

데이터베이스 의미론의 기초: 자질 구조에서 테이블로*

이기용
klee@mail.korea.ac.kr
고려대학교 언어과학과

A Basis of Database Semantics: from Feature Structures to Tables

Kiyong Lee
Department of Linguistics, Korea University

요 약

오늘날 전산망을 통해 대량의 다양한 언어 정보가 일상 언어로 교환되고 있다. 따라서 대량의 이러한 정보를 효율적으로 처리할 수 있는 언어 정보 처리 시스템이 필요하다. Hausser (1999)와 이기용 (1999)은 그러한 언어 정보 처리 시스템으로 데이터베이스 의미론을 주장하였다. 이 의미론의 특징은 자연언어의 정보 처리 시스템 구축에 상업용 데이터베이스 관리 시스템을 활용한다는 점이다. 이때 야기되는 문제 중의 하나가 표상(representation)의 문제이다. 그 이유는 언어학의 표상 방법이 데이터베이스 관리 시스템의 표상 방법과 다르기 때문이다. 특히, 관계형 데이터베이스 관리 시스템(RDBMS)에서는 테이블(table) 형식으로 각종 정보를 표시한다.

따라서, 이 논문의 주안점(主眼點)은 언어학에서 흔히 쓰이는 표상 방법, 즉 문장의 통사 구조를 표시하는 수형(tree)이나 의미 구조를 표시하는 논리 형태(logical form), 또는 단어나 구의 특성을 나타내는 자질 구조(feature structure)를 테이블 형식으로 대체하는 방법을 모색하는 것이다. 더욱이, 관계형 데이터베이스 관리 시스템에서는 테이블에 대한 각종 연산, 특히 두 테이블을 연결(link)하는 작업이 가능하고 이런 연산 과정을 통해 정보를 통합하거나 여과할 수 있기 때문에 관련 정보를 하나의 테이블에 표상하거나 정보 자료의 분산 저장과 자료의 순수성을 유지하는 것이 용이하다. 이 논문은 끝 이러한 점을 가급적 간단한 예를 들어 설명하는 데 그 목적이 있다.

1. 머리말

일상 생활에서는 사람들이 특별한 표상(representation) 방법을 쓰지 않고도 일상 언어만으로 충분한 언어 생활을 누리고 있다. 그러나 인간의 그러한 언어 사용에 대해 명시적으로 논하는 일, 특히 컴퓨터를 통해 그런 작업의 타당성을 실험하는 일은 형식화된 표상 언어가 없으면 불가능하다.1 수형(tree)을 통해 문장의 통사 구조를 표시하거나 술어 양화 논리나 람다 연산 언어(Lambda Calculus)로 문장의 의미 구조를 나타내 주는 것이 그 좋은 예이다.2 자질 구조(feature structure)를 통해 문장의 형태, 통사, 의미 구조에 대한 보다 섬세한 정보를 기술하려는 것도 매우 좋은 예

이다.3

이 논문은 언어학에서 중전에 사용되어 온 수형이나 논리 언어, 또는 자질 구조와 같은 표시 방법 대신에 테이블의 형식으로 문장의 통사적 정보나 의미 정보를 표시하는 방법을 논하려는 것이 그 목적이다.

오늘날 전산망을 통해 대량의 정보가 빠른 속도로 교환되고 있으며 이러한 정보의 매개체로 영어나 한글과 같은 자연언어가 흔히 쓰이고 있다. 따라서 이런 정보를 체계적으로 처리하고 관리하기 위해서는 경영 정보 등의 관리에 쓰여 온 Oracle이나 Informix와 같은 상업용 데이터베이스 관리 시스템을 일상 언어의 정보 처리에도 도입할 필요가 있다. 더욱이 이런 시스템에서 운용되는 언어로 SQL (Structured Query Language)과 같은 표준화된 질의 언어가 있으므로 이런 언어를 통해 자연 언어의 질의 응답 시스템을 비교적 쉽게 개발할 수 있으리라 추정된다. 테이블 형식으로 자연언어의 각종 정보를 표상하려는 것은 언어 정보

* 이 논문은 1996년도 한국학술진흥재단의 외국석학과의 공동연구과제(연구책임자: 이기용) 연구비에 의하여 연구되었음. 이 논문의 수정에 고려대 최재용 교수와 국립국어연구원 이준석 박사의 큰 도움이 있었다.

