

시맨틱 웹서비스 조합을 위한 지식 전제조건 (Knowledge Preconditions for Composition of Semantic Web Services)

김 상 균 [†] 이 규 철 ^{**}
(Sang-Kyun Kim) (Kyu-Chul Lee)

요 약 행동 또는 계획이 인식적으로 실행 가능한지에 대한 지식 전제조건 문제는 그 동안 여러 연구들을 통해 정형화되어 왔다. 하지만 지금까지의 연구에서는 이러한 행동 및 계획이 실행 가능한지를 만드는 시점에서만 검사하며, 실제 수행될 때도 실행 가능할 것이라는 것을 가정하기 때문에, 시맨틱 웹에서 여러 에이전트들이 웹서비스 조합을 실행하기 위해 제한된 자원들을 공유해야 하는 분산 환경에서는 제대로 동작하지 않게 된다. 따라서 본 연구에서는 웹서비스 조합의 원자성을 보장할 수 있도록 트랜잭션적으로 수행 가능한 웹서비스 조합을 정의하고 이를 정형화한다. 이를 위해서 TL-ALCF에서 **K**와 **A** 모달 연산자를 추가하여 확장한 TL-ALCF_{KNF}를 제안함으로써 웹서비스 조합을 표현할 수 있는 언어인 TL-ALCF에서 인식 추론이 어떻게 수행될 수 있는지를 보인다.

키워드 : 웹서비스 조합, 지식 전제조건, 시맨틱 웹

Abstract Several researches have been proposed to formalize the knowledge preconditions problem - i.e., an action or a plan is epistemically feasible. However, since the feasibility is only checked at design-time and is assumed that it will also be feasible at run-time, it is not suitable in the context of Semantic Web services composition, where many agents should share the limited resources required for the execution of Web services composition. Therefore, in this paper, we formalize a transactionally executable Web services composition which enables to guarantee its atomicity. Moreover, in order to formalize the transactional executability, we propose TL-ALCF_{KNF} which extends TL-ALCF with the modal operators **K** and **A**. Based on TL-ALCF_{KNF}, we show how to carry out the epistemic reasoning with TL-ALCF as a language to represent Semantic Web services composition.

Key words : Web Services Composition, Knowledge Preconditions, Semantic Web

1. 서 론

최근 분산된 웹 어플리케이션을 구축할 수 있는 새로운 패러다임으로 웹서비스가 출현하였다. 하지만 인터넷 상에서 이용 가능한 웹서비스들의 수가 증가함에 따라 자신이 원하는 서비스를 직접 찾는 것이 점점 어려워지고 있기 때문에 각각의 서비스에 시맨틱을 기술함으로써 에이전트가 사람의 개입 없이 서비스를 자동으로 검색하고 실행할 수 있는 기술이 요구되고 있다. 이러한 비전을 실현하기 위해 최근 시맨틱 웹(Semantic Web)

이 출현하였다[1,2]. 시맨틱 웹에서는 컴퓨터가 이해할 수 있는 형태로 웹 자원을 기술하기 위해 OWL과 같은 언어를 제공하고 있으며, 또한 이러한 OWL에 기반을 두어 웹서비스 온톨로지들을 표현함으로써 웹서비스의 발견, 조합, 실행 등 다양한 작업을 자동화할 수 있도록 OWL-S가 제안되었다.

본 논문에서는 이러한 시맨틱 웹의 특징들 중에서 웹서비스 조합(Web Services Composition)과 관련된 이슈들을 다룬다. 그 중에서도 특히 행동(action) 또는 계획(plan)이 인식적으로 실행 가능한지(epistemic feasibility)[3] 즉, 에이전트가 계획을 수행하기에 충분히 알고 있는지에 대한 문제로 알려진 지식 전제조건(knowledge preconditions) 문제를 웹서비스 조합에 대해 적용한다.

인식적으로 실행 가능한 웹서비스 조합에 대해 추론

[†] 학생회원 : 충남대학교 컴퓨터공학과
ychnupo23@yahoo.co.kr

^{**} 종신회원 : 충남대학교 컴퓨터공학과 교수
kcleee@cnu.ac.kr

논문접수 : 2005년 4월 18일
심사완료 : 2005년 8월 11일

을 수행하기 위해서는 시간에 대한 추론(temporal reasoning)과 지식에 대한 추론(reasoning about knowledge)을 같이 이용해야 한다. 즉, 에이전트가 웹서비스 조합을 수행할 수 있을 만큼 충분히 알고 있는지를 결정하기 위해서는 웹서비스 조합이 시작할 때 무엇을 알고 있는지 그리고 웹서비스 조합을 수행함으로써 어떻게 월드 상태(world state)와 에이전트의 지식이 바뀌는지를 알 수 있어야 한다.

Artale과 Franconi는 최근 기술 로직(Description Logics)에 기반을 두어 시간, 행동, 계획을 표현하기 위해서 시간 간격에 기반한(interval-based temporal) 언어인 TL-ALCF를 제안하였다[4]. TL-ALCF에서 행동과 계획은 Allen의 시간 관계[5]에 따라 특정 시간 동안 나타나는 월드 상태로 표현하였으며, TL-ALCF의 포함(subsumption) 문제에 대해 사운드(sound)하고 완전(complete)한 추론 절차를 소개하고, 이 절차가 결정 가능(decidable)하다는 것을 보였다. 또한 Donini et al.은 프레임 기반 시스템(frame-based system)에서 단조롭지 않은(nonmonotonic) 특징들을 정형화하는 연구로써 ALCK_{NF}를 제안하였다[6]. 본 논문에서는 이 ALCK_{NF}를 이용하여 월드 상태와 에이전트의 지식을 표현할 수 있음을 보인다.

지금까지 계획에 대한 지식 전제조건 문제는 여러 연구를 통해 정형화되었는데, 특히 Davis[3]는 인식적으로 실행 가능한 결정적 또는 비결정적 계획들에 대해 정형화하였다. 하지만 Davis는 이러한 계획의 인식적 실행 가능성을 계획을 만들 때만(design-time) 검사하며, 한번 검사된 계획은 실행할 때도(run-time) 성공적으로 수행될 것이라는 것을 가정하였다. 그러나 웹서비스¹⁾는 주로 인터넷 분산 환경에서 수행되며 이러한 환경에서는 많은 에이전트들이 특정 웹서비스를 동시에 호출할 수 있기 때문에 만약 웹서비스 조합을 만들 때 실행 가능하다고 하더라도 웹서비스 조합을 실행하는 동안 다른 에이전트들이 웹서비스 조합을 구성하는 각각의 웹서비스에 대한 월드 상태를 바꾸게 되면 정작 실행할 때에는 실행 가능하지 않을 수 있다. 사실 인공지능 분야에서는 그 동안 조건적 계획(conditional planning) 또는 불확정 계획(contingency planning)에 대한 연구를 통해 이와 같은 문제를 해결하기 위해 노력해왔다. 이러한 조건적 또는 불확정 계획 시스템에서는 계획이 실행되는 도중에 나타나는 불확실성(uncertainty)을 해결하기 위해 계획의 목적을 이룰 수 있도록 보장한다기 보다는 부분적으로 성공한다 하더라도 가장 최선의 또

는 가장 높은 확률을 가지는 행동을 선택하도록 한다. 반면에 웹서비스 조합에서는 가장 최선의 방법보다는 일반적으로 웹서비스의 원자성을 중요시 여기기 때문에, 만약 일부만 수행되었을 경우 트랜잭션의 원자성(atomicity)을 보장하기 위해서 보정(compensation)과 같은 트랜잭션 관리가 필요하다.

