

지식베이스의 확장과 다중행위자 시스템에서의 협상

계명대학교 전자계산학과

이 명진*, 김 진상

The Extension of Knowledge Base and Negotiation in Multi-Agent Systems

Department of Computer Science, Keimyung University

Myung-Jin Lee, Jin-Sang Kim

요약

행위자는 관찰 혹은 결론에 대해 최대한으로 일관성이 있는 지식베이스의 확장을 필요로 한다. 이와 같은 지식베이스의 확장은 상호간에 부합하지 않고 서로 충돌하는 결론을 가지는 모순되는 많은 확장들을 수반한다. 본 논문에서는 abduction의 개념을 이용하여 지식베이스를 확장시키는 방법을 소개한다. 한편 다중 행위자 시스템에서 문제 해결자로 부르는 행위자들이 효율적으로 상호작용하도록 자동화된 행위자들을 어떻게 설계할 것인가는 매우 중요한 사항이다. 따라서 행위자는 각자의 목표를 통신하여 상호 도움이 되는 일치 상황에 도달하기 위해 타협하도록 설계되어야 한다. 본 논문에서는 argumentation 시스템을 이용하여 믿음, 소망, 그리고 의지를 가진 BDI 행위자 구조를 위한 기초적인 자동화된 협상 모델을 소개한다.

1. 서론

행위자는 자신의 불완전한 지식베이스로부터의 관찰 혹은 다른 행위자들이 전송한 정보에 대해 추측하기 위해서 가능한 설명으로써 가설들의 집합을 제공할 필요가 있으며, 이러한 가설들의 집합을 추론하기 위해서 자신의 지식베이스와 가설이 무결성 제약조건을 만족해야 함을 의미하는 abduction의 선언적 의미론과 가설들을 생성하는 작용적 의미론을 갖고 있어야 한다.

한편 다중 행위자 시스템은 주어진 문제를 해결하려 하는 자율적인 행위자들의 행위 그리고 행위자들이 효율적으로 상호작용하기 위해 행위자들을 어떻게 설계할 것인가에 관심을 가진다 [4]. 협상은 혼자 힘으로 달성할 수 없는 목표를 달성하려 하는 두 개 혹은 그 이상의 행위자들 사이에서 발생하는 과정이다. 이러한 목표들은 충돌할 수 있고 행위자들은 목표를 달성하기 위해 서로 의존할 수 있다. 따라서 행위자는 각자의 목표를 통신하여 상호 도움이 되는 일치에 도달하기 위해 타협하도록 설계되어야 한다.

본 논문에서는 abduction의 선언적 의미론으로 선택된 확장 그리고 수용가능성 의미론을 가지는 행위자를 소개하고, 작용적 의미론의 역할을 소개한다. 또한 행위자들이 표현력이 있는 통신언어를 사용하여 서로의 믿음과 의도를 변화시키기 위해서 설득력이 있는 협상을 수행할 수 있게 하는 argumentation 시스템에 기초한 행위자들의 협상에 관하여 연구한다.

2. 행위자의 지식베이스의 확장

2.1 Abduction의 선언적 의미론

Abduction은 비단조 추론 중의 하나이며, 불완전한 지식베이스로부터의 관찰에 대해 추측하기 위해서 그 관찰의 설명의 일부분으로써 가설들의 집합을 제공한다. 일반적으로 abduction 추론 이론은 $\langle KB, H, IC \rangle$ 로 정의하는 abduction 프레임워크에 기초한다. 여기서 KB 는 지식베이스를 표현하는 일차 이론이며, H 는 가능한 가설들을 표현하는 일차 식들의 집합이며, 그리고 IC 는 받아들일 수 있는 설명들을 결정하기 위해서 사용하는 무결성 제약조건들을 표현하는 일차 식들의 집합이다.

