> Ron Kaplan will tell you more about that some day.



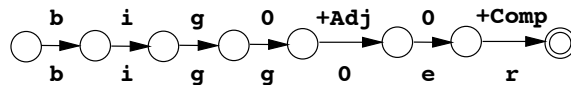
original source lexicon. Figure 3 is a sketch of that process. Just intersecting the rule



**Figure 3**  
Intersecting and Composing Two-level Rules with a Lexicon

automata by themselves was barely possible for us then because of the exponential worst-case complexity of the intersection algorithm. We assumed that the composition with a large lexicon might make the computation even harder to carry out. In fact the opposite happened. The reason should have been obvious from the beginning. The intersection of a set of two-level rules explodes because it has to compute a result for *any* lexical string. But if the set of lexical inputs is restricted to the forms that actually exist in the language, there is no blowup. The same applies to the composition of transducers derived from rewrite rules. If the rule cascade is computed starting with the lexicon, the “overanalysis” problem illustrated in Figure 1 never arises.

The surface forms that Microlytics wanted for the Finnish spell-checker were easily extracted from the transducer, but we realized that keeping the lexical forms and their surface realizations in a single network would be even more valuable. I created a transducer for English with a small number of two-level rules. It consisted of mappings such as in Figure 4. Annie Zaenen and Carol Neidle created with a large number of



**Figure 4**  
A Path in a Lexical Transducer

rules a much more ambitious proof-of-concept, a lexical transducer for French, mapping lemmas such as *vouloir+Verb+IndP+Sg+P3* to the corresponding surface form *veut*. Such a transducer is the ultimate “two-level model” for a language as it compactly encodes

- all the **lemmas** (lexical forms with morphological tags);
- all the inflected surface forms;
- all the mappings between lexical forms and surface forms.

A comprehensive analyzer such as we built for English and French consists of tens of thousands of states and hundreds of thousands of arcs; but physically they can be quite small, a couple of megabytes in size. The same network can be applied in two ways: to provide an analysis for a surface form or to generate a surface from a lexical form in a tiny fraction of a second. Karttunen, Kaplan, and Zaenen (1992) and Karttunen (1994) are the first published reports on lexical transducers.

In 1993 Xerox established a new European research center (XRCE) near Grenoble, France. Annie Zaenen and I went there to launch the Center's research on natural language. We started with a couple of employees in an unfinished building with three empty floors, an elevator, and a pile of Sun workstations stacked at the entrance. Not knowing a word of French made it a hardship assignment for me, but in every other respect it was a lucky break. Because XRCE was a startup as a research center in need of visibility and recognition on the level of the Xerox Corporation, I got more resources and help for my work than I could possibly have had at PARC. A Xerox business unit made a contract with XRCE to produce morphological analyzers and disambiguators (= "taggers") for six European languages. Kenneth R. Beesley, who had worked for Microlytics, came to Grenoble to manage the development effort. I headed a small finite-state team of researchers and programmers charged with the mission of creating better development and runtime tools such as XFST (Xerox Finite State Tool) and LEXC (Lexicon Compiler).

It became evident that large systems of two-level rules were difficult to debug. We concluded that lexical transducers are easier to construct with sequentially applied rules than with the parallel two-level rules. André Kempe and I therefore developed a compiler for **replace rules** (Karttunen 1995, 1996; Kempe and Karttunen 1996).<sup>11</sup> The XFST **regular expression** language now includes a large set of different types of replace expressions: parallel replacement, replacement with multiple contexts, replacement with left-to-right or right-to-left shortest or longest match constraints in addition to the usual finite-state operations union, intersection, composition, and negation.

Ken Beesley and I managed to get permission from the XRCE management to release to researchers most of the tools that were developed in Grenoble for creating and applying finite-state networks, not just for morphological analysis but also for other useful NLP tasks such as tokenization and named-entity recognition. Ken and I wrote a book, *Finite State Morphology* (Beesley and Karttunen 2003), a pedagogical text that explains and documents the tools that come with the book. There have been many improvements in the software since then. A new edition of the book is in the making.