이와 같이 기존의 계획 시스템에서는 중요하지 않게 고려해 왔던 트랜잭션 문제가 웹서비스 조합에서는 중요하기 때문에, 본 연구에서는 기존의 지식 전제조건 문제에 트랜잭션에 대한 개념을 추가하고 이를 정형화한다. 이를 위해 트랜잭션적으로 수행 가능한(transactionally executable) 웹서비스 조합을 정의함으로써 웹서비스 조합의 실행 가능성을 정형화하며, 이에 앞서 트랜잭션적으로 수행 가능한 웹서비스 조합을 정형화하기 위한 기술 로직 언어으로써 위에서 언급한 TL-ALCF와 ALCK_{NF}를 결합한 TL-ALCFK_{NF}를 제안한다. 사실 본 연구에서 웹서비스 조합의 원자성을 보존하는 방법은 그 동안 분산 트랜잭션에 대해 제안된 방법[9]과 비슷하다. 그러나 본 연구의 목적은 새로운 분산 트랜잭션 방법을 제안하는 것이 아니며, 기존 방법들을 기초로 하여 시맨틱 웹에서 자동화된 웹서비스 조합의 실행을 위해 지식을 어떻게 표현하고 추론하는지를 제안하는 데 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 기초가 된 TL-AKCF와 지식 전제조건 문제를 소개하며, 3장에서는 본 논문에서 사용하는 TL-ALCFK_{NF}를 제안한다. 또한 4장과 5장에서는 이 언어를 기반으로 실행 가능한 웹서비스와 웹서비스 조합을 정형화하며, 6장에서 관련 연구들과 본 연구를 비교하고, 마지막으로 7장에서 결론을 맺는다.

2. 기존 연구

2.1 TL-ALCF

최근 Artale과 Franconi는 시간 간격에 기반한 로직의 한 종류인 TL-ALCF를 제안하였으며, 또한 TL-ALCF의 포함 문제에 대해 사운드하고 완전한 절차를 소개하고 이 절차가 결정 가능하다는 것을 보였다. TL-ALCF는 크게 시간을 표현할 수 있는 시간 로직(temporal logic)인 TL과, 기능적 룰(functional role)을 나타내는 특징 로직(feature logic)을 ALC에 추가한 ALCF로 구성된다. 또한 이 연구에서는 행동과 계획을 시간 간격에 일어나는 것들로 정의하는데 즉, 행동은 해당 도메인에서 특정 시간에 가지고 있는 프로퍼티 집합인 월드 상태에 시간 제약(temporal constraint)을 부여한 것으로 표현하며, 계획은 시간적으로 관련된 행동과 상태(state)들로 표현한다. 이러한 TL-ALCF를 설명하

1) 본 논문에서는 특별한 의미의 구별이 필요 없는 한 액션이라는 용어 대신에 웹서비스라는 용어를 사용한다.

기 위해 다음과 같이 비행기 표를 예약하는 간단한 행동을 보자.

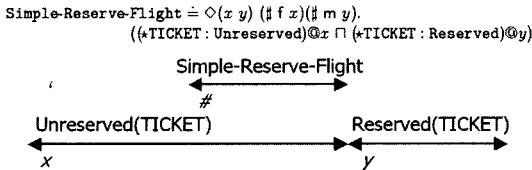


그림 1 공리에서 각 개념들의 시간적 의존 관계

그림 1은 Simple-Reserve-Flight 개념(concept)이 가지고 있는 서브 개념들의 시간적 의존 관계를 보인 것으로 Simple-Reserve-Flight는 특정 시간 간격 동안 Unreserved 상태에 있다가 Reserved 상태에 있게 되는 행동을 표현하며, 이 행동은 *TICKET이라는 파라미터를 가진다. 시간 변수들은 행동이 정의되는 시간을 표현하는 특별한 시간 변수인 # (now라고 부른다)를 제외하고, 시간 존재 한정자(temporal existential quantifier)인 "◇"에 함께 표현되며, (# f x)(# m y)와 같은 시간 제약들은 시간 간격 #가 시간 간격 x와 같이 끝나고, 시간 간격 #는 시간 간격 y와 만난다는 것을 나타낸다. 또한 파라미터 특징(parametric feature) *TICKET은 시간에 상관 없이 Simple-Reserve-Flight 타입의 개체를(individual) Simple-Reserve-Flight 행동에 매핑하는 파라미터의 역할을 한다. 이 외에도 어떤 개념이 특정 시간 간격에 일어난다는 것을 나타내기 위해서 bindable 이라고 부르는 @연산자를 이용하는데, 예를 들어 위 예제의 (*TICKET:Unreserved)@x와 (*TICKET:Reserved)@y에서는 *TICKET:Unreserved가 시간 x에 한정되고 *TICKET:Reserved가 시간 y에 한정된다는 것을 나타낸다.

2.2 지식 전제조건 문제

에이전트가 행동 또는 계획을 수행할 수 있는지에 대한 문제는 에이전트가 행동 또는 계획을 수행할 수 있을 정도로 충분히 이능지에 대한 인식적 실행 가능성과 행동 또는 계획을 물리적으로 수행할 수 있는지에 대한 물리적 실행 가능성(physical feasibility) 이렇게 크게 두 가지로 생각할 수 있다. 특히 행동 또는 계획의 인식적 실행 가능성에 대한 것은 지식 전제조건 문제라고 하여, 처음에 McCarthy와 Hayes[7]에 의해서 간단하게 언급되었으며 이후에 Moore[8]와 Davis[3]에 의해 다음과 같이 정형화되었다.

특히 Moore는 가능한 세계(possible world) 이론을 처음으로 적용해 행동의 인식적 실행 가능성을 다음과 같이 정형화하였다.

“만약 에이전트 A가 시간 T에 행동 E를 구성하는 특정 행동들을 안다면, E는 시간 T에 A에게 인식적으

로 실행 가능하게 된다.”

