A Communicating Process Model of Language

Siamak Rezaei (siamak@mcgill.cs.ca)
School of Computer Science
McGill University

Abstract

We introduce a communicating model for representing linguistic processes. The framework introduces the basic notion of a linguistic process and the communication between these processes. These linguistic processes can communicate with each other and compete for the limited resources available. We have applied the model for different aspects of grammatical modeling and language modeling. In this work, we report on the application of the model for grammatical modeling and we will also discuss the possible application of the model for modeling gene scrambling. Finally we describe a formal specification of a process grammar for this purpose.

1 Introduction

The idea of brain as a computer or a processing machine has been the topic of research in cognitive science. The traditional approach to modeling brain has focused on brain as a set of dynamic evolving mental states. A different approach to the study of brain is to consider brain activities as a set of dynamic communicating processes. The shift from a fine-grained notion of *mental state* to a more coarse-grained notion of *mental process* is one aspect of such a model, but the notion of communication among mental processes is another aspect of such a dynamic approach.

Language is an area of research for the study of brain processes and psycholinguists have looked at different aspects of processing language by humans. In psycholinguistics, a focus of research on human parsing system has been the discovery of constraints on the human parser [Crocker, 1996]. Most of the basic models have presupposed the existence of traditional tree diagrams in linguistics. The psycholinguistic constraints are defined and imposed on these structures.

In our work we take a different view to language and we model grammar as a set of communicating process structures. Our focus will be the study of constraints on communication. A shift from static tree structures in linguistics to dynamic *linguistic resources* for which the processes can compete. For this we will turn to the dynamic models in computer science and look into the problem of representing linguistic processes and the communication between them.

In this paper we will first look briefly at the current work in formal modeling of communicating processes in computer science, then we will explain a possible approach for modeling linguistic processes, linguistics channels and the linguistic resources. Based on this model a set of resource constraints on word order have been discovered that will be presented. Then we will discuss the application of this model for modeling non-cognitive domain of gene scrambling. Finally we describe a formal specification of a process grammar for this model.

2 Process and Communication

Communicating Sequential Processes (CSP) [Hoare, 1985] provides a formal model for the interaction of processes through communication channels. Communicating processes have also been modelled by Calculus of Communicating Systems (CCS) [Milner, 1989] and its mobile extension π-Calculus [Milner, 1993].

In a process model or algebra, the fundamental notion is *process*. Examples of a basic process are a simple event (e.g. the utterance of a letter, dropping of a coin) and termination (e.g. the end of utterance)[1].

A process can also be more complex: and it can be a sequence of processes or a parallel combination of processes where the order of execution is not specified. The "." operator is used for prefixing or sequence. P.Q specifies the behaviour of a process which behaves as sequence of P and Q. An example is the utterance of two letters or words by a single person e.g. "He slept".

"|" is the parallel operator. P|Q specifies the behaviour of a process which behaves as P and Q in any order. These processes are run concurrently and can start and terminate at any time independently from each other. An example of which might be the utterance of two words by two persons. The order of uttering these words is not specified.

"Choice" is another operator for joining two processes. The choice operator "+" ensures only one action among several candidates is performed and it eliminates the others. P+Q depicts a process that can behave as P or as Q, but not both. A person cannot utter two letters at the same time and he has choice over the letters of the alphabet. Similarly one cannot insert two coins simultaneously into a machine with only one slot.

What makes a "process" more powerful than a "state" in a finite state automata is the fact that processes can communicate with each other and they can send or receive data over communication channels. An illustration of this is when two humans speak to each other. The channel for communication might be a telephone line or open air. One process emits a value or message over a channel and the other process having access to the same channel will receive the message. When the sender sends the message **m** over channel **c** by $\overline{c}\langle m\rangle$, the receiver will receive it by executing the process c(x) which will receive the message and upon receiving the message, will bind its free variable **x** to it, that is $\{m/x\}$. The sender channel is marked with a line

[1] There are other basic processes for *no-action* (analogous to pause in an utterance) and *deadlock*.

over it marking it as negative polarity, that communicates with a positive polarity channel with the same name, but no extra marking.

