지능 에이전트 구현의 인지적 접근

Cognitive Approach for Building Intelligent Agent

태 강 수*
Kang-Soo Tae

요 약

에이전트가 지각이나 행위의 표상을 이해할 수 없는 이유는 의미론적 자질을 문자열로 변환하는 구문론적 표상방식에 의해서 일어난다. 자율적으로 학습하는 인지 에이전트를 구현하기위해 코헨은 에이전트가 sensor와 effector를 사용하여 주위환경과 물리적으로 직접적인 상호작용을 통하여 물리적 스키마의 의미 표상을 학습하는 의미론적 방법을 제안하였다. 본 논문에서는 부정(negation)은 그러한 물리적 스키마를 인식하게 하는 메타 스키마임을 제안한다. 최근에 Graphplan은 계획 시스템의 성능을 향상하기 위하여 inconsistency를 이용하는 제어규칙을 사용하지만, 구문론적으로 접근하여서 부정의 의미 개념을 이해하지 못하고 중복표현의 문제를 야기한다. IPP는 부정 함수인 *not*을 도입하여 중복문제를 해결하지만 여전히 구문론적으로 접근하며 또한 시간과 공간에서 비효율적이다. 본 논문에서는 의미론적인 접근법을 도입하여 부정을 위해서 반대 개념이라는 긍정 아톰(atom)을 사용하는 것이 지능 에이전트를 구현의 효율적 기법이라고 제안하고, 이 가설을 지지하는 실험적 결과를 제시한다.

Abstract

The reason that an intelligent agent cannot understand the representation of its own perception or activity is caused by the traditional syntactic approach that translates a semantic feature into a simulated string. To implement an autonomously learning intelligent agent, Cohen introduces a experimentally semantic approach that the system learns a contentful representation of physical schema from physically interacting with environment using its own sensors and effectors. We propose that negation is a meta-level schema that enables an agent to recognize its own physical schema. To improve the planner's efficiency, Graphplan introduces the control rule that manipulates the inconsistency between planning operators, but it cannot cognitively understand negation and suffers from redundancy problem. By introducing a negative function not, IPP solves the problem, but its approach is still syntactic and is inefficient in terms of time and space. In this paper, we propose that, to represent a negative fact, a positive atom, which is called opposite concept, is a very efficient technique for implementing an cognitive agent, and demonstrate some empirical results supporting the hypothesis.

Keyword : Cognitive Agent, Machine Learning, Planning, Schema

1. 서 론

에이전트가 자율적으로 지식의 표상을 학습하고 이해한다는 인지적 목표는 현실적으로 달성하기 극히 어려운 문제이다. 지식 표상 언어 (knowledge representation language)는 에이전트가 처리할 수 있는 형식으로 지식을 표현하기 위하여 사용된다. 하지만 현재 대다수의 지능적 에이전트들은 지식공학자가 의도하는 대로 나타나는 표상을 활용한다[1]. 따라서 표상의 의미는 에이전트에 대해 내재적이라기 보다 외재적인 존재이다. 또한 기존의 지식표상 방식은 주위에 존재하는 개체들의 의미론적 자질을 단순히 문자열로 변환하는 구문론적 방식을 이용한다. 이러한 접근방식은 에이전트의 지식표상을 의미(meaning)와 연결할 수 없다. 이러한 난제를 해결하기 위하여, 코헨은 에이전트가 자신의 센서와 모터와 언어를 사용하여 세계와 상호 작용함으로써 표상을 직접적으로 학습할 수 있는 의미론적 방식을 제안하였다[1]. 따라서 에이전트는 인간 전문가에 의해서 자의적으

* 정 회 원 : 전주대학교 정보기술컴퓨터공학부 조교수
kstae@jeonju.ac.kr(제 1저자)

로 제공되는 문자열이 아닌 자신의 센서를 통해서 직접 들어오는 감각적 자료들의 모음으로부터 자신의 세계를 감지한다. 이러한 방식은 원천적으로 개념소자(conceptual primitive)로서의 물리적 스키마(physical schema)를 전제한다. 물리적 스키마는 추상적 개념들의 기초 단위를 형성하며 인지적 에이전트의 개발의 중추를 형성한다.

본 논문에서는 에이전트가 이러한 물리적 스키마를 인지할 수 있게 하는 메타 수준의 개념소자가 있어야 한다고 제안한다. 하나의 대상을 인식하기 위해서 기본적으로 부정(negation)이라는 개념소자는 에이전트가 관심 대상이 아닌 객체들로 구성된 집합을 인지할 수 있도록 해준다. 우리는 집합과 여집합의 개념을 이용하여 부정의 개념의 정의하며 계획(planning)의 분야에서 부정의 몇 가지 측면을 고찰할 것이다.