By the time I left Grenoble to come back to PARC in 2001, Inxight, a Xerox spinoff company in California, was marketing finite-state morphological analyzers and stemmers for about three dozen languages. From a computational point of view morphology was a solved problem.

#### 4.4 Computational vs. Paper-and-Pencil Morphology

Historically, computational linguists and their "paper-and-pencil" counterparts in linguistics departments have been curiously out of sync in their approach to phonology and morphology. When computational linguists implemented parallel two-level models in the 1980s, paper-and-pencil linguists were stuck in the sequential Chomsky–Halle paradigm. Many arguments had been advanced in the phonological literature in the 1970s to show that phonological alternations could not be described or explained adequately without sequential rewrite rules. The idea of rules as constraints between a lexical symbol and its surface realization was seen as misguided. It went unnoticed that two-level rules could have the same effect as ordered rewrite rules because the

---

11 Our compilation algorithm was inspired by the landmark article of Kaplan and Kay (1994). We found a way to express the constraints on replacement using fewer auxiliary symbols than Kaplan and Kay. The compilation time and the size of the intermediate networks is very sensitive to the size of the auxiliary alphabet.

realization of a lexical symbol could be constrained by the lexical side and/or by the surface side. The standard arguments for rule ordering were based on the *a priori* assumption that a rule could refer only to the input context (Karttunen 1993).

But in the mid 1990s when most computational linguists working with the Xerox tools embraced the sequential model as the more practical approach, a two-level theory took over paper-and-pencil linguistics by storm in the guise of *Optimality Theory* (OT) (Prince and Smolensky 1993; Kager 1999; McCarthy 2002). In just a few years virtually all working phonologists switched into the OT paradigm. From my perspective OT is a two-level model where the ranking of the constraints plays the role that rule-ordering has in the sequential model.

If one believes, as I do, that the mapping from lexical forms to inflected surface forms is basically a regular relation, then the choice between the two ways of decomposing it, either as a composed cascade of replace operations or as an intersection of parallel rules, has important practical consequences but it is not a deep theoretical divide. In fact, the two-level analyzer for French discussed in (Karttunen, Kaplan, and Zaenen 1992) combined parallel rules with composition. It is unclear to me why my paper-and-pencil colleagues seem to think that it has to be absolutely one or the other.

I have written several papers in the hope of getting my paper-and-pencil colleagues interested in, or at least aware of, what is happening in computational morphology (Karttunen 1993, 1998, 2003, 2006). I have not succeeded. Paper-and-pencil morphologists in general are not interested in creating complete descriptions for particular languages. They design formalisms for expressing generalizations about morphological phenomena commonly found in all natural languages. Practical issues that arise in the context of real-life applications such as completeness of coverage, physical size, and speed of applications are irrelevant from an academic morphologist's point of view. The main purpose of a morphologist writing for an audience of fellow linguists is to be convincing that his theory of word formation provides a more insightful and elegant account of this aspect of the human linguistic endowment than the competing theories and formalisms.

My frustration is best summed up in a fable that I attached to my paper on a finite-state implementation of Gregory Stump's **realizational morphology** (Stump 2001; Karttunen 2003).

Time after time, from Johnson (1972) to Ellison (1994) to Eisner (2002), computational knights have presented themselves at the Royal Court of Linguistics, rushed up to the Princess of Phonology and Morphology in great excitement to deliver the same message:

Dear Princess. I have wonderful news for you: You are not like some of your NP-complete sisters. You are *regular*. You are *rational*. You are **finite-state**. Please marry me. Together we can do great things.

And time after time, the put-down response from the Princess has been the same:

Not interested. You do not understand Theory. Go away, you geek.