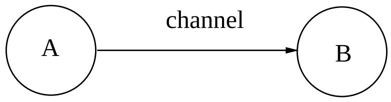

Figure 1: Communication Over Channels

A channel is illustrated in Figure 1.

The channel is the medium through which names can be exported, e.g. c in $\overline{c}(m)$, or imported, e.g. c in c(x).

In this work, we introduce the notion of communication for “linguistic” processes. By *linguistic process*, we have the domain of grammatical representation in mind. Like CSP, the communication channels in linguistic processes connect only two processes together. But what are these processes and what is their internal structure? We will answer these questions and define the notion of channel of communication for these linguistic processes.

3 Communicating Linguistic Processes

In [Rezaei, 1997] the basic notion of a linguistic process and its structure are defined. By using a sequence operator, the linguistic processes are constructed out of smaller units of words which are themselves constructed out of smaller morphological units. The more complex a linguistic process becomes, the more complex its communication can be.

In our model, a linguistic process is defined as a communicating structure that has an associated time-span. Using this notion of time-span, the temporal relation of linear precedence for these processes has been defined.

[Rezaei and Crocker, 1997] introduces the notion of linguistic communication channel by a dynamic representation of grammatical relations. Linguistic constraints on grammatical trees can be re-expressed to introduce constraints on communication. Whether we consider grammatical relations as primitive elements in the theory or we derive them from other principles of the theory, this decision will have further consequences for our future expansion of the theory. Here our main attempt is to add a dynamic aspect to the present approaches for linguistic modeling and we have decided to follow theories like LFG [Bresnan and Kaplan, 1982] in considering grammatical relations as primitive elements in our theory. In our dynamic model we take them as communication channels between process structures (i.e. constituents) which can communicate with other processes. This is depicted in Figure 2.

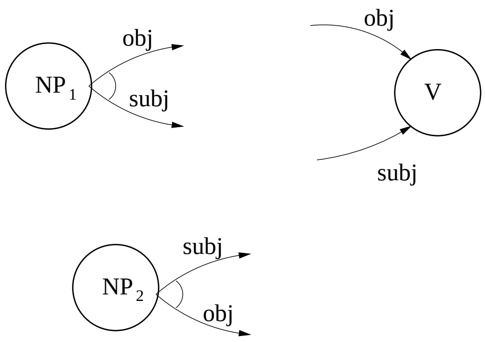

Figure 2: Communicating Linguistic Processes

In this figure, the channels are marked by potential grammatical roles that they require or provide[2]. This is especially important for modeling free word order languages in which different permutations of the sentence constituents are possible. This flexibility in word order is called *scrambling*.

For channels, we have considered two stages: marking and acquisition. The first step is channel marking, when the pre/post positions case mark a bare NP, at this stage the possible channels that an NP-process can compete for are specified. At the final stage, only one of these channels can be acquired by the process and this stage is channel acquisition.

It is at this stage that the actual act of communication happens between two processes over a single channel. In Figure 2 only one of the NPs can communicate with the **obj** channel of the verb.

There are certain constraints on channel marking and acquisition in different languages. For fixed word order languages the location and positions are very important parameters for channel marking.

For cases where there are multiple choices, competition determines which channel is acquired by a process. A choice operator guarantees that only one of the candidates can participate and the constraints on channels guarantee a single winner. Using a *path-set* data structure, a computational model for implementing these linguistic processes can be implemented [Rezaei, 2000b]. Figure 3 shows the result of parsing after competition for resources is done.

[2]The choice operator for channels of the NPs in the figure is highlighted with an arc.

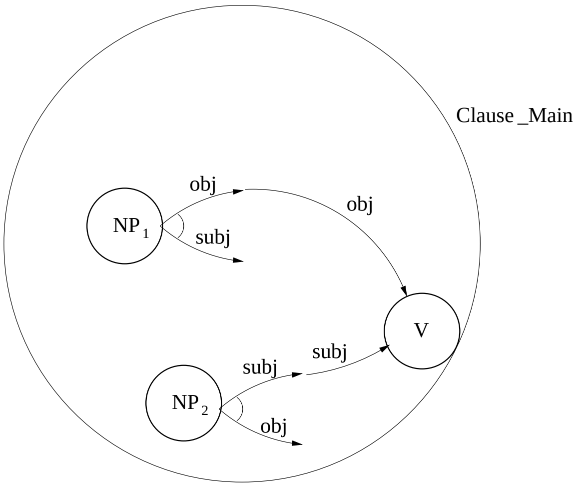

Figure 3: After Communication of Local Resources

To deal with Long Distance Scrambling (LDS)[3] we use a mechanism which is analogous to functional uncertainty in theories like LFG. Under certain conditions (e.g. barriers theory of GB) some channels can be passed or exported from one clause to an embedded one.

