Thai Syntactic PART Parser Using Least-Exception Logic (LEL)

Ampai Pornprasertsakul
Vilas Wuwongse
Kriengchai Chansaenwilai

Division of Computer Science
Asian Institute of Technology
P.O. Box 2754, Bangkok 10501, Thailand
Fax no. : (66-2) 516-2126, e-mail : ampai@ait.th

Abstract

The syntax of Thai sentences and its features are explained using PATR grammar formalism and a modified CHART parser written in LPA-Prolog was implemented . Some sentences were tested and resulted in highly ambiguous parsed trees due to the following characteristics: (1) a sequence of characters without blanks may form more than one word; (2) a sequence of words without blanks may form a phrase or sentence; (3) modifiers may follow the word modified; (4) a word may have more than one lexical categories; (5) word agreement is not available; (6) word order is not so strict and (7) embedded noun phrases are allowed in verb phrases and vice versa. This study is to reduce syntactic ambiguity by introducing Least-Exception Logic (LEL) into the present parser. The least exception ambiguous parsed tree is reduced by minimizing the total weight of exception expressed in objective function subject to grammar-rules constraints using Integer Linear Programming (ILP). The success of the parser depends on the weight of exception assigned to ambiguous lexicons and syntactic grammar rules, which are needed to be further explored.

I. Introduction

Human language understanding is apparently incremental in the sense of proceeding in a piece-meal fashion, carried out in small gradual steps as each words is encountered. One body of work which appears to be useful in incremental parsing and interpretation is reason (or truth) maintenance. A reason-maintenance system supports incremental formation and revision of beliefs [Wiren, 1990]. By viewing the construction of partial analyses of a text as analogous to forming beliefs about the meanings of its parts, a relation between parsing and reason maintenance can be conceived.

II. ATMS-Style (Asummption-based Truth Maintenance System) Approaches

The natural language understanding also can be further thought as an assumption-based reasoning process. Since there are various kinds of ambiguities and indeterminacies involved in natural language, and often these ambiguities or indeterminacies cannot be solved as soon as they are detected. Without making assumptions, it is difficult for a natural language understanding system when ambiguities or indeterminacies occur.

On the other hand, since many assumptions may be made during the process, there will be many possible directions in which the process can be continued, each of them indicated by a subset of the assumptions. It is not efficient to let the system consider all the possible directions then a selection criteria to provide the system the plausible direction should be introduced. An assumption-based reasoning can guide the processing in the following manner:

- make assumptions when needed
- maintain consistency of the belief set
- tell how plausible each possible direction is according to the plausibilities assigned to assumptions
- maintain plausibilities of assumptions

III. Least Exception Logic (LEL)

Least exception logic (LEL) is a model for default reasoning that is similar to ATMS in many regards. But LEL goes beyond ATMS that it helps to decide which assumptions to accept. LEL decomposed resolution into unification and solution, and performs the solution as an integer linear program (ILP) [Post, 1990]. The system operates as a nonmonotonic theorem prover where knowledge is stated in the predicate calculus and an ILP makes conclusions, maintains logical consistency, and orders the multiple extensions such that the extension that includes the least exceptions, in the form of defeated beliefs, is selected. The beliefs that are potential unsound are disjoined with propositions about their exception.

The LEL interleaves the unification and solution processes; the unification steps adds new constraints and objective function terms, and the solution step adjusts the solution for the newly instantiated data. The model of unification interleaved with solution via ILP is shown in the following Fig. 1.

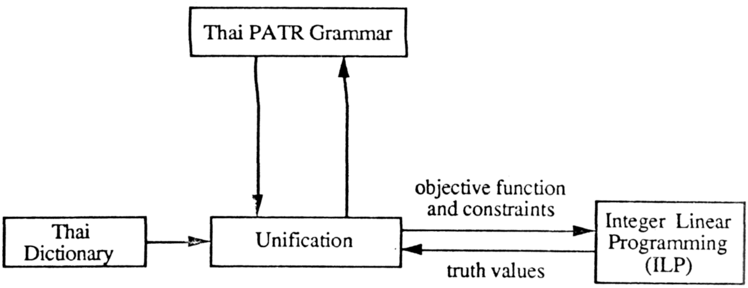

Fig. 1 The unification interleaved with solution via ILP

3.1 Reasoning with Parametric Variables

LEL reasons with parametric variables as an ILP. An arbitrary set of ground formulas can be transformed into an equivalent system of integer linear constraints. Each ground atomic *formula is considered to be a zero-one variable. The transformation ensures that the truth* values that satisfy the ground formulas are identical to the truth values that satisfy the linear constraints. The transformation of propositional formulas to an ILP is relatively straightforward.

