A Deterministic Parser For Partially Free Word Order Languages

Key-Sun Choi

1. Introduction

Choi(1984) proposed a model and its algorithms for parsing natural languages based on Petri nets (Peterson 1981). Its underlying grammar is called *Semantic Petri Net Grammar* (SPNG), whose grammatical framework is isomorphic to that of Lee's 1983 *Augmented Montague Grammar* (AMG). Each grammar rule of SPNG consists of a syntactic part that specifies an acceptable fragment of a parsing tree, and a semantic part that specifies how the logical formulas corresponding to the constitutents of the fragment are to be combined to yield the logical formula for the fragment in Intensional Logic.

The parsing mechanism for SPNG resembles the execution rules of its underlying Petri net. For the control of parsing, each rule is fired after checking its *common conditions* and *special conditions,* and then *common actions* and *special actions* are applied to the input for the rule. *Common conditions* are checked before firing every rule. There are two types of common condition: *case-matchability* conditions and *priority* conditions. *Special conditions* are rule-dependent. Different rules have different special conditions. If a rule has a special condition, it is of higher priority over other rules that have no special condition. These mechanisms guarantee the deterministic parsing.

Common actions are applied to every rule, whose roles are:(1) concatenation of expressions in input places, (2) derivation of the category of output expresisons, and (3) translation of the logical formulas of input expressions. *Special actions* are rule-dependent and some rules take no special action.

A *dynamic rule-packeting* allows the SPNG parser to expect the next possible input structures and firable rules; which is a *top-down* mechanism. However, since the dynamic rule-packeting is made based on input strings, it is partially *bottom-up.* Input places of a rule transition play the role of the look-ahead facility.

The further detailed descriptions of SPNG and its parsing algorithms will not be presented in this paper, since they are well described in Choi(1984). Here, only the deterministic aspect of SPNG parser will be discussed; that is, the parsing with SPNG satisfies the *Determinism Hypothesis* (Marcus 1980).

2. Determinism Hypothesis

The *Determinism Hypothesis* is formulated as follows (Marcus 1980): "the syntax of any natural language can be parsed by mechanism which operates *strictly deterministically* in that it does not simulate a nondeterministic machine."

Marcus presented three specific properties of the parser such that they prevent the parser from simulating nondeterminism by blocking the implementation of either backtracking or pseudo-parallelism (Marcus 1980, Tennant 1981):

(1) In order to eliminate the possibility of backtracking, all syntactic substructures created by the parser are permanent. A top-down parsing with *augmented transition networks* (ATNs) (Woods 1970) makes extensive use of backtracking.

(2) In order to eliminate the possibility of simulating nondeterminism via pseudo-parallelism, all syntactic substructures created by the parser for a given input must be output as part of the syntactic structure assigned to that input. Bottom-up parsers make extensive use of parallelism in the form of keeping several candidate parses or partial parses active simultaneously.

(3) The internal state of the mechanism must be constrained in such a way that no temporary syntactic structures are encoded within the internal state of the machine. That is, any structure is not hidden in the state of the machine.

Taking the Determinism Hypothesis as a given, Marcus proposes a further set of three properties that any deterministic parser must embody: (1) it must be partially data driven; (2) it must be able to reflect expectations that follow from general grammatical properties of the partial structures built up during the parsing process; and (3) it must have some sort of looking-ahead facility, even if it is basically left-to-right.

3. Partially Free Word Order Languages and Determinism Hypothesis

Fig. 1 shows examples explaining three properties of any deterministic parser for Korean, a partially free word order language.

First, before the parser recognizes the case marker, it can not determine the case for 'Mary' in (1a) and (1b) of Fig. 1. A deterministic parser must

determine the case, after it recognizes the case marker postpositioned after 'Mary'. In a hypothesis-driven parser, it may first assume that the first term phrase has a nominative case marker among other alternatives. If that rule is applied to (1b), then the parser comes to find that it is the wrong hypothesis, and must make a backtracking in order to apply an alternative rule. Hence, any deterministic parser must be partially data driven.

The parser must:

Be partially data driven

(1a) $[[\text{Mary}]_{n*}\ [\text{ka}]_{n=1}]_{1*}$ (nominative case)

(1b) $[[\text{Mary}]_{n*}[\text{lil}]_{n=2}]_{2*}$ (accusative case)

Reflect expectations

(2a) $[\text{Mary-ka}]_{1*}[\text{non-ta}]_{10}$ 'Mary plays'

(2b) $[\text{Mary-ka}]_{1*}[\text{note-lil}]_{2*}$ $[\text{John-eke}]_{3*}$ $[\check{\text{c}}\text{unta}]_{3*2*10}$
'Mary gives John a note'

Have some sort of look-ahead

(3a) [Mary-ka John-il čo^{h}ahanin sasil-il]$_{2*}$ nae-ka anta.
'I know the fact that Mary likes John'

(3b) Mary-ka [John-il čo^{h}ahanin namča-lil]$_{2*}$ salaŋhanta.
'Mary loves a man who likes John'

Fig. 1. Some examples which motivates the structure of a deterministic parser for Korean.