또한 Davis는 결정적 계획에 대한 Moore의 연구를 확장하여 비결정적 계획에 대한 지식 전제조건 문제를 연구하였는데, 특히 Davis는 비결정적 계획의 경우를 고려하기 위해 계획이 인식적으로 실행 가능이라는 용어 대신에 계획이 수행 가능(executable)이라는 용어를 사용하였다. 다음은 이러한 Davis의 수행 가능한 계획에 대한 정의이다.

계획 P는 만약 다음과 같은 조건들을 만족하면 시간 T에 에이전트 A에게 수행 가능하다.

1. P는 시간 T에 시작해 수행된 후에 종료한다.
2. P가 시간 T에 시작한 후에
 - A. A는 P가 성공적으로 끝날 것인지를 안다.
 - B. A는 모든 행동들이 P의 다음 단계(next step) 인지 아닌지를 안다.
 - C. 모든 P의 다음 단계는 실행 가능하다.

2.3 MKNF-DLs

인공지능 분야에서는 최근 프레임 기반 시스템 또는 지식 기반 어플리케이션의 여러 가지 단조롭지 않은 특징들을 정형화하는 노력이 수행되어 왔다. 이러한 특징들에는 절차 규칙(procedure rules), 디폴트 규칙(defaults rules), 인식 질의(epistemic queries), 무결성 제약(integrity constraints), 롤 폐쇄(role closure), 개념 폐쇄(concept closure)등이 포함되는데, 이들은 프레임 기반 시스템의 주요 기능들 중의 하나로써, 특정 도메인에 대한 지식을 표현하고 추론할 수 있도록 한다. 하지만 이들은 일반적인 기술 언어와 달리 1차 로직을 사용하여 정형화 할 수 없기 때문에, 기존 기술 언어의 시맨틱 외에 확장된 방법이 필요했다.

이를 위해 기존에는 기술 언어에 K 모달 연산자²⁾를 추가하여 확장함으로써, 인식 질의와 절차 규칙을 표현할 수 있도록 하였으나 그 외의 다른 단조롭지 않은 특징들을 다루지 못했다. 이에 최근 Donini et al.[6]는 기술 언어에서 단조롭지 않은 특징들을 다루기 위한 새로운 프레임워크인 MKNF-DLs(Description logics of minimal knowledge and negation as failure)을 제안했으며, 이 프레임워크에서는 기존의 K 모달 연산자 외에 새로운 A 모달 연산자³⁾를 추가함으로써 위의 모든 단조롭지 않은 특징들을 다룰 수 있도록 하였다. 또한 이 프레임워크에서 사용되는 두 모달 연산자를 ALC 기술 언어에 추가함으로써 ALCK_{NF} 라는 기술 언어를 만들었으며, 이를 이용해 실제 프레임 기반 시스템에서 단조롭지 않은 특징들이 어떻게 표현되는지를 보였다.

2) 최소 지식 연산자(minimal knowledge operator)라고도 함.
3) 자동 인식 연산자(autoepistemic operator)라고도 함.

3. TL-ALCFK_{NF}

3.1 TL-ALCFK_{NF}의 구분과 시맨틱

TL-ALCFK_{NF}는 TL-ALCF에 **K**와 **A** 모달 연산자를 추가하여 확장한 것으로써, 웹서비스 조합을 표현할 수 있는 TL-ALCF에서 인식 추론이 가능하게 한다. 다음은 이러한 TL-ALCFK_{NF}의 구분을 보인 것이다.

TL	$E, F \rightarrow C$	(비시간 개념)	
	$E \cap F$	(합집)	
	$E \circ X$	(한정자)	
	$E[Y] \circ X$	(대체 한정자)	
	$\diamond(X) \overline{E} E$	(존재 한정자)	
	$T_e \rightarrow (X(U)Y) (X(U) \text{!}) (\text{!}(U)Y)$	(시간 제약조건)	
	$\overline{T}_e \rightarrow T_e \overline{T}_e \overline{T}_e$		
	$U, V \rightarrow U, V$	(이점)	
	$b a m mi oi oi s si$	(Allen의 관계)	
	$d di f fi =$		
	$X, Y \rightarrow x y z \dots$	(시간 변수)	
	$\overline{X} \rightarrow X \overline{X}$		
ACCFK _{NF}	$C, D \rightarrow A$	(원자 개념)	
	$T \perp \neg C \cap D C \cup D$		
	$\forall RC \exists RC$		
	$p \downarrow q$	(동의)	
	$p \uparrow q$	(비동의)	
	$p \parallel$	(미정의)	
	$p : C$	(선택)	
	$KC AC$		
	$R \rightarrow P KP AP$	(롤)	
	$p, q \rightarrow f$	(원자 특징)	
	$*g$	(원자 파라미터 특징)	
	$p \circ q$	(패스)	
		$Kp Ap$	

그림 2 TL-ALCFK_{NF}의 구분

TL-ALCFK_{NF}의 구분은 **K**와 **A** 모달 연산자를 가지는 표현식을 제외하고는 TL-ALCF와 똑같으며 이 모달 연산자는 위 그림에서 보이는 것과 같이 비시간(non-temporal) 개념인 C와 D, 원자 롤(atomic roles)인 P, 패스(path)인 p, q와 함께 사용된다.

TL-ALCFK_{NF}의 시맨틱⁴⁾은 개념, 롤, 특징을 시간 해석(temporal interpretation)과 인식 해석(epistemic interpretation)에 따라 해석함으로써 얻을 수 있다. 즉, TL-ALCFK_{NF}에서의 해석은 I, M, N 이렇게 세 가지로 표현하는데, 여기에서 I는 TL-ALCF에서의 시간 해석인 $I \equiv \langle \Delta, \overset{I}{V}, I_H \rangle$ 이며, M, N은 도메인 Δ 에서 정의되는 시간 해석 I의 집합이 된다. 다음 그림 3은 이러한 TL-ALCFK_{NF}에서의 시맨틱을 보인 것이다.

