JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 27 No. 7, pp. 35-48, July 2022
https://doi.org/10.9708/jksci.2022.27.07.035

Knowledge Representation and Reasoning using Metalogic in a Cooperative Multiagent Environment

Koono Kim*

*Professor, Global Leadership School, Handong Global University, Pohang, Korea

[Abstract]

In this study, it propose a proof theory method for expressing and reasoning knowledge in a multiagent environment. Since this method determines logical results in a mechanical way, it has developed as a core field from early AI research. However, since the proposition cannot always be proved in any set of closed sentences, in order for the logical result to be determinable, the range of expression is limited to the sentence in the form of a clause. In addition, the resolution principle, a simple and strong reasoning rule applicable only to clause-type sentences, is applied. Also, since the proof theory can be expressed as a meta predicate, it can be extended to the metalogic of the proof theory. Metalogic can be superior in terms of practicality and efficiency based on improved expressive power over epistemic logic of model theory. To prove this, the semantic method of epistemic logic and the metalogic method of proof theory are applied to the Muddy Children problem, respectively. As a result, it prove that the method of expressing and reasoning knowledge and common knowledge using metalogic in a cooperative multiagent environment is more efficient.

▸ **Key words**: model theory, proof theory, metalogic, cooperative multiagent

[요 약]

본 연구에서는 멀티에이전트 환경에서 지식을 표현하고 추론함에 있어서 증명 이론적 방법을 제안한다. 이 방법은 논리적 결과를 기계적 방법으로 결정하므로 초기 인공지능 연구부터 핵심 분야로 발전해 왔다. 하지만 임의의 닫힌 문장들의 집합에서 항상 명제가 증명할 수 있지 않기에 논리적 결과가 결정할 수 있어지려면 절 형식의 문장으로 그 표현 범위를 제한한다. 그리고 절 형식의 문장들에서만 적용 가능한, 단순하면서도 강력한 추론 규칙인 비교흡수 원리(Resolution principle)를 적용한다. 또한 증명이론을 메타술어로 표현할 수 있으므로 증명이론의 메타논리로 확장 가능하다. 메타논리가 모델 이론의 인식 논리(epistemic logic)보다 향상된 표현력을 기반으로 실용적인 면과 효율면에서 우월할 수 있다. 이를 입증하기 위해 인식 논리의 의미론과 증명이론의 메타논리 방식으로 각각 Muddy Children 문제에 적용한다. 그 결과 협력적 멀티에이전트 환경에서 메타논리를 사용하여 지식과 공통지식을 표현하고 추론한 방법이 더 효율적임을 증명한다.

▸ **주제어**: 모델 이론, 증명이론, 메타논리, 협력적 멀티에이전트

• First Author: Koono Kim, Corresponding Author: Koono Kim
*Koono Kim (kok@handong.edu), Global Leadership School, Handong Global University
• Received: 2022. 06. 14, Revised: 2022. 07. 04, Accepted: 2022. 07. 07.

Copyright © 2022 The Korea Society of Computer and Information
http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

4차 산업혁명은 지능(intelligence)과 정보가 융합된 지능정보사회에 초점을 맞춘 것이라 할 수 있다[1]. 지능정보시대에 핵심 융합기술로 부상한 인공지능은 1956년에 John McCarthy를 중심으로 연구가 시작되었으며 이 연구는 크게 기호 기술(symbolic) 기반과 비기호(sub-symbolic) 기술 기반의 연구로 발전되어 왔다. 전문가의 지식을 논리적으로 생각하고 표현할 수 있는 전문가 시스템 연구는 대표적인 기호 기술 기반 연구로 분류되고 사람의 뇌 신경 세포 구조를 모방한 신경망이나 기계학습 등은 비기호 기술 기반 연구로 분류된다[2]. 최근 딥러닝(deep learning)과 빅데이터(big data) 기술 등, 비기호 기술 기반의 괄목할 만한 성과로 인해 전 사회적으로 인공지능에 관한 관심이 거의 폭발적이라고 할 수 있다.

최근 지능형 시스템이 우리 사회에 적용되기 위해서는 지능형 시스템의 결정을 설명할 필요성이 새로운 문제로 대두하게 되었다. 설명할 수 있는 인공지능(eXplainable Artificial Intelligence)의 접근 방식으로 보면 비기호 기술 방식은 장벽에 부딪히게 되었고 이 문제를 해결하기 위한 큰 노력이 필요하게 되었다[3]. 최근 자율주행차에 적용된 딥러닝 기술의 결과 오류에 의한 사고[4]에서도 알 수 있지만 기계학습으로 출력된 결과를 설명할 수 있어야 하고 잘못된 결과라면 어떤 부분에서 문제가 있었는지를 설명할 수 있어야 한다는 점이다. 반면, 전통적인 논리 기반 기술인 기호 기반 기술 방식은 이해할 수 있고 설명할 수 있는 지능형 시스템을 구축하기에 가장 유망한 기술 중 하나로 다시 주목받기 시작했으며 향후 지속적인 발전이 기대되는 분야로 평가되고 있다[2].

전통적으로 지식의 표현은 기호 기술 기반의 연구 분야이며 그 가운데서도 논리 기반 기술을 중심으로 발전되어 왔다. 논리 기반 기술은 다시 일차논리(first order logic)[5, 6, 7]와 메타논리 그리고 모달논리(modal logic)[8]와 인식논리(epistemic logic)[9] 형식으로 표현되어 왔는데 본 연구에서는 모달, 인식논리의 가상세계 의미론으로 지식의 의미를 표현할 때 제한적인 한계점을 좀 더 향상된 표현력을 가지는 증명이론의 메타논리 방식을 사용하여 지식을 표현하고자 한다. 또한 한 에이전트에서 지식을 표현하는 것에 멈추지 않고 변화하는 환경에 맞게 여러 에이전트 사이에서 공통지식(common knowledge)을 공유하고 추론할 수 있도록 표현하고자 한다.

멀티에이전트 환경은 4차 산업혁명의 핵심 기술인 지능형 시스템 구축에 있어서 필수 요소라 할 수 있는데 일반적으로 개별 에이전트는 서로 경쟁(competitive)하거나 협력(cooperative)할 수 있다. 경쟁하는 에이전트들은 개인의 이익을 최대화하려고 독자적인 목표를 추구하는 반면, 협력하는 에이전트들은 공동의 목표를 추구하고 개별 지식이나 정보들을 공유하고 다른 에이전트의 행동에 반감을 품지 않고 선의의 의도로 받아들인다. 본 연구에서는 인공지능 분야의 지식 표현과 추론에 널리 알려져 있는 '진흙 묻은 아이(Muddy Children)' 문제[10, 11, 12]를 협력적 멀티에이전트 환경에서 인식 논리의 가상 세계 의미론 표현 방식보다 더 향상된 표현력으로 변화하는 환경에 적합한 메타논리 방식으로 형식화하여 지식을 표현하고 추론하여 문제를 해결하고자 한다.

본 논문은 총 Ⅴ장으로 구성되었다. Ⅱ장에서는 지식 표현의 전통적 방법인 모델 이론을 소개하고 한계점을 보완할 수 있는 증명 이론적 방법을 소개한다. Ⅲ장에서는 에이전트 환경에서 증명 이론적 에이전트가 모델 이론적 방식보다 실용적인 부분에서 더 적합함을 분석하였고 Ⅳ장에서는 증명 가능성을 메타술어로 표현할 수 있기에 메타논리 방식의 지식 표현을 소개하고 메타논리가 모델 이론의 인식 논리 표현방식보다 실용적, 효율적인 면에서 더 우월함을 '진흙 묻은 아이(Muddy Children)' 문제를 비교 분석함으로 이 주장을 입증하고자 한다. Ⅴ장은 결론과 향후 연구 방향을 소개한다.

II. Model Theory and Proof Theoretic Methods

1. Model theory

전통적으로 지식과 믿음을 표현하는 연구는 모델 이론(model theory)을 이용하는 접근 방법이다. 모델 이론은 형식 언어와 그 해석(interpretation), 특정 형식 언어가 만들 수 있는 분류 유형에 관한 연구에서 시작되었으며 집합 이론 구조를 사용하여 형식적이든 자연적이든 모든 언어의 해석에 대한 연구이다. 일차언어 L의 해석 I는 다음처럼 구성된다:

- L이 표현하려는 객체들의 집합 $\triangle$(공집합 아님)를 정의역(domain)이라 한다.
- L의 상수에 대해 $\triangle$의 원소를 각각 하나씩 대응시킨다.
- L의 n항 함수 기호에 대한 $\triangle^n \rightarrow \triangle$의 n항 함수를 각각 하나씩 대응시킨다.
- L의 n항 술어 기호에 대한 $\triangle^n$의 부분집합 관계를 각각 하나씩 대응시킨다.