Graphplan은 두 계획 연산자(operator)들 사이의 모순관계를 추론함으로써 탐색공간의 크기를 절반으로 줄여주는 mutex라는 부정적 함수를 도입하였다[3]. 그러나 부정적 용어를 단순히 문자열로 다루는 구문적인 접근방식은 의미론적 부정의 개념을 이해하지 못한다. 예를 들어 계획연산자가 P가 거짓임을 요청할 때 Graphplan은 새로운 명제 즉 not-P를 정의한다. 이러한 표기법은 하나의 상태 변화가 effect에서 add(not-in(x))와 del(in(x))와 같은 두 가지 프로세스들에 의해 처리되는 중복성 문제를 야기할 수 있다. IPP나 Blackbox와 같은 계획기는 부정어를 처리하기 위하여 부정함수 not을 도입한다. 이러한 부정 함수는 계획연산자의 efftct 상태를 add(*not* in(x))와 같은 단일한 프로세스에 의해서 처리할 수 있도록 한다. 하지만 술어를 부정하기 위하여 not을 사용하는 것은 술어의 앞부분에 부정적 기호를 덧붙이는 구문론적 수준의 기법에 불과하다. 따라서 에이전트가 on과 *not* on과 같은 구문론적 수준의 반대 용어들을 쉽게 구분할 수 있지만, on과 off와 같은 의미론적 반대 용어들은 구별할 수 없다. 인간은 이러한 긍정 용어(positive term)들 사이의 반대 관계를 즉시 추론할 수 있으며 또한 부정적 표현은 긍정 술어에 의하여 표현하는 경향이 있다. 예를 들어서, 인간은 dry를 부정하기 위해서 not dry 보다는 wet 라는 보다 짧은 용어를 선호한다. 본 논문에서는 not-P 표기법이 (not P) 표기법 보다 효율적이며 강력한 기법이라고 제안한 다음, 처리 시간 및 공간의 면에서 우리의 가설을 지지하는 경험적 자료를 제시한다.

2. 구문론과 의미론적 표상

지식 표상 언어의 구문론은 제대로 구성된 문장을 작성하는 법을 명시하는 반면, 언어의 의미론은 세계의 사실들을 다룬다. 문장은 사실을 지시한다. 문장과 사실 사이의 대응관계는 저자의 해석에 의해 정해진다. 기존에는 해석의 저자는 인간 전문가에 국한되었었지만, 앞으로 에이전트 스스로 그러한 해석의 저자가 되는 것이 바람직할 것이며, 코헨의 연구의 주요한 목표중의 하나가 그러한 자율성을 추구하는 것이었다. 문장의 의미는 문장의 세계에 대한 기술이다. 사실은 세계의 부분이지만, 저자의 마음에서는 사실의 표상이 지식베이스에 코드화 되어 있다. 따라서 에이전트의 내부에서 일어나는 추론은 사실 자체가 아니라 사실의 표상에 대해서 작동한다[7,10]. 이러한 논점은 물자체에 대한 칸트의 인식론에 근거한다고 할 수 있다.

논리학에서 어떤 하나의 사실이 또 하나의 다른 사실로부터 반드시 발생한다는 의미론적 인과론은 어떤 하나의 문장이 또 하나의 다른 문장으로부터 필연적으로 연역된다는 논리적 규칙에 반영되어 있다. 논리적 추론은 문장들 사이에 존재하는 이러한 필연적 인과 관계를 추구하는 과정이다. 문장 P가 하나의 사실 Fp를 지시하고 문장 q가 또 하나의 사실 Fq를 지시한다고 하자. 만일 Fq가 필연적으로 Fp로부터 발생한다고 하면 논리적으로 q는 p로부터 연역된다. 이러한 예는 구문론적 표상이 세계와 어떻게 논리적으로 연결되는

지 보여준다. 그러나 이러한 유형의 표상에 존재하는 문제점은 에이전트가 여전히 세계를 이해하지 못하고 있다는 점이다. p와 Fp 사이의 대응관계는 에이전트에 의해 내재적으로 습득된 것이 아니라 인간 저자에 의해서 프로그래밍된 것이며 인공적 에이전트에게 외재적으로 주어진 것이다. 인간이 제시한 이러한 규칙과 기호들은 문자열로 되어있으며 에이전트의 지식베이스에 저장된다. 에이전트는 단순히 저자의 마음 안에서 세계와 연결되어 있다. 이러한 오류는 많은 문제를 야기할 수 있다. 대다수의 인공지능 시스템은 지식공학자가 의미하기로 기도한 표상을 조작한다. 따라서 표상의 의미는 에이전트 시스템에 대해서 외생적이다.