TL-ALCFK_{NF}의 시맨틱에서 개체(individual) $d \in \Delta$ 는 KC (즉, $d \in (KC) \overset{I, M, N}{I}$) 개념의 인스턴스가 되는데, 여기서 C는 M의 모든 해석에서 $d \in C \overset{I, M, N}{I}$ 을 만족하면, 시간 간격 t에 C타입을 가지는 엔터티들의 집합이

$$\begin{aligned}
 \overset{I, M, N}{V}_{t, \mathcal{H}} &= \Delta \\
 \overset{I, M, N}{I}_{t, \mathcal{H}} &= \emptyset \\
 \overset{A, T, M, N}{A}_{t, \mathcal{H}} &= \{a \in \Delta \mid (t, a) \in A \overset{I, M, N}{I}\} = A_t^I \\
 \overset{F, T, M, N}{F}_{t, \mathcal{H}} &= \hat{F}_t \subseteq \Delta \times \Delta \mid \forall a, b, (a, b) \in \hat{F}_t \leftrightarrow (t, a, b) \in F \overset{I, M, N}{I} = F_t^I \\
 (\neg C) \overset{I, M, N}{V}_{t, \mathcal{H}} &= \Delta \setminus C_t \overset{I, M, N}{V} \\
 (E \circ X) \overset{I, M, N}{V}_{t, \mathcal{H}} &= E \overset{I, M, N}{V}_{t, \mathcal{H}}(X)_{t, \mathcal{H}} \\
 (E[Y] \circ X) \overset{I, M, N}{V}_{t, \mathcal{H}} &= E \overset{I, M, N}{V}_{t, \mathcal{H}}(Y \cup \{y \rightarrow y(X)\}) \\
 (\diamond(X) \overline{E} E) \overset{I, M, N}{V}_{t, \mathcal{H}} &= \{a \in \Delta \mid \exists W, W \in (\overline{X}, \overline{E}) \overset{I, M, N}{V}_{t, \mathcal{H}} \cup \{t \rightarrow t\} \wedge a \in E \overset{I, M, N}{V}_{t, \mathcal{H}}(W)\} \\
 (E \cap F) \overset{I, M, N}{V}_{t, \mathcal{H}} &= E \overset{I, M, N}{V}_{t, \mathcal{H}} \cap F \overset{I, M, N}{V}_{t, \mathcal{H}} \\
 (C \cap D) \overset{I, M, N}{V}_{t, \mathcal{H}} &= C_t \overset{I, M, N}{V} \cap D_t \overset{I, M, N}{V} \\
 (C \cup D) \overset{I, M, N}{V}_{t, \mathcal{H}} &= C_t \overset{I, M, N}{V} \cup D_t \overset{I, M, N}{V} \\
 (\exists RC) \overset{I, M, N}{V}_{t, \mathcal{H}} &= \{a \in \Delta \mid \exists b, (a, b) \in R_t \overset{I, M, N}{V} \wedge b \in C_t \overset{I, M, N}{V}\} \\
 (\forall RC) \overset{I, M, N}{V}_{t, \mathcal{H}} &= \{a \in \Delta \mid \forall b, (a, b) \in R_t \overset{I, M, N}{V} \rightarrow b \in C_t \overset{I, M, N}{V}\} \\
 (p \downarrow q) \overset{I, M, N}{V}_{t, \mathcal{H}} &= \{d \in \text{dom}_{p_t \overset{I, M, N}{V}} \cap \text{dom}_{q_t \overset{I, M, N}{V}} \mid p_t \overset{I, M, N}{V}(a) = q_t \overset{I, M, N}{V}(a)\} \\
 &= (p \downarrow q) \overset{I, M, N}{V}_t \\
 (p \uparrow q) \overset{I, M, N}{V}_{t, \mathcal{H}} &= \{d \in \text{dom}_{p_t \overset{I, M, N}{V}} \cap \text{dom}_{q_t \overset{I, M, N}{V}} \mid p_t \overset{I, M, N}{V}(a) \neq q_t \overset{I, M, N}{V}(a)\} \\
 &= (p \uparrow q) \overset{I, M, N}{V}_t \\
 (p \parallel) \overset{I, M, N}{V}_{t, \mathcal{H}} &= \Delta \setminus \text{dom}_{p_t \overset{I, M, N}{V}} \\
 (p : C) \overset{I, M, N}{V}_{t, \mathcal{H}} &= \{d \in \text{dom}_{p_t \overset{I, M, N}{V}} \mid p_t \overset{I, M, N}{V}(a) \in C_t \overset{I, M, N}{V}\} \\
 f_t \overset{I, M, N}{V} &= f_t : \Delta \xrightarrow{\text{partial}} \Delta \mid \forall a, (a \in \text{dom}_{f_t} \leftrightarrow (t, a) \in \text{dom}_{f \overset{I, M, N}{I}}) \\
 &\quad \wedge f_t(a) = f \overset{I, M, N}{I}(t, a) \\
 (p \circ q) \overset{I, M, N}{V}_t &= p_t \overset{I, M, N}{V} \circ q_t \overset{I, M, N}{V} \\
 *g_t \overset{I, M, N}{V} &= *g \overset{I, M, N}{V}_t \\
 (KC) \overset{I, M, N}{V}_{t, \mathcal{H}} &= \bigcap_{J \in M} C_t \overset{I, M, N}{V} \\
 (AC) \overset{I, M, N}{V}_{t, \mathcal{H}} &= \bigcap_{J \in N} C_t \overset{I, M, N}{V} \\
 (KP) \overset{I, M, N}{V}_{t, \mathcal{H}} &= \bigcap_{J \in M} P_t \overset{I, M, N}{V} \\
 (AP) \overset{I, M, N}{V}_{t, \mathcal{H}} &= \bigcap_{J \in N} P_t \overset{I, M, N}{V} \\
 (Kp) \overset{I, M, N}{V}_t &= \bigcap_{J \in M} p_t \overset{I, M, N}{V} \\
 (Ap) \overset{I, M, N}{V}_t &= \bigcap_{J \in N} p_t \overset{I, M, N}{V}
 \end{aligned}$$

그림 3 TL-ALCFK_{NF}의 시맨틱

된다. 다시 말해서 만약 어떤 개체가 모든 가능한 세계 M의 개념 해석에 속한다면 이 개체는 시간 간격 t에 개념 C의 인스턴스가 된다고 알게(“known”)되는 것이다.

또한 개체 $d \in \Delta$ 는 AC (즉, $d \in (AC) \overset{I, M, N}{I}$) 개념의 인스턴스가 되는데, 여기서 N의 모든 해석에서 $d \in C \overset{I, M, N}{I}$ 을 만족하면, 시간 간격 t에 C타입을 가지는 엔터티들의 집합이 된다. 다시 말해서 만약 어떤 개체가 모든 가능한 세계 N의 개념 해석에 속한다면 이 개체는 시간 간격 t에 개념 C의 인스턴스가 된다고 가정(“assumed”)되는 것이다. 이러한 해석과 비슷한 방법으로 롤 P와 패스 p도 해석될 수 있다. 또한 인식 해석은 시간과 무관하기 때문에, 시간 부분과 비시간 부분을 구별하여 해석하는 Artale과 Franconi의 방법에 따라 시간 한정자(temporal quantifier) 또는 시간 대체 한정자(temporal substitutive quantifier)를 인식 해석에서 사용할 수 없도록 한다.