1 이에 대한 더 자세한 논의는 이기용 (1989)을 참조 바람.

2 강범모 외 (1999)에 실린 논문들은 주로 형식 논리를 통해 자연언어의 의미론을 다루는 방법을 소개했다.

3 Carpenter (1992)는 자질구조의 수리적 특성을 밝힌 다음, 자연언어의 문법 기술에 활용할 수 있는 가능성을 논했다.

의 처리와 관리 그리고 추론의 체계화에 관계형 데이터베이스 관리 시스템(Relational Database Management System: RDMS)을 활용하기 위한 실용적인 이유 때문이다.

2. 테이블이란?

어떤 의미에서 테이블은 가장 간단한 형식의 정보를 담는 그릇이다. 우선, 테이블을 가리키기 위해서 테이블의 이름(name)이 있어야 한다. 둘째, 테이블은 약 상자나 격자의 형식을 띠므로 상단부를 여러 영역(field)으로 분류할 필요가 있다. 셋째로, 각 영역에 해당하는 값(value)이 주어져야 한다. 다음은 전화 번호에 관한 정보를 담은 테이블이다.

(1) Table: 전화_번호부

교수명	학과	연구실전화
강명윤	언어	3290-2174
이승환	철학	3290-2020
송하춘	국문	3290-1964

이 테이블은 전화 번호부라는 이름과, 교수명, 학과, 연구실 전화번호의 세 속성으로 구성된 영역(field)과, 각 영역에 해당하는 3개의 기록(record)으로 구성되어 있다. 즉 언어학과 강명윤 교수의 전화 번호와 철학과 이승환 교수의 전화 번호, 그리고 국문과의 송하춘 교수의 전화 번호에 대한 기록이 적혀 있다.

테이블의 장점은 테이블에 대한 정보를 서로 연결(link)시킬 수 있다는 점이다. 예를 들면, 위의 전화 번호부를 다음의 주소록과도 연결시킬 수 있다.

(2) Table: 주소록

이름	주소	집전화
강명윤		
이기용	서초구반포본동반포아파트62-105	535-6256
이승환		907-2894
송하춘	서초구방배4동881-7	3476-2609

이때 전화번호부 속의 송하춘과 주소록의 송하춘이 일치하므로 송하춘에 대한 개인 정보가 늘어난다. 주소록에는 강명윤과 이승환에 대한 정보가 일부만 주어졌고 전화번호부의 명단에 없는 이기용에 대한 정보가 주어져 있다. 따라서 2개 또는 그 이상의 테이블을 연결할 때 부분적이지만 확실한 정보만을 서로 교환하게 되므로 정보 기록의 잉여성과 모순성을 배제할 수 있다.

그리고, 이 테이블을 연결하여 다음과 같은 새로운 테이블을 만들 수도 있다.

(3) Table: 개인정보

교수명	학과	연구실전화	주소	집전화
강명윤	언어	3290-2174		
이승환	철학	3290-2020		907-2894
송하춘	국문	3290-1964	서초구방배4동881-7	3476-2609

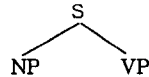
관계형 데이터베이스 관리 시스템의 한 가지 장점은 복잡한 정보를 한 테이블에 집중시키지 않고 분산처리할 수 있다는 점이다. 일상적인 정보는 그 유형이 복잡하고, 사용자의 관점에 따라 정보를 보는 초점이 달라진다. 예를 들면, 연구실 전화 번호에만 관심이 있는 사람은 정보를 많이 담고 있는 개인 정보 테이블보다는 오히려 연구실의 전화 번호부 테이블이 더 간편한 것이다. 따라서, 일상 언어의 정보도 관계형 데이터베이스 관리 시스템에서 테이블 형식으로 표시되었을 때 그 정보가 쉽게 분산처리될 수 있다.⁴

3. 수형에서 테이블로

수형은 일종의 그래프(graph)로서 언어학에서 구조를 표시하는 가장 기본적인 형식이다. 다음은 구구조 규칙 S → NP VP에 의해 도출되는 하향식(top-down) 수형의 예이다.