에이전트 컴퓨터 시스템은 지식베이스에 존재하는 것 이외에는 세계에 대해서 아는 바가 전혀 없음으로 인간이 하듯이 세계에 대해서 추론할 수가 없다. 세계에 대한 지식이 없는 상태에서 임의의 p라는 문장을 증명하기 위해서 에이전트가 선택할 수 있는 유일한 방법은 자신의 지식베이스에서 p가 필연적으로 연역된다는 것을 증명하는 것이다. 이러한 결론은 에이전트의 세계의 해석에 대한 지식과는 관계없이 모든 세계에서 항상 옳다. 그러한 결론이 시스템에게는 무의미한 문자열에 불과하지만, 인간은 그러한 해석을 알고 있으므로 유의미하다. 추론에서 항진 명제를 사용하는 장점은 탐색공간의 지수적 증가라는 단점을 수반한다. 이러한 문제를 해결하기 위해서 planner 시스템은 제어규칙을 사용한다. 예를 들어 Prodigy는 탐색공간을 감소하기 위하여 계획연산자들의 선택규칙을 사용하며 Graphplan은 연산자들 사이의 불가능한 구조들을 찾아내는 mutex를 사용한다[3,9]. 주의깊게 생각해보면 이러한 구문론적 규칙은 두 가지 반대되는 사실들이 동시에 세계에 존재할 수 없다는 의미론적 규칙을 반영한 것임을 알 수 있다.

지식의 구문론적 표상 방식의 문제점들 중의 하나는 중복표현의 문제이다. Graphplan은 두 연산자들이나 술어들 사이의 모순관계를 추론할 수 있지만 부정의 의미를 이해하지 못하고 부정어를 단지 문자열로 다룬다. 명제 p의 부정이 필요하면 Graphplan은 (not P)에 상응하는 not-P라는 새로운 술어를 정의하지만 둘 사이의 상응관계를 이해하지 못한다. 이러한 단순한 부정적 표기법은, 에이전트가 in과 not-in은 반대임을 인지할 수 있는 특별한 지식을 가지고 있지 않는 한, 하나의 상태변화가 add(not-in(x))와 del(in(x))와 같은 두 가지 프로세스에 의해서 기술된다는 중복표현의 문제를 일으킨다. Graphplan을 확장하여 이러한 부정적 사실들을 일괄적으로 처리하기 위해, IPP는 not이라는 부정함수를 도입한다. not-p는 더 이상 사용하지 않으므로, 연산자의 부정적 결과는 {add(not-p)와 del(p)} 대신에 {add(not p)}로서 통일적으로 처리될 수 있다.

3. 부정의 스키마적 표상

개념의 기원이 무엇인가 하는 문제는 여전히 뜨거운 논쟁의 대상이다. 이성주의에서는 인간의 마음은 출생 시 이미 일정한 구조를 가지고 태어난다고 주장하지만, 경험주의에서는 개념은 활동의 표상들을 추상화함으로써 학습될 수 있다고 주장한다. 어쨌든 심리학, 철학, 언어학, 로봇공학 등에서는 인간의 이성이 개념적 소자들에 의존한다는 증거들이 나타나고 있다[1]. Mandler는 어린이가 감각-모터의 상호작용으로부터 공간구조의 이미지-스키마적 재구성을 통하여 개념 구조를 생성할 수 있다고 주장한다. 이러한 이미지-스키마는 감각 흐름을 부분적으로 표상으로 사상(map)하는 일종의 패턴 감지자 또는 필터이다. 이러한 개념 소자는 세계와의 물리적 상호작용에 기반을 둔 스키마적 구조이다[5].

이상적인 측면에서는 에이전트가 데이터베이스에 저장된 기호가 무슨 뜻인지 스스로 파악해야 한다. Mandler의 영향을 받아서 코헨은 인공적인 에이전트가 어떤 종류의 의미를 학습할 수 있다고 가정