하나의 해석 (I, M, N)은 모든 해석 (I, M, N)과 모든 시

4) 위의 그림 3에서 **K**와 **A** 모달 연산자를 가지는 6개의 표현식을 제외한 나머지 표현식은 TL-ALCF의 시맨틱과 똑같기 때문에 본 절에서는 이 6개의 표현식에 대해서만 설명한다.

간 간격 t에 대해서 만약 $A_t^{I,M,N} = C_t^{I,M,N}$ 이면, 공리 $A \equiv C$ 를 만족시킬 수 있으며, 모든 해석 (I,M,N)과 모든 시간 간격 t에 대해서 만약 $C_t^{I,M,N} \subseteq D_t^{I,M,N}$ 이면, 개념 C는 개념 D를 포함(subsume)하게 된다($C \sqsubseteq D$). 또한 해석 (I,M,N)은 어떤 시간 t에 만약 $C_t^{I,M,N} \neq \emptyset$ 이면, 개념 C에 대한 모델이 되며, 만약 어떤 개념이 모델을 가지면 만족 가능(satisfiable)하다고 하고, 모델을 가지지 않으면 만족 가능하지 않는다(unsatisfiable)고 한다.

어서션(assertion)들의 시멘틱은 집합 멤버십(set membership)을 이용해 다음과 같이 해석된다. 즉, $C(i,a)$ 는 만약 $a^{I,M,N} \in C_t^{I,M,N}$ 이면 해석 (I,M,N)에 의해서 만족되며, $p(i,a,b)$ 는 만약 $D_t^{I,M,N}(a^{I,M,N}) = b^{I,M,N}$ 이면 (I,M,N)에 의해서 만족된다. 또한 $*g(a,b)$ 는 만약 $*g^{I,M,N}(a^{I,M,N}) = b^{I,M,N}$ 이면 (I,M,N)에 의해서 만족되고, $R(i,j)$ 는 만약 $\langle i^t, j^t \rangle \in R^t$ 이면 (I,M,N)에 의해서 만족된다.

3.2 TL-ALCFK_{NF} 예제

이 절에서는 간단한 웹서비스 예제를 통해 TL-ALCF에서 모달 연산자를 사용함으로써 (즉, TL-ALCFK_{NF}를 사용하여) 어떻게 에이전트의 지식을 표현할 수 있는지를 보인다. 이를 위해 우선 다음과 같은 지식 베이스(Knowledge Base) $\Sigma = \langle T, A \rangle$ 가 존재한다는 것을 가정한다.

지식 베이스에서 TBox T는 하나의 공리를 가지며 이 공리는 어느 시간 동안 두 *TICKET이 예약 가능한 상태로 있다가 예약이 수행되는 행동을 나타낸다. 이 예제에서는 이 공리가 어떤 비행기의 티켓 두 장을 예약하는 간단한 웹서비스를 표현한다고 가정한다.

$$\text{Reserve-Flight} \doteq \Diamond(x y u v) (\# f x) (\# m y) (\# f u) (\# m v). \\ ((*TICKET : \text{Unreserved})@x \sqcap (*TICKET : \text{Reserved})@y \sqcap \\ (*TICKET : \text{Unreserved})@u \sqcap (*TICKET : \text{Reserved})@v)$$

이러한 TBox에 대한 어서션들인 ABox A는 티켓들이 예약 가능한 상태로 있다가 예약이 수행되는 상태를 기술하게 되며, 이 어서션에서 개체 a는 ticket₁과 ticket₂를 파라미터로 가지고 시간 i_a동안 발생하는 웹서비스 인스턴스를 나타낸다.

$$*TICKET(a, ticket_1), \text{Unreserved}(i_1, ticket_1), f(i_1, i_a), *TICKET : \text{Reserved}(a), \\ *TICKET(a, ticket_2), \text{Unreserved}(i_2, ticket_2), f(i_2, i_a), f(i_3, i_a)$$

위와 같은 TBox의 공리에서 각 개념들 간의 시간 관계는 다음 그림과 같은 형태로 표현된다.

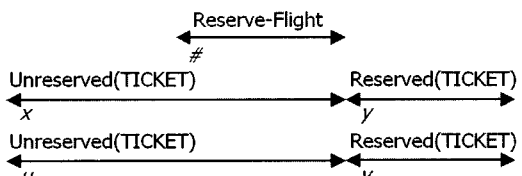


그림 4 공리에서 각 개념들 간의 시간 관계

위의 지식 베이스에서 TBox와 어서션들 중 *TICKET (a, ticket₁), f(i₁, i_a), Unreserved(i₁, ticket₁)을 보면, 행동 a의 파라미터인 ticket₁이 시간 i₁동안 예약이 되지 않고 있다는 것을 알 수 있다. 또한 *TICKET: Reserved(a), m(i₃, i_a)을 보면, 명확히 무엇인지는 모르지만 (이 개체가 ABox에서 언급되지 않은 것일 수도 있다.) 적어도 한 장의 예약된 티켓이 존재한다는 것을 알 수 있다.⁵⁾ 따라서 만약 이 예제에서 티켓이 ABox에 명확하게 나타나도록 하려면 Abox에 있는 어서션들의 가능한 상태에 대해 제약조건들을 표현할 수 있어야 한다.

또한 위 예제가 여행을 예약하는 어떤 웹서비스 조합에서 비행기 티켓을 구매해 주는 하나의 웹서비스이며, 인터넷 분산 환경에서 여러 에이전트들이 이 웹서비스 조합을 수행한다고 가정하자. 이러한 상황에서는 여러 에이전트들이 시간 i_a동안 ticket₁과 ticket₂를 예약하려고 할 것이기 때문에, 어떤 에이전트가 이 두 티켓이 예약될 수 있을 것이라는 것을 웹서비스 조합을 수행하기 전에 알고 있다 하더라도, Reserve-Flight를 실제 수행하기 전에 다른 에이전트들이 만약 이 두 티켓을 예약했다면 이 에이전트는 티켓을 예약할 수 없게 되며 결국 이 웹서비스는 실패하게 된다. 따라서, 이러한 문제를 해결하기 위해서는 Reserve-Flight를 실행하기 전에 티켓들이 존재하는지 검사(check)해야 한다. 본 연구에서는 이러한 문제를 해결하기 위해서 다음과 같이 자동 인식 연산자 A를 사용한다.

$$\mathbf{KReserve-Flight} \doteq \Diamond(x y u v) (\# f x) (\# m y) (\# f u) (\# m v). \\ ((*A * TICKET : \text{AUnreserved})@x \sqcap (*A * TICKET : \text{AReserved})@y \sqcap \\ (*A * TICKET : \text{AUnreserved})@u \sqcap (*A * TICKET : \text{AReserved})@v)$$

위의 공리로부터 우리는 모든 가능한 세계에서 Reserve-Flight의 파라미터로 알려진(known) 티켓들이 시간 i₁과 i₂동안 예약이 되지 않고 있다는 것을 알고, 시간 i₃에 예약된 상태로 된 것을 알고 있는지 검사(check)할 수 있다. 따라서 이 공리를 통해 에이전트는 Reserve-Flight가 인식적으로 실행 가능한지 알 수 있게 된다. 이와 같이 본 논문에서는 에이전트의 지식을 표현하기 위해 기존 연구에서 많이 사용해왔던 최소 지식 연산자 K가 아니라 자동 인식 연산자 A를 사용하는데, K를 사용하게 되면 K의 의미가 알려진 티켓이 특정 시간 동안 예약 안 된 상태로 있다가 예약되는 상태를 검사한다기 보다는 강요(force)하는 의미를 가지기 때문에 올바른 방법이 될 수 없기 때문이다.