(4) $S \rightarrow NP VP$

(5) 하향식 수형



이 수형에서 S는 NP와 VP의 모점(mother)이며 NP와 VP는 S의 딸(daughter)이 된다. 이 관계를 테이블로 나타내면 다음과 같다.

(6) Table: tree0

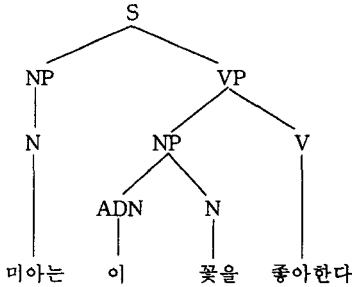
mother	daughter1	daughter2
S	NP	VP

⁴ 테이블의 특성에 대해서는 데이터베이스 관리시스템, 특히 관계형 시스템을 소개하고 있는 Elmasri와 Novathe (1994), Ramakrishnan (1998), Rob과 Coronel (1997)에서 자세히 설명되고 있다.

여기서 다음과 같은 규칙에 의해 NP와 VP는 다시 확장될 수 있다.

- (7) 규칙: NP → ADN N
 NP → N
 VP → NP V

(8) 수형



위의 수형에서 어휘 부분을 뺀 나머지 부분을 테이블로 옮기면 다음과 같다.

(9) Table: tree01

mother	daughter1	daughter2	mother level
S	NP	VP	0
NP	N		1
VP	NP	V	1
NP	ADN	N	2
N	△		2
V	△		2
ADN	△		3
N	△		3

이 테이블을 보면 S뿐만 아니라 NP와 VP도 확장되었음을 알 수 있다. 그런데 NP는 두 번 확장되었는데 한번은 N으로, 두 번째는 ADN과 N으로 확장되었다. 첫 번째 NP는 딸이 N밖에 없으므로 둘째 딸은 빈 공간(null)으로 표시되었다.

테이블에서 층위(mother level)의 값은 모점의 층위를 말하며, 딸들의 층위는 모점의 층위에서 1을 빼면 된다. 예를 들면, 층위 0인 S의 딸인 NP와 VP의 층위도 각각 1이다. 이러한 층위의 표시는 수형의 도출에 매우 중요한 역할을 한다. 층위 0은 첫째, 수형의 뿌리(root)를 가리킨다. 따라서, 하향식(top-down) 수형의 도출은 여기서 시작한다. 수형의 정형 조건으로, 수형에는 반드시 뿌리가 있어야 하되, 오직

뿌리가 하나이어야 한다. 그렇지 않으면 수형의 단일근조건(單一根條件, single-rooteness)을 어기게 된다. 위의 테이블에는 층위가 0인 모점이 오직 하나이므로 이 조건을 충족시켜 주고 있다.

둘째, 층위는 국부 수형(local tree)들을 연결시키는 역할을 한다. 만일 층위에 대한 정보가 없으면 위의 경우에 어느 NP가 어떤 식으로 확장되었는지 명확히 알 수 없으며, 그 결과로 여러 형태의 수형이 도출될 수 있다. 위 테이블에서 둘째 기록의 층위는 1이므로 층위 0인 S의 딸 NP를 확장했음을 알 수 있다. 반면에, 셋째 기록의 NP는 층위가 2이므로 층위 1의 VP의 딸 NP를 확장한 것이다.

테이블에는 수형의 종단 마디(terminal node)는 △로 표시되어 있다. 이 경우에 다른 범주들의 경우와 같이, 동일한 테이블에 어휘 범주인 N, ADN, V 등을 모점으로 하고 어휘들을 딸들로 직접 다룰 수도 있다. 그러면 테이블이 너무 커지기 때문에 정보의 분산 원칙에 의해 어휘는 별도의 테이블에 수록되도록 하고 어휘 삽입의 과정을 거치도록 한다.

다음은 위의 테이블에서 딸 범주들이 찾아가는 테이블이다. 딸 범주 중에 ADN, N, V는 해당하는 항목이 있으므로 그와 관련된 각각의 사전과 연결된다.