Because the most suitable suitor has always been rejected, I suspect that the Princess has a vested interest in making simple things appear more complicated than they really are. The good news that the computational knights are trying to deliver is unwelcome. The Princess prefers the pretense that phonology/morphology is a profoundly complicated subject, believing herself to be shrouded by veils of theories. If that is the correct analysis

of the situation, computational linguists should adopt a different strategy. Instead of being the eternal rejected suitor at the Royal Court, they should adopt the role of the innocent boy in the street shouting

The Princess has no clothes! The Princess has no clothes! . . .

That was my conclusion in the 2003 paper.

## 5. Epilogue

I am very happy to see that the topics I worked on at the very beginning of my career have finally become relevant in NLP. To quote again the opening paragraph of my 1970 ACL presentation (Karttunen 1971b):

It is evident that logical relations between main sentences and their complements are of great significance in any system of automatic data processing that depends on natural language. For this reason a systematic study of such relations, of which this paper is an example, will certainly have a great practical value, in addition to what it may contribute to the theory of the semantics of natural languages.

This 37-year-old prediction of semantics having *practical value* is becoming a reality in the context of automated question answering and reasoning initiatives such as the PASCAL Textual Entailment Challenge (Dagan, Glickman, and Magnini 2005) and the ARDA-sponsored AQUAINT project (Karttunen and Zaenen 2005; Zaenen, Karttunen, and Crouch 2005). The first computational implementation of textual inferences arising from the six types of implicative constructions in Tables 1 and 2, and their interaction with factive verbs, is presented in Nairn, Condoravdi, and Karttunen (2006). We may soon see search engines that actually make use of semantic processing in addition to simple string matching. The ability to draw textual inferences will significantly improve the quality of question answering and web searches.

From a linguistic perspective, this is an auspicious time to take a fresh look at issues such as the classification of complement constructions. The availability of search engines such as Google makes it possible to check the linguist's semantic intuitions against actual usage. One question I have always had about the classification of implicative constructions is whether the commitment to the truth or falsity of the complement clause is always based on a semantic entailment or whether some of these cases should be looked upon as a usage convention. For example, if you google the pattern *didn't hesitate to*, it is immediately evident that *hesitate to* really is one of the rare -+implicatives.

- (20) a. Head Coach Jon Gruden didn't hesitate to share interesting Buccaneer information with the two Chamber of Commerce crowds on Friday.  
 b. John didn't hesitate to refer the file to CIB and from there it went to the Victoria Police.  
 c. George Patton with all his swagger and confidence didn't hesitate to throw himself and his men into the teeth of the German offensive and won the day.

When you see examples such as in (20) in their full context, it is obvious that the author presents the complement clause of *hesitate to* as a fact. But it is difficult to explain why things should be this way starting from the concept of *hesitation* or the semantics of the verb *hesitate*.

As a sample of things I am planning to do next, I will leave you two little puzzles to solve. The construction *didn't wait to* is ambiguous. Here are a couple of examples from Google to illustrate the ambiguity.

- (21) a. Deena did not wait to talk to anyone. Instead, she ran home.  
 b. It hurt like hell, but I'm glad she didn't wait to tell me.

(21a) implies *Deena did not talk to anyone*. But (21b) implies *She told me something right away*.

**Question 1:** How does it come about that *X didn't wait to do Y* means either that X did Y right away or that X didn't do Y at all?

When you look at examples with *didn't wait to* in their full context, it is nearly always possible to tell which of the two meanings the author has in mind. In (21a), for instance, the negative polarity item *anyone* and the word *instead* are telltale indicators. In (21b), the cataphoric pronoun *it* indicates that a telling event took place. I am sure that it is possible to learn to pick the intended meaning by statistical techniques. But statistics alone will not give you an answer to Question 1, nor will it solve the related problem in Question 2.

**Question 2:** Why is it not possible to translate expressions such as *Neil didn't wait to take off his coat* to other languages in a way that preserves the ambiguity the sentence has in English?

In languages such as Dutch, Finnish, French, German, Hungarian, and Japanese among others, it is of course possible to express the two meanings of *X did not wait to Y* but not in one and the same sentence. My answer to these two questions will have to wait until my next semantics paper.