3.3 TL-ALCFK_{NF}에서의 추론

이 절에서는 TL-ALCFK_{NF}에서 추론을 어떻게 수행하는지를 보인다. Artale과 Franconi는 기술 로직 부분

5) 사실 만약 *TICKET:Reserved(a), f(i₃, i_a) 대신에 Reserved(i₃, ticket₁), m(i₃, i_a), Reserved(i₄, ticket₁), m(i₄, i_a)와 같은 어서션이 존재한다면 예약된 티켓이 ticket₁과 ticket₂라는 것을 알 수 있다.

(ALCF)[13]에 대한 추론과 시간 부분(TL)에 대한 추론을 각각 분리함으로써 TL-ALCF의 포함 문제가 비시간 부분인 ALCF의 포함 문제로 축소될 수 있다는 것을 보였다. 따라서 TL-ALCFK_{NF}에서의 포함 문제도 ALCFK_{NF}의 포함 문제로 축소될 수 있다는 것을 알 수 있다. 또한 Donini et al.은 ALCK_{NF} 지식 베이스에 제약을 주어 만든 지식베이스를 주체적으로 한정된 (subjectively quantified) ALCK_{NF} 지식 베이스라고 정의하였으며, 이 지식 베이스가 ALCK_{NF}의 각 모델 M에 대한 특성을 기술하기 위한 필요충분조건을 만족한다는 것을 보였다. 그리고 주체적으로 한정된 ALCK_{NF} 지식 베이스에서 한정된 개수의 모델의 개수를 가지도록 제약을 주어 만든 지식베이스를 단순(simple) ALCK_{NF} 지식 베이스라고 정의하였으며, 이 지식 베이스를 통해 결정 가능한 추론을 수행할 수 있음을 보였다.

이러한 특징들에 따르면 TL-ALCFK_{NF}에서의 추론은 ALCK_{NF}에서의 추론에 특징("F" : feature) 해석을 추가함으로써 수행할 수 있다는 것을 알 수 있다. 따라서 본 논문에서는 TL-ALCFK_{NF}에 대한 일반적인 추론 방법을 모두 보이지는 않으며, 단지 TL-ALCK_{NF}에서 어떻게 특징 해석이 추가될 수 있는지를 보인다. 이를 위해 우선 주체적으로 한정된 ALCFK_{NF} 지식 베이스를 정의하기 위해서 주체적으로 한정된 ALCK_{NF}의 정의에서 선택(selection) 연산자('·')를 다음과 같이 추가한다.

정의 1. 주체적으로 한정된 ALCFK_{NF} 지식 베이스는 ALCFK_{NF} Σ 의 포함문(inclusion), 정의문(definition) 또는 인스턴스 어서션에서 나타나는 각 개념 표현식 C가 $\exists R.D, \forall R.D, p:D$ 의 형태를 가지는 한정된 서브표현식(quantified subexpression)이며, 이 한정된 서브표현식이 다음과 같은 조건을 만족하는 ALCFK_{NF} Σ 이다.

- (1) R은 원자 롤이며 p는 패스이고 D는 ALCF 개념이다.
- (2) R은 KR' 또는 AR'형태의 인식 롤이며, p는 Kp' 또는 Ap'형태의 인식 패스이고 D는 KD', $\neg KD'$, AD', $\neg AD'$ 형태 중에 하나이다.

또한 단순 ALCFK_{NF} 지식 베이스는 단순 ALCK_{NF} 지식 베이스의 정의에서 ALC 개념 대신에 ALCF 개념으로 바꿈으로써 간단하게 재정의될 수 있다.

정의 2. 단순 ALCFK_{NF} 지식 베이스 (T, A)는 다음과 같은 조건들을 만족하는 ALCFK_{NF} 지식 베이스이다. (T는 (T', Γ)의 쌍으로 구성된다.)

- (1) T'는 ALCF TBox이다.
- (2) Γ 는 ALCFK_{NF} 단순 포함문 (정의문을 포함)의 집합이다. 다시 말해서, Γ 는 $KC \sqsubseteq D$ ($KA \sqsubseteq D$ 의 형태를 포함) 형태의 포함문 어서션의 집합으로, 여기에서 C는 $T' \neq T \in C$ 를 만족하는 ALCF 개념이며, A는 원자 개념이고, D는 한정자의 범위 안에 K 연

산자가 나오지 않는 주체적으로 한정된 개념 표현식이다.

- (3) A는 단순 인스턴스 어서션들의 집합으로써, A에서 나타나는 모든 개념 서브표현식들은 T의 타입을 가진다.

이와 같은 정의에 따르면 3.2절의 예제에서 보인 지식 베이스는 하나의 정의문을 가지며 정의문의 한정된 서브표현식은 Ap'형태의 인식 패스와 AD'형태의 ALCF 개념이 존재하기 때문에 주체적으로 한정된 지식 베이스인 것을 알 수 있으며, 또한 이 정의문은 $KA \sqsubseteq D$ 의 형태를 가지고 개념 D는 K연산자를 포함하지 않으며 모든 A에 나타나는 모든 인스턴스 어서션들이 단순하므로 단순 ALCFK_{NF} 지식 베이스인 것을 알 수 있다. 따라서 이 예제는 단순 ALCFK_{NF}가 된다.

이제 이러한 단순 ALCFK_{NF} 지식 베이스에서 특징에 대한 추론을 수행하는 테이블로우 계산법(tableau calculus)을 보이도록 한다. 이 테이블로우 계산법의 기본 아이디어는 ALCK_{NF}의 방법에 따르며, 이 ALCK_{NF}에 대한 테이블로우 계산법에 특징 로직에 대한 부분을 추가하여 ALCFK_{NF}에 대한 테이블로우 계산법을 얻을 수 있다.

