

데이터베이스 의미론을 위한 한국어 피동형의 전산적 처리*

홍정하[†], 최승철[†], 이기용^{††}
(kleist[†], klee^{††})@mail.korea.ac.kr, whywest@yahoo.com[†]
고려대학교 언어과학과

Processing Korean Passives for Database Semantics

Jungha Hong, Seungchul Choe, Kiyong Lee
Department of Linguistics, Korea University

요 약

Hausser (1999)와 이기용 (1999a, 1999c)에서는 데이터베이스 관리 시스템(DBMS)을 이용하여 자연언어의 의미를 다루는 데이터베이스 의미론을 제안하였다. 특히 이기용 (1999c)에서는 수행도(trec), 논리 형태(logical formulas), 자질 구조(feature structure)와 같은 다양한 언어 표상 형식들을 관계형 데이터베이스 관리 시스템(RDBMS)의 표상 형식인 테이블 형식으로 전환 가능함을 보임으로써 데이터베이스 의미론에 관계형 데이터베이스 관리 시스템을 도입할 수 있음을 제시하였다.

한편, Lee (2000)에서 제시한 데이터베이스 의미론 모형에서는 데이터베이스 관리 시스템과 사용자(end-user)를 연결하는 언어 정보 처리 시스템(LIPS: Linguistic Information Processing System)을 제안하였다. 이 언어 정보 처리 시스템은 사용자에게 의해 입력된 언어 자료를 처리하여 그 분석 결과를 데이터베이스 관리 시스템에 전달하고, 이를 통해 구축된 데이터베이스에서 추출한 정보를 다시 사용자에게 전달하는 시스템이다.

이 논문은 한국어 '이, 히, 리, 기' 피동형을 전산처리를 할 수 있도록, 데이터베이스 의미론에서 핵심 요소인 언어정보 처리 시스템과 데이터베이스 관리 시스템을 구현하는 것이 목적이다.

1. 서론

이 논문은 한국어 '이, 히, 리, 기' 피동형을 전산적으로 처리할 수 있도록, 데이터베이스 의미론에서 핵심 요소인 언어 정보 처리 시스템(LIPS: Linguistic Information Processing System)과 데이터베이스 관리 시스템(DBMS: Database Management System)을 구현하는 것이 목적이다.

이를 위해 언어 정보 처리 시스템은 한국어 '이, 히, 리, 기' 피동형을 형태·통사적으로 처리할 수 있도록 표상 모형으로는 자질 구조를¹⁾, 구현 도구로는 Malaga²⁾를

사용하여 구현하며, 데이터베이스 관리 시스템은 피동문의 의미 정보를 테이블 형식으로 처리할 수 있도록 Oracle8을 이용하여 구현한다.

2. 데이터베이스 의미론

Hausser (1999)와 이기용 (1999a, 1999c)에서는 데이터베이스 관리 시스템(DBMS)을 이용하여 자연언어의 의미를 다루는 데이터베이스 의미론을 제안하였다.

이 이론에서 구현의 중심점은 데이터베이스이다. 태

* 이 논문은 1999년도 한국학술진흥재단 선도연구자 지원사업(연구책임자: 이기용)의 연구비를 지원 받아 이루어졌다.

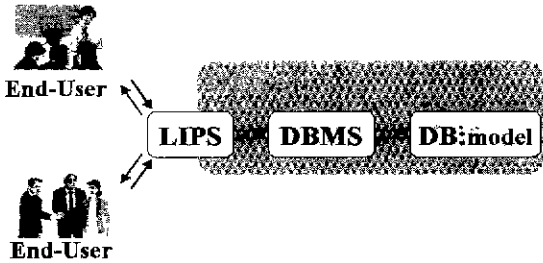
1) 이 논문의 자질 구조는 Sag & Wasow (1999)를 참고로 하여 수정하였다.