### Acknowledgments

Many thanks to Kenneth R. Beesley, Daniel G. Bobrow, Robert Dale, Aravind K. Joshi, John T. Maxwell III, Bonnie Webber, and Annie Zaenen for their help on the style and content of this paper.

### References

- Antworth, Evan L. 1990. *PC-KIMMO: a two-level processor for morphological analysis*. Number 16 in Occasional publications in academic computing. Summer Institute of Linguistics, Dallas.
- Armstrong, Susan. 1996. Multext: Multilingual text tools and corpora. In H. Feldweg and E. W. Hinrichs, editors, *Lexikon und Text*. Max Niemeyer Verlag, Tuebingen, pages 107–112.
- Beaver, David I. 1995. *Presupposition and Assertion in Dynamic Semantics*. Ph.D. thesis, University of Edinburgh, Edinburgh, Scotland.
- Beesley, Kenneth R. and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications, Stanford, California.
- Black, A., G. Ritchie, S. Pulman, and G. Russell. 1987. Formalisms for morphographemic description. In *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics*, pages 11–18, Copenhagen, Denmark.
- Carter, D. 1995. Rapid development of morphological descriptions for full language processing systems. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*, pages 202–209, Dublin, Ireland.
- Chomsky, Noam. 1957. *Syntactic Structures*. Mouton, Gravenhage, Netherlands.

- Chomsky, Noam. 1973. Conditions on transformations. In Steven Anderson and Paul Kiparsky, editors, *A Festschrift for Morris Halle*. Holt, Reinhard, and Winston, New York, New York, pages 232–286.
- Chomsky, Noam and Morris Halle. 1968. *The Sound Pattern of English*. Harper and Row, New York.
- Dagan, Ido, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 1–8, Southampton, U.K.
- Eisner, Jason. 2002. Phonological comprehension and the compilation of optimality theory. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 56–63, Washington, DC, July. Association for Computational Linguistics.
- Ellison, T. Mark. 1994. Phonological derivation in Optimality Theory. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 1007–1013, Kyoto, Japan.
- Fillmore, Charles. 1971. Verbs of judging: An exercise in semantic description. In Charles J. Fillmore and Terence Langendoen, editors, *Studies in Linguistic Semantics*. Holt, Rinehart and Winston, Inc., New York, New York, pages 273–289.
- Frege, Gottlob. 1892. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und Philosophische Kritik*, pages 25–50. English translation: "On Sense and Meaning", in Brian McGuinness (ed), *Frege: Collected Works*, pages 157–177, Basil Blackwell, Oxford.
- Gajek, Oliver, Hanno T. Beck, Diane Elder, and Greg Whittemore. 1983. LISP implementation. In Mary Dalrymple, Edit Doron, John Goggin, Beverly Goodman, and John McCarthy, editors, *Texas Linguistic Forum*, Vol. 22. Department of Linguistics, The University of Texas at Austin, Austin, TX, pages 187–202.