Figure 1 compares other aspect of our framework with concepts in Government Binding (GB) theory and Lexical Functional Grammar (LFG).

Table 1: A Comparison with GB and LFG.

GB	LFG	Our CLP
Case	?	Channel marking
θ theory	Functional uncertainty	Channel acquisition
Move$-\alpha$	Functional uncertainty	(Mobile) channels
Barriers	?	Mobility Constraints
Control	Control	Conditional channels
Optimality	probabilistic feature str.	Fuzzy channels

One additional advantage of a communication model of language is the ability to specify resource based constraints.

[3]In LDS a constituent will be moved across the boundary of two clause boundaries.

4 Resource Constraints on Parsing

Over the last few years a different conceptualisation concerning 'resource sensitivity' has emerged in several disciplines connected to the study of language. This idea has been explored within categorial grammar in [Carpenter, 1996] and [Morill, 1994]. More recently [Johnson, 1997a] and [Johnson, 1997b] introduce a resource-based conceptualisation of LFG. In [Johnson, 1996] the approach is illustrated with a view of characterising constructions in terms of 'plugging'. A set of objects are constructed and some of these objects need to combine with other objects to become saturated, and rules determine what can be 'plugged into' what.

Phenomena such as argument attachment in natural languages are inherently resource based and most linguistic theories use some mechanism of resource sensitivity for argument attachment. From a resource-based perspective, other constraints related to performance and parsing can be defined that restrict the possible word orders.

A computational model for implementing these linguistic processes has been elaborated in [Rezaei, 2000b]. In our model we use a dynamic structure to allow a set of paths or threads to progress in parallel and to compete for the limited grammatical resources. This mechanism allows a process to compete for one or more channels at the same time.

Based on this model, we have introduced a notion of limited resources by the Resource Limitation Principle (RLP) for representing local scrambling in some free constituent order languages.

Resource Limitation Principle
No two NPs can exist in a clause with the same grammatical functions.

By using blocking word order rules we blocked the progress of paths which contained two instances of a grammatical resource.

We also discovered another blocking constraint, the Resource Barrier Principle (RBP) for long distance scrambling (LDS) and export of resources. In LDS a constituent will be moved across the boundary of two clause boundaries.

Resource Barrier Principle
If a resource exists in a clause, it acts as a barrier in front of the resources with the same grammatical functions which want to scramble into lower level embedded clauses from higher ones.

The notion of limited resources for scrambling is a grammatical constraint that hasn't been investigated earlier. This aspect of the communication model interacts with the notion of competition among the linguistic processes.

Competition can be modelled in an algebraic way that can be implemented with fewer problems than in other models for competition such as optimality theory [Smolensky et al., 1992]. Based on the notion of "algebraic optimality", the competition aspect of the model has been implemented [Rezaei, 2000b].

5 From Word to Gene Scrambling

The communicating linguistic processes model that we have discussed is a general model and one can specify constraints on the number of channels for each process and come up with different classes of communicating linguistic processes (CLP). In this section we will discuss the CLP-II model where each process has access to two instances of linguistic channels. These binary processes form a string or pipeline of processes, but the processes are further constrained by the word order constraints.

The CLP-II model can be used for modeling the traditional cursive English handwriting system, where each letter or glyph is connected to its immediate right and left neighbour, and for each letter we have initial form, medial form and final form. We have used CLP-II for modeling the connectivity of font graphs for Arabic cursive script [Rezaei, 2000a].

Arabic cursive script is used for writing different languages. Figure 4 shows an example of cursive font graph for Kurdish language which is an an Indo-European language. The script is written and read from right to left. and each letter can have three forms of initial, medial and final. The first column on the right in the figure shows the initial forms for some letters (h,l,y,b,d,z,j,s,sh). The second column shows the corresponding medial forms and the left-most column shows the corresponding final forms.