2) Malaga는 독일 Erlangen-Nürnberg 대학교 컴퓨터언어학과에서 좌연접문법(Left-Associative Grammar)을 표상할 수 있도록 C-언어를 보장하여 개발한 프로그래밍 언어이다(Beutel 2000 참조). 좌연접문법에 대해서는 Hausser (1989, 1999), 이기용 (1999b), 홍정하 (1998) 참조.

데이터베이스에는 언어를 분석한 정보가 일정한 양식으로 저장된다. 데이터베이스는 기능과 구조, 목적과 체계가 있어서 자료들의 비교 분석, 첨삭이나 검색 등을 효율적으로 처리할 수 있는 관리 시스템을 필요로 한다. 바로 데이터베이스 관리 시스템이 이러한 데이터베이스를 관리하게 된다.

Lee (2000)에서 제시한 데이터베이스 의미론 모형은 다음과 같다.

(1) 데이터베이스 의미론 모형



데이터베이스 의미론 모형은 데이터베이스(DB), 데이터베이스 관리 시스템(DBMS), 그리고 데이터베이스 관리 시스템과 사용자(end-user)를 연결하는 언어 정보 처리 시스템(LIPS: Linguistic Information Processing System)으로 구성된다. 이 언어 정보 처리 시스템은 사용자에게 의해 입력된 언어 자료를 처리하여 그 분석 결과를 데이터베이스 관리 시스템에 전달하고, 이를 통해 구축된 데이터베이스에서 추출한 정보를 다시 사용자에게 전달하는 시스템이다.

이 논문에서 언어 정보 처리 시스템은 Malaga를 이용하여 구현되었으며, 입력된 언어 자료를 처리하여 그 결과 중 의미 정보를 데이터베이스 관리 시스템에 전달하게 된다.

3. 데이터베이스 의미론에 적용: 자질구조에서 테이블로

이기용 (1999a, 1999c)에서는 수형도, 논리 형태, 자질구조와 같은 다양한 언어 표상 형식들을 관계형 데이터베이스 관리 시스템(RDBMS)의 표상 형식인 테이블 형식으로 전환가능함을 보임으로써 데이터베이스 의미론에 관계형 데이터베이스 관리 시스템을 도입할 수 있음을 제

시하였다.

관계형 데이터베이스 관리 시스템의 표상 형식인 테이블은 특히 속성(attribute)과 그것의 값(value)으로 구성된 자질 구조를 표상하기에 적합하다. 다시 말해서 테이블의 필드(field), 데이터(data), 레코드(record)는 자질구조의 언어 정보를 표상하기에 충분하다.

이기용 (1999c)에서는 (2)와 같은 자질구조의 언어정보가 (3-4)와 같은 테이블 형태로 전환 가능함을 보였다.

(2) 자질구조의 언어정보

- 가. [Word_form: “미아가”, POS: noun, Case: nom]
- 나. [Word_form: “미아를”, POS: noun, Case: acc]
- 다. [Word_form: “뛰다”, POS: verb, Case_frame: <nom>]
- 라. [Word_form: “사랑하다”, POS: verb, Case_frame: <nom, acc>]

(3) Table: word_form_for_nouns

Word_form	POS	Case
“미아가”	noun	nom
“미아를”	noun	acc

(4) Table: word_form_for_verbs

Word_form	POS	Case_frame
“뛰다”	verb	nom
“사랑하다”	verb	nom_acc

(2-가, 나, 다, 라)의 자질 구조는 각각 테이블에서 하나의 레코드를 형성하며, 자질 구조의 속성은 테이블의 필드, 자질 구조의 값은 테이블의 데이터에 해당된다.

따라서 이 논문에서는 데이터베이스 의미론에 적용할 수 있도록 언어 정보 처리 시스템에서의 표상 모형을 자질 구조로 사용한다.