Abduction 프레임워크 $\langle KB, H, IC \rangle$ 가 주어지면 가설들의 집합 $E \subseteq H$ 가 질의어 Q 에 대한 abduction 해답일 필요충분조건은 다음과 같다:

$$KB \cup E \vdash Q \text{ 그리고 } KB \cup E \text{가 } IC \text{를 만족.}$$

Abduction 프로그램 KB 의 시나리오는 일차 이론 $KB \cup H$ 이다. 여기서 H 는 가설들의 집합이고 $KB \cup H \cup IC$ 는 일관성이 있다. 프로그램 KB 의 선택된 확장은 최대한으로 수용가능한 KB 의 시나리오이다 [1]. 가설을 수용하는 경우는 그 가설의 반대의 증거가 없을 때이다. "반대의 증거가 없다"는 $Atom A$ 의 증거 E 는 $S \vdash B$ 가 되는 $not-B \in E$ 가 존재한다면 시나리오 S 에 의해서 패배된다고 한다. 그래서 가설이 수용가능할 필요충분조건은 그 가설의 반대의 증거가 패배되는 경우이다.

2.2 Abduction의 작용적 의미론

Abduction 프로그램 KB 에 관하여 질의어 Q 의 abduction 해답이 존재한다면 KB 의 선택된 확장 $KB \cup H(E \subseteq H)$ 가 존재한다. Abduction 증명 프로시저는 아래의 두 가지의 단계들로 구성된 SLDNF-resolution의 확장으로 생각할 수 있다:

1. 반박을 유도하기 위해 후향추론하여 필요한 가설들을 수집 (abduction 유도)
2. 수집된 가설들이 일관성이 있는가를 검사(일관성 유도)

3. Argumentation에 기초한 BDI 행위자의 협상

지능적인 행위자는 동적인 환경에 존재하며, 그러한 환경에 관하여 불완전한 지식을 갖고있으며, 그리고 행위의 수행을 통해 그러한 환경에 부분적으로 영향을 미칠 수 있는 하나의 시스템이다. 전형적으로 행위자들은 충돌할 수도 있는 여러 개의 작업들을 수행하며, 행위자의 결정이 유용한 결과를 가지게 하기 위해서 작업들을 어떻게 달성할 것인가에 대해 결정을 내릴 필요가 있다. 이와 같은 요구사항들을 만족할 수 있는 소프트웨어 구조를 개발하기 위한 방법들 중의 하나는 BDI 접근방법이다. 전형적인 BDI 구조는 다섯 가지의 지식구조들을 포함한다: 행위자의 믿음, 소망, 의지, 계획 라이브러리, 그리고 해석기 [5].

3.1 Argumentation 시스템

Argumentation 시스템에서 argument는 임시적인 증명이다. 왜냐하면 이러한 argument는 다른 행위자들에 의해 평가되기 때문이다. Argument에 관한 추론은 복잡한 사실에 대한 argument를 만들기 위한 다른 argument들의 결합, 하나의 사실에 대해 경쟁하는 argument들의 단일화, 그리고 argument의 수용가능성을 결정하기 위한 argument 구조의 분석을 가능하게 한다. Argumentation 시스템은 관심있는 명제에 찬성 혹은 반대하는 일련의 논리적인 단계들 즉, argument들을 만들어낸다. 고전적인 논리에서 argument는 결론에 이르게 하는 추론들의 나열이다. Argument가 옳다면 그 결론은 참이다. 간단한 지식베이스 Δ_1 을 생각해보자:

Δ_1 : $human(socrates), \quad f1$
 $human(X) \rightarrow mortal(X). \quad r1$

위의 지식베이스에서 argument ' $\Delta_1 \vdash mortal(socrates)$ '는 옳게 만들어질 수 있다. 왜냐하면 $mortal(socrates)$ 는 Δ_1 으로부터 증명가능하기 때문이다. 그래서 옳은 argument는 "f1과 r1의 문맥에서 $mortal(socrates)$ 가 참이다"라는 결론을 만들어낸다[3]. 찬성 혹은 반대의 정도는 그 유도에서 사용된 명제들을 검사함으로써 결정할 수 있다.

행위자는 자신의 지식베이스에 있는 정보를 사용하여 명제에 찬성 혹은 반대하는 argument를 만들어내기 위해서 적당한 argument 결과 관계 ' \vdash_{ACR} '을 사용할 수 있다. Argumentation의 형태를 다음과 같이 정의할 수 있다:

$Database \vdash_{ACR} (Sentence, SIR)$

여기서 ' \vdash_{ACR} '은 적당한 argument 결과 관계를 나타내며, 'SIR'은 'Sentence'를 추론하는데 사용된 추론 규칙들의 집합이다. 모든 argument들을 수용가능성의 종류들로 분류함으로써 경쟁하는 argument들을 단일화할 수 있다.