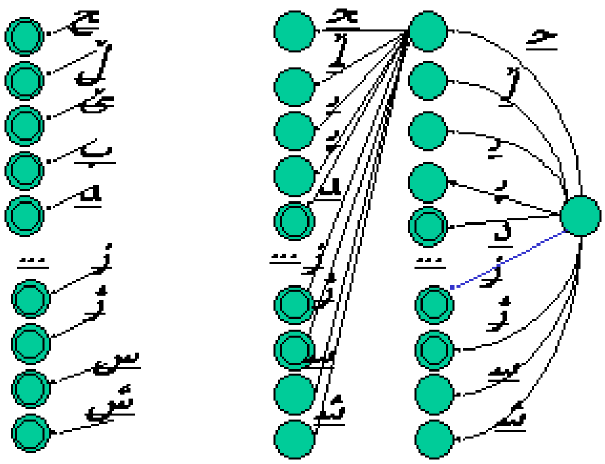

Figure 4: Font Graph for an Arabic cursive Script

In FIGURE 4 we have shown possible medial connections attaching to initial **h** on its left. These connections can be shown for most other initial forms and the number of possible connections increases as more letters are attached. The problem of testing all possible connections for this font graph is a combinatorial problem.

In our work, we have represented each glyph form (initial, medial and final) as a linguistic process that can communicate with its left and right neighbouring glyph. The interface parameters for each specific connection has been considered for matching the connections or glyph channels. Glyph channels are instances of linguistic channels.

By using a CLP representation, a finite state model based on communication channels has been derived and the connectivity of the font has been tested.

The CLP model for cursive font graphs is a CLP-II model where each glyph process has access to its immediate right and left neighbour. Another example of application of CLP-II is for modeling DNA strings and gene scrambling. Figure 5 shows an example of a scrambled gene. The DNA segments should be unscrambled to come up with the actual order for the gene (cf. canonical word order in grammar).

Each gene segment has boundary pointers in its left and right that give clues about the possible canonical order for a gene. An example of these pointers for ordered gene segments (based on [Landweber et al., 2000]) is shown next.

Table 2: DNA Segments (DSs) and boundary pointers

Left boundary	DSs	Right boundary
CAAAA	1	TA
TA	2	ATCTAAGAATGATGA
...	...	...
CTTGAAAATCAA	16	ATGCTTGAaTC
ATGCTTGAAcTC	17	AAGAAAAC
AAGAAAC	18	ATGATCTT
...	...	...
TTcCCAAaAAGGAAA	47	TCATG
TCATG	48	n.d.

We use a notion of gene process as an example of linguistic process that can communicate with its left and right neighbour. We model gene pointers as communication channels.

A gene process like a glyph process has access to two channels, but a gene process

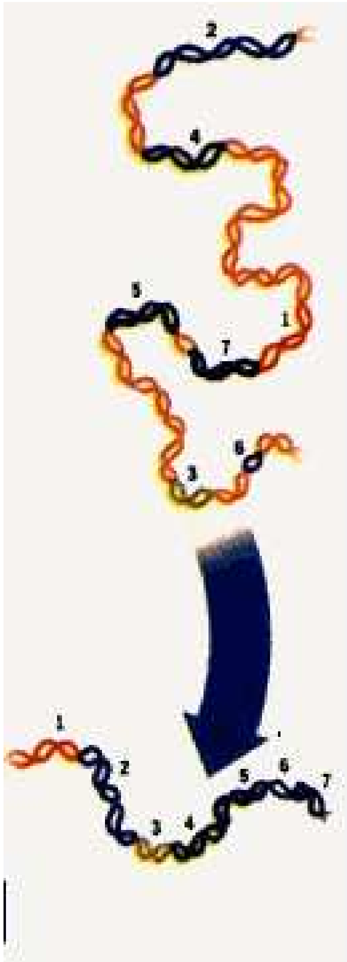

Figure 5: Parsing Gene Scrambling

can scramble in a gene string. This contrasts with the strict ordering constraints on communication for glyphs. A glyph in an string can only communicate with its immediate left and right neighbour in the string. In this regard, the word order constraints for glyph processes are analogous to the word order constraints in fixed word order languages, while the word order constraints for gene processes allow more flexibility, analogous to the free constituent languages.