4. 한국어 피동형의 특징

영어와 같은 인구어의 피동(passive)은 주로 통사적 차원의 문제로 다루어지지만, 한국어의 피동문은 일차적으로 문장의 서술어에 따라 결정되기 때문에 형태적 차원의 문제라 할 수 있다. 지금까지의 연구에서 피동문을 구성하는 한국어 서술어의 피동형식을 형태적 차원에서 네

가지 유형으로 정리할 수 있는데, 그 유형과 예는 다음과 같다³⁾.

(5) 피동형식의 유형

- 가. 타동사 어간 + '이, 히, 리, 기'
도둑이 순경한테 잡혀었다.
- 나. 자동사
소가 농부에게 맞았다.
- 다. 용언어간 + '아/어' + '지다'
수양제가 을지문덕에게 깨뜨려지다.
- 라. 명사성어근 + '되다'
그 사람의 글이 신문에 게재되었다.

이 네 가지 피동형식의 유형 중에서 타동사 어간에 피동접사 '이, 히, 리, 기'가 결합한 형태가 한국어에서 피동문을 구성하는 가장 대표적인 서술어이다. 그리고 이렇게 형태적으로 피동접사 '이, 히, 리, 기'와 결합한 어형 중에서 통사적으로 행위자와 행위를 받는 대상을 모두 나타낼 수 있는 것을 피동성이 강하다고 하여 진피동으로 분류한다.

한편, 표준국어대사전에서 표제어로 수록된 피동사 중에 '이, 히, 리, 기'와 결합하는 표제어의 빈도적 분포는 다음과 같다.

(6) '이, 히, 리, 기'의 분포⁴⁾

형태소	피동사
-이-	139 (38.4%)
-히-	91 (25.1%)
-리-	102 (28.2%)
-기-	29 (8.0%)
-우-	1* (0.3%)
계	362 (100.0%)

(6)에서 피동사를 구성하는 형태소 중에서 '이, 히, 리'가 '기'에 비해 보다 넓게 사용됨을 보이고 있다.

그러나 피동접사 '이, 히, 리, 기'와 결합할 수 있는 타동사 어간의 어휘 수는 360여 개로 극히 제한되어 있고⁵⁾, 실제 피동형을 형성하는 것과 그렇지 못한 것간의

규칙성을 포착하기 어렵다. 다음의 (7-가, 나, 다. 라)와 (7-가', 나', 다', 라')는 각각 피동형이 되는 것과 불가능한 것인데 그 기준 설정이 쉽지 않다⁶⁾.

(7) 피동형의 불규칙성

- 가. 뽑다-뽑히다
- 가'. 줍다-*줍{이/히/리/기}다
- 나. 팔다-팔리다
- 나'. 사다-*사{이/히/리/기}다
- 다. 몰다-몰리다
- 다'. 넣다-*넣{이/히/리/기}다
- 라. 때먹다-때먹히다
- 라'. 홀겨보다-*홀겨보이다

이 논문에서는 타동사 어간에 피동접사 '이, 히, 리, 기'가 결합한 진피동형을 연구 대상으로 한다.

5. 피동형의 형태적 처리

언어 정보 처리 시스템에서 피동형의 형태적 처리는 홍정하 (2000)을 기반으로 하여 이형태의 설정 및 처리 과정을 수정 보완한 것이다.

언어 정보 처리 시스템은 피동형 구성이 가능한 형태소들을 이용하여 정형의 피동형들을 생성해내는데, 그 결과는 다음과 같다.

(8) 정형(well-formed) 피동형의 생성

malaga > mg 4 먹 보 잘 찢 물 이 히 리 기 다

- 1: “먹히다”
- 2: “먹다”
- 3: “보이다”
- 4: “보다”
- 5: “찢기다”
- 6: “찢다”
- 7: “물리다”
- 8: “물다”

여기서 mg(morphological generate)는 구축된 결합규칙

하다(우인혜, 1995).

6) 이러한 피동형 불규칙성 처리에 대해서는 홍정하 (2000) 참조.

3) 박양규 (1990), 배희임 (1988) 참조.