일차언어로 표현한 문장들의 모델 이론적 방법의 의미를 통해 절 형 문장들의 의미를 설명한다. 문장이 참이거나 거짓임을 결정하기 위해 문장에 표현된 기호들에게 각각의 의미를 부여해야 한다. 여기서 논리 연결사와 수량사의 의미는 바뀌지 않고 고정되지만, 상수나 함수 혹은 술어 기호에 부여되는 의미는 언제든 바뀔 수 있다. 문장들은 관심을 가지는 세상에 관한 표현이고, 관심 대상의 객체 집합을 여기서 정의역이라 정의하고, 여기서 변수는 정의역을 대상으로 하고 상수는 정의역 원소에 대응되며 술어 기호는 정의역에서의 관계로, 함수 기호는 정의역 위에서의 대응으로 각각 대응된다. 이런 설명이 모델 이론에서 해석의 기본적 개념이다.

해석 I가 일차언어 L의 해석일 경우, 닫힌 문장 A가 해석 I에 관해 참(truth)이면 해석 I를 닫힌 문장 A의 모델(model)이라 정의하고 $\models_I A$ 형식으로 표현한다. 그리고 일차언어 L의 닫힌 문장들의 집합 T의 모든 요소가 해석 I에 대해 참(truth)이면 해석 I를 닫힌 문장 집합 T의 모델이라 하고, $\models_I T$ 형식으로 표현한다.

일차언어 L의 닫힌 문장 집합인 T 그리고 또 다른 문장 P가 있다고 가정할 때, 문장 P가 집합 T의 논리적 결과가 되는 필요충분조건은 일차언어 L의 모든 해석 I에 관해 해석 I가 집합 T의 모델이면 해석 I 또한 문장 P의 모델인 경우이다. 닫힌 문장 P는 닫힌 문장들의 집합 T의 논리적 결과라는 사실을 $T \models P$ 형식으로 표현하고 T가 P를 "논리적으로 함의 한다(logically entail)" 혹은 P는 T의 "논리적 함의" 라고 말한다. $T \models P$임을 증명하는 방법은 $T \cup \{\neg P\}$가 모순이라는 것을 증명하면 되는데 그 까닭은 $T \models P$가 의미하는 내용은 어떤 해석이든지 집합 T의 모델이면서 동시에 P의 모델이 된다는 의미이기 때문에, $T \cup \{\neg P\}$가 참(truth)인 해석은 존재할 수 없다. 그러나 $T \cup \{\neg P\}$가 모순이라는 사실을 증명하기 위해서는 정의역 $\triangle$를 바탕으로 어떤 해석도 모델이 존재할 수 없음을 보여야 하지만, 정의역 $\triangle$가 공집합이 아니라는 조건만으로 모순이라고 증명할 알고리즘이 존재하지 않는다. 그러므로 정의역을 설정하는 알고리즘으로 잘 알려진 Herbrand 해석을 사용한다.

일차언어 L의 Herbrand universe(집합) H_U는 일차언어 L에 나타난 상수와 함수 기호를 사용하여 생성할 수 있는 모든 기초항의 집합으로 정의한다. 만약 일차언어 L에 상수가 없을 경우 H_U에 임의 기호를 상수로 추가해서 기초항들을 생성한다. 일차언어 L의 Herbrand base(기저)는 일차언어 L에 표시된 술어 기호에 H_U의 기초항들을 인수로 사용해 생성할 수 있는 모든 기초문장들의 집합을 말한다. 그리고 일차언어 L의 Herbrand 해석은 정의역을 Herbrand 집합 H_U로 하고 일차언어 L의 n항 함수 기호 f에 대해 $(t_1, ..., t_n) \rightarrow f(t_1, ..., t_n)$로 정의하는 매핑 $H_U{}^n \rightarrow H_U$를 함수 기호 f에 대응시키는 Herbrand 기저의 부분집합으로 정의한다.

예를 들어, 두 혼(Horn)절의 모임 T가 다음과 같이 있다고 가정하자.

$$human(aristotle) \leftarrow$$
$$mother(mom(X), X) \leftarrow human(X)$$

두 혼절의 모임 T의 Herbrand 집합은 다음처럼 표현될 수 있다.

$$H_U = \begin{Bmatrix} aristotle, mom(aristotle), \\ mom(mom(aristotle)), ... \end{Bmatrix}$$

그리고 Herbrand 기저도 다음과 같이 나타낼 수 있다.

$$\begin{Bmatrix} human(aristotle), \\ human(mom(aristotle)), ..., \\ mother(aristotle, aristotle), \\ mother(mom(aristotle), aristotle), ... \end{Bmatrix}$$

Herbrand 집합과 Herbrand 기저는 모두 무한집합이다. Herbrand 해석 중에서

$$\begin{Bmatrix} human(aristotle), human(mom(aristotle)), \\ mother(aristotle, aristotle) \end{Bmatrix}$$

는 두 혼절의 모임 T의 모델이 될 수 없지만

$$\begin{Bmatrix} human(aristotle), \\ mother(mom(aristotle), aristotle) \end{Bmatrix}$$

는 T의 모델이므로 Herbrand 모델이 될 수 있다.

Herbrand 해석에서 함수 기호들과 상수의 대응은 항상 고정되어있으므로 Herbrand 기저의 부분집합 하나는 Herbrand 해석이 될 수 있다. 반면 Herbrand 기저의 임의의 한 부분집합을 택하면, 부분집합의 원소인 각각의 기초문장을 참으로 하는 Herbrand 해석 한 개가 대응된다. 그러

나 일반적으로 집합 T에 관한 해석과 Herbrand 해석 모두 무한의 가짓수가 존재하므로 한정된 시간 안에 $T \models P$임을 결정할 수 없다. 즉 닫힌 문장 P가 닫힌 문장들의 집합 T의 논리적 결과임을 설명할 수 있는 모델 이론적 방법이 직관적으로는 설득력이 있어 보이지만, 계산 가능성(computability) 면에서 볼 때 닫힌 문장들의 집합 T에 관한 유한의 해석이 가능할 때에만 $T \models P$를 증명할 수 있다. 그러므로 $T \models P$일 경우는 항상 유한 단계 안에서 이것을 증명할 수 있는 증명 이론적인 방법이 필요함을 알 수 있다.

2. Proof Theoretic

일반적으로 닫힌 문장들의 집합 T와 하나의 닫힌 문장 P에 대해 $T \models P$ 경우 항상 유한 단계 안에서 이것을 증명할 수 있다. 집합 T에서부터 문장 P의 증명(proof)은 유한의 문장 나열로 마지막에 P가 나타나는 문장이며, 처음부터 문장 P가 나타날 때까지 각각의 문장은 집합 T에 포함되거나, 논리적 공리(logical axiom) 또는 이미 생성된 문장들에서 MP(modus ponens) 추론 규칙을 적용한 결과여야 한다. 여기서 논리적 공리란 문장이 논리적 형식으로 인해 모든 해석에서 참인 것을 말하며, 논리적 공리의 개수는 무한이다. 그러므로 임의의 문장을 나타내는 패턴 변수를 이용하여 유한의 공리 스키마를 사용하는데, $\alpha \rightarrow (\beta \rightarrow \alpha)$와 같은 함축 소개 스키마[13]가 하나의 예이다. α와 β는 패턴 변수를 나타낸다. 이 외에도 네 개의 공리 스키마를 더 사용한다. 이같이 논리적 공리를 사용하므로 T를 일차논리의 공리적 이론 혹은 일차이론이라 말한다. 증명에 사용되는 유일한 추론 규칙 MP는 다음과 같이 분수 모양으로 표현한다.

$$\frac{\alpha \rightarrow \beta,\ \alpha}{\beta}$$

이는 $\alpha \rightarrow \beta$와 α 같은 형식의 문장이 있을 경우 β와 같은 문장을 유도할 수 있다는 의미이다.

닫힌 문장들의 집합 T에서 논리적 공리들과 추론 규칙 MP를 이용한 닫힌 문장 P의 증명이 존재하면, 문장 P는 집합 T로부터 증명가능하다(provable)라고 정의한다. 또한 문장 P는 집합 T의 정리(theorem)라고 말하고 $T \vdash P$ 형식으로 표시한다. 증명은 논리적 결과를 기계적 방법으로 결정하므로 초기 인공지능 연구부터 핵심 분야로 발전해 왔다. 그러나 임의의 T에 대해 어떤 P나 $\neg P$가 항상 증명 가능하지는 않기 때문에 논리적 결과가 항상 결정 가능하지 않다. 현실적으로 기계적인 방식의 처리를 위해서는 임의의 닫힌 문장들이 아니라 절 형식의 문장으로 그 표현의 범위를 제한하게 된다.

임의의 두 개의 절 C_1, C_2에서, C_1의 어떤 리터럴 L_1 그리고 C_2의 어떤 리터럴 L_2가 서로 부정이라면 C_1, C_2에서 L_1, L_2를 각각 삭제한 후 그 결과를 논리합으로 연결한 절 형식으로 유도할 수 있는데 이러한 처리 과정을 비교흡수 원리(Resolution principle)라 한다. 비교흡수 원리는 절 형식의 문장들에서만 적용 가능한 아주 단순하면서도 강력한 추론 규칙이다. 논리프로그래밍은 절 형식의 표현 범위를 더욱 제한하여 혼(horn) 절 형식의 문장들에 대한 비교흡수 원리를 적용한 프로그래밍 언어이다. 지금도 논리프로그래밍의 문법과 추론의 기능 그리고 성능을 향상 시키려는 계산 논리(computational logic)[14] 연구가 다양하게 진행 중이다.