The first step in the transformation of logical linear constraints is to convert the formulas to conjunctive normal form. Conjunctive normal form is a conjunction of clauses, where each clause is a disjunction of literals. A literal is an atomic formula or a negated atomic formula. LEL shares the use of conjunctive normal form with nearly all logic-based reasoning systems. In particular, Horn clause theorem provers, such as Prolog, and other resolution theorem provers use conjunctive normal form.

An algorithm to convert formula to conjunctive normal form is based on moving negation in and repeatedly distributing AND over OR. The following Table 1 shows some conversion examples.

Table 1

Formula	Conjunctive Normal Form
Q(X)	Q(X)
R(X) → T(X)	¬R(X) ∨ T(X)
P(X) ∧ ¬T(X)	P(X) ¬T(X)
P(X) ∨ Q(X) → R(X) ∨ S(X)	¬P(X) ∨ R(X) ∨ S(X) ¬Q(X) ∨ R(X) ∨ S(X)

Next step is to transform the ground clauses, into linear constraints. Letting P be a ground atomic formula, or Boolean variable, P becomes (1-P) and over the closed range i = 1 and 1 = n, the expression OR P_i becomes $\sum P_i \geq 1$. Negation is enforced through substitution of (1-P) for ¬ P, which map 0 to 1 and 1 to 0. Disjunction is enforced by constraining the sum of the terms to be greater than or equal to 1. Since the variables are Boolean this virtually defines disjunction. The detailed conversion of the one of the clauses is accomplished as follows:

$$\neg P(X) \vee R(X) \vee S(X) \text{ becomes } (1-P(X) + R(X) + S(X) \geq 1$$
$$\text{or } P(X) - R(X) - S(X) \leq 0.$$

The following Table 2 shows some conversion examples.

Table 2

Conjunctive normal form	Constraints
Q(X)	Q(X) = 1
¬R(X) ∨ T(X)	R(X) - T(X) ≤ 0
P(X)	P(X) = 1
¬T(X)	T(X) = 0
¬P(X) ∨ R(X) ∨ S(X)	P(X) - R(X) - S(X) ≤ 0
¬Q(X) ∨ R(X) ∨ S(X)	Q(X) - R(X) - S(X) ≤ 0

3.2 Representation Thai Sentence Structures as an ILP

LEL decomposes resolution into unification and solution, which are interleaved rather than performed concurrently. The unification phase produced a system of ground clauses, which is then solved by the ILP phase. The ILP phase produces solutions in the form of truth value assignments, which can be taken as ground singletons, or clauses with one literal, to support further unification, and the cycle repeats.

LEL handles default reasoning through an ILP, which has an objectives function as well as a system of constraints. An ILP is solved by optimizing the objective function subject to simultaneous satisfaction of the constraints. In case of inference involves potential exceptions, the objective function will be defined and constraints will be formulated. Here, we represent sound inference as an ILP to prepare for the later inclusion of defaults.

minimize $C = 3.\text{ exception }(VP,3) + 2.\text{ exception }(VP,2) + 1.\text{ exception }(VP,1)$

Subject to:

$$V + NP - VP - \text{exception}(VP,3) \leq 1$$
$$V + VP1 - VP - \text{exception}(VP,2) \leq 1$$
$$V + PsV - VP - \text{exception}(VP,1) \leq 1$$

The first constraint is equivalent to the grammar rule VP $\rightarrow$ V NP. If V = 1 and the word followed is a noun then this constraint will be satisfied and a verb phrase will be formed. The guidance of the ILP is that the rule with the highest weight will be tried first in order to minimized the objective function. If it is true then the highest weight of 3 will be eliminated and hence this objective function will be minimized. Similar interpretation is applied to the constraints remained.

3.3 Minimizing Exceptions in the ILP

The instantiated through unification define an ILP in boolean variables. These constraint the exceptions just as they constrain other atomic formulas. In LEL, the exception define the objective function, which is to minimize the total weight of exceptions that are set true, where the weight is given by the second parameter in the exception. That is, the objective function is the weighted sum of the exceptions. The solution to the ILP is the assignment of Boolean values of propositions such that each constraint is satisfied and the least exceptions are allowed, by weight.

minimize $C = 2.\text{ exception }(X,\text{weight}) + 1.\text{ exception }(Y,\text{weight})$
Subject to : ...

The second parameter of the exception is the weight of the exception. The exception weights order the individual exceptions, and the objective function orders the extensions.

For Thai language, the weight will be given based on the frequency of use which is shown in the following examples:

3.4.1) Lexical Ambiguity

Ex. 'Kon' (noun, 2), (verb, 1)

It means that the word 'Kon' is play the role as noun rather than verb.