4) *'-우-'로 파생된 피동사는 '불리우다'로서 '불리다'의 잘못된 쓰임으로 사용되는 단어

5) 우인혜 (1993)에 따르면 타동사에 '이, 히, 리, 기'가 결합하는 피동사의 개수는 약 150여 개이지만, 실제 국어사전에 공통으로 피동사로 기술되는 어휘수가 95개에 불과

에 따라 mg 뒤의 형태소로 결합가능한 어형을 모두 생성하는 명령어이다. 또한 숫자 4는 최대의 형태소 결합 개수를 제한하는 것이다. (8)은 이러한 어휘 생성 명령에 따라 형태소 ‘먹, 보, 잘, 찢, 돌, 이, 히, 리, 기, 다’가 결합하여 생성가능한 정형의 피동형과 능동형을 제시하고 있다.

한편, ‘나누다/나뉘다, 모으다/모이다, 잡그다/잡기다, 자르다/잘리다’와 같은 피동형은 (8)의 피동형과는 다른 결합방식을 취한다. 즉, 어간과 피동접사가 결합할 때 어형변화를 수반하기 때문에 ‘모으’ → ‘모으’, ‘잡그’ → ‘잡기’, ‘자르’ → ‘잘리’와 같은 이형태를 생성하여 결합해야 한다.

그러나 이러한 유형의 피동형의 처리에서 다음과 같은 비정형의 어형 분석 및 생성은 방지해야 한다

- (9) 가. *나누이, *모으이, *자르리, *잡그기
나. *모으다, *잘리다, *잡기다

즉, 이러한 유형의 어간은 반드시 이형태가 피동접사와 결합해야 하며, 이형태는 직접 종결어미와 결합할 수 없는 제약이 있어야 한다.

이러한 유형의 어형들을 언어정보 처리 시스템에서 생성하면 (10)과 같은 정형의 어형들이 생성된다.

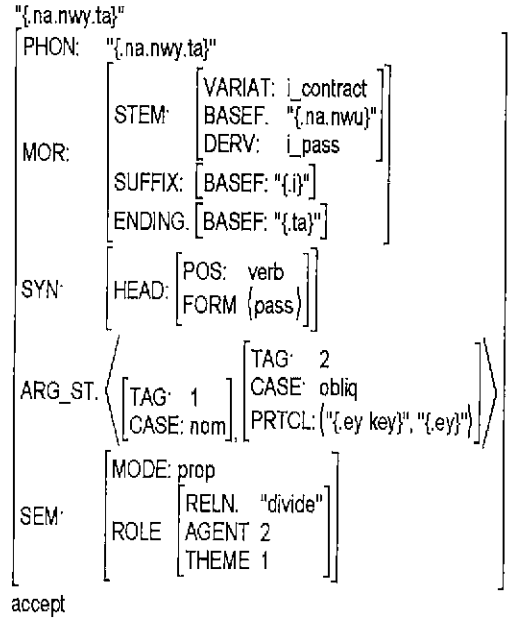
(10) 이형태를 이용한 정형의 피동형 생성

```
malaga> mg 4 나누 모으 모으 잡그 잡기 자르 잘리
이 | 히 리 기 다
```

- 1: "나뉘다"
- 2: "나누다"
- 3: "모으다"
- 4: "모이다"
- 5: "잡그다"
- 6: "잡기다"
- 7: "자르다"
- 8: "잘리다"

이러한 유형의 피동형 중 언어 정보 처리 시스템에서 분석한 ‘나뉘다’의 분석 결과는 (11)과 같다. 이 분석 결과는 자질구조로 표상되며, 음운(PHON), 형태(MOR), 형태(MOR), 통사(SYN), 논항구조(ARG_ST), 의미(SEM) 정보로 구분된다⁷⁾.