3.2 통신 언어

사회적인 문맥에서 행위자들의 협상은 일반적인 통신 언어 CL에서 메시지의 교환을 통해 달성된다[4]. 메시지의 실제적인 교환은 참여하는 행위자들의 개인적인 필요와 목표에 의해 유도된다. 협상은 행위자들 사이의 통신을 필요로 하고, 애매함을 없애기 위해서 각각의 행위자는 고유한 식별자를 가져야 한다. 협상에 관련된 행위자들의 식별자들의 집합을 Agents로 표시한다. BDI 행위자들이 부족한 자원들에 대해 협상한다고 가정하면 행위자들은 자신의 목표를 달성하기 위해서 자신에게 필요한 부족한 자원의 할당을 요구한다. 자원들의 집합을 Resources로 표시한다. 이러한 경우에 행위자들이 사용하는 통신 언어 CL을 아래와 같이 간단히 정의할 수 있다:

정의 1. $a, b \in Agents$ 와 $r \in Resources$ 라 할 때 통신 언어 CL을 다음과 같이 가정한다:

1. $Request(a, b, Give(b, a, r), (\xi, \varphi), t) \in CL$
2. $Accept(a, b, Give(a, b, r), t) \in CL$
3. $Reject(a, b, Give(a, b, r), (\xi, \varphi), t) \in CL$

여기서 t 는 시간 변수이고 (ξ, φ) 는 argument이다. 즉, φ 는 행위자 a 가 식 ξ 를 추론하는데 사용한 추론 규칙들의 집합이다. ' $Request(a, b, Give(b, a, r), (\xi, \varphi), t)$ '를 "시간 t 에서 행위자 a 가 ξ 를 달성하기 위해서 행위자 b 에게 부족한 자원 r 을 요청한다"라고 해석할 수 있다. 행위자의 실제적인 제안에 해당하는 ' $Give(b, a, r)$ '은 특정 영역에 따라 여러 형태로 확장할 수 있

다.

3.3 Argument의 평가

Argumentation 개념의 기초는 argument가 다른 argument를 공격할 수 있다는 것이다. Argument φ 가 식 ξ 를 지원한다는 사실을 (ξ, φ) 로 표현하고, argument 쌍 (ξ_1, φ_1) 이 (ξ_2, φ_2) 를 공격한다는 사실을 $Attacks((\xi_1, \varphi_1), (\xi_2, \varphi_2))$ 로 표현한다. 공격의 정확한 의미는 사용하는 구체적인 언어 L 에 강하게 의존한다. 본 논문에서는 BDI 행위자 구조를 사용하기 때문에 다음과 같이 공격을 정의한다:

정의 2. Argument (ξ_1, φ_1) 과 (ξ_2, φ_2) 에서 φ_1 과 φ_2 를 각각 $M^1(p_1) \wedge \dots \wedge M^n(p_n)$ 와 $M^1(q_1) \wedge \dots \wedge M^m(q_m)$ 형태의 문장이라 가정한다. 여기서 M^i 와 M^j 은 믿음, 소망, 혹은 의지를 나타낸다.

1. $p_i \equiv \neg q_i$ 이면 (ξ_1, φ_1) 는 (ξ_2, φ_2) 를 공격한다.
2. $l \in \xi_2, (l, \varphi_3)$, 그리고 $\varphi_3 = M^1(r_1) \wedge \dots \wedge M^m(r_m)$ 이면 $p_i \equiv \neg r_k$ 인 경우에 (ξ_1, φ_1) 는 (ξ_2, φ_2) 를 공격한다.

예를 들어, 첫 번째 경우에서 "Socrates는 철학자가 되려고 한다"는 "Plato는 Socrates가 철학자가 되지 않았으면 한다"를 공격한다. 두 번째 경우에서 "Plato가 신의 존재를 믿고있다고 Socrates는 믿는다"는 "Plato는 신의 존재를 믿지 않는다"를 공격한다.