- Gazdar, Gerald. 1979. Conventional implicature. In Choon-Kyu Oh and David A. Dinneen, editors, *Syntax and Semantics, Volume 11: Presupposition*. Academic Press, New York, New York, pages 57–89.
- Geis, Michael and Arnold Zwicky. 1971. On invited inferences. *Linguistic Inquiry*, 2:561–565.
- Geurts, Bart. 1999. *Presuppositions and Pronouns*. Elsevier, Cambridge, Massachusetts.
- Ginzburg, Jonathan. 1992. *Questions, Queries and Facts: a Semantics and Pragmatics for Interrogatives*. Ph.D. thesis, Stanford University, Stanford, California.
- Ginzburg, Jonathan. 1996. Interrogatives: Questions, facts, and dialogue. In Shalom Lappin, editor, *Handbook of Contemporary Semantic Theory*. Blackwell Publishers, Oxford, United Kingdom, pages 385–422.
- Grice, H. Paul. 1975. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Speech Acts*. Academic Press, New York, NY, pages 41–58.
- Groenendijk, Jeroen and Martin Stokhof. 1984. On the semantics of questions and the pragmatics of answers. In Fred Landman and Frank Veltman, editors, *Varieties of Formal Semantics*. Foris Publications, Dordrecht, Germany, pages 143–170.
- Hamblin, C. L. 1973. Questions in Montague English. *Foundations of Language*, 10:41–53.
- Harris, Randy Allen. 1995. *The Linguistics Wars*. Oxford University Press, Oxford, United Kingdom.
- Hausser, Roland and Dietmar Zaefferer. 1979. Questions and answers in a context dependent Montague grammar. In Siegfried Josef Schmidt and Franz Guenther, editors, *Formal Semantics and Pragmatics for Natural Languages*. Reidel, Dordrecht, Germany, pages 339–358.
- Hausser, Roland R. 1983. The syntax and semantics of English mood. In Ferenc Kiefer, editor, *Questions and Answers*. Reidel, Dordrecht, Germany, pages 97–158.
- Hays, David G. 1967. *Introduction to Computational Linguistics*. American Elsevier, New York, N.Y.
- Heim, Irene. 1982. *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, University of Massachusetts, Amherst, Massachusetts.
- Heim, Irene. 1983. On the projection problem for presuppositions. In *West-Coast Conference on Formal Linguistics*, volume 2, pages 114–126, Stanford, California.
- Heinämäki, Orvokki. 1974. *Semantics of English Temporal Connectives*. Ph.D. thesis, University of Texas, Austin, Texas.
- Johnson, C. Douglas. 1972. *Formal Aspects of Phonological Description*. Mouton, The Hague.
- Joshi, Aravind K. and Ralph M. Weischedel. 1973. Some frills for the modal tic-tac-toe of Davies and Isard: Semantics of predicate complement constructions. In *IJCAI*, pages 352–355, Stanford, California.
- Kager, René. 1999. *Optimality Theory*. Cambridge University Press, Cambridge, England.
- Kamp, Hans. 1981. Événements, représentation discursive et référence temporelle. *Langages*,