For parsing scrambled genes we use the notion of path-set. The use of time-span for each process guarantees that only one instance of a specific gene segment in gene string will be used in each path. After parsing, depending on ambiguity in the gene representation, we will obtain one or few paths that correspond to the canonical order for the gene. The parsing program should also be robust to take into account the fuzziness in the gene left or right boundaries that might not match exactly with other boundaries. The notion of *fuzzy communication channel* [Rezaei, 1999] helps us to represent such a notion.

6 From π-calculus to π-grammar

In the previous sections we gave an outline of the basic concepts for a communicating process model of language and some of its non-linguistic applications. Figure 6 puts the present work in a background of the previous work in process algebra.

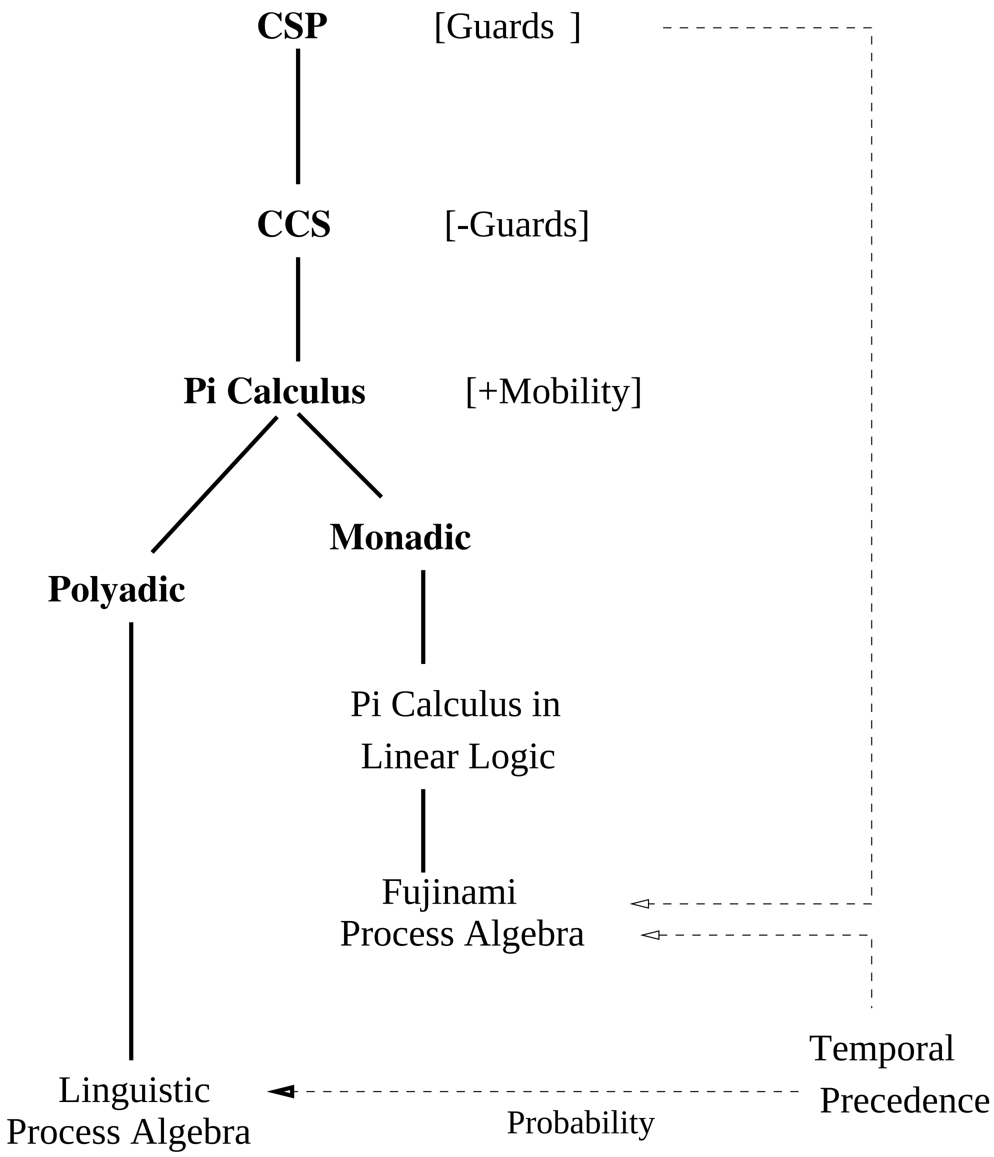

Figure 6: Process Algebras

In this process algebra background we see the evolution of process algebras into more complex algebras which allow the specification of mobility of processes in addition to the communication between processes. The minimal operators needed for our model are listed in the following:

$$
\begin{array}{lll}
P \longrightarrow & 0 \text{ (termination)} & \\
 & \mid \alpha & \text{(event)} \\
 & \mid \alpha \cdot Q & \text{(prefix)} \\
 & \mid [\mathrm{x{=}y}]\mathrm{P} & \text{(match]} \\
 & \mid \mathrm{P} \mid \mathrm{Q} & \text{(parallel)} \\
 & \mid \mathrm{P} + \mathrm{Q} & \text{(choice)} \\
 & & \\
\alpha \longrightarrow & \overline{x}(y) & \text{(exporting)} \\
 & \mid \mathrm{x(y)} & \text{(importing)} \\
 & \mid \tau & \text{(silent action)}
\end{array}
$$

Still these formal algebras lack the flexibility that we require for modeling the linguistic processes. Notions such as *ambients* [Gordon and Cardelli, 1998] need to be incorporated into these algebras to model locality. Stochastic and temporal extensions of these process algebras are also not easy to develop.

Another way to formalize our process model is to develop a grammar of communicating processes by incorporating a minimal set of operators and features for parallelism, communication and competition into the existing formal grammatical models in Computer Science such as Categorial Grammar(CG). The result will be a *process grammar* in which the notion of *process structure* – in contrast to a category – plays a key role. Major aspects of this process grammar are described in the remaining of this section.

- The morphology of a process[4] is described by prefix (.) and CG slash operators (\ and /).
- + and - are used for specifying the communication over a channel (e.g. *obj*). For example NP-obj and V+obj combine to form a communication link V=obj=NP.
- [ ] is used for temporal time stamp for each process. V[1] and S[1-6] depict processes that exist in first time period and first through sixth time periods respectively.
- {} is used to depict multiple communication channels for a process, e.g. V{+obj, +subj}; and multiple communication, e.g. S{V[3]=obj=NP[1],V[3]=subj=NP[2]}. The name before {} is used to specify a locality[5] or ambient.
- | is used for describing alternatives and competition at channel level and inter-process communication level. The competing NP-obj and NP-subj are depicted as NP{-obj | -subj}. The competing V=obj=NP[1] and V=obj=NP[1]

[4] Note that the operators we use in this subset are different from the operators that we used in the previous section based on the work in process algebra research.

[5] || operator is used for flexible word order inside a locality.

for an ambiguous case marking of an NP is depicted as S{V=obj=NP[1] | V=subj=NP[1]}.

- Process creation or reductions are depicted by $\rightarrow$. S $\rightarrow$ S S_{embed} is an example of process creation. In the case of localities or ambients, the export of mobile channels is possible during the process creation.

- The stochastic measures for activation are specified by # at all levels of specifications (channels, communication, localities, alternatives, ...). These measures are *possibility* values borrowed from fuzzy logic and fuzzy sets[6].

- The precedence operator ($\prec$) and its stochastic version[7] with an associated # value impose *optimal* linear constraints on channels.

The process grammar (π-grammar) that we described provides the notion of *imperfect channel* by the activation measure #. But the specification of π-grammar in its present form does not provide *conditional* channels. This needs further research. But π-grammar is expressive and flexible to be extended. It provides operators for expressing a large set of constraints and principles. For example the Resource Limitation principle that we specified earlier has been implemented by using a stochastic precedence operator [Rezaei, 2000b].

7 Conclusion

Starting from the communicating "sequential" processes in CSP, one can reach a richer notion of communicating "linguistic" processes (CLP). The main task is the extension of syntax and semantics of a formal model such as CSP or Pi-calculus in a minimal way to accommodate the required linguistic counterparts.