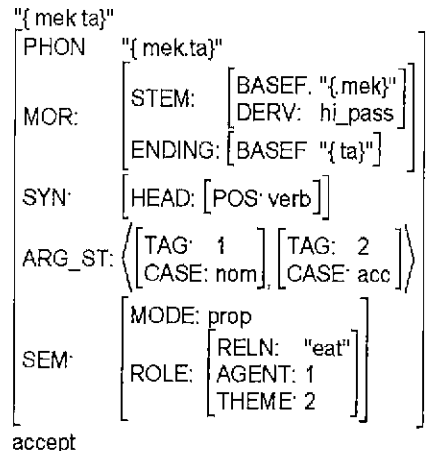
(11) '나뉘다'의 형태 분석 결과



6. 피동형의 통사적 처리

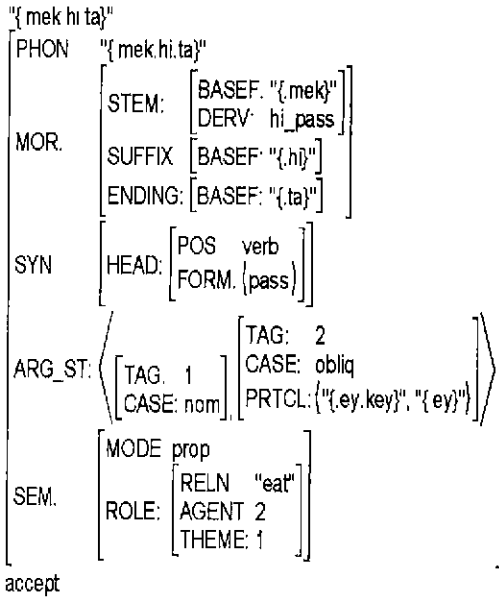
피동문을 능동문과 비교해 보면, 행위자와 행위의 대상은 동일하지만 해당하는 각 논항의 위치와 위치에 따른 격의 차이가 있다는 것이다. 따라서 타동사의 피동형은 논항 구조의 차이를 반영하면서도 동일한 의미 관계를 유지해야 한다.

(12) 타동사 ‘먹다’의 분석 결과



7) 자세한 것은 홍정하 (2000) 참조.