- 64:39–64.
- Kamp, Hans. 2001. The importance of presupposition. In Christian Rohrer, Antje Roßdeutscher, and Hans Kamp, editors, *Linguistic Form and its Computation*. CSLI, Stanford, California, pages 207–254.
- Kamp, Hans and Uwe Reyle. 1993. *From Discourse to Logic*. Kluwer, Dordrecht, Germany.
- Kaplan, Ronald M. and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.
- Kaplan, S. Jerrold. 1977. *Cooperative Responses from a Natural Language Query System*. Ph.D. thesis, University of Pennsylvania, Philadelphia, Pennsylvania.
- Karttunen, Lauri. 1969a. Discourse referents. In *Proceedings of the International Conference on Computational Linguistics COLING'69*. Preprint 70.
- Karttunen, Lauri. 1969b. *Problems of Reference in Syntax*. Ph.D. thesis, Indiana University, Bloomington, Indiana.
- Karttunen, Lauri. 1969c. Pronouns and variables. In *CLS 5: Proceedings of the Fifth Regional Meeting*, pages 108–116, Chicago, Illinois. Chicago Linguistic Society.
- Karttunen, Lauri. 1971a. Implicative verbs. *Language*, 47:340–358.
- Karttunen, Lauri. 1971b. The logic of English predicate complement constructions. The Indiana University Linguistics Club. Bloomington, Indiana.
- Karttunen, Lauri. 1973. Presuppositions of compound sentences. *Linguistic Inquiry*, 4:167–193.
- Karttunen, Lauri. 1974. Presupposition and linguistic context. *Theoretical Linguistics*, 1(1):181–194.
- Karttunen, Lauri. 1976. Discourse referents. In James D. McCawley, editor, *Syntax and Semantics Volume 7, Notes from the Linguistic Underground*. Academic Press, New York, pages 363–385.
- Karttunen, Lauri. 1977. Syntax and semantics of questions. *Linguistics and Philosophy*, 1:1–44.
- Karttunen, Lauri. 1983. KIMMO: a general morphological processor. In Mary Dalrymple, Edit Doron, John Goggin, Beverley Goodman, and John McCarthy, editors, *Texas Linguistic Forum, Vol. 22*. Department of Linguistics, The University of Texas at Austin, Austin, TX, pages 165–186.
- Karttunen, Lauri. 1984. Features and values. In *COLING'84*, pages 28–33, Stanford, California.
- Karttunen, Lauri. 1986. D-PATR: A development environment for unification-based grammars. In *COLING'86*, pages 74–80, Bonn, Germany.
- Karttunen, Lauri. 1993. Finite-state constraints. In John Goldsmith, editor, *The Last Phonological Rule*. University of Chicago Press, Chicago, Illinois.
- Karttunen, Lauri. 1994. Constructing lexical transducers. In *COLING'94*, pages 406–411, Kyoto, Japan.
- Karttunen, Lauri. 1995. The replace operator. In *ACL'95*, Cambridge, Massachusetts. cmp-ig/9504032.
- Karttunen, Lauri. 1996. Directed replacement. In *ACL'96*, Santa Cruz, CA. cmp-ig/9606029.
- Karttunen, Lauri. 1998. The proper treatment of optimality in computational phonology. In *FSM/NLP'98. International Workshop on Finite-State Methods in Natural Language Processing*, Bilkent University, Ankara, Turkey, June 29. cmp-ig/9804002.
- Karttunen, Lauri. 2003. Computing with realizational morphology. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2588 of *Lecture Notes in Computer Science*. Springer Verlag, Heidelberg, pages 205–216.
- Karttunen, Lauri. 2006. The insufficiency of paper-and-pencil linguistics: the case of Finnish prosody. In Miriam Butt, Mary Dalrymple, and Tracy Holloway King, editors, *Intelligent Linguistic Architectures*. CSLI Publications, Stanford, California.
- Karttunen, Lauri and Kenneth R. Beesley. 1992. Two-level rule compiler. Technical Report ISTL-92-2, Xerox Palo Alto Research Center, Palo Alto, CA, October.
- Karttunen, Lauri, Ronald M. Kaplan, and Annie Zaenen. 1992. Two-level morphology with composition. In *COLING'92*, pages 141–148, Nantes, France, August 23–28.
- Karttunen, Lauri and Martin Kay. 1985a. Parsing in a free word order language. In David R. Dowty, Lauri Karttunen, and Arnold Zwicky, editors, *Natural Language Parsing*. Cambridge University Press, Cambridge, United Kingdom, pages 279–306.
- Karttunen, Lauri and Martin Kay. 1985b. Structure sharing with binary trees. In *Proceedings of the 23rd Meeting of the Association for Computational Linguistics*, pages 133–136, Chicago, Illinois.
- Karttunen, Lauri, Kimmo Koskenniemi, and Ronald M. Kaplan. 1987. A compiler for two-level phonological rules. In Mary Dalrymple, Ronald Kaplan, Lauri Karttunen, Kimmo Koskenniemi, Sami Shaio, and Michael Wescoat, editors, *Tools for Morphological Analysis*, volume 87-108 of *CSLI Reports*. Center for the Study of Language and Information, Stanford