일반적으로 테이블로우 트리의 한 가지(branch) B는 $\langle w|C(x) \rangle$ 형태의 접두 멤버십(prefixed membership) 어서션들의 집합으로, w는 월드(world)를 나타내며 이는 자연수로 표시된다. 단순 ALCFK_{NF} 지식 베이스 $\Sigma = \langle T, A \rangle$ (T는 (T', Γ)의 쌍)에 대한 테이블로우의 경우에는 $\{ \langle 1|K\varphi \rangle, \varphi \in A \}$ 와 같은 접두식(prefixed formula)의 집합을 가지고 시작하는데, 여기에서 O_B 는 B에서 언급된 개체들의 집합으로 표시된다. 또한 Σ 에 있는 어서션들은 부정 정규형(Negation Normal Form)이라는 것을 가정하며, M은 $M \in \{K, A\}$ 와 같은 모달 연산자, 그리고 N은 $N \in \{K, A, \neg K, \neg A\}$ 와 같이 부정(negation)을 포함하는 모달 연산자를 나타낸다. 이러한 표기 방법에 따라 구성된 테이블로우 계산법은 다음과 같다.

선택-규칙 : 만약 $\langle w|Mf:NC(x) \rangle \in B$ 이고, $\langle 1|Mf(x, y) \rangle \in B$ 와 $\langle 1|NC(y) \rangle \in B$ 와 같은 y가 없다면, $z \in O_B \cup \{ \iota \}$ 이고 $\iota \notin O_B$ 인 어떤 z에 대해서 $\langle 1|Mf(x, z) \rangle$ 와 $\langle 1|NC(z) \rangle$ 를 B에 추가한다.

↓-규칙 : 만약 $\langle w|Mp \equiv Mq(x) \rangle \in B$ 이고, $\langle 1|Mp(x, y) \rangle \in B$ 와 $\langle 1|Mq(x, y) \rangle \in B$ 와 같은 y가 없다면, $z \in O_B \cup \{ \iota \}$ 이고 $\iota \notin O_B$ 인 어떤 z에 대해서 $\langle 1|Mp(x, z) \rangle$ 와 $\langle 1|Mq(x, z) \rangle$ 를 B에 추가한다.

↑-규칙 : 만약 $\langle w|Mp \neq Mq(x) \rangle \in B$ 이고, $\langle 1|Mp(x, y) \rangle \in B$, $\langle 1|Mq(x, z) \rangle \in B$, $y \neq z \in B$ 와 같은 y'와 z'가 없다면, $y \in O_B \cup \{ \iota \}$, $z \in O_B \cup \{ \iota \}$, $\iota \notin O_B$ 인 어떤 y와 z에

대해서 $\langle \text{IM}p(x,y) \rangle$ 와 $\langle \text{IM}q(x,z) \rangle$ 를 B 에 추가한다.

패스-규칙 : 만약 $\langle w|\mathbf{M}f\mathbf{M}p(x) \rangle \in B$, $\mathbf{M}p \neq \epsilon$ 이고 $\langle \text{IM}f(x,y') \rangle \in B$ 와 $\langle \text{IM}p(y',z) \rangle \in B$ 와 같은 y' 가 없다면, $y \in O_B \cup \{ \iota \}$ 이고 $\iota \notin O_B$ 인 어떤 y 에 대해서 $\langle \text{IM}f(x,y) \rangle$ 와 $\langle \text{IM}p(y,z) \rangle$ 를 B 에 추가한다.

합수-규칙 : 만약 $\langle w|\mathbf{M}f(x,y) \rangle \in B$, $\langle w|\mathbf{M}f(x,z) \rangle \in B$, $y \neq z$ 이면 B 에서 y 를 z 로 바꾼다.

위의 규칙들은 기존에 재한된 ALCF의 완전한 특징(feature completion) 규칙을 바탕으로 인식 모델을 처리할 수 있도록 수정된 것으로써, 직관적으로 알 수 있듯이 이 규칙들은 기존의 ALCK_{NF}의 테이블로우 계산법에 영향을 미치지 않기 때문에 ALCK_{NF}에 대한 테이블로우 계산법에 추가하여 ALCFK_{NF}에 대한 테이블로우 계산법을 구성할 수 있다. 또한 지식베이스에서 이러한 ALCFK_{NF}에 대한 테이블로우 계산법의 규칙을 모두 적용한 후에 만들어진 것들이 유한하다면, ALCFK_{NF} 개념이 만족 가능(satisfiable)한지 아닌지를 결정할 수 있게 된다. 다음은 이러한 특징들에 대한 정리들이다.

정리 1. $\Sigma = \langle T, A \rangle$ 를 단순 ALCFK_{NF} 지식 베이스라고 했을 때, 만약 Σ 에 대한 ALCFK_{NF}의 최선의 가치 B 가 존재한다면 Σ 는 만족 가능하다.

증명. 만약 S 를 단순 ALCFK_{NF}의 지식 베이스에 테이블로우 계산법을 적용하여 얻어진 것들의 집합이라 가정한다. 만약 S 에서 최선의 가치가 존재하지 않는다면 분명히 만족하지 않는다.

또한 S 가 완전한 집합이라면, 즉 모든 테이블로우 계산법을 적용하여 더 이상 적용할 것이 없다면, S 에는 모든 변수 x 와 모든 특징 f 에 대해서 xfy 와 같은 변수 y 가 기껏해야 하나 존재하게 된다. 따라서 이러한 S 에 존재하는 모든 변수를 해석할 수 있는 최선의 가치가 존재하게 되며 ALCFK_{NF}의 지식 베이스가 만족 가능함을 알 수 있다.

정리 2. $\Sigma = \langle T, A \rangle$ 를 단순 ALCFK_{NF} 지식 베이스라고 하면 Σ 의 만족 가능성을 입증하는 문제는 결정 가능하다.

증명. ALCFK_{NF} 테이블로우 계산법을 적용하여 만들어진 집합에서 개체가 새로 생성되는 경우는 선택-규칙, ↓-규칙, ↑-규칙, 패스-규칙을 적용할 때이다. 따라서 $\mathbf{M}f\mathbf{N}C(x)$, $\mathbf{M}p \neq \mathbf{M}q(x)$, $\mathbf{M}p \neq \mathbf{M}q(x)$, $\mathbf{M}f\mathbf{M}p(x)$, $\mathbf{M}P\mathbf{N}C(x)$ 와 같은 형태의 표현식이 발생하는 개수를 $Q(A)$ 라고 하고(단, $\mathbf{M}p \neq \mathbf{M}q(x)$ 가 나타나는 경우 $Q(A)=Q(A)+1$ 한다), O_Σ 를 ABox에 명시적으로 나타난 개체의 집합이라고 하면, 새로 생성되는 개체의 최대 숫자는 $2^{Q(A)} + O_\Sigma$ 와 같이 지수적으로 증가하게 되지만 표현식의 개수가 한정되어 있기 때문에 결국 테이블로우 계산법은 유한

한 시간내에 끝날 수 있게 된다.