한다. 이를 위해서 그는 개념적 지식의 기원을 알고자 한다. 스키마는 에이전트가 객체의 패턴이나 클래스를 인지하게 해주는 개념적 소자이다. 물리적 스키마는 움직임이나 힘을 사용하는 등의 에이전트와 객체들 사이의 기본적인 관계나 상호작용의 궁극적인 물리적 프로세스에 기반을 둔 추상적이고 도메인 독립적인 기술이다. 코헨은 어떻게 에이전트가 세계와의 센서와 모터의 상호작용에 의해 자신의 의미론적 표상을 개발할 수 있는지에 관심이 있다. 에이전트가 자신의 지각내용을 이해할 수 없다는 문제에 대해서 코헨은 에이전트의 세계에 대한 연결은 센서를 통해서 이루어져야한다고 생각한다. 그러면 에이전트의 지각적 표상은 센서에 기반을 두게 된다. 에이전트의 주변 환경에 대한 직접적 연결은 표상에 대해서 의미를 부여할 수 있다. 에이전트가 학습한 이러한 내재적 의미는 인간 저자가 의도하는 대로 작동하는 시스템을 구축하는 전통적인 인공지능의 외재적 의미에 반대한다. 세계 내에서 작동(embedded)하는 로봇 에이전트인 코헨의 Baby는 이미지-스키마에 유사한 규칙을 사용하여 객체, 행동, 범주 등의 표상을 학습한다. 아이가 태어날 때 아이는 행동하고 지각하는 등의 비교적 단순한 일을 하는데 표상, 개념 및 언어는 이러한 행위와 지각에서 비롯된다. 코헨의 방법에서는 에이전트는 물리적 스키마를 선험적 인지구조로 가정하며 이러한 구조를 통계적 관찰을 통해서 학습한다.

본 논문에서는 특히 부정적 구문적 연산자의 의미론적 양상을 이해하는데 관심이 있다. 예를 들어서, ¬p는 부정연결자인 ¬와 p를 결합하여서 생성된다. 부정의 구문론적 양상은 술어나 문장에 부정 연결자를 결합하는 것이다. 코헨은 센서와 모터의 활동을 통해 패턴을 인지하는 물리적 스키마만을 연구하였다. 본 논문에서는 물리적 스키마와 관련지어 부정의 의미론적 양상을 연구하는 것이다. 스키마는 에이전트가 객체의 패턴이나 클래스를 인지하게 해주는 개념적 소자이다. 우리는 부정의 정의를 에이전트가 스키마 자체를 인지하게 해주는 개념적 소자라고 규정하고, 부정의 이러한 양상을 증명한다.

세계는 객체들이 혼돈스럽게 불규칙적으로 존재하는 것이 아니라 일종의 놀라운 규칙성을 나타내고 있다. 그래서 이러한 세계의 규칙성을 이해하기 위해서는 객체와 행동을 범주화하는 온톨로지가 에이전트의 필수적인 요소이다. 객체의 클래스는 색상이나 크기와 같은 객관적 성질이나 또는 graspable이나 fit-in-my-hand와 같은 상호 작용하는 성질에 근거하여 구분이 된다. 귀납적 학습 프로그램은 (xi, yi) 형태의 데이터로부터 yi = f(xi)와 같은 함수를 배운다. yi는 클래스라고 불리고 f는 각 xi에 적절한 클래스를 부여한다. 오직 두개의 yi 값이 존재할 때 그러한 시스템은 개념을 학습한다고 하고 각 xi는 그러한 개념의 긍정적 또는 부정적 사례에 해당한다. f는 개념의 정의라고 간주할 수 있다. 의미론적 관점에서 개념은 f를 만족하는 긍정사례들의 집합을 지시한다. 데이터의 집합을 두 분할로 나누어서 개념을 학습할 수 있는 기계학습 기법은 다수 존재하지만 그 기능들은 대개가 긍정사례에 대한 개념을 학습하는 것에 국한된다. 이러한 기법은 자료 수준에서의 학습이라고 한다. 예를 들어서, C4.5는 게임을 하기에 좋은 날을 학습할 수 있다[6].

여기에서 우리의 연구와 관련 있는 질문이 있다: 부정사례의 집합을 위한 개념은 무엇인가? 현재까지는 이러한 질문에 대해서 심각하게 대답하는 기계학습 기법은 존재하지 않는다. 예를 들어서, C4.5는 게임을 하기에 적합하지 않은 날들에 대한 개념을 학습할 필요가 없다. 이점은 주로 기계학습 시스템이 문제해결의 효율성에 중점을 두고 있으면 부정데이터와 긍정데이터 사이에 존재하는 관계를 탐구하는 데에는 관심이 없기 때문이다. 이러한 유형의 시스템은 메타 개념을 학습한다고 할 수 있다. 명백하게, 부정개념 자체는 긍정개념 자체에 속하지 않는 객체들의 집합을 지칭하기 때문에 아주 간단하게 처리할 수 있다. 그러나 에이전트가 실생활에 적응하기 위해서는 부정이나 반대개념을 인지하는 등의 보다 복잡한

인지능력이 필요할 것이다. 이점은 인간이 긍정적 사실과 부정적 사실을 함께 다루거나 나아가서 부정적 사실을 긍정적 술어에 의해 표현하는 등의 경향을 살펴볼 때 특히 자연어 이해 등의 분야에서 중요하다고 할 수 있다. 예를 들면, 우리는 초등학생이 쉽다와 어렵다, 또는 차갑다와 뜨겁다와 같은 반대 개념을 암기하는 것을 볼 수 있는데 이것은 어렵지 않다 나 차갑지 않다와 밀접한 관계가 있다.