- University, Palo Alto, CA, pages 1–61.
- Karttunen, Lauri and Stanley Peters. 1979. Conventional implicature. In Choon-Kyu Oh and David A. Dinneen, editors, *Syntax and Semantics, Volume 11: Presupposition*. Academic Press, New York, pages 1–56.
- Karttunen, Lauri, Hans Uszkoreit, and Rebecca Root. 1981. Morphological analysis of Finnish by computer. In *Proceedings of the 71st Annual Meeting of SASS*, Albuquerque, New Mexico.
- Karttunen, Lauri and Annie Zaenen. 2005. Veridicity. In Graham Katz, James Pustejovsky, and Frank Schilder, editors, *Annotating, Extracting and Reasoning about Time and Events*, number 05151 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany.
- Keenan, Edward L. 1971. Two kinds of presupposition in natural language. In Charles J. Fillmore and Terence Langendoen, editors, *Studies in Linguistic Semantics*. Holt, Rinehart and Winston, Inc., New York, New York, pages 45–54.
- Kempe, André and Lauri Karttunen. 1996. Parallel replacement in finite-state calculus. In *COLING'96*, Copenhagen, August 5–9. cmp-ig/9607007.
- Kiparsky, Paul and Carol Kiparsky. 1971. Fact. In D. Steinberg and L. Jakobovits, editors, *Semantics. An Interdisciplinary Reader*. Cambridge University Press, Cambridge, England, pages 345–369.
- Koskenniemi, Kimmo. 1983. Two-level morphology: A general computational model for word-form recognition and production. Publication 11, University of Helsinki, Department of General Linguistics, Helsinki.
- Koskenniemi, Kimmo. 1986. Compilation of automata from morphological two-level rules. In Fred Karlsson, editor, *Papers from the Fifth Scandinavian Conference on Computational Linguistics*, pages 143–149.
- Langendoen, Terence and Harris B. Savin. 1971. The projection problem for presuppositions. In Charles J. Fillmore and Terence Langendoen, editors, *Studies in Linguistic Semantics*. Holt, Rinehart and Winston, Inc., New York, New York, pages 55–62.
- Lewis, David. 1979. Scorekeeping in a language game. *Journal of Philosophical Logic*, 8:339–359.
- McCarthy, John J. 2002. *The Foundations of Optimality Theory*. Cambridge University Press, Cambridge, England.
- McCawley, James D. 1970. Where do noun phrases come from? In Roderick A. Jacobs and Peter S. Rosenbaum, editors, *Readings in English Transformational Grammar*. Ginn and Company, Waltham, Massachusetts, pages 166–183.
- Montague, Richard. 1970a. English as a formal language. In B. Visentini et al., editor, *Linguaggi nella Società e nella Tecnica*. Edizioni di Comunità, Milan, Italy, pages 189–224.
- Montague, Richard. 1970b. Universal grammar. *Theoria*, 36:373–398.
- Montague, Richard. 1973. The proper treatment of quantification in ordinary English. In P. Suppes K. J. J. Hintikka, J. M. E. Moravcsik, editor, *Approaches to Natural Language*. Reidel, Dordrecht, Germany, pages 221–242.
- Moshagen, Sjur, Pekka Sammallahti, and Trond Trosterud. 2006. Twol at work. In Antti Arppe, Lauri Carlson, Krister Lindén, Jussi Piitulainen, Mickael Suominen, Martti Vainio, Hanna Westerlund, and Anssi Yli-Jyrä, editors, *Inquiries into Words, Constraints and Contexts*. CSLI, Stanford, California, pages 94–105.
- Nairn, Rowan, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Inference in Computational Semantics (ICoS-5)*, Buxton, UK.
- Oflazer, Kemal. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2):137–148.
- Partee, Barbara. 1995. Montague grammar and transformational grammar. *Linguistic Inquiry*, 6:203–300.
- Potts, Christopher. 2004. *The Logic of Conventional Implicatures*. Oxford University Press, Oxford, United Kingdom.
- Prince, Allan and Paul Smolensky. 1993. Optimality Theory: Constraint interaction in generative grammar. Technical report, Rutgers University, Piscataway, NJ. RuCCS Technical Report 2. Rutgers Center for Cognitive Science.
- Prince, Ellen. 1978. A comparison of *wh*-clefts and *it*-clefts in discourse. *Language*, 54:883–906.
- Pulman, Stephen. 1991. Two level morphology. In H. Alshawi, D. Arnold, R. Backofen, D. Carter, J. Lindop, K. Netter, S. Pulman, J. Tsujii, and H. Uszkoreit, editors, *ET6/1 Rule Formalism and Virtual Machine Design Study*. CEC, Luxembourg, chapter 5.
- Ritchie, G., A. Black, S. Pulman, and G. Russell. 1987. The Edinburgh/Cambridge morphological