(10) Table: parts_of_speech

POS	lexicon_type
A	lex_adjective
ADN	lex_adnominal
N	lex_noun
V	lex_verb

이 테이블은 단순히 수형의 종단 마디를 품사(POS: parts of speech)별로 나누어, 품사별 사전을 찾아가도록 유도하는 역할만을 하고 있다.

어휘 삽입의 간단한 예를 들면 다음과 같다.

(11) Table: lex_verb

POS	base_form	present_decl	past_decl
V	사랑하	사랑한다	사랑하였다
V	놀	논다	놀았다
V	좋아하	좋아한다	좋아하였다

여기서 동사 V의 현재 서술형(present_decl)인 "좋아한다"를

택하여 Δ 와 대치하면 본래의 수행에 주어진 대로 동사 V의 종단 마디가 “좋아한다”가 된다.

어휘를 삽입할 때, V의 Δ 자리에 “사랑한다”나 “사랑하였다”를 삽입한다면 별 문제가 없지만 “논다”나 “놀았다”를 삽입하면 이 수행의 VP 구조에는 NP, 즉 목적어가 있기 때문에 문제가 생긴다. 그러나 이런 문제를 해결하기 위해 어떤 제약을 가해야 하는 것은 수행이나 테이블의 문제가 아닌 별도의 문제이므로 현재의 논의 대상에서 제외된다. 그리고, 사전의 테이블 구조도 더 섬세한 기술이 있어야 할 것이지만, 이것 또한 현재의 논의 대상이 아니므로 여기서 논하지 않겠다.

4. 논리 형태에서 테이블 형태로

이 논문의 주장 중의 하나는 자연언어를 통한 정보도 자질 구조에 의한 표시 방법보다 테이블에 의한 표시 방법이 편리하고 간결할 뿐 아니라, 보다 더 섬세한 정보를 표시해 놓을 수 있다는 것이다. 예를 들어, 다음 두 문장에 나타나는 대립관계(opposition)를 나타내는 방법을 비교해 보기로 한다.

(12) 준은 미아를 사랑하지만 미아는 준을 미워한다

여기서 “준”을 j로, “미아”를 m으로 옮기고, “사랑한다”와 “미워한다”를 각각 love'와 hate'으로 옮기면 다음과 같은 술어 논리의 형식문을 얻을 수 있다.

(13) [love'(j,m) \wedge hate'(m,j)]

이 논리 형태에서는 사랑 love'와 미움 hate'의 관계가 포착되어 있지 않다. 이것들의 대립관계를 표시하기 위하여 다음과 같은 의미 공준(meaning postulate)을 도입할 수 있다.

(14) 의미 공준

$$\square \forall x \forall y [[\text{love}'(x,y) \rightarrow \neg \text{hate}'(x,y)] \wedge [\text{hate}'(x,y) \rightarrow \neg \text{love}'(x,y)]]$$

위의 의미 공준은, 필연적으로 누구든지 사랑하면 미워하지 않고 미워하면 사랑하지 않음을 말한다.

이 논리 형식들을 테이블로 바꾸면 다음과 같다.

(15) Table: love_hate

Relation	Agent	Patient	Consequent
love	jun	mia	not hate
hate	mia	jun	not love

사랑과 미움의 테이블 속에 준과 미아의 관계와 그 결과를 표시하였다. 필요할 경우에는 사랑과 미움의 대립 관계를 별도의 테이블로 표시할 수도 있다.

(16) Table: opposition

Relation	Consequent
dead	not alive
fail	not pass
hate	not love
love	not hate
pass	not fail

앞의 테이블 (15)와는 달리, 대립 관계의 테이블 (16)은 술어들의 대립 관계를 좀 더 일반화시켜서 표현한 것이기 때문에 (14)와 같은 의미 공준을 일반화하여 테이블로 표시한 것으로 볼 수 있다.

5. 자질 구조에서 테이블로

단어나 구의 결합 관계를 제약하기 위해서는 그것들의 품사나 범주뿐 아니라 통사 및 의미적 특성을 밝혀주어야 한다. 이를 위해 언어학에서 흔히 쓰는 방법은 자질 구조로 그 특성들을 표시하는 것이다. 예를 들면, 다음과 같은 문장들의 정형성(well-formedness)을 논할 때 자질구조의 표시가 필요하다.