A parallel research in computational semantics has argued for the existence of channels of communication for semantic modeling. The introduction of a syntactic process algebra or a syntactic channel algebra will complement the work in semantics and provide a communication model for linguistic modeling [Rezaei, 1999]. Such a formal framework for modeling human communication apparatus is general enough and can be applied to other non-linguistic cognitive-biological domains, such as genetic modeling.

The formal specification of the model in terms of process algebras remains a complex task, but instead we have incorporated the communication notions into an existing grammatical framework. The result is Π-grammar, a process grammar.

[6] In a feature based specification of the grammar, we have for each feature or set, a corresponding # value or # set respectively. This requires the incorporation of fuzzy sets into feature unification.

[7] See [Rezaei, 2000b] for further detail.

References

[Bresnan and Kaplan, 1982] Bresnan, J. and Kaplan, R. M. (1982). Lexical-Functional Grammar: A Formal System for Grammatical Representation. In *Joan Bresnan,(editor) The Mental Representation of Grammatical Relations.* MIT Press.

[Carpenter, 1996] Carpenter, B. (1996). *Lectures on Type-Logical Semantics.* MIT Press, Cambridge, Massachusetts.

[Crocker, 1996] Crocker, M. W. (1996). *Computational Psycholinguistics: An Interdisciplinary Approach to the Study of Language.* Kluwer Academic Publishers.

[Gordon and Cardelli, 1998] Gordon, A. D. and Cardelli, L. (1998). Mobile ambients. In *Foundations of System Specification and Comutation Structures, Lecture Notes in Computer Science*, Springer Verlag.

[Hoare, 1985] Hoare, C. A. (1985). *Communicating Sequential Processes.* Printice-Hall, Englewood Cliffs, NJ.

[Johnson, 1996] Johnson, M. (1996). Resource Sensitivity in Grammar and Processing. In *The Ninth Annual CUNY Conference on Human Sentence Processing*, New York.

[Johnson, 1997a] Johnson, M. (1997a). Features as Resources. In *Proceedings of the LFG Conference*, University of California, San Diego.

[Johnson, 1997b] Johnson, M. (1997b). A resource-Sensitive Interpretation of Lexical Functional Grammar. In *WWW (Available online)*, Brown University.

[Landweber et al., 2000] Landweber, L. F., Kuo, T.-C., and Curtis, E. A. (2000). Evolution and assembly of an extremely scrambled gene. *PNAS*, 97(7):3298–3303.

[Milner, 1989] Milner, R. (1989). *Calculus of Communicating Systems.* Printice-Hall, Englewood Cliffs, NJ.

[Milner, 1993] Milner, R. (1993). The Polyadic π-Calculus: a Tutorial. In *F. L. Bauer, W. Brauer, and H. Schwichtenberg, editors, Logic and Algebra of Specification*, pages 203–246, Springer Verlag.

[Morill, 1994] Morill, G. V. (1994). *Type-logical Grammar: Categorial Logics of Signs.* Kluwer academic Publishers, Dordrecht.

[Rezaei, 1997] Rezaei, S. (1997). Linguistic Communicating Process Structures. In *Australasian Natural Language Processing workshop (ANLPW)*, Sydney.

[Rezaei, 1999] Rezaei, S. (1999). *Linguistic and Computational Analysis of Word Order and Scrambling in Persian.* PhD thesis, Division of Informatics, University of Edinburgh.

[Rezaei, 2000a] Rezaei, S. (2000a). Font connectivity and testing,. School of Computer Science, McGill University.

[Rezaei, 2000b] Rezaei, S. (2000b). Parsing Scrambling with Path Set: A Graded Grammaticality Approach. In *ACL International Workshop on Parsing Technology (IWPT2000)*, Trento, Itlay.

[Rezaei and Crocker, 1997] Rezaei, S. and Crocker, M. W. (1997). A Dynamic Representation of Grammatical Relations. In *Proceedings of the LFG Conference*, University of California, San Diego.

[Smolensky et al., 1992] Smolensky, P., Legendre, G., and Miyata, Y. (1992). Principles for an integrated connectionist/symbolic theory of higher cognition. Technical Report Report CU-CS-600-92, Computer Science, University of Colorado at Boulder.