(13) 피동형 '먹히다'의 분석 결과

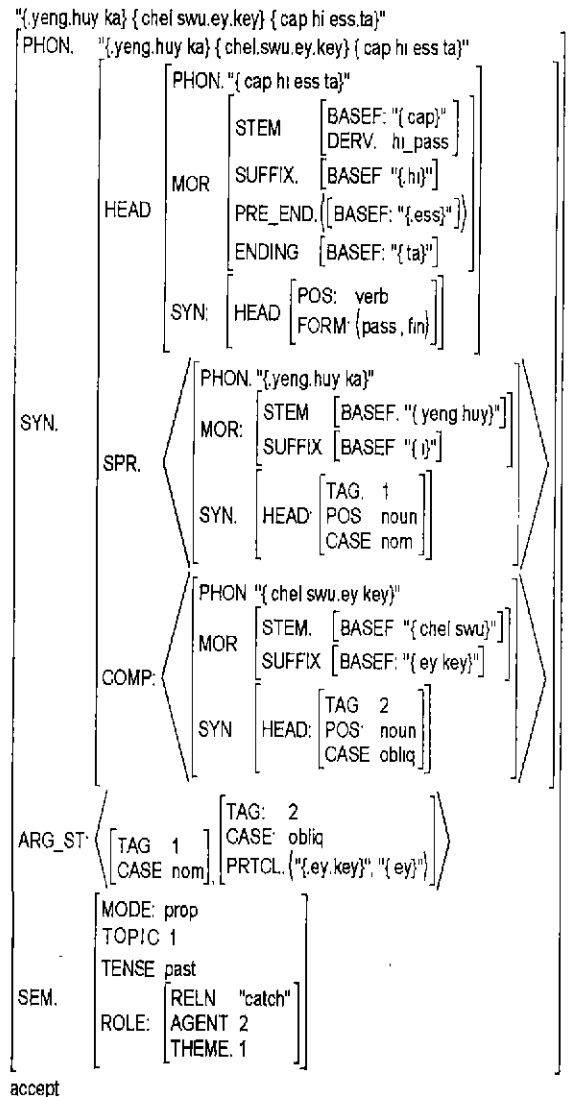


타동사 '먹다'의 자질구조 (12)와 이것의 피동형 '먹히다'의 자질구조 (13)을 비교해 보면, 형태 정보를 표상하는 MOR에 접사 '히'가 결합되어 있다는 것, 통사정보를 표상하는 SYN에서 동사가 피동형이라는 것, 그리고 논항구조 AGR_ST와 의미정보 SEM에서 AGENT, THEME의 값들에 차이가 있다는 것이다. 여기서 (12)와 (13)의 AGR_ST와 SEM을 자세히 살펴보면, AGR_ST의 TAG 값에 의해 행위역 AGENT와 대상역 THEME으로 지시되는 논항의 위치와 격이 달라졌다는 것뿐이다. 즉, (12)에서는 AGENT의 값 1과 동일한 TAG 값을 취하는 AGR_ST의 첫 번째 논항의 격은 주격 nom이고, THEME의 값 2와 동일한 TAG 값을 취하는 AGR_ST의 두 번째 논항의 격은 대격 acc이다. 그러나 (12)의 피동형인 (13)에서는 AGENT의 값 2와 동일한 TAG 값을 취하는 AGR_ST의 두 번째 논항의 격은 사격 obliq이고 조사 PRCTL이 '에게, 에' 중 하나이어야 함을 나타낸다. 반면에 (13)에서 THEME의 값 1과 동일한 TAG 값을 취하는 AGR_ST의 첫 번째 논항의 격은 주격이다. 따라서 각 의미역을 표상하는 논항의 위치의 차이와 그 논항의 격이 틀리지만 두 의미역은 그대로 유지된다.

이러한 피동형의 자질구조는 (14)와 같은 피동문의

분석을 허용한다. (14)는 동사 '잡히었다'가 HEAD이고, 명사구 '영희가'는 SPR, 명사구 '철수에게'는 COMP로 분석되었음을 나타내고 있다. 또한 SPR로 분석된 명사구 '영희가'는 SEM의 THEME 값 1과 동일한 TAG 값을 취하고, COMP로 분석된 명사구 '철수에게'는 AGENT 값 2와 동일한 TAG 값을 취하므로 각각 대상역과 행위역의 논항임을 나타내고 있다.

(14) '영희가 철수에게 잡히었다'의 분석 결과



8) 피동문에서 행위역 논항의 조사는 이 밖에도 여러가지가 가능하나, 이 논문에서는 '에게', '에'로 한정한다.

한편, (14) 의미 정보 SEM의 다른 정보는 능동문과 동일하지만 주제 TOPIC 정보는 능동문과 다르다. 즉, 피동문에서는 대상역의 논항이 문장의 주제가 되지만, 능동문에서는 행위역의 논항의 문장의 주제가 된다. 따라서, (14)은 피동문이므로 대상역 THEME의 값 1이 주제 TOPIC의 값이 된다.

7. 의미 정보의 추출

언어정보 처리 시스템에서 분석된 결과를 모두 관계형 데이터베이스 관리 시스템에 적용하는 것이 아니라, 이 연구에서는 분석 결과 중에서 의미 정보만을 추출한다. 즉, (14)와 같은 문장 분석 결과에서 의미 정보 SEM의 정보를 추출하면 다음과 같다.