수신한 메시지를 어떻게 해석하고 송신할 메시지를 어떻게 만들어낼 것인가를 결정하기 위해서 행위자가 사용하는 수단인 argumentation이다. 현재의 이론으로부터 만들어진 어떤 argument 쌍 (ξ_2, φ_2) 에 의해 공격을 당하지 않는 argument 쌍 (ξ_1, φ_1) 를 받은 경우에 행위자는 간단히 argument φ_1 과 식 ξ_1 을 자신의 이론에 추가할 수 있다.

3.4 메시지의 해석과 생성

행위자들의 쌍 사이의 협상의 역사를 포함하기 위해서 행위자의 메모리를 협상 상태로 표현한다[4].

정의 3. 시간 t 에서의 행위자 a 와 b 사이의 협상 스래드(\mathcal{S}_{a-b} 혹은 \mathcal{S}_{b-a} 로 표기)는 $\langle x^j_{dir-ei} : t_j \leq t \rangle$ 형태의 유한한 나열이다. 여기서,

1. $x^j_{dir-ei} \in CL$
2. $d_i, e_i \in (a, b)$
3. $d_i \neq e_i$

정의 4. 일차 언어의 정렬식들의 집합 W 에서 믿음, 소망, 혹은 의지를 나타내는 M 으로 확장한 언어 $M(W)$ 와 시간 t 에서 행위자 a 에 대한 협상 상태 s 는 다음과 같다: $s = \langle \mathcal{Q}, T, H \rangle$

1. \mathcal{Q} 는 협상이 가능한 자원들의 모임
2. $T \subseteq M(W)$ 는 이론
3. 협상의 역사 H 는 행위자 a 를 포함하는 모든 협상 스래드들의 집합 즉, $H = \{\mathcal{S}_{r-a} \mid i \in Agents\}$.

메시지의 의미론의 정의를 연산자 I 와 G 로 구분한다. I 는 협상 상태 전위를 구현하고 G 는 특별한 상태에서 취할 메시지 행위를 결정한다. 메시지의 실제적인 결과는 행위자의 해석에 의존한다. 이러한 해석 과정은 주로 특정 영역에 의존하고 또한 행위자의 특정한 내부 구조에 의존한다.

정의 5. 메시지의 해석: 통신 언어 CL과 행위자 b 의 가능한 모든 협상 상태들의 집합 S_b 가 주어지면, 행위자 b 를 위한 해석 함수 $I : CL \times S_b \rightarrow S_b$ 를 다음과 같이 정의할 수 있다:

$I(Request(a, b, Give(b, a, r), (\xi, \varphi), t), s) = \langle \mathcal{Q}, T, H - \mathcal{S}_{r-a} + \mathcal{S}'_{r-a} \rangle$

이때 $\theta'_{b \rightarrow a} = \theta_{b \rightarrow a} \wedge Request(a, b, Give(b, a, r), (\xi, \varphi), t)$
 $I(Accept(a, b, Give(a, b, r), t), s) = \langle Q + r, T', H - \theta_{b \rightarrow a} \rangle$
 이때 $T' = T - B(Have(a, r)) + B(Have(b, r))$
 $I(Reject(a, b, Give(a, b, r), (\xi, \varphi), t), s) = \langle Q, T, H - \theta_{b \rightarrow a} + \theta'_{b \rightarrow a} \rangle$
 이때 $\theta'_{b \rightarrow a} = \theta_{b \rightarrow a} \wedge Reject(a, b, Give(a, b, r), (\xi, \varphi), t)$

정의 6. 메시지의 생성: 행위자 b 의 협상 상태 $s = \langle Q, T, H \rangle$ 가 주어지면, 행위자 a 에 응답하는 행위자 b 를 위한 생성 함수 G 를 다음과 같이 정의할 수 있다:

If $\exists (\xi', \varphi')$ such that $Attacks((\xi', \varphi'), (\xi, \varphi))$
 $G(Request(a, b, Give(b, a, r), (\xi, \varphi), t_1)) = Accept(b, a, Give(b, a, r), t_2)$
 If $\exists (\xi', \varphi')$ such that $Attacks((\xi', \varphi'), (\xi, \varphi))$ and $A(\xi', \varphi') \geq A(\xi, \varphi)$ and \exists scarce resource r'
 $G(Request(a, b, Give(b, a, r), (\xi, \varphi), t_1)) = Request(b, a, Give(a, b, r'), (\xi', \varphi'), t_2)$
 else $\exists (\xi', \varphi')$ such that $Attacks((\xi', \varphi'), (\xi, \varphi))$ and $A(\xi', \varphi') \geq A(\xi, \varphi)$ and \exists scarce resource
 $G(Request(a, b, Give(b, a, r), (\xi, \varphi), t_1)) = Reject(b, a, Give(b, a, r), (\xi', \varphi'), t_2)$
 If $\exists (\xi', \varphi')$ such that $Attacks((\xi', \varphi'), (\xi, \varphi))$ and $A(\xi', \varphi') < A(\xi, \varphi)$
 $G(Request(a, b, Give(b, a, r), (\xi, \varphi), t_1)) = Accept(b, a, Give(b, a, r), t_2)$
 If $\exists (\xi', \varphi')$ that supports (ξ, φ)
 $G(Reject(a, b, Give(a, b, r), (\xi, \varphi), t_1)) = Request(b, a, Give(a, b, r), (\xi', \varphi'), t_2)$
 else $G(Reject(a, b, Give(a, b, r), (\xi, \varphi), t_1)) = Send\ negotiation\ termination\ message$
 If \exists scarce resource r'
 $G(Accept(a, b, Give(a, b, r), t_1)) = Request(b, a, Give(a, b, r'), (\xi, \varphi), t_2)$
 else $Send\ negotiation\ termination\ message$

추가적으로 생성 함수 G 의 결과가 *accept*인 경우에 행위자 b 는 자신의 이론 T 와 협상이 가능한 자원들의 집합 Q 를 다음과 같이 변경하여야 한다:

$$T = T - B(Have(b, r)) + B(Have(a, r)), \quad Q = Q - r$$

위와 같은 해석 함수 I 와 생성 함수 G 를 포함하는 BDI 행위자 구조에서 argument를 이용하여 협상이 어떻게 진행되는가를 다음절에서 예제를 통해 살펴본다.

3.5 예제: BDI 행위자

BDI 행위자 구조에서 실제로 argumentation 시스템이 어떻게 작용하는지를 보이기 위해서 다른 목표들과 다른 자원들을 가진 집안 일을 도와주는 두 개의 행위자들의 경우를 생각해 보자. 행위자 a 는 그림을 걸려고 하고 그림, 나사못, 스크루드라이버, 그리고 망치를 갖고 있다. 행위자 b 는 행위자 a 가 못을 갖고 있다고 믿고 있고, 그림을 거는 방법과 거울을 거는 방법을 알고 있다:

$\Delta_a:$ $I_a(Can(a, hang_picture))$
 $B_a(Have(a, picture))$
 $B_a(Have(a, screw))$
 $B_a(Have(a, hammer))$
 $B_a(Have(a, screwdriver))$
 $B_a(Have(b, nail))$
 $B_a(Have(X, hammer)) \wedge B_a(Have(Y, nail)) \wedge B_a(Have(Z, picture))$
 $\rightarrow B_a(Can(W, hang_picture))$
 $B_a(Have(X, screwdriver)) \wedge B_a(Have(X, screw)) \wedge B_a(Have(X, mirror))$
 $\rightarrow B_a(Can(X, hang_mirror))$
 $B_a(Can(X, Y)) \wedge I_a(Do(X, Y)) \rightarrow B_a(Do(X, Y))$

관련 행위자 b 는 거울을 걸려고 하고, 거울과 못을 갖고 있고, 그리고 못을 사용하여 거울을 거는 방법을 알고 있다:

$\Delta_b:$ $I_b(Can(b, hang_mirror))$
 $B_b(Have(b, mirror))$
 $B_b(Have(b, nail))$
 $B_b(Have(X, hammer)) \wedge B_b(Have(Y, nail)) \wedge B_b(Have(Z, mirror))$
 $\rightarrow B_b(Can(X, hang_mirror)) \wedge B_b(\neg Have(Y, nail))$
 $B_b(Can(X, Y)) \wedge I_b(Do(X, Y)) \rightarrow B_b(Do(X, Y))$