다음은 비교흡수 추론의 예를 들어 보고자 한다. 다음 두 혼 절이 있다고 가정하자.

$$human(aristotle) \leftarrow \qquad \text{2-①}$$

$$mother(mom(aristotle), aristotle) \leftarrow human(aristotle) \qquad \text{2-②}$$

2-②를 절 형식으로 나타낸 문장은 다음과 같다.

$$mother(mom(aristotle), aristotle) \vee \neg human(aristotle) \qquad \text{2-③}$$

리터럴 $\neg human(aristotle)$이 2-①번과 서로 부정이므로 비교흡수 추론을 실행하면 다음 2-④의 결과 문장으로 유도된다.

$$mother(mom(aristotle), aristotle) \qquad \text{2-④}$$

3. Epistemic Logic and Possible Worlds Semantics

모델 이론을 사용한 대표적인 방법이 모달 연산자로 확장한 인식 논리이다. 하나의 에이전트일 경우 인식 논리로 정의하지만, 의미적 특성에서는 모달 논리의 가능성과 필연성의 의미를 가능 세계(possible worlds) 의미론으로 발전시킨 크립키(Kripke) 구조를 사용한다. 멀티에이전트 환경에서는 인식 논리의 의미론으로 정의한다.

인식 논리는 명제논리에 모달 연산자를 사용해 확장한 형태로 볼 수 있고, 정렬 식 $K\alpha$은 "α를 안다"로 읽는다. 믿음 논리는 모달 연산자 K 대신 믿음 연산자 B를 사용하는 것 외에 모두 동일하고, 식 $B\alpha$은 "α를 믿는다"로 읽는다. 인식 논리는 몇 가지의 공리 스키마를 만족한다.

$K(\alpha\rightarrow\beta)\rightarrow(K\alpha\rightarrow K\beta)$ 2-⑤

$K\alpha\rightarrow\alpha$ 2-⑥

$K\alpha\rightarrow KK\alpha$ 2-⑦

$\neg K\neg\alpha\rightarrow K\neg K\neg\alpha$ 2-⑧

2-⑤은 "α이면 β이다를 안다면 α를 알면 β를 안다"라는 의미의 문장이고, 2-⑥은 "α을 안다면 α는 참이다"라는 의미이고, 2-⑦과 2-⑧은 자기성찰에 관한 문장으로, 각각 "어떤 α라는 사실(α라는 사실이 아니란 것)을 안다면(모른다면) α를 안다는(모른다는) 사실을 안다"라고 읽을 수 있다. 그리고 인식 논리의 추론 규칙에는 다음 두 가지가 있다.

$$\frac{\alpha,\ \alpha\rightarrow\beta}{\beta}$$ 2-⑨

$$\frac{\alpha}{K\alpha}$$ 2-⑩

2-⑨ 추론 규칙은 일차논리의 추론 규칙인 MP와 동일하고, 2-⑩ 추론 규칙은 α 가 주어지면 $K\alpha$을 유도 가능하다는 인식의 필연성(epistemic necessitation) 규칙이다.

지식에 대한 인식 논리의 특성은 단일 에이전트에 관한 분석이다. 하지만 인공지능과 컴퓨터 과학 영역에서는 본질적인 지식의 특성보다 지식과 행동 간의 관계인 실용적인 면을 더 의미 있게 다룬다. 예를 들어, "자율 주행차가 교차로를 지날 때 무엇을 알아야 하며, 신호등이 없는 교차로에서는 어떻게 운행하는지 알 수 있는가?"처럼 지식과 행동은 상호보완적 관계인 경우가 많다. 더 나아가 멀티에이전트 환경인 경우, 주어지는 입력의 추론 외에도 다른 에이전트에 대한 추론은 훨씬 복잡함을 알 수 있다.

멀티에이전트 환경에서는 인식 논리의 가능 세계 의미론으로 정의한다. 멀티에이전트 환경에서 에이전트들의 집합 $G=\{1,2,...,n\}$ 있고, 단순명제 $p,q,r,\ldots$ 로 구성된 집합 Γ가 서술하는 세계에 대해 집합 G의 에이전트들이 추론한다. 각 에이전트에 관한 모달 연산자 $K_1,...,K_n$을 사용해서 인식 논리의 식을 구성하고, 정렬 식 K_ip로 "에이전트 i은 명제 p을 안다"처럼 문장을 표현할 수 있다. 집합 Γ 위에서 G를 위한 크립키(Kripke) 구조 $M=\langle W,v,\omega_1,...,\omega_n\rangle$형식과 같은 튜플이다. 여기서 W는 가능 세계의 집합 혹은 상태의 집합이며 v는 가능 세계 집합 W의 각 세계에서 Γ의 명제들에게 진리값을 대응시키는 해석을 말한다. 즉, 각 세계 $w\in W$에 관해 $v(w):\Gamma\rightarrow\{true,false\}$이다. 그리고 ω_i는 가능 세계 집합 W 위에서의 이항관계, $W\times W$의 부분집합이다.

$v(w)$는 진리값 함수이며 w에서 p가 참 또는 거짓인지를 배정한다. 예를 들면, 명제 p가 "포항에 눈이 온다"라는 명제이면 $v(w)(p)=true$는 크립키 구조 M의 세계 w에서 포항에 눈이 온다는 상황을 나타낸다. 이항관계 ω_i는 i 에이전트가 w 세계에서 주어진 정보를 기반으로 x라는 세계가 가능하다고 생각하면 $(w,x)\in\omega_i$이기에 가능성 관계라고 정의한다. 가능성 관계 ω_i는 가능 세계의 집합 W 위에서 반사적(reflexive), 추이적(transitive), 대칭적(symmetric)인 합동 관계 특징을 가진다. 가능 세계 의미론은 구조 M의 w에서 p의 참/거짓을 배정하는 해석이고, 구조 M의 세계 w에서 포항에 눈이 온다는 상태를 $(M,w)\models p$처럼 표현한다. $(M,w)\models p$은 "(M,w)에서 p는 참이다(성립한다, 만족한다)"로 읽는다.

멀티에이전트 환경에서 가능 세계 의미론의 핵심은 에이전트 i는 M의 세계 w에서 α라는 사실을 안다는 것은 i 에이전트가 w 세계에서 가능하다고 생각하는 모든 세계에서 α가 참이라는 개념을 형식화한 것이다. 여기서 α는 명제논리의 정렬 식이고 M은 해석을 위하여 임의로 설정한 고정된 정적인 집합이라는 한계점이 있다.

III. Proof Theoretic Agent

II장 1절에서 설명한 모델 이론에서의 핵심 기능은 참의 개념을 문장과 의미 구조 간의 관계로 추정해서 언어와 경험 사이의 관계를 명확히 하는 것이다. II장 3절에 소개한 가능 세계 의미론에서도 역시 언어와 경험의 관계에 관한 설명으로서, 모델 이론의 의미 구조 역시 독립적으로 존재하는 현실에 의해 경험이 발생한다는 가설을 전제로 하는데, 이런 사실은 수학적으로 현실을 이상화한 것이다. 모델 이론의 또 다른 목적은 문장 집합 T와 문장 P 사이의 논리적 귀결에 대한 것이다. II장 1절에서 설명한 논리적 귀결의 모델 이론적 정의는 문장 집합 T와 하나의 문장 P에 적용된 기호들의 의미와 상관없이 집합 T가 성립될 때마다 문장 P가 성립한다는 직관적이고 일반화한 개념으로 형식화한 것이다.

하지만 이런 정의는 모델 이론이 경험과 언어 간의 관계에 관한 설명을 제공한다는 첫째 기능을 완벽하게 실행하지 못한 것으로 본다. 또한 논리적 귀결의 개념을 구체화한 둘째 기능 역시 부분적인 성공으로 분석한다. 그 이유

는 두 기능 모두 언어의 구문적 분석과 상관없이 개체와 함수 그리고 관계로 구성된 정의역(실체)이 존재한다는 모델 이론의 가설 때문에 여러 문제를 일으킨다. 이 장에서는 모델 이론의 문제점들을 찾고 증명 이론적 방법이 해결 방법이 될 수 있음을 분석하고자 한다.

1. A Practical Approach to Model Theory

세상과 상호작용하고 이런 맥락에서 논리를 사용하여 추론하는 에이전트는 경험과 언어 간의 관계에 대해 특별하고 보다 실용적 관점을 제공한다. 에이전트에 의해 구성 및 조작되는 기호들의 표현은 에이전트 내부의 '정신적 언어(mental language)'로 형식화한 지식 베이스의 문장들로 볼 수 있다. 정신적 언어의 개념은 일반 프로그래밍 언어가 이해하는 방법과 유사하다. 고급 언어로 작성된 프로그램은 낮은 수준의 언어, 기계어로 컴파일되므로 처음 고급 언어로 인식할 수 없지만 프로그램 동작과 의미를 이해하고 추론하려면 여전히 고급 언어로 작성되고 적합한 수준의 가상 머신에서 실행되는 것으로 보아야 한다.