(17)

- a. 미아가 된다
- b. 미아를 사랑한다
- c. *미아를 된다

여기서 (a)와 (b)의 정문은 허용하고 (c)의 비문을 막기 위한 장치가 필요하다.

그 한 가지 방법은 첫째, CASE나 CASE_FRAME과 같은 속성을 도입하고 그것들의 값으로 nom, acc 등을 도입한다.

둘째, 이 속성과 그 값을 이용하여 명사와 동사의 자질을 다음과 같이 표시한다.

- (18)
- a. [Word_form: “미아가”, POS: noun, CASE: nom]
 - b. [Word_form: “미아를”, POS: noun, CASE: acc]
 - c. [Word_form: “땀다”, POS: verb,
CASE_FRAME: <nom>]
 - d. [Word_form: “사랑한다”, POS: verb,
CASE_FRAME: <nom, acc>]

셋째, 명사와 동사의 결합 조건으로 명사의 CASE 값이 동사의 CASE_FRAME 속에 있으면 결합이 가능하고, 그렇지 않으면 불가능한 것으로 정한다. 그러면, “미아가”는 “땀다”나 “사랑한다”와 결합할 수 있지만, “미아를”은 “사랑한다”와는 결합이 가능하지만 “땀다”와는 결합이 불가능하다. 따라서, 정문인 (17a)와 (17b)는 허용되고 비문인 (17c)는 제거된다.

위의 정보를 테이블 형식으로 바꾸는 과정을 보이면, 다음과 같다. 첫째, 어형의 특성을 다음과 같은 테이블로 표시한다.

(19) Table: word_form_for_nouns

Word_form	POS	CASE
“미아가”	N	nom
“미아를”	N	acc

(20) Table: word_form_for_verbals

Word_form	POS	CASE_FRAME
“땀다”	verb	nom
“사랑한다”	verb	nom_acc
“예쁘다”	adjective	nom
“준다”	verb	nom_acc_dat

표준관계형 데이터베이스 관리 시스템은 원자치(atomic value) 이외에는 영역의 값으로 취할 수 없다.⁵ 따라서,

5 여기서 말하는 표준 관계형은 제1정규형(first normal form)의 관계만을 다루는 시스템을 말한다. 제1정규형 관계에서는 어떤 영역의 값으로 다중치나 또 하나의 관계를 택할 수 없다. 객체지향형(object-oriented)인 경우에는 비정규형(non-first normal form) 관계를 다룰 수 있다. 자세한 것은 Elmasri Novathe (1994) 등을 참조하기 바람.

<nom, acc>과 같은 목록은 그 값이 될 수 없다.

그러나 다음과 같이 다중기호(multi-symbol)를 정의함으로써 간접적으로 목록과 같은 대상물이 영역의 값이 되도록 할 수 있다.⁶ 다음은 그 예이다.

(21) Table: multi_symbols_for_cases

Multi_symbol	Case1	Case2	Case3
nom_acc	nom	acc	
nom_acc_dat	nom	acc	dat

관계형의 테이블과는 달리, 자질 구조의 속성은 그 값으로 원자치나 목록뿐 아니라 자질 구조 자체를 값으로 취할 수 있다. 다음은 그 예이다.

- (22) [Word_form: “사랑한다”,
POS: verb,
CASE_FRAME: <nom, acc>,
SEM: [Tense: present,
ARGUMENT_FRAME: <agent, patient>,
CONTENT: “love”]]

위의 예를 보면, CASE_FRAME의 속성치는 nom과 acc으로 구성된 목록이며, SEM의 속성치는 3개의 자질로 구성된 자질 구조이다.

그러나 앞의 예에서와 같이 다른 테이블로 정보를 분산하여 연결시키는 방법이 있다. 목록이 영역의 값일 때는 목록의 항목들을 여러 영역으로 나누는 테이블을 작성하면 된다. 속성치가 자질 구조일 때도 별도의 테이블을 만들면 된다. 다음은 그 예이다.