(15) '영희가 철수에게 잡혔다'의 의미 정보

MODE: prop				
TOPIC: 1				
TENSE: past				
ROLE:	<table border="1"> <tr> <td>RELN: "catch"</td> </tr> <tr> <td>AGENT: 2</td> </tr> <tr> <td>THEME: 1</td> </tr> </table>	RELN: "catch"	AGENT: 2	THEME: 1
RELN: "catch"				
AGENT: 2				
THEME: 1				

그러나 (15)와 같이 의미 정보로만 구성된 자질구조에서는 논항의 의미역 AGENT와 THEME의 값처럼 숫자로 된 의미 정보는 해당하는 논항을 지시할 수 없으므로 의미 정보 추출시 해당하는 논항으로 대치시켜야 한다. (16)은 (15)의 각 숫자가 지시하는 논항을 대치시킨 결과이다

(16) '영희가 철수에게 잡혔다'의 대치된 의미정보

MODE: prop				
TOPIC: "영희"				
TENSE: past				
ROLE:	<table border="1"> <tr> <td>RELN: "catch"</td> </tr> <tr> <td>AGENT: "철수"</td> </tr> <tr> <td>THEME: "영희"</td> </tr> </table>	RELN: "catch"	AGENT: "철수"	THEME: "영희"
RELN: "catch"				
AGENT: "철수"				
THEME: "영희"				

(16)과 같은 의미 정보는 데이터베이스 구현시 테이블 형태로 전환된다.

8. 데이터베이스 구현

이기용(1999c)에서는 (15)와 같은 자질 구조를 (16, 17)과 같은 테이블로 분산하는 것을 제안하였다. 그러나 관계형 데이터베이스 관리 시스템(RDBMS)에서는 중첩 테이블(nested table)을 지원하지 않으므로 (16, 17)과 같이 구현하기에는 어려움이 있다.

(16) Table : sem

mode	topic	tense	role
prop	영희	past	*

(17) Table : role

reln	agent	theme
catch	영희	철수

(16, 17)과 같은 두 테이블로 분산 시키기 위해서는 객체관계형 데이터베이스 관리 시스템(ORDBMS)이나 객체지향 데이터베이스 관리 시스템(OODBMS)을 사용해야만 가능하다. 그러나, 객체지향 데이터베이스 관리 시스템은 기본적인 데이터베이스 관리 기능에 있어서의 안정성 및 성능의 검증 미비로 인하여 아직 제대로 인정받지 못하고 있다. 반면, 비록 표준화가 미비되어 제품마다 조금씩 차이가 있지만 객체관계형 데이터베이스 관리 시스템의 경우 현재 상용화되어 사용되고 있는 관리 시스템이다.

따라서, 객체지향 데이터베이스 관리 시스템을 바탕으로 중첩 테이블(Nested Table)을 지원하는 Oracle8을 사용하면 (15)와 같은 자질구조를 (16, 17)과 같은 테이블로 변환시킬 수 있다. Oracle8을 이용하여 이를 구현하는 순서는 (18)과 같으며, 그 구현과정을 제시하면 (19)와 같다.

(18) Oracle8을 이용한 구현 순서

가. role 객체 role_type을 만든다.

나. 객체 테이블을 만들어 객체 role_table_type을 만든다.

다. 중첩 테이블을 포함하고 있는 테이블 sem을 만든다.

(19) Oracle8을 이용한 (16)의 구현

```

SQL> CREATE TABLE role_type (reln VARCHAR2(20), agent NUMBER(2), theme(2));
Type created.
SQL> CREATE TYPE role_table_type AS TABLE OF role_type;
Type created.
SQL> CREATE TABLE sem (phon VARCHAR2(20), smode VARCHAR2(10),
  2 topic NUMBER(2), tense VARCHAR2(10), role role_table_type)
  3 NESTED TABLE role STORE AS ts_role;
Table created.
SQL> INSERT INTO sem VALUES ('영희가 철수에게 잡혔다', 'prop', '영희', 'past',
  2 role_table_type(role_type('catch', '철수', '영희')));
1 row created.
SQL> SELECT * FROM sem;
PHON                                SMODE TOPIC TENSE ROLE(RELN, AGENT, THEME)
-----
영희가 철수에게 잡혔다 prop   영희   past   ROLE_TABLE_TYPE(ROLE_TYPE(catch, 철수, 영희))
SQL> SELECT s.phon, s.smode, s.topic, s.tense, r.reln, r.agent, r.theme
  2 FROM sem s, table(s.role) r;

PHON                                SMODE TOPIC TENSE RELN  AGENT THEME
-----
영희가 철수에게 잡혔다 prop   영희   past   catch 철수   영희