초기상태에서 행위자 a 의 협상 상태 $s_a = \langle Q_a, T_a, H_a \rangle$ 이며 행위자 b 의 협상 상태 $s_b = \langle Q_b, T_b, H_b \rangle$ 이다. 여기서

$$Q_a = \langle picture, screw, hammer, screwdriver \rangle, T_a = \Delta_a, H_a = \emptyset$$

$$Q_b = \langle mirror, nail \rangle, T_b = \Delta_b, H_b = \emptyset$$

이다. 행위자 a 는 자신의 목표를 달성하기 위해서 $B_a(Do(a, hang_picture))$ 에 찬성하는 argument를 만들어내려 한다. 즉,

$$\Delta_a \vdash_{ACR} (B_a(Do(a, hang_picture)), \{SIR\})$$

행위자 a 는 행위자 b 가 갖고있는 못을 사용하지 않는다면 $B_a(Do(a, hang_picture))$ 에 찬성하는 argument를 만들 수 없기 때문에 스스로 그림을 걸 수 없다는 것을 알게 된다. 그러나 행위자 a 는 못의 사용이 포함된 $B_a(Do(a, hang_picture))$ 에 찬성하는 아래의 argument를 만들 수 있다:

$$\Delta_a \vdash_{ACR} (B_a(Do(a, hang_picture)), \{(1), (9), (7), (4), (6), (2)\})$$

4. 결론

Sierra, Jennings, Noriega, 그리고 Parsons는 세 가지의 메시지들(threats, rewards, appeals)을 사용한 argumentation 시스템을 통해 설득력이 있는 협상을 수행할 수 있는 형식적인 프레임워크를 제안하였다[4]. 그러나 위의 연구는 행위자 자신의 관찰에 대한 argument를 제공하지 않고, 행위자의 선언적 의미론을 고려하지 않고, 그리고 단순히 다중 행위자 시스템에서 행위자들 사이의 협상을 위한 argumentation 결과 관계들을 사용하였다.

한편 Kowalski와 Toni는 실제적인 추론 특히 법률추론에 유용하고 default 추론으로의 여러 접근방법들을 일반화시키는 argumentation에 기초한 추론의 형식적인 이론을 제안하였다. 이들은 argumentation 이론이 이론가(theorist), 논리 프로그래밍, default 논리, 자기인식(autoepistemic) 논리, 그리고 비단조 모달(modal) 논리들을 일반화하고 단일화함을 보였다[3].

본 논문에서는 argumentation 시스템을 이용하여 BDI 행위자를 위한 간단한 협상 모델을 소개하였다. Sierra는 특정 영역에 편중된 협상 모델을 제안했지만 본 논문에서는 BDI라는 보다 일반적인 상황에 적합한 협상 모델을 제안했다. 다른 행위자들에 관한 부분적인 지식을 가진 행위자는 argument를 통해 전달된 정보를 사용하여 자신의 관점에서 전달된 정보를 평가하고 자신의 목표를 달성하려 한다. 행위자는 해석 함수와 생성 함수를 사용하여 자신의 지식베이스를 수정하고 전송할 메시지를 결정한다. 간단한 BDI 행위자 구조를 가정하여 위와 같은 협상 모델이 적절히 작용함을 보였지만 다음과 같은 요소들이 본 논문에 포함될 필요가 있다: 관찰을 믿음의 일부분으로 생각한 Kowalski의 연구[2]를 다중 행위자 시스템에 적용하여 보다 정교한 지식베이스의 확장을 통한 협상 모델의 개발, 행위자들 사이의 역할 관계, 행위들에 대한 선호관계, 보다 정확하고 명료한 메시지의 도입과 이에 대한 해석 함수와 생성 함수의 도입 등이다.

참고문헌

- [1] Dung, P. M., An abductive foundation of logic programming, J. of Logic Programming, 22(2), 1995.
- [2] Kowalski, R. A., and Sadri, F., From logic programming towards multi-agent systems, To appear in: Annals of Mathematics and Artificial Intelligence, 1999.
- [3] Kowalski, R. A., and Toni, F., Argument and reconciliation, Int. Symp. on 5th Generation Computer Systems, 1994.
- [4] Sierra, C., Jennings, N. R., Noriega, P. and Parsons S., A Framework for Argumentation-Based Negotiation. Proc. Fourth Int. Workshop on Agent Theories, Architectures and Languages, 1997.
- [5] Wooldridge, M., A logic of BDI agents with procedural knowledge, Proceedings of 2nd Workshop of the MODELAGE Project. Sesimbra, Portugal, 15-17th, 1996.