Second, in (2a) and (2b) of Fig. 1, 'Mary-ka' belongs to the 1* category. In SPNG, an input place m* of the rules R4 and R5 has a token of 1*. The place m* is one of the input places of R4 or R5. Having that information, we can expect that the next input word belongs to either 10 or K*10 category, and R4 or R5 may be the next firing rule. On the other hand, in (2b), after recognizing 'John-eke', the place m* forms a stack of three tokens $<t_{3*}, t_{2*}\ t_{1*}>$, where t_{1*},t_{2*} and t_{3*} stand for $[\text{Mary-ka}]_{1*}$, $[\text{note-lil}]_{2*}$ and $[\text{John-eke}]_{3*}$ respectively. Hence, it can be expected that its main verb belongs to 3*2*10 place as one of the input places. This property says that any deterministic parser can not be entirely *bottom-up*. It must reflect expectations on the basis of the information which follows from the partial structures built up during the parsing process.

Finally, if a deterministic parser is to correctly analyze such a pair of sentences as (3a) and (3b) of Fig. 1, it must have a sufficient look-ahead facility. After only 'Mary-ka' is recognized, a parser cannot determine whether it is the subject of the verb 'salaŋha' or not. The syntactic struc-

tures can not be determined until a word after 'salaŋhanin' is recognized. Thus a deterministic parser must have a large enough *window* on the clause to see sufficient input data.

4. A Deterministic Parser with SPNG

We have discussed the necessity of three characteristics that follow from the *Determinism Hypothesis.* In this section, the structure of SPNG embodies these principles in the following ways:

(1) *A deterministic parser must be at least partially data driven.* A rule schema of SPNG is a kind of *pattern/action rule.* SPNG parser generates a token for each lexical item and then the corresponding basic place is filled with that token. Similarly, a token is generated as a result of rule-firing and then the corresponding derived place gets that token. Thus, the parser is directly responsive to the input which it receives.

(2) *A deterministic parser must be able to reflect expectations that follow from the partial structures built up during the parsing process.* If a place p_j, is filled with a token, and the first input place of a rule R_i is p_j then such rules become contained in the dynamic rule-packet. SPNG parser only attempts to match rules that are in the dynamic-packet, and rules in the rule-packet and just expected rules which follow from the partial structures built up during the parsing process. Rules are dynamically activated and deactivated according to the distribution and the flow of tokens in SPNG.

(3) *A deterministic parser must have some sort of constrained look-ahead facility.* Input places of SPNG provides this constrained look-ahead. The number of input places in a rule is limited to three. In order to fire a rule, all of its input places must be filled with tokens. Each token of a place can hold a single complete constituent, regardless of that constituents' size.

5. The Structure of SPNG Parser and Determinism Hypothesis

It has been demonstrated above that SPNG parser embodies the charcteristics that follow from the *Determinism Hypothesis.* In this section, it will be shown that SPNG parser operates *strictly deterministically.*

First, once a token is generated, the information in that token is not eliminated; all information of tokens in input places of a rule is inherited and transformed into the output place of that rule. Their syntactic and logical forms are combined and form a token in an output place. Hence, the structures generated by SPNG parser are permanent; it does not allow any backtracking.

Second, an ordered bag of input places of a rule behaves like a *pattern.*

If one of the rules in the rule-packet is fired, other rules that have common input places are automatically deactivated. One may suspect that the dynamic rule-packeting is a kind of parallelism. But, rules in the dynamic packet are only active, and any inactive rules are not applied. Only one of competing rules in the packet can be fired, and all of the information in input tokens will be kept on until one sentence is completely parsed.

Finally, SPNG has no hidden structures. An input bag of places plays the role of a *pattern* explicitly. It hides nothing. Every information for parsing is explicitly contained in tokens. It is not hidden in any extra-storage out of SPNG.

References

Choi, K.-S. (1984) 'Petri Net Grammars for Natural Language Analysis', *Language Research* **20**(2), 181-202, Language Research Institute, Seoul National University, Seoul.

Lee, K. (1983) 'AMG: Case Theory in Montague Grammar', *Proc. of '83 Kyoto Workshop on Formal Grammar,* Feb. 19-20, Kyoto.

Marcus, M.P. (1980) *A Theory of Syntactic Recognition for Natural Language,* MIT Press, Massachusetts.

Peterson, J.L. (1981) *Petri Net Theory and the Modeling of Systems,* Prentice-Hall Inc., Englewood Cliffs, NJ.

Tennant, H. (1981) *Natural Language Processing,* Petrocelli Books, Inc., New York.

Woods, W.A. (1970) 'Transition Network Grammars for Natural Language Analysis', *Comm. ACM* 13 (Oct.) 591-606.

Department of Computer Science
Korea Advanced Institute of Science and Technology
P.O. Box 150, Cheongryang
Seoul 131
Korea