에이전트가 어떤 객체들의 클래스를 하나의 개념으로 인식하면, 그것은 하나의 클래스를 그 클래스에 속하지 않는 나머지 객체들로부터 분별한다는 것을 의미한다. 이것은 에이전트가 세계를 두개의 클래스로 분할하는 것을 암시하며 또한 그것이 하나의 주어진 객체가 두 개 중 어느 클래스에 속하는지를 알고 있다는 것을 의미한다. 만일 에이전트가 하나의 객체에 대한 개념을 파악한다면 그것은 임의의 어떤 객체가 그 개념에 속하는지 여부도 또한 알 수 있다는 것을 의미한다. 따라서 이분법은 세계에 대한 개념화를 향한 초기적 수준을 말한다.

어떤 영역에서 전체집합 U가 두 집합 A와 B로 분할되었다고 가정하자. A의 여집합은 전체 U에는 소속되지만 A에는 속하지 않는 원소들의 집합이며 B는 A의 여집합이다. 역으로 A는 B의 여집합이다. A와 B는 둘 다 개념의 정의를 만족한다. 한 개념의 부정은 그 개념에 속하지 않는 모든 나머지 객체들의 집합을 지칭하기 때문에 부정의 관계는 실제로는 여집합 관계이다. 따라서 만일 에이전트가 개념 A를 인지한다면 그것은 개념 B를 알아야 한다. B의 존재는 어느 영역의 U에서나 A를 정의하거나 인식하는데 필요한 배경지식으로 작용한다. 그리고 B는 A의 부정이라고 부른다. 인지적 에이전트는 이러한 여집합적인 관계를 알 수 있는 지적 능력을 소유하여야 한다. 이러한 능력이 없다면 하나의 클래스를 인지하는 것이 불가능하기 때문이다. 따라서 어떤 세계를 두 부분으로 분할하는 능력은 인지적 소자라고 할 수 있다. 이러한 논지에서 우리는 부정의 스키마적 정의를 증명한다.

정리 : 부정은 인지적 에이전트의 기본적 인지 스키마이다.

증명 : 인지적 에이전트가 존재한다고 하자. 에이전트가 자신의 온톨로지가 없이 추론하는 것은 불가능하다. 귀납법의 기본으로 그러한 에이전트의 지각하는 세계에는 오직 하나의 클래스가 존재한다고 하자. 그것은 에이전트에게는 세계를 분류하여 인지하게 하는 어떠한 온톨로지가 존재할 수 없음을 나타낸다. 그리고 에이전트는 세계에 대해서 이해할 수도 추론할 수도 없다. 따라서 세계에는 하나 이상의 클래스가 존재해야 한다. 에이전트의 지각에 오직 두개의 클래스가 존재할 때, 세계에 존재하는 모든 객체는 두개의 분할로 나누어지며 그중에 하나의 분할 A는 어떤 성질 또는 함수 p를 만족하며 다른 분할 B는 그 성질을 만족시키지 않는다. 따라서 에이전트는 B를 개념에 대한 부정으로 인식하는 한, 분할 A를 개념으로 인식할 수 있다. 그것은 하나의 클래스를 그에 대한 부정이 없이는 인식하는 것이 불가능하기 때문에 부정은 개념적 소자이다. 따라서 부정은 인지적 에이전트의 기본적 인지 스키마이다.

클래스 내부에 어떠한 조직이 없는 하나의 클래스만 있다는 것은 실제로는 세계에는 인식할 클래스가 없다는 것을 의미한다. 상대주의는 어떤 현상을 이해하는 하나의 기본적 접근법이다. 만일 우리가 의자에 앉아 있을 때 우리는 우리가 움직이고 있다는 사실을 알지 못하지만, 지구 밖의 외계에 있는 사람은 우리가 지구와 같이 움직이고 있음을 지각할 수 있을 것이다. 우리는 지구의 일부가 아닌 곳에서의 관점을 가졌을 때만 우리가 움직이고 있다는 사실을 지각할 수 있듯이 우리는 오직 또 다른 개념을 비교하기 위해서 가지고 있을 때만 하나의 개념을 파악할 수 있으며 우리는 이러한 가장 초보적인 개념을 그 개념의 부정이라고 한다. 따라서 하나의 개념을 인식한다는

것은 그 부정을 인식하고 있다는 점을 암시한다.