논리 프로그램의 기본 이론은 적합한 형식의 논리가 고급 프로그래밍 언어의 역할도 가능하고 지식 베이스의 역할 역시 할 수 있다. 데이터베이스는 프로그램의 경우와 같이 다양한 레벨에서 이해할 수 있는데 외부 레벨은 사용자가 보는 관점이고 물리적 레벨은 컴퓨터가 보는 관점이며 개념 레벨은 데이터베이스의 설계자가 데이터베이스의 의도한 동작을 이해하고 추론할 때 보는 관점이다. 이 개념 레벨에서 데이터베이스는 논리적 형식 문장들의 집합인 논리적 이론으로 설명할 수 있다. 또한 에이전트는 정신적 언어로 형식화한 문장들의 지식 베이스 혹은 이론의 형식으로 세상과 상호작용을 통한 신념을 표현하는 것으로 볼 수 있다.

일반적으로 에이전트의 지식 베이스는 관찰 문장과 이론 문장으로 구성될 수 있는데 전자는 입력을 기록하고 경험에 대응하는 문장이고 후자는 경험에 직접 대응하지 않는 문장이다. 관찰 문장은 개체를 식별하고 분류하며 기록하는 기본 원자 문장으로 변수를 가지지 않는다. 이론적 문장의 도움으로 입력 관찰 문장에서 다른 기본 원자 문장이 파생될 수 있는데 이 문장은 이전 혹은 미래의 관찰 문장들과 비교 가능하다. 기본 원자 문장이 과거나 미래의 입력 관찰과 일치할 때 '참'으로 간주할 수 있다.

예를 들어, "부엌에서 연기가 발생한다." 라는 입력 관찰을 지식 베이스는 기록할 수 있다. '부엌'과 '연기'를 표현하는 기호로 적당한 '정신적 상수'가 사용된다. 과거의 '지식'이 존재하는 경우 이 상수들은 지식 베이스에 이미 존재할 수 있고, 아니면 새로운 상수일 수 있다. 관찰 기록이 '~에서 ~가 발생한다.'는 관계를 표현하기 위해 술어 기호를 사용하고 관찰 시간은 어떤 형식의 정신적 타임 스탬프에 의해 기록할 수 있다:

$$isa(\text{연기}_1, \text{연기}) \quad isa(\text{부엌}_1, \text{부엌})$$
$$\text{발생하다}(\text{연기}_1, \text{부엌}_1, \text{시간}_1)$$

시간1은 연기가 발생한 시간을 표현한 정신적 타임 스탬프이고 지식 베이스의 이론적 문장은 어떤 형식으로든 다음처럼 지식을 나타낼 수 있다:

"언제 어디서나 연기가 발생하면 그 전에 연기가 발생하게 된 발화사건이 먼저 발생한다."

그리고

"언제 어디서나 발화 사건이 발생하면 곧 화재 상태가 된다."

이론적인 문장을 사용해 부엌에 불이 났거나 아니면 곧 일어날 것이라는 미래의 결론을 유도할 수 있고 이 결론은 동일한 시간 또는 다른 시간, 다른 관찰의 결과로 나타난 다른 관찰 문장들과 비교 가능하다. 관찰이 지식 베이스의 정신적 언어로 표현되고 기록되면 유도된 문장들과 입력 문장들 사이의 비교는 순전히 정신적 언어에 따라 구문적으로 처리된다.

관찰 문장의 집합이 완전무결하고 이상적이라면 이 개념은 모델 이론에서 모델의 개념과 거의 유사하다. 모델 이론에는 실제의 개체, 함수, 관계로 구성된 세계가 있으나 좀 더 실용적 이론은 에이전트가 동화되도록 요구받는 관찰 문장들의 연속적이고 피할 수 없는 입력 스트림만 존재한다는 것이다. 에이전트 환경에서 입력 스트림의 소스를 조사하고 그 특성에 대해 예측하는 과정은 불필요하고 전혀 도움 되지 않는다. 모델 이론에서 '참'이란, 문장과 주어진 세계의 상태 간의 정적인 대응이다. 하지만 컴퓨팅을 추구하는 실용주의 이론에서 가장 중요한 요소는 '참'과 언어와 경험 간의 대응이 아닌, 계속해서 변화되는 관찰 문장의 피할 수 없고 연속적으로 끊임없이 진행되는 입력 흐름을 지식 베이스에 적절히 동화시키는 일이어야 한다.

2. Agent's Knowledge Assimilation

에이전트의 지식 베이스가 문장들의 집합으로 구성되었을 때 입력 문장이 동화되는 방법에 관하여 계산적으로 실현가능한 알고리즘을 지식동화(knowledge assimilation)의 과정에서 제안하였다[15]. 지식동화 알고리즘은 에이전트의 지식 베이스에 관찰을 동화시키는 것 외에도 데이터베이스 갱신 및 자연어 문자 이해 그리고 과학 이론 검증

및 확장 등과 같이 여러 방면으로 사용할 수 있다. 지식동화의 추상 알고리즘은 다음과 같다:

임의의 한 에이전트 지식 베이스의 지식동화에서 현재 주어진 상태를 Kb로, 입력 문장을 P로, 그 이론의 후속 상태를 Kb'로 두었을 때 그사이의 관계는 자원 제한적 연역으로 다음 ㉠-㉤ 경우 중 하나에 의해 결정된다:

㉠ P은 Kb의 논리적 귀결(logical consequence)이다.

㉡ Kb의 일부분은 P 그리고 Kb 나머지 부분을 이용하여 논리적으로 함축된다. 즉, $Kb = Kb_1 \cup Kb_2$이고 Kb_2 은 $Kb_1 \cup \{P\}$ 논리적 귀결이다.

㉢ P와 Kb는 모순적(inconsistent)이다.

㉣ ㉠-㉢ 중에 어떤 관계도 아니다.

모델 이론을 지식동화 알고리즘과 비교할 때, 모델 이론에서 입력 문장은 세계에 대한 완전하고 정확한 설명이고 모든 것들이 미리 주어지는 특별하고 제한적인 경우들을 다룬다고 볼 수 있다. 지식동화의 경우에서 ㉠-㉣은 모델 이론에서 '참'의 재귀적 정의와 비슷하지만 가장 큰 차이는 모델 이론이 언어의 구문 구조와는 별도로 관계, 속성, 개체를 포함하는 의미 구조의 존재를 가정하고 있다는 점이다. 반면 지식동화는 지식 베이스에 입력된 문장이 동화되어야 하는 연속적이고 지속적인 흐름만 있다고 가정한다는 점이다.

결론적으로 지식동화는 언어와 경험 간의 관계에 관한 설명으로서 모델 이론에 대한 구문론적 실용적 대안을 제공할 수 있다. 경험은 끊임없이 변화되는 에이전트의 지식 베이스에 동화되는 입력 문장의 피할 수 없는 연속적 흐름의 형태를 취한다고 가정하고 있다. 지식 베이스는 유용한 정보들을 구성하고 효율적인 접근을 제공한다. 그 결과 에이전트의 지식 베이스는 최대한 경험과 일치해야만 한다. 그리고 이상적으로는 일관된 설명도 제공해야만 한다.

3. Logical Consequences' Proof Theoretical Specification

위에서 분석한 사항들은 언어와 경험 간의 관계를 설명하는 데 있어서 모델 이론이 가지는 한계점을 지적한 것이다. 모델 이론은 정신적 언어와 세계 간의 관계 그리고 자연어와 의미 사이의 관계를 설명하지 못하거나 큰 도움이 되지 않는다. 그러나 모델 이론의 중요한 성취는 논리적 귀결의 명세를 제공 가능하다는 점인데, 이것은 단순히 증명 과정을 제공하는 것보다 좀 더 설득력이 있다. 이것과 관련해서 모델 이론은 비 구성적인 형태로 프로그램 사양이 프로그램을 명시하는 방법과 거의 같은 방식으로 논리적 귀결의 개념을 명시하고 있다. 증명이론은 이와 대조적으로, 비결정적이지만 논리적 귀결에 관한 구성적인 정의를 제공하고 있으며, 이것은 비결정적 프로그램과 비슷하다.

이제 결점 없고 완전하며 이상적인 관찰 문장의 입력을 해석으로 이해하여 순수하게 구문론적 개념으로 논리적 귀결의 구문적 명세를 형식화하는 것을 설명하고자 한다. 이 명세는 절 형식의 형식화한 문장일 때 더욱 분명할 수 있는데 이 형식은 비교흡수 추론과 논리프로그래밍 둘 모두의 기초가 된다. 일반적으로 일차논리의 구현으로 간주되는 절 형식은 일차논리와 비교하여 몇 가지 단점이 있지만, 절 형식 자체로 지식 표현의 방법론으로 볼 수 있다. 절 형식의 주요 장점은 표준 형식을 사용하여 간단하게 문장 사이의 중요하지 않은 구문 차이를 피할 수 있는 점이다. 예를 들면, 문장 $A \wedge B$는 $\{A, B\}$ 집합으로 표현하고, $\neg\neg A$와 같은 이중부정은 자동 제거되며, 원자 문장에만 부정이 적용될 수 있다.

스콜렘(Skolem) 상수와 함수 기호를 이용해 존재 수량사는 없애는데, 이 이유는 존재 수량사보다 존재적 의미를 더 명확하게 하기 때문이다. 예를 들면, 문장 $\forall X \exists Y mother(Y, X)$은 $\forall X mother(mom(X), X)$과 동일한 절 형식이 되는데, 여기서 사용한 mom 함수 기호는 다른 곳의 함수 기호와는 구분되어야 한다.