(23) Table: sem_for_verbals

Verbal_key	Word_form	Tense	Arg_Frame	Content
a_0030	“예쁘다”	present	theme	“pretty”
v_0015	“땀다”	present	agent	“run”
v_0030	“사랑한다”	present	agent_patient	“love”
v_0150	“주었다”	past	agent_recipient_patient	“give”

위의 테이블을 보면 Verbal_key라는 영역란이 있다. 이 영역란은 테이블들을 연결시켜 주는 연결 열쇠(key)가 된다.

6 말라가(Malaga)라는 언어 처리 도구 언어에서는 다중기호가 사용되고 있다. 이에 대한 설명은 이기용 (1999)와 Beutel (1998)을 참조하기 바람.

따라서, "사랑한다"에 대한 테이블 (20)을 위의 테이블에 연결시키려면 (24)와 같이 이 테이블에 Verbal_key가 추가로 표시되어 있어야 한다.

(24) Table: word_form_for_verbals(revised)

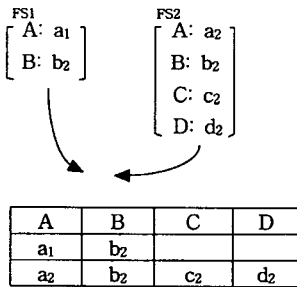
Verbal_key	Word_form	POS	Case_Frame
a_0030	"예쁘다"	adjective	nom
v_0015	"된다"	verb	nom
v_0030	"사랑한다"	verb	nom_acc
v_0150	"준다"	verb	nom_acc_dat

이 테이블과 앞에 주어진 테이블을 서로 연결하면 동사 "사랑한다"의 자질 구조 (22)에 해당하는 정보를 얻을 수 있다.

6. 테이블의 장점

그러면 관계형 시스템상에서 테이블 형식의 표상 방법이 자질 구조의 표상 방법과 비교하여 어떠한 장점을 갖는 것일까? 첫째, 테이블은 각 자질 구조들의 관련된 정보를 통합하여 표상하는 장점이 있다.

(25) 통합

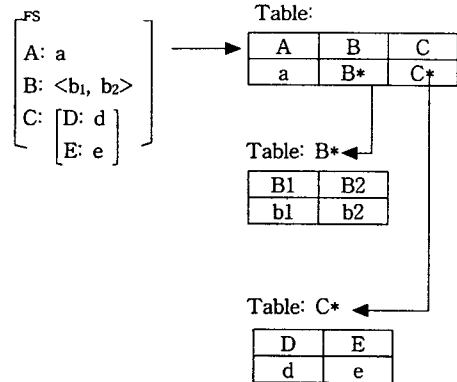


(25)에서 보는 바와 같이 각 테이블의 기록은 하나하나의 자질 구조에 해당한다. 테이블은 각 자질 구조들의 공통된 부분, 여기서는 공통된 속성에 따라 해당 자질 구조들을 간결하게 표상한다. 즉, 하나의 테이블이 여러 자질 구조를 표상할 수 있다.

둘째, 내부 구조를 갖는 자질 구조의 정보를 여러 테이블로 분산하여 처리할 수 있다. 자질 구조는 원자치 이외에

목록이나 자질 구조 자체를 속성치로 허용한다. 따라서, 자질 구조의 표상력은 강화되지만 자질 구조 자체는 매우 복잡해질 수 있다. 그러나 표준형의 테이블식 표상에서는 원자치만을 허용하기 때문에 테이블이 내부 구조를 가질 수 없다. 그 대신 (26)에서와 같이 내포된 관계(nested relation)를 허용함으로써 다른 테이블과 연결되어 정보의 관련성을 포착할 수 있다.

(26) 분산



셋째, 테이블 형태에서는 영역의 순서를 조정함으로써 정보의 초점을 바꿀 수 있다.⁷

(27) 영역 순서 조정

Table: Relation

Relation	Agent	Patient
love	Jun	Mia

Table: Agent

Agent	Relation	Patient
Jun	love	Mia

(27)에서 두 테이블은 동일한 정보를 표상하고 있다. 그러나 이 테이블들을 어떻게 해석하느냐에 따라 그 정보의 달라질 수 있다. 예를 들면, Table: Relation에서는 Relation에, Table: Agent에서는 Agent에 초점을 두는 것

⁷ 영역에 초점(focus)의 난을 추가하여 초점이 어디에 있는가를 명시적으로 표시할 수도 있다.