4. 부정의 긍정표상으로서의 반대개념

부정 술어를 사용하는 문제는 왜 부정에 대한 지식을 명시하는 것이 기계에는 필요한데 인간에게는 불필요한지에 대한 의문을 제기한다. 상태 S1에 두개의 술어 p와 q가 기술되어 있다고 가정하자. 만일 규칙 R : p → q 가 어떤 시스템 A에는 알려져 있다면, 또 하나의 상태 S2가 S1에서 q를 제거함으로써 생성될 수 있으며 S1과 S2는 그 규칙 R과 관계 하에서는 서로 동등하다. 반면에 규칙 R이 또 하나의 시스템 B에는 알려지지 않았다고 가정하자. B는 p로부터 q를 추론할 수 없기 때문에, S1과 S2는 동등할 수 없으며 S2는 B에게 noise 등의 문제를 야기할 수 있다. 우리는 아직 인간의 마음을 과학적으로 이해하는 수준에 이르지 않았기 때문에 인간의 지식체계를 분석할 수는 없으며, 따라서 이러한 애매한 지식을 기계에 명확하게 코드화 할 수 없다. 그러나 그러한 첫 단계로서 우리는 반대 개념을 이해하는 다소 단순한 문제에서 시작해 보고자 한다. 이러한 반대 개념 R은 긍정사실에서 부정사실을 기계가 추론하도록 하는데 이용될 수 있다.

인간은 주어진 사실을 부정하기 위해서 반대개념이라고 하는 새로운 술어를 도입하여 활용한다는 점을 관찰함으로써, 본 논문에서는 보다 지능적인 에이전트를 구현하기 위한 기법으로서 부정적 사실에 대해 부정 함수 not을 쓰기 보다는 새로운 술어를 사용할 것을 제안한다. 그리고 시간과 공간의 관점에서 우리의 주장을 지지하는 실험적 결과를 IPP 영역에서 제시한다. 계획의 영역이론은 해당 업무 영역에 대한 에이전트의 지식을 표현한다. 하나의 계획 연산자는 로봇의 하나의 행동 모듈에 해당한다[9]. 각 모듈은 다른 모듈과는 독립적으로 처리되기 때문에 각 연산자도 또한 영역이론에서 각각 독립된 모듈이다. 그러나 각 연산자들이 표면적으로는 서로 관련이 없을지라도 인간 지각의 기저구조에서는 밀접하게 연관되어 있을 수 있다. 예를 들어서, open-dr와 close-dr 연산자들은 개념적으로는 반대로 간주되고 있다. 연산자는 에이전트의 행동을 전제조건, pre(op)과 결과 상태 effects(op)로 모델링 하는데 결과 상태는 다시 add(op)과 del(op)으로 구성되어 있다. 연산자의 부정적 결과는 del(p)나 add(not p)로 표현될 수 있다. 표상을 이해하는 인간의 인지능력을 모사하기 위하여, 우선 계획의 분야에서 부정적 사실을 표현하는 두 가지 방식을 소개하고 각 접근방식에서의 문제점들을 토론한 후 그 해결책으로 인간과 유사한 반대개념의 도입을 제안한다.

Graphplan은 not을 추론할 수 없으며 단지 문자열로 처리한다. 만일 연산자가 술어 p가 거짓임을 전제조건에 필요로 한다면 (not P)에 상응하는 not-p를 새로운 명제로 정의할 필요가 있다. 예를 들어서, p가 (on-ground <y>)라고 하면, 우리는 not-p가 (not on-ground <y>)나 (not-on-ground <y>) 또는 (up-in-the-air<p>)로 표기될 수 있다.

에이전트의 팔이 실제로 비어있을 때 센서가 부정확하여서 동시에 어떤 객체를 잡고 있다고 믿는다고 가정하자 : {arm-empty, holing-x}. 불완전한 영역지식을 가진 에이전트는 이러한 상태는 불가능하다는 점을 파악할 수가 없다. 그러나 정상적인 인간은 만일 arm-empty가 사실이면 그 상태에서는 ¬holding-x가 동시에 사실이어야 한다고 추론할 수 있다. 따라서 그는 위와 같은 믿음은 반대되는 사실들을 포함하는 모순된 상태라고 쉽게 추론할 수 있다 : {arm-empty, holing-x, ¬holding-x}. 위와 같이 긍정 술어에서 부정 술어를 추론하는 과정이 인간에게는 단순 명백해 보이지만, 기계에서는 핵심적인 제어지식으로 활용될 수 있다.