절 형식의 의미론은 Herbrand 해석으로 정의할 수 있다. S는 절들의 집합이라 가정하고 S에 관한 Herbrand 해석은 절들의 집합 S에 나타나는 상수 기호, 함수 기호, 술어 기호의 어휘로 구성되는 기초 원자 문장의 집합이다. 그러므로 Herbrand 해석은 결함 없고 완전하며 가능한 관찰 문장의 집합이라고 구문적으로 이해 가능하다. 언어의 모든 기초항은 '상상 가능한' 관찰할 수 있는 개체의 이름으로 여기고 모든 술어 기호는 관찰 가능한 관계 혹은 술어의 이름으로 간주한다. 또한 절의 집합인 Kb가 문장 P을 '논리적으로 함축한다'라는 것을 나타내기 위해 문장 P의 부정 $\neg P$을 집합 P^*으로 변환해서 $Kb \cup P^*$가 모순이라는 사실을 밝히면 되는데 이를 위해서 $Kb \cup P^*$의 모든 Herbrand 해석이 $Kb \cup P^*$의 절 일부분을 거짓으로 만든다는 사실을 증명하면 된다. Herbrand 해석 I가 하나의 절 C을 거짓으로 만드는 경우는 절 C의 기초 사례 $A_1 \wedge ... \wedge A_m \rightarrow B_1 \vee ... \vee B_n$ 식을 거짓으로 하는 경우이다. 여기서 모든 $A_1, ..., A_m$이 해석 I에 포함되고 어떠한 $B_1, ..., B_n$도 해석 I에 포함되지 않을 경우, 절 C는 I에 의해 거짓이 된다.

이상으로 모델 이론의 핵심적인 기능인 '참' 개념과 논

리적 결과의 구문적 명세는 동적으로 변화하는 환경과 에이전트 사이의 상호작용 환경에 잘 맞지 않고 정신적 언어와 세계 사이의 관계 그리고 자연어와 의미 사이의 관계를 잘 설명하지 못한다. 그리고 논리적 귀결 또한 모델 이론은 비 구성적 방식으로 개념을 명시하고 있기 때문에 알고리즘화하기 힘들다. 모델 이론이 가지는 이런 문제점들을 해결하는 방법으로 증명 이론적 방법이 하나의 해결 방법이 될 수 있음을 설명하였다.

IV. Knowledge Representation and Reasoning using Metalogic

앞에서 설명한 증명 가능성을 메타술어로 표현할 수 있으므로 에이전트 문장의 집합을 지식 베이스로 가정한다면 증명이론의 메타논리가 모델 이론의 인식 논리보다 향상된 표현력을 기반으로 실용적인 면과 효율면에서 우월할 수 있다. 본 절에서 메타논리 방식으로 지식을 표현하는 것과 멀티에이전트 환경에서 공통지식을 표현하여 '진흙 묻은 아이(Muddy Children)' 문제를 해결해 보고자 한다.

1. Knowledge Representation of Metalogic

언어 L의 문장들에 '관한(aboutness)' 어떤 문장 P가 존재할 때, 문장 P을 주어진 언어 L에 관한 '메타레벨(meta level)' 문장이라 정의하고, 언어 L의 문장들은 '객체(object) 레벨' 문장이라 정의한다[5]. 메타논리를 이용한 메타프로그래밍은 전문가시스템, 프롤로그(Prolog) 및 논리프로그래밍 언어의 컴파일러 구현, 지식동화, 자연어 처리, 그리고 멀티에이전트 시스템의 지식 표현에도 사용되는 보편적이고 강력한 기술이다. 이러한 활용 대부분은 객체언어와 메타언어의 결합을 필요로 한다.

메타언어와 객체언어가 서로 결합한 시스템에 관해서 두 가지 반대가 제기되었는데, 하나는 객체 레벨과 메타 레벨 사이의 이름을 구별하는데 사용하는 명명 규칙이 복잡하다는 것과 다른 하나는 내포적 개념을 나타내기 위하여 구문적인 표현을 사용하는 방법이 너무 세분되었다는 문제 제기이다. 첫 반대에 대한 해결책으로는 명명규약을 포기하고 구문 표현을 자체 이름으로 나타내는 항으로의 기능을 할 수 있도록 허용한 점이다. 자신 이름의 구문적 표현을 하면 $\forall X, Y(X \leftarrow believes(Y, X) \wedge wise(Y))$처럼 객체와 메타 레벨의 결합된 문장 표현이 가능하며 전체 한정된 다른 문장처럼 다음과 같이 모든 기초 사례의 집합을 표현하는 것으로 이해할 수 있다.

$$likes(john, mary) \leftarrow$$
$$believes(john, likes(john, mary)) \wedge wise(john)$$

두 번째 반론은 Ⅲ장 3절에서 부분적으로 해결할 수 있었다. 여기서 절 형식은 제거하지 않으면 같은 문장 간의 사소한 구문적 차이를 제거한다. 이런 점에서 절 형식과 일차논리의 표준형과의 관계는 자연어 문장의 '의미'를 나타내는 '심층구조'와 문장 자체가 나타내는 '표면구조'의 관계와 비슷하다고 볼 수 있다. 예를 들어, 다음 두 문장은 모달논리 측면에서 논리적으로 합동이다.

$$believes(john, \forall X(human(X) \rightarrow mortal(X)) \wedge human(john))$$
$$believes(john, \forall X(human(X) \rightarrow mortal(X)) \wedge human(john) \wedge mortal(john))$$

그 이유는 만약 $believes$가 모달 연산자라면 다음 두 문장은 논리적으로 합동이기 때문이다.

$$\forall X(human(X) \rightarrow mortal(X)) \wedge human(john)$$
$$\forall X(human(X) \rightarrow mortal(X)) \wedge human(john) \wedge mortal(john)$$

그러나 만약 $believes$가 메타논리 측면에서 메타술어라면 두 문장은 합동이 아니다. 다만 다음처럼 비논리적 공리 두 개의 문장이 주어지면 합동이 된다.

$$believes(T, P) \leftarrow$$
$$believes(T, P \leftarrow Q) \wedge believes(T, Q)$$
$$believes(T, P \wedge Q) \leftarrow$$
$$believes(T, P) \wedge believes(T, Q)$$

메타논리 프로그래밍에서 메타술어 $demo(T, P)$는 결론 P가 이론 T에서 시연 가능함(demonstrable)을 나타낸것이다[6]. 메타술어 $believes$를 증명술어 $demo(T, P)$로 자연스럽게 해석할 수 있다. 여기서 첫 인수 T는 에이전트의 지식 베이스(이론)을 나타내고 P는 에이전트가 믿는 문장 이름을 나타낸다.

술어 $demo$는 다음처럼 비논리적 공리로 정의할 수 있는데 $believes$와 같다.

$demo(T, P) \leftarrow$ 4-①
$demo(T, P \leftarrow Q) \wedge demo(T, Q)$
$demo(T, P \wedge Q) \leftarrow$ 4-②
$demo(T, P) \wedge demo(T, Q)$

계산 논리에서 술어 $demo$의 이론 T는 절의 집합 이름을 나타낸다. 절 형식의 유한집합은 집합 원소를 논리곱이나 리스트 형태로 표현할 수 있다. 논리곱 표현

$C_1 \wedge ... \wedge C_{n-1} \wedge C_n$ 과 표준형

$C_1 \wedge (... \wedge (C_{n-1} \wedge (C_n \wedge true))...)$은 같다.

$\wedge$은 리스트의 중위 구성자이고 $true$는 리스트 종결자를 나타낸다. 문장 집합에 포함된 하나의 문장이 증명 가능하다는 사실은 4-③과 4-④와 같이 비 논리적 공리로 나타낼 수 있다.

$demo(P \wedge Q, P)$ 4-③
$demo(P \wedge Q, R) \leftarrow demo(Q, R)$ 4-④

그러므로 다음 예제 $demo((p \leftarrow q) \wedge (q \wedge true), p)$는 4-③과 4-④ 공리를 적용하면 다음과 같고, 4-① 공리를 적용하면 증명된다.

$demo((p \leftarrow q) \wedge (q \wedge true), p \leftarrow q)$
$demo((p \leftarrow q) \wedge (q \wedge true), q)$

이론과 구문적 객체를 나타내는 데 더 유용한 방법은 상수의 사용이다. 상수를 표현한 문장의 집합으로 구성된 이론에서 어느 문장이 원소인지 아닌지의 여부는 적절한 비 논리적 공리를 통하여 나타낼 수 있다. 따라서 상수 c가 이론 $\{c_1, ..., c_n, ...\}$의 이름이면, 집합의 원소인지 아닌지는 $demo(c, c_1) \cdots demo(c, c_n) \cdots$ 나열로 정의할 수 있다.