으로 해석하게 만들 수 있다. 즉 특정한 정보에 초점을 맞추기 위하여 테이블 영역의 순서를 조정할 수 있다.

끝으로, 관계형 데이터베이스 관리 시스템의 장점을 최대한 살릴 수 있다. 자료의 오류와 정보의 잉여성을 쉽게 배제할 수 있으므로 자료의 온전성(integrity)을 유지할 수 있다. 더욱이, 시스템 전체가 자료와 구조의 변화에 영향을 받지 않는다. 따라서, 테이블의 구조를 바꾸거나 단계적으로 증축하기가 쉽다는 시스템상의 장점이 있다.

7. 맺음말

수행과 논리 형식과 자질 구조는 언어학 기술의 기본 표상체들이다. 이 논문은 이러한 표상체들을 관계형 데이터베이스 관리 시스템에서 일반적으로 쓰이는 테이블 형태로 바꾸는 방법을 보였다. 될수록 쉽게 설명하기 위하여, 수리적 정의나 설명을 피하고 구체적인 예를 들었다.

언어학에서도 테이블이 많이 쓰여 왔다. 라틴어나 불어와 같은 굴절어에서 명사나 동사의 변화를 보여 줄 때도 테이블을 사용한다. 영어 사전은 보더라도 동사의 변화표가 반드시 부록으로 실려 있다. 우리말 사전에는 이런 표가 별로 없지만, 조사 목록이나 어미 목록, 또는 파생어의 목록을 일정한 표로 보일 수 있으며, 더 나아가 논항이나 연어 관계에 대한 용언들의 특성도 표로 제시할 수가 있을 것이다. 그러면, 언어 기술의 많은 부분이 표의 형태로 표시될 수 있을 것이며 이에 대한 장단점을 더욱 체계적으로 논의할 수 있을 것이다.

이 논문의 주 목적은 관계형 데이터베이스 관리 시스템을 활용하는 데이터베이스 의미론의 표상 방법으로 테이블 방식을 도입하자는 것이었다. 의미 정보뿐 아니라 더 나가서 언어 기술 전반에 테이블의 표상 방법을 도입할 수 있음을 이 논문에서 또한 논하였다. 자연언어의 통사 분석에 쓰이는 차트 파싱(chart parsing)이나 음운론에서의 격자(grid) 등이 테이블의 한 형태이다. 앞으로 자연언어 기술에서 테이블을 어떻게 활용할 것인지에 대한 본격적인 논의가 필요할 것이다.

참고문헌

- [1] 강범모 외. 1999. 형식 의미론과 한국어 기술. 서울: 한신문화사.
- [2] 이기용. 1989. "자연언어처리에 있어서의 의미의 문제". 어학연구 25.1: 225-238.
- [3] 이기용. 1999. "전산언어학", 강범모 외(1999)에 게재.
- [4] Beutel, Björn. 1998. "Malaga 4.2". Abteilung Computerlinguistik, Universität Erlangen-Nürnberg. Unpublished.
- [5] Carpenter, Bob. 1992. *The Logic of Typed Feature Structures with Applications to Unification Grammars, Logic Programs and Constraint Resolution*, Cambridge.
- [6] Elmasri, Ramez, and Shamkhe B. Navathe. 1994. *Fundamentals of Database Systems*. 2nd edition. New York: Addison-Wesley.
- [7] Hausser, Roland. 1999. *Foundations of Computational Linguistics: Man-Machine Communication in Natural Language*. Berlin: Springer-Verlag.
- [8] Ramakrishnan, Raghu. 1998. *Database Management Systems*. Boston: McGraw-Hill.
- [9] Rob, Peter, and Carlos Coronel. 1997. *Database Systems: Design, Implementation, and Management*. New York: Course Technology, a Division of International Thomson Publishing.
- [10] Stephens, Ryan K. Ronald R. Plew, Bryan Morgan, and Jeff Perkins. 1997. *Teach Yourself SQL in 21d Days*. 2nd edition. Indianapolis: Sams Publishing.