만일 술어 (on-ground <y>)가 연산자를 적용해서 effects에서 제거되어야 한다면, 에이전트의 인식에서는 (on-ground <y>)가 (not-on-ground <y>)에 반대되는 개념이 아니기 때문에 Del(on-ground

<y>)와 Add(not-on-ground <y>)가 둘 다 동시에 effects 리스트에 명시되어야 한다. 이러한 중복표현의 문제를 제거하기 위해서 IPP에서는 p를 부정하기 위해서 부정함수 not을 도입한다. not-p가 더 이상 사용되지 않기 때문에 부정효과는 {Add (not-p), Del(p)} 대신에 {Add(not p)}로 통일적으로 사용될 수 있다. 예를 들어서, IPP에서 사용되는 Brief-case 영역에서 Take-out 연산자가 원래 전제조건 {in(x), is-at(loc)}, add-list {not in(x)}, delete-list {in(x)}를 가진다고 하자. 연산자의 결과상태는 중복표현이 되었으며 add-list에서 추론될 수 있으므로, 그러한 연산자는 전제조건 {in(x), is-at (loc)}, add-list {not in(x)}로 간결하게 표현될 수 있다.

비록 IPP가 not을 사용하여 중복표현의 문제를 해결하였지만 그러한 부정어 사용의 장점은 메모리 공간과 처리시간의 측면에서 단점을 수반한다. not-P는 긍정 아톰(atom)이지만 (not P)는 복합항(composite term)이라는 점에 유의해야 한다. 당연히, 하나의 개념을 한 단어가 아니라 두개의 단어로 표현하는 것은 불편한 일이다. 또한 긍정 항보다 부정 항을 다루는 것이 더 복잡하고 어렵다. 따라서 지능적인 시스템을 구성하기 위해서는 부정적 복합 항 보다는 긍정적 단일 항을 사용하여 개념을 표현하는 것이 보다 효율적이라고 합리적으로 가정할 수 있다.

인간의 사고의 독창성의 한 면모는 부정 개념을 표현하기 위하여 단일 항적 긍정 표현법을 창안함으로써 복잡성과 불편성을 제거하는 능력이며 인간의 이성은 최소묘사길이(Minimum Description Length)의 원리에 따라 작동하는 것으로 여겨진다. 우리는 먼저 인간이 어떻게 개념 및 그 반대개념을 학습하고 부정개념을 표현하기 위해서 새로운 긍정 항을 만들어 내는지 추론을 할 것이다. 아이는 다양한 속성을 지닌 어떤 객체의 집합을 동일한 속성을 지닌 동일 객체들의 부분집합으로 구분함으로써 개념을 배울 수 있다. 아이가 물을 만지고 젖다(wet)의 개념을 배운다고 가정하자. 또한 마르다(dry)의 개념은 아직 모르고 있다고 하자. 그러한 현상을 매번 젖지 않다 (not wet)를 사용하여 표현하는 것은 꽤 불편할 것이다. 마찬가지로 새로운 조어가 없이 not만을 사용하는 시스템은 그 표현력에서 한계가 있다. 보다 표현력이 있는 시스템은 연산자의 전제조건이나 결과를 표현하기 위해서 arm-empty의 부정으로 not arm-empty 보다는 holding(x)로 표현해야 한다. not P를 표현하기 위해서 긍정 항을 사용하는 우리의 방법론은 2치 시스템(two-valued system)에 적용되며 다치 시스템(multi-valued system)에는 적용되지 않는다. 예를 들어서, {white, red, blue, black}의 색채 관련 속성집합이 있을 때, (not white)는 black 보다는 (red ∨ blue ∨ black)의 이접적 항(disjunctive term)을 의미한다.

우리의 가설을 검증하기 위해서 각각 긍정 및 부정 표현을 사용하는 두 유형의 연산자를 사용하는 IPP 영역의 계획 문제들을 풀어 보았다. 예를 들어서, 긍정표현을 가진 Briefcaseworld 영역에서 Put-in 연산자는 전제조건 {(not-in(x), at(x, loc)}와 결과 add-list {in(x)}와 del-list {(not-in(x)}를 가지고 있으며, 부정표현을 가진 Negated Briefcaseworld 영역에서 Put-in 연산자는 전제조건 {(not in(x), at(x, loc)}와 결과 add-list {in(x)}를 가지고 있다. 우리는 Briefcaseworld와 Negated Briefcaseworld의 두 영역에서 다섯 문제를 해결하기 위해서 똑같은 연산자를 사용하였다. 예를 들어서, ex3a.fct 문제는 paycheck, dictionary, ticket 등의 객체들로 구성되어 있다. 초기상태에서 에이전트는 집에 있으며, 수표는 은행에 있고, 사전은 사무실에 있고 기차표는 역에 있다. 에이전트의 목표는 가방을 가지고 돌아다니면서 위 물건들을 집으로 가져오는 것이다. 다른 문제들은 위의 첫 번째 문제를 다소 변경하여 보다 복잡한 초기상태나 목표를 가지고 있다. IPP는 두 영역 모두에서 같은 수의 action을 시험하여 정확히 같은 계획을 생성하지만, 아래 표에 나타난 바와 같이 공간 및 시간의 측면에서 긍정영역이 둘째의 부정

영역보다 더 효율적이었다. 아래 표에서 초 단위로 두 영역에서 처리 시간을 측정하였다.