에이전트가 고유한 지식 베이스를 가졌을 때 편리성을 위해 에이전트의 이름과 지식 베이스의 이름을 겹칠 수 있다. 따라서 메타 문장들

$demo(john, p \leftarrow q),\ demo(john, q)$

은 $john$이 $p \leftarrow q$을 믿고 q 또한 믿는다는 뜻으로 해석가능하다. 이 문장들과 4-① 공리로부터 다음과 같은 결론을 유도할 수 있다.

$demo(john, p)$

2. Common Knowledge of Cooperative Multiagent Environments

이 절에서는 에이전트들이 경쟁하지 않고 서로 협력하여 공동의 목표를 이루어 나가는 예제를 실험하려고 한다. 우호적이고 협력적인 멀티에이전트 환경에서 공통지식을 이용하여 문제를 해결하는 과정을 '진흙 묻은 아이' 문제를 통해 형식화한다. 지식의 표현은 모델 이론의 인식 논리보다 향상된 표현력을 기반으로 실용적인 면과 효율면에서 우월한 증명이론적 메타논리 방식을 사용한다. 다음은 '진흙 묻은 아이(Muddy Children)' 문제이다:

n명의 아이들이 함께 노는 모습을 상상해보라. 이 아이들의 어머니는 아이들이 더러워지면 심각한 결과를 초래할 것이라고 말했다. 물론 각 어린이는 깨끗한 상태를 유지하기를 원하지만 다른 아이가 더러워지는 것을 보고 싶어 한다. 이제 놀이를 하는 동안 일부 어린이, 즉 k가 이마에 진흙이 묻는 일이 발생한다. 각자 다른 아이의 이마에 묻은 진흙은 볼 수 있지만 자신의 이마는 볼 수 없다. 그러니 당연히 아무도 뭐라 하지 않는다. 아버지는 "너희 중 적어도 한 사람은 이마에 진흙이 묻었다"라고 말하면서 말하기 전에($k > 1$인 경우) 그들 각자가 알고 있는 사실을 표현했다. 그런 다음 아버지는 다음과 같은 질문을 반복해서 한다. "너희 중에 이마에 진흙이 묻었는지 아는 사람이 있느냐?" 모든 아이들은 지각력이 있고 지적이며 진실하고 동시에 대답한다고 가정한다.

지금까지 '진흙 묻은 아이' 문제의 해결은 대부분 귀납법으로 증명하였지만, 본 연구에서는 메타논리로 답을 형식화하고자 한다. 귀납법을 대신하여 간결한 설명을 위하여 $n = k = 2$ 경우로 한정한다.

2.1 Common Knowledge Representation and Attainment

증명술어 $demo$를 계산 논리의 효율성을 위해 에이전트의 믿음과 같은 것으로 가정한다. 에이전트의 지식을 "참인 믿음"이라는 일반적인 설명을 이용하면 다음과 같이 정의할 수 있다.

$demo(A, P) \leftrightarrow believes(A, P)$
$believes(A, P) \wedge P \leftrightarrow knows(A, P)$

에이전트 A가 P 문장을 믿는다는 것은 에이전트 A가

문장 P를 증명할 수 있을 때이고, 에이전트 A가 문장 P를 안다는 것은 에이전트 A가 문장 P를 믿고 문장 P가 참인 경우이다.

에이전트 $a_1, ..., a_n (n \geq 2)$의 모임 G의 개별 에이전트가 문장 P를 아는 경우 $every(G, P)$로 나타내고, 그룹 G의 모든 에이전트에게 문장 P가 공통지식[16, 17]일 경우에는 $common(G, P)$로 표현하면 4-⑤ 규칙이 성립한다. 그리고 그룹 G에서 공통지식 P를 4-⑥과 4-⑦로 정의할 수 있다.

$$knows(A,P) \leftarrow every(G,P) \wedge member(A,G) \quad 4\text{-⑤}$$
$$every(G,P) \leftarrow common(G,P) \quad 4\text{-⑥}$$
$$common(G, every(G,P)) \leftarrow common(G,P) \quad 4\text{-⑦}$$

두 에이전트 집합 $\{a_1, a_2\}$를 all이라는 상수로 대체하고 명제 p가 all의 공통지식이라면 다음처럼 표현할 수 있다.

$common(all, p)$

4-⑥ 식에 의해

$every(all, p)$

유도할 수 있고, 에이전트 a_1과 a_2가 상수 all의 멤버이기에 4-⑤로부터 다음처럼 유도 가능하다.

$$knows(a_1, p), \quad knows(a_2, p)$$

먼저 4-⑦ 공리를 적용하면, "각 에이전트가 명제 p를 안다는 사실은 모든 에이전트 사이에서 공통지식이다"를 의미하는

$common(all, every(all, p))$

을 유도할 수 있고, 다시 4-⑥ 공리에 의해

$every(all, every(all, p))$

도 성립한다. 그리고 4-⑤ 공리 및 에이전트 a_1과 a_2가 all의 구성원이라는 사실에서 다음처럼 유도 가능하다.

$$knows(a_1, knows(a_1, p)), knows(a_1, knows(a_2, p)),$$
$$knows(a_2, knows(a_1, p)), knows(a_2, knows(a_2, p))$$

메타논리 프로그래밍 측면에서 보면 4-⑤ ~ 4-⑦ 공리들은 순환 프로그램이기에 4-⑦ 공리를 반복 적용할 경우 공통지식의 무한계층 개념도 손쉽게 표현할 수 있다.

2.2 Muddy Children Problem

진흙 묻은 아이 문제의 해답을 형식화하기 위한 첫 단계는 공통지식을 생성하는 것이다. $n = k = 2$로 가정하였기에, 두 아이 $child1$과 $child2$의 집합을 $all = \{child1, child2\}$이라 한다. 모든 아이의 이마에 진흙이 묻어 있으므로 $muddy1$과 $muddy2$로 표현하고 참이 된다. 따라서 $muddy1 \vee muddy2$("너희 중 적어도 한 사람은 이마에 진흙이 묻었다")라는 말도 참이다. 즉 다음 4-⑧ 식과 같다.

$$muddy1 \vee muddy2 \leftarrow muddy1 \quad 4\text{-⑧}$$
$$muddy1 \vee muddy2 \leftarrow muddy2$$

동일한 장소에 있는 모든 에이전트에게 신뢰할 수 있는 에이전트 X가 동시에 P라는 사실을 말하고 P라는 사실이 참이면 다음 4-⑨와 같이 가정할 수 있다.

$$common(all, P) \leftarrow said(X, all, P) \wedge trustworthy(X) \wedge P \quad 4\text{-⑨}$$

그러므로 동일한 장소에 있는 두 아이에게 $muddy1 \vee muddy2$라고 말한 아버지의 말

$said(father, all, (muddy1 \vee \ muddy2))$

은 지식 $trustworthy(father)$와 4-⑧에 의해 다음 4-⑩이 된다.

$$common(all, (muddy1 \vee muddy2)) \quad 4\text{-⑩}$$

그리고 공통지식 4-⑤, 4-⑥, 4-⑦ 공리들에 의해 다음처럼 유도된다.

$knows(child1, (muddy1 \vee \ muddy2))$,
$knows(child1, knows(child2, (muddy1 \vee muddy2)))$
, ...
$knows(child2, (muddy1 \vee \ muddy2))$,
$knows(child2, knows(child1, (muddy1 \vee \ muddy2)))$
, ...

한편 에이전트 A가 문장 P에 대한 질의를 받고 에이전트 A가 문장 P를 안다면 4-⑪처럼 나타낼 수 있다.

$$tells(A,P) \leftarrow asked(A,P) \wedge knows(A,P) \quad 4\text{-}⑪$$

즉, 협력적인 에이전트 A는 P에 관한 질의에 응답한다. 이 공리에 대한 변형으로 에이전트 A가 P라는 사실을 모른다면 다음 4-⑫처럼 침묵할 수 있다.

$$silent(A,P) \leftarrow asked(A,P) \wedge \neg knows(A,P) \quad 4\text{-}⑫$$

더 나아가 에이전트가 질문에 응답하지 않고 침묵하고 있다면 다음 4-⑬처럼 명시적으로 모른다고 가정하여 표현할 수 있다.

$$ignorant(A,P) \leftarrow asked(A,P) \wedge silant(A,P) \quad 4\text{-}⑬$$

처음 질문 "너희 중에 이마에 진흙이 묻었는지 아는 사람이 있느냐?"에 $child1$이 침묵한 이유는 4-⑫에 의해
$asked(child1, muddy1)$이지만
$\neg knows(child1, muddy1)$이기 때문이다. 그러므로 $ignorant(child1, muddy1)$이 된다. $child2$ 역시 동일한 이유로 침묵하게 되고 결과적으로 다음 식처럼 $ignorant(child2, muddy2)$가 된다. 따라서 아이들은 각자 상대방의 얼굴을 관찰하고, 침묵을 관찰하므로 다음 4-⑭와 같은 지식을 가지게 된다.

$$knows(child1, muddy2), \quad 4\text{-}⑭$$
$$ignorant(child1, muddy1)$$
$$knows(child2, muddy1),$$
$$ignorant(child2, muddy2)$$

에이전트가 사실 P에 대해 완전한 지식을 가진다는 의미는 $complete(A,P)$로 표현하고 사실 P을 알든지 또는 $\neg P$을 알든지로 정의한다:

$$complete(A,P) \leftrightarrow knows(A,P) \vee knows(A,\neg P) \quad 4\text{-}⑮$$