```

9. 맺음말

이 논문에서는 한국어 '이, 히, 리, 기' 피동형을 처리할 수 있도록 구현된 언어정보 처리 시스템에서 이러한 피동형의 형태·통사적 처리과정을 살펴보고, 언어정보 처리 시스템을 통한 분석 결과 중 의미 정보를 추출하여 데이터베이스로 구현하는 방법을 제시하였다.

이 논문에서 제시한 언어 정보 처리 시스템과 데이터베이스 관리 시스템을 이용한 데이터베이스의 구현은 데이터베이스 의미론의 기본 틀이라 할 수 있다. 구축된 데이터베이스는 더 다양한 언어 정보 관리 및 검색, 추출 등을 가능케 한다.

앞으로 언어정보 처리 시스템을 더 다양한 언어현상들을 처리할 수 있도록 개발·확장하여하며, 또한 그 처리 결과를 효과적이고 체계적으로 데이터베이스로 구현하고 확장하는 방법을 논의해야 할 것이다.

참 고 문 헌

[1] 강명윤. 1997. "한국어의 형태론적 피/사동 현상의 소형구절구조 이론적 접근", 「어학연구」 33-1: 79-100.

[2] 국립국어연구원. 2000. 「표준국어대사전」.

[3] 박양규. 1990. "피동법", 서울대학교 대학원 국어연구회 편, 국어연구 어디까지 왔나, 동아출판사 493-499.

[4] 배희임. 1988. 「국어피동연구」, 고려대학교 민족문화연구소.

[5] 오라클. 1999a. "객체 관계형 데이터 모델", Oracle Korea Magazine, 1999 여름호, 한국 오라클(주).

[6] _____. 1999b. "객체 관계형 데이터베이스의 적용 가이드", Oracle Korea Magazine, 1999 가을호, 한국 오라클(주).

- [7] 우인혜. 1993. 「국어의 피동법과 피동 표현의 연구」, 한양대학교 박사학위논문.
- [8] _____. 1995. “국어 피동의 범위”, 「국어학」 26: 99-124.
- [9] 이기용. 1999a. “전산의미론”, 감범모 외 공저, 「형식 의미론과 한국어 기술」, 서울: 한신문화사.
- [10] _____. 1999b. 「전산형태론」, 고려대학교 출판부.
- [11] _____. 1999c. “데이터베이스 의미론의 기초: 자질 구조에서 테이블로”, 「1999년도 한글 및 한국어 정보처리 학술대회 논문집」.
- [12] 홍정하. 1998. 「좌연접문법을 이용한 영어명사구의 전산처리」, 고려대학교 석사학위 논문.
- [13] _____. 2000. “한국어 피동형의 전산처리”, 「2000년도 한국 언어정보학회 학술대회 발표논문집」.
- [14] Beutel, Björn & Roland Hausser. 2000. "Malaga 4.3" (unpublished), Abteilung für Computerlinguistik, Universität Nürnberg-Erlangen, Germany.
- [15] Hausser, Roland. 1989. *Computation of Language: an Essay on Syntax, Semantics and Pragmatics in Natural Man-Machine Communication*, Berlin: Springer-Verlag.
- [16] _____. 1999. *Foundations of Computational Linguistics: Man-Machine Communication in Natural Language*, Berlin: Springer-Verlag.
- [17] Lee, Kiyong. 2000. “Developing Database Semantics as a Computational Model”, Akira Ikeya and Masahito Kawamori (eds.), *Proceedings of 14th Pacific Asia Conference on Language, Information and Computation*.
- [18] Sag, Ivan & Thomas Wasow. 1999. *Syntactic Theory: A Formal Introduction*, CSLI.