- analyser and dictionary system (version 3.0) user manual. Technical Report Software Paper No. 10, Department of Artificial Intelligence, University of Edinburgh.
- Ritchie, G., G. Russell, A. Black, and S. Pulman. 1992. *Computational Morphology: Practical Mechanisms for the English Lexicon*. MIT Press, Cambridge, Mass.
- Russell, Bertrand. 1905. On denoting. *Mind*, 14:479–493.
- Russell, Bertrand. 1957. Mr. Strawson on referring. *Mind*, 66:385–389.
- Schiller, Anne. 1996. Deutsche Flexions- und Kompositionsmorphologie mit PC-KIMMO. In Roland Hausser, editor, *Linguistische Verifikation: Dokumentation zur Ersten Morpholympics 1994*, number 34 in Sprache und Information. Max Niemeyer Verlag, Tübingen, Germany, pages 37–52.
- Schützenberger, Marcel-Paul. 1961. A remark on finite transducers. *Information and Control*, 4:185–196.
- Shieber, Stuart, Hans Uszkoreit, Fernando Pereira, Jane Robinson, and Mabry Tyson. 1983. The formalism and implementation of PATR-II. In Barbara J. Grosz and Mark Stickel, editors, *Research on Interactive Acquisition and Use of Knowledge*. SRI International, Menlo Park, CA, techreport 4, pages 39–79.
- Soames, S. 1982. How presuppositions are inherited: A solution to the projection problem. *Linguistic Inquiry*, 13:483–545.
- Stalnaker, Robert. 1973. Presuppositions. *The Journal of Philosophical Logic*, 2:447–457.
- Strawson, Peter. 1950. On referring. *Mind*, 59:320–344.
- Strawson, Peter. 1964. Identifying reference and truth values. *Theoria*, 30:320–344.
- Stump, Gregory T. 2001. *Inflectional Morphology. A Theory of Paradigm Structure*. Cambridge University Press, Cambridge, England.
- Uibo, Heli. 2006. Eesti keele morfoloogia modelleerimisest lõplike muundurite abil (On modelling the Estonian morphology by the means of finite-state transducers). In M. Koit, R. Pajusalu, and H. Õim, editors, *Keel ja arvuti*, number 6 in Tartu Ülikooli üldkeeleteaduse õppetooli toimetised. Max Niemeyer Verlag, Tübingen, Germany.
- van der Sandt, Rob. 1992. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9:333–377.
- van der Sandt, Rob A. and Bart Geurts. 1991. Presupposition, anaphora, and lexical content. In *Text Understanding in LiLOG, Integrating Computational Linguistics and Artificial Intelligence, Final Report on the IBM Germany LiLOG-Project*, pages 259–296, London, UK. Springer Verlag.
- Vendler, Zeno. 1967. *Linguistics and Philosophy*. Cornell University Press, Ithaca, New York.
- Wall, Robert E. 1972. *Introduction to Mathematical Linguistics*. Prentice Hall, Englewood Cliffs, New Jersey.
- Webber, Bonnie L. 1978. *A Formal Approach to Discourse Anaphora*. Ph.D. thesis, Harvard University, Cambridge, Massachusetts.
- Weischedel, Ralph. 1975. *Computation of a Unique Class of Inferences: Presupposition and Entailments*. Ph.D. thesis, University of Pennsylvania, Philadelphia, Pennsylvania.
- Zaenen, Annie, Lauri Karttunen, and Richard Crouch. 2005. Local textual inference: can it be defined or circumscribed? In *Workshop on the Empirical Modeling of Semantic Equivalence and Entailment*, pages 31–36, Ann Arbor, MI.
- Zeevat, Henk. 1992. Presupposition and accommodation in update semantics. *Journal of Semantics*, 9:379–412.

|

|