그리고 이 공리에 대한 부정으로 불완전한 지식을 가질 때 4-⑯처럼 정의할 수 있다.

$$incomplete(A,P) \leftrightarrow ignorant(A,P) \wedge ignorant(A,\neg P) \quad 4\text{-}⑯$$

지식의 완전성에 대한 정의 4-⑮ 공리를 논리프로그래밍 형식으로 나타내면 다음처럼 표현할 수 있다.

$$knows(A,\neg P) \leftarrow complete(A,P) \wedge ignorant(A,P) \quad 4\text{-}⑰$$

$$knows(A,P) \leftarrow complete(A,P) \wedge ignorant(A,\neg P) \quad 4\text{-}⑱$$

관찰에 의해 4-⑭와 동일한 도메인 지식을 가지는 것과 함께, 첫 질문에 따른 침묵의 관찰로 인해 다음 4-⑲와 같이 추가로 가정할 수 있다.

$$common(all, complete(child1, muddy2)) \quad 4\text{-}⑲$$
$$common(all, complete(child2, muddy1))$$
$$common(all, ignorant(child1, muddy1))$$
$$common(all, ignorant(child2, muddy2))$$

즉, $child1$이 $muddy2$에 관해 완전한 지식을 가진다는 사실과 $child2$가 $muddy1$에 관해 완전한 지식을 가진다는 사실은 공통지식이다. 그리고 $child1$이 $muddy1$에 대해 모르는 것과 $child2$가 $muddy2$에 대해 모른다는 사실 또한 공통지식이다. 왜냐하면 문제에서 "모든 아이들은 지각력이 있고 지적이고 진실하며 동시에 대답한다"라고 하였기에 $child1$과 $child2$는 모두 서로의 얼굴을 관찰할 수 있으며 동시에 대답하지 않은 사실 역시 관찰할 수 있기 때문이다.

이제 진흙 묻은 아이 문제의 해답을 형식화하는 데 필요한 마지막 공리는 4-⑳과 같다.

$$knows(A,Q) \leftarrow knows(A,(P \vee Q)) \wedge ignorant(A,P) \wedge complete(A,Q) \quad 4\text{-}⑳$$

에이전트 A가 $P \vee Q$에 대해 완전한 지식을 가지고 있고, 사실 P는 모르지만 Q에 관해 완전한 지식을 가지고 있다면 에이전트 A는 사실 Q를 안다를 나타낸 공리이다. 이 문제에서 첫 질문에 관해 $child2$의 추론은 4-⑩ 공리에 의해
$knows(child1, (muddy1 \vee muddy2))$
이고, 4-⑭ 공리에 의해
$ignorant(child1, muddy1)$
이며, 4-⑲ 공리에 의해서
$complete(child1, muddy2)$

이므로 4-⑳ 공리로부터 다음처럼 추론할 수 있다.

$knows(child1, muddy2)$

그리고 $child1$ 아이가 또한 동일한 능력의 추론을 할 수 있기에 다음을 유도할 수 있다.

$knows(child2, muddy1)$

다시 말하면, 첫 질문에 대한 침묵으로 $child1$, $child2$ 각각은 다음처럼 알게 된다.

$knows(child1, knows(child2, muddy1))$

$knows(child2, knows(child1, muddy2))$

동일한 질문을 두 번째 되풀이하게 될 경우에 $child1$과 $child2$는 동시에 "예"라고 답을 할 수 있다.

지금까지 메타논리를 이용하여 지식의 표현 및 추론을 형식화하였다. 본 연구에서는 증명술어 $demo$를 지식 혹은 믿음과 동일한 의미로 간주하였다. 인식 논리의 $knows$, $believes$와 메타논리의 $demo$ 술어는 증명 이론적 관점에서 볼 때 동일한 의미라고 가정하였다. 이런 가정이 철학적 측면에서는 논란의 소지가 있을 수 있으나, 실용성을 추구하는 계산 논리의 측면에서는 응용성과 효율성 둘 모두에서 유용하기 때문이다. 에이전트가 우호적이고 협력적일 때 공동 목표의 달성을 위해 에이전트들은 공통지식을 생성하고 그 지식을 기반으로 '진흙 묻은 아이(muddy children)' 문제를 해결할 수 있다는 것을 메타논리를 통해 확인하였다.

3. Comparison Analysis

앞 절에서 진흙 묻은 아이 문제를 메타논리 방식으로 해결하였지만, 이 문제는 고전적으로 Ⅱ장 3절에서 소개한 인식 논리의 가상 세계 의미론을 통해 표현하고 해결하여 왔다. 이 절에서는 모달 논리의 의미론을 통한 해결 방법을 알아보고 메타논리 방법과 비교 분석하고자 한다.

인식 논리의 의미론은 주어진 식이 참인지 거짓인지를 결정하는 형식적인 방법을 사용하는데 크립키 구조라고도 부르는 가능 세계 의미론을 사용한다. 진흙 묻은 아이 문제에서는 일종의 model selection(문제 조건에 대해 참인 경우를 선택)방식으로 해결한다. 여기서는 세 명의 아이(에이전트)들이 있다고 가정하자. 문제 해결을 위한 식들은 다음과 같이 표현한다.

$$M = (S, \pi, k_1, k_2, k_3) \quad 4\text{-}㉑$$

$$S : (x_1, x_2, x_3) \quad 4\text{-}㉒$$

$$\Phi = \{p_1, p_2, p_3, p\} \quad 4\text{-}㉓$$

$$(M, (x_1, x_2, x_3)) \models p_i \ \ iff \ x_i = 1 \quad 4\text{-}㉔$$

$$(M, (x_1, x_2, x_3)) \models p \ \ iff \ x_i \text{ 중 하나가 } 1 \quad 4\text{-}㉕$$

4-㉑은 크립키 구조를 표현한 것이고 k_1, k_2, k_3은 세 명의 아이들을 나타낸다. 4-㉒는 가능한 세계를 나타낸 식이고 $x_i = 1$은 i번째 아이의 이마에 진흙이 묻어 있음을 나타내고 $(1, 0, 1)$은 첫 번째와 세 번째 아이가 이마에 진흙이 묻어 있는 세계를 나타낸다. 따라서 S에는 8가지, $(0, 0, 0) \sim (1, 1, 1)$의 가능한 세계가 있다. 4-㉓은 단순 명제들의 집합을 표현한 것으로 p_i는 i번째 아이의 이마에 진흙이 묻어 있음을 나타내고 p는 적어도 한 명의 아이의 이마에 진흙이 묻어 있다는 명제를 나타낸다. 4-㉔는 p_i가 참일 필요충분조건은 $x_i = 1$일 때를 나타내고 4-㉕는 p가 참일 필요충분조건은 x_i 중 하나가 1일 때를 나타낸다.

진흙 묻은 아이 문제를 해결하기 위해 인식 논리의 가상 세계 의미론에서는 Fig. 1처럼 크립키 구조 그래프를 사용한다. 만약 실세계가 $(1, 0, 1)$이라 가정하면 다음과 같은 식을 유도할 수 있다.

$$(M, (1, 0, 1)) \models K_1 \neg p_2 \quad 4\text{-}㉖$$

$$(M, (1, 0, 1)) \models K_1 p_3 \quad 4\text{-}㉗$$

$$(M, (1, 0, 1)) \models \neg K_1 p_1 \quad 4\text{-}㉘$$

$$C(\neg p_2 \rightarrow K_1 \neg p_2) \quad 4\text{-}㉙$$

$$(M, (1, 0, 1)) \models Ep \quad 4\text{-}㉚$$

4-㉖은 에이전트1이 상상하는 두 세계 $(1, 0, 1)$과 $(0, 0, 1)$에서 모두 두 번째 원소 x_2의 값이 0이므로 즉, 에이전트1은 에이전트2의 이마에 흙이 묻어 있지 않다는 사실을 알고 있다. 4-㉗은 에이전트1이 상상하는 두 세계 $(1, 0, 1)$과 $(0, 0, 1)$에서 모두 세 번째 원소 x_3의 값이 1이므로 에이전트1은 에이전트3의 이마에 흙이 묻었다는 것을 안다. 4-㉘은 에이전트1은 자기 자신이 이마에 흙이 묻었는지 모른다는 것을 알 수 있다. 4-㉙는 만약 에이전트2의 이마에 진흙이 묻지 않았다면 에이전트1도 그 사실을 알고 있다는 것은 모든 에이전트들이 다 아는 공통지식(C 모달 연산자는 공통지식을 나타냄)이라는 의미이다. 4-㉚은 모든 에이전트들(E 모달 연산자는 모든 에이전트을 나타냄)이 적어도 하나의 에이전트의 이마에 진흙이 묻었다는 사실을 안다라는 의미이다.

인식 논리의 의미론에서는 몇 가지 문제점을 나타낸다.