Name of Problem	Briefcaseworld Domain	Negated Briefcase Domain
Ex3a.fct	0.25	0.37
Ex3b.fct	0.08	0.33
Ex4a.fct	1.86	2.29
Ex5max.fct	3.29	5.28
Ex5d.fct	52.89	68.21

또한 아래 표에 나타난 바와 같이 K Bytes 단위로 두 영역에서 사용된 메모리 공간을 측정하였다. 실험결과는 공간의 측면에서 부정적 표현보다는 부정에 대한 긍정적 표현이 보다 효율적임을 보여준다.

Name of Problem	Briefcaseworld Domain	Negated Briefcase Domain
Ex3a.fct	2295.0	3475.7
Ex3b.fct	457.4	808.6
Ex4a.fct	4864.7	7286.2
Ex5max.fct	10848.0	15950.9
Ex5d.fct	10848.0	15950.9

부정은 구문론적 함수이지만 반대개념은 의미론적 과정이며 반대의 도입은 에이전트 시스템의 성능을 향상할 수 있음을 유의해야 한다. IPP와 같은 최근의 시스템은 P와 not-P, 즉 on 과 off, 가 서로 반대된다는 점을 이해할 수 없기 때문에, 이러한 능력을 지닌 에이전트는 복잡한 동적인 영역에서는 확장이 불가능하다. 비유적으로, Graphplan을 확장하기 위해서 IPP에 not이 이식되었듯이 반대함수가 부정을 추론하기 위해 인지에이전트에 이식될 수 있을 것이며, 반대개념을 에이전트 내부에서 어떻게 처리해야 하는지 하는 구현 방법은 영역이론에서 숨겨져 있어야 한다.

5. 결 론

현재 지능적 에이전트는 지식 표상의 구문론적 표기법으로 인하여 자신의 지각과 행동의 의미를 이해할 수 없다. 이러한 문제를 해결하기 위해서 코헨은 물리적 스키마가 세계와의 상호작용을 통해서 학습될 수 있다는 가설을 제시하였다. 본 논문은 부정은 물리적 스키마를 인지하기 위해서 반드시 필요한 메타 스키마임을 증명하고, 부정적 사실을 표현하는 두 가지 방법을 비교하였다. 또한 부정적 사실을 표현하기 위해서 반대개념이라고 하는 긍정 아톰을 사용하는 것이 지능 에이전트를 구현하는 보다 효율적인 방법이라는 가정을 하며 이것을 입증하는 실험 결과를 제시하였다. 의미론적으로 반대개념을 학습하는 능력은 에이전트가 실제세계의 복잡한 문제를 해결하는데 있어서 근본적인 중요한 문제이다.

참 고 문 헌

[1] Cohen, P. R., Atkin, M. S., Oates, T., Neo: Learning Conceptual Knowledge by Sensorimotor Interaction with an Environment, *Proceedings of the 6th International Conference on Intelligent Autonomous System*, 2000

[2] Kaelbling, L. P., Oates, T., Hernandes, N., and Finney, S., Learning in Worlds with Objects, 2001 AAAI Spring Symposium Workshop, Stanford University, 2001

[3] Kambhampati, R. and Lambrecht, E, Parker, E. Understanding and extending Graphplan, Proc. 4th European Conference on Planning, 1997.

[4] Koehler, J. Extending Planning Graphs to an ADL Subset, Proc. 4th European Conference on Planning, 1997.

[5] Mandler, J. M., How to build a baby: Conceptual primitives. Psychological Review 99(4), 1992

[6] Quinlan, J. R., C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993

[7] Russell, S. Norvig, P., Artificial Intelligence: A Modern Approach, Prentice-Hall International, 1995

[8] Searle, J. R., Minds, Brains, and Programs, Behavioral and Brain Science, 1980.

[9] Weld, D. Recent Advances in AI Planning, *AI Magazine*, 1999.

[10] Wittgenstein, L., Philosophical Investigations, Macmillan, London, 1953.

◐ 저 자 소 개 ◑

태 강 수

1983년 전북대학교 영어영문학과 (학사)
1991년 University of North Texas 컴퓨터과학과 (이학석사)
1991년 미 IBM사 근무
1997년 University of Texas 컴퓨터공학과 (공학박사)
1997년~1998년 성덕대학 전자계산학과 전임강사
1998년~현재 : 전주대학교 정보기술컴퓨터공학부 조교수
관심분야 : 기계학습, 계획, 인공지능, 인지과학
E-mail : kstae@jeonju.ac.kr