먼저 4-㉙에서 크립키 그래프에서 $\neg p_2 \rightarrow K_1 \neg p_2$에 해당하는 노드가 존재하지 않는다는 점이다. 따라서 그 노드에서 도달 가능한 모든 노드에서 역시 $\neg p_2 \rightarrow K_1 \neg p_2$이 성립한다는 공통지식의 성질을 추론할 수 없다는 의미론의 문제점이 발생한다. 또한 4-㉚은 그립키 그래프에서 이 성질의 추론이 비효율적이라는 점이다. $(1,0,1)$은 에이전트1은 에이전트3이 그렇다는 것을 안다; 에이전트2는 1과 3이 그렇다는 것을 안다; 에이전트3은 1이 그렇다는 것을 안다; 따라서 각 에이전트는 적어도 한 명이 그렇다는 것을 안다라는 의미인데 비효율적이고 부자연스럽다. 결론적으로 인식논리의 의미론은 무엇이 참인지를 규정하는 이론을 강조하다 보니 기계적인 추론 기능이 모호하거나 아예 안되거나, 된다고 하더라도 비효율적인 면이 나타난다. 첫 번째 아이(에이전트)가 모르겠다고 했을 때 두 번째 아이의 지식의 변화 그리고 두 번째 아이가 모르겠다고 했을 때 세 번째 아이의 결론 유도 과정을 구체적으로 처리할 수 없다는 점이 가장 큰 문제점이라 하겠다. 이러한 문제의 해결은 Ⅳ장 2절에서 메타논리적 방법의 표현과 추론으로 가능함을 증명하였다.

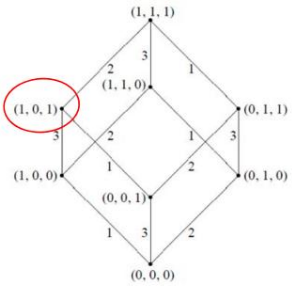

Fig. 1. The Kripke structure graph

Ⅴ. Conclusions

최근 딥러닝과 신경망, 빅데이터 기술 등, 비기호 기술 기반의 괄목할 만한 성과로 인해 인공지능에 대한 관심과 연구가 다시 주목받고 있다. 더불어 설명할 수 있는 인공지능(eXplainable Artificial Intelligence) 연구의 필요성이 나타나면서 비기호 기술 방식은 이 장벽을 넘어야 하는 목표가 생겼다. 반면, 전통적인 논리 기반 기술인 기호 기반 기술 방식은 이해할 수 있고 설명할 수 있는 지능형 시스템을 구축하기에 가장 유망한 기술 중 하나로 다시 주목받기 시작했으며 향후 지속적인 발전이 기대되는 분야로 평가되고 있다.

전통적으로 지식의 표현은 기호 기술 기반의 연구 분야이며 그 가운데서도 논리 기반 기술을 중심으로 발전되어 왔다. 모달 연산자를 사용한 인식논리가 주류를 이루고 발전되어 왔다. 모델 이론을 기반으로 하나의 에이전트에서 지식을 표현하는 인식논리와 멀티에이전트 환경에서 인식논리의 가상 세계 의미론으로 표현하는 방법은 논리적 귀결의 명세를 제공하는 점과 이 명세가 단순히 증명 과정을 제공하는 것보다 더 설득력이 있다는 장점을 가지지만 정신적 언어와 세계 간의 관계 그리고 자연어와 의미 사이의 관계를 설명하지 못하거나 큰 도움을 주지 않는다는 점을 분석하였다. 이는 동적으로 변화하는 환경과 에이전트 사이의 상호작용 환경에 잘 맞지 않고 논리적 귀결 또한 비구성적 방식으로 개념을 명시하고 있기 때문에 알고리즘화하기 힘든 점을 분석하였다.

본 연구에서는 모델 이론의 방식보다 증명이론 방식이 해결책이 될 수 있음을 설명하였다. 증명 이론적 방법을 사용한 지식의 표현 및 추론을 실험하기 위해 증명 가능성을 메타술어로 쉽게 표현할 수 있기에 에이전트 문장의 집합을 지식 베이스로 가정하고 증명이론의 메타논리가 모델 이론의 인식 논리보다 향상된 표현력을 기반으로 실용적인 면과 효율 측면에서 우월할 수 있다는 점을 확인하였다. 실험으로는 메타논리 방식으로 지식을 표현하는 것과 에이전트들이 공동의 목표를 위해 서로 협력하는 멀티에이전트 환경에서 공통지식을 표현하여 '진흙 묻은 아이(Muddy Children)' 문제를 해결해 보았다.

향후 연구 과제로는 멀티에이전트 환경에서 메타논리를 이용한 지식표현 및 공통지식을 형성하여 자율 주행차나 로봇들이 협업하여 하나의 목적을 달성하도록 돕는 의사결정 시스템의 한 부분을 적용해보는 것이다[18]. 현실에서 실제 적용되기 위해서는 관련 연구들이 더 많이 진행되어야 할 것으로 판단한다.

REFERENCES

[1] K. Schwab, "*The fourth industrial revolution*," Crown Business, pp. 6-27, 2016.

[2] R. Calegari, G. Ciatto, V. Mascardi, and A. Omicini, "Logic-based technologies for multi-agent systems: a systematic literature review," Autonomous Agents and Multi-Agent Systems, Vol. 35, No. 1, pp. 1-67, April 2021. DOI:10.1007/s10458-020-09478-3

[3] A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, et al, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward

responsible AI," Information Fusion, Vol. 58, pp. 82-115, June 2020. DOI:10.1016/j.inffus.2019.12.012

[4] K. Kim, "A Study on the Use of Common Knowledge among Autonomous Driving Agents - Based on the CBK Model," The Korean Society of Culture and Convergence, Vol. 43, No. 12, pp. 971-988, December 2021. DOI: https://doi.org/10.33645/cnc.2021.12.43.12.971

[5] D. Perlis, "Meta in Logic," Meta-Level Architectures and Reflection, pp. 37-49, 1988.

[6] R. A. Kowalski, and J. S. Kim, "A metalogic programming approach to multi-agent knowledge and belief," Artificial Intelligence and Mathematical Theory of Computation, pp. 231-246, 1991. DOI: 10.1016/B978-0-12-450010-5.50019-0

[7] P. Mancarella, A. Raffaeta, and F. Turini, "Knowledge representation with multiple logical theories and time," Journal of Experimental & Theoretical Artificial Intelligence, Vol. 11, No. 1, pp. 47-76, January 1999. DOI: 10.1080/095281399146616

[8] S. A. Kripke, "A completeness theorem in modal logic," The Journal of Symbolic Logic, Vol. 24, No. 1, pp. 1-14, March 1959, Published online by Cambridge University Press, 2014. DOI: https://doi.o rg/10.2307/2964568

[9] W. H. Holliday, "Epistemic Logic and Epistemology," In: S. O. Hansson, and V. F. Hendricks, editors, *Introduction to Formal Philosophy*, Springer, October 2018. DOI: https://doi.org/10.1007/978-3- 319-77434-3_17

[10] N. Gierasimczuk, and J. Szymanik, "A Note on a Generalization of the Muddy Children Puzzle," Proceeding of ACM International Conference, pp. 257-264, July 2011. DOI: 10.1145/2000378.2000409

[11] J. J. Kline, "Evaluations of epistemic components for resolving the muddy children puzzle," Economic Theory, Vol. 53, No. 1, pp. 61-83, May 2013. DOI: 10.1007/s00199-012-0735-x

[12] X. Huang, and R. Meyden, "Symbolic Synthesis of Knowledge-based Program Implementations with Synchronous Semantics," Proceedings of the 14th Conference on Theoretical Aspects of Rationality and Knowledge, pp. 121-130, January 2013.

[13] M. R. Genesereth, N. J. Nilsson, "*Logical Foundations of Artificial Intelligence*," Morgan Kaufmann Publishers, pp. 45-62, 2012.

[14] R. A. Kowalski, "*Computational Logic and Human Thinking: How to be Artificially Intelligent*," Cambridge University Press, pp. 60-76, 2011.

[15] H. Decker, "Abduction for knowledge assimilation in deductive databases," Proceedings 17th International Conference of the Chilean Computer Science Society Computer Science Society, pp.48-57, November 1997. DOI: 10.1109/SCCC.1997.636868

[16] K. Kim, "Achieving and reasoning about common beliefs based on social networking services: on the group chatting model of kakaotalk," Journal of The Korean Institute of Intelligent Systems, Vol. 27, No. 1, pp. 7-14, February 2017. DOI: 10.5391/jkiis.2017.27.1.007

[17] K. Kim, "Common knowledge attainment and diffusion in group chatting model of kakaotalk using logic programming," Journal of The Korean Institute of Intelligent Systems, Vol. 28, No. 3, pp. 294-303, June 2018. DOI: 10.5391/JKIIS.2018.28.3.294

[18] K. Kim, "A Study on the Use of Common Knowledge among Autonomous Driving Agents - Based on the CBK Model," The Korean Society of Culture and Convergence, Vol. 43, No. 12, December 2021. DOI: https://doi.org/10.33645/cnc. 2021.12.43.12.971

Authors

Koono Kim received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Hannam University and Keimyung university, Korea, in 1993, 1995 and 2022, respectively.

Prof. Kim joined the faculty of the Global Leadership School at Handong University, Pohang, Korea, in 2000. He is currently a associate professor in the same university. He is interested in common knowledge, Metalogic programming, Multi-agent system.