사실 Hollunder와 Donini et al.의 논문에서 따르면 ALCF 개념과 단순 ALCK_{NF} 개념 모두 만족 가능한지 아닌지를 결정할 수 있다고 하였기 때문에, 이 언어를 조합한 단순 ALCFK_{NF} 또한 결정 가능하다는 것을 쉽게 알 수 있다. 이와 같이 단순 ALCK_{NF} 개념이 결정 가능하다는 것은 실제 이 개념을 사용해 구축된 지식 베이스에서 사용자들이 이 지식 베이스에 대한 질의를 했을 때 질의에 대한 응답을 줄 수 있음을 의미하며,⁶⁾ 따라서 다음 절에서 언급할 웹서비스와 웹서비스 조합의 실행 가능성을 결정할 수 있게 된다.

4. 웹서비스⁶⁾

Artale과 Franconi는 시간, 행동, 계획을 표현할 수 있는 정형화된 방법을 제안했지만, 이 방법에서는 행동과 계획이 시작할 때 알고 있는 것, 즉 전제조건(precondition)과 행동 또는 계획을 실행한 결과로 인해 바뀐 월드 상태와 에이전트의 지식, 즉 효과(effect)에 대해서는 명확하게 명시하지 않았다. 하지만 일반적으로 웹서비스 조합에서는 웹서비스 조합을 구성하는 웹서비스들이 각각 자치성을 가지며 분산 트랜잭션에 따라 수행되기 때문에 이전에 수행된 웹서비스가 실패하면 현재 수행될 웹서비스가 수행될 수 없는 것과 같이 상호간에 영향을 미치게 된다. 이와 같은 이유로 웹서비스 조합에서는 일반적인 계획 시스템보다 전제조건과 효과가 중요하게 된다.

따라서 다음 절에서 웹서비스 조합에 대해 설명하기 전에, 이 절에서는 웹서비스의 전제조건과 효과, 그리고 웹서비스의 실행 가능성에 대한 정의를 명확히 한다. 특히 본 논문에서 수행되는 웹서비스는 일정 시간 간격(finite time interval) $\# = [t_1, t_2]$ 에서 원자 단위로 수행된다는 것을 가정한다.

정의 3. 에이전트가 시간 $\# = [t_1, t_2]$ 동안 수행하는 웹서비스 WS의 전제조건은 시간 t_1 에 에이전트가 알고 있는 월드 상태로 표현한다.

명제 1. 단순 TL-ALCFK_{NF} 지식 베이스에서 시간 $\# = [t_1, t_2]$ 동안 수행되는 웹서비스 WS가 있을 때, 에이전트가 시간 t_1 에 WS에 대해 알고 있는 월드 상태는 다음과 같은 Allen의 시간 제약을 가지는 단순 TL-ALCFK_{NF} 개념과 어서션들이다.

$$\begin{aligned} Tc &\rightarrow (X(U) \#) \mid (\#(V) X) \\ U &\rightarrow fi \mid di \mid m \mid o \mid s \mid si \mid = \\ V &\rightarrow f \mid d \mid mi \mid oi \mid si \mid s \mid = \end{aligned}$$

6) 결정 가능하지 않은 지식베이스에서는 질의의 결과를 결정할 수 없기 때문에 정확한 응답을 줄 수 없게 된다.

이와 같이 TL-ALCFK_{NF}에서 웹서비스의 전제조건은 시간 t1에 이 웹서비스의 월드 상태들과 최소한 부분적으로 겹치는 월드 상태로 표현될 수 있다. 예를 들어 3.2절에서 보인 예제의 전제조건은 다음과 같이 표현될 수 있다.

$\diamond(x\ u)(\#f\ z)(\#f\ u).((\text{A}^*\text{TICKET} : \text{AUnreserved})\otimes z \sqcap (\text{A}^*\text{TICKET} : \text{AUnreserved})\otimes u) \\ *TICKET(a, ticket_1), \text{Unreserved}(i_1, ticket_1), f(i_1, i_a), \\ *TICKET(a, ticket_2), \text{Unreserved}(i_2, ticket_2), f(i_2, i_a)$

정의 4. 에이전트가 시간 # = [t1,t2] 동안 수행하는 웹서비스 WS의 효과는 시간 t2에 에이전트가 알고 있는 월드 상태로 표현하며, 이 월드 상태는 WS를 수행한 결과로 바뀐 것을 의미한다.

명제 2. 단순 TL-ALCFK_{NF} 지식 베이스에서 시간 t = [t1,t2] 동안 수행되는 웹서비스 WS가 있을 때, 에이전트가 시간 t2에 WS를 수행한 결과로 알고 있는 월드 상태는 다음과 같은 Allen의 시간 제약 을 가지는 단순 TL-ALCFK_{NF} 개념과 어서션들이다.

$Tc \rightarrow (X(U) \#) | (\#(V) X)$
 $U \rightarrow m_i | o_i$
 $V \rightarrow m | o$

일반적으로 웹서비스는 캡슐화된 객체로 다루어지기 때문에, 에이전트는 웹서비스를 실행하는 동안 웹서비스의 내부에서 무슨 변화가 일어나는지 또는 어떤 로직으로 수행되는지에 대해서 알 필요가 없다. 대신 에이전트는 웹서비스의 실행을 위해 무엇이 필요한지, 그리고 실행의 결과로써 월드 상태와 에이전트의 지식이 어떻게 바뀌는지 아는 것이 중요하다. 따라서 단순 TL-ALCFK_{NF} 지식 베이스에서 웹서비스의 효과는 시간 t1과 t2 사이에 일어났다가 없어진 월드 상태를 가지지 않으며, 시간 t2에 웹서비스의 월드 상태와 부분적으로 겹치는 월드 상태로 표현된다. 예를 들어 3.2절에서 보인 예제의 효과는 다음과 같이 표현될 수 있다.

$\diamond(y\ v)(\#m\ y)(\#m\ v).((\text{A}^*\text{TICKET} : \text{AReserved})\otimes y \sqcap (\text{A}^*\text{TICKET} : \text{AReserved})\otimes v) \\ *TICKET(a, ticket_1), \text{Reserved}(i_3, ticket_1), m(i_3, i_a), \\ *TICKET(a, ticket_2), \text{Reserved}(i_4, ticket_2), m(i_4, i_a)$

정의 5. 시간 t = [t1,t2] 동안 수행하는 웹서비스 WS가 있을 때, 만약 에이전트 A가 시간 t1에 WS의 전제조건과 효과를 안다면 WS는 시간 t1에 에이전트 A에게 **인식적으로 실행 가능하다**.

정의 5는 위의 정의 3과 정의 4를 이용하여 웹서비스의 인식적 실행 가능성을 정의한 것이다. 이 정의는 Moore가 정의한 인식적으로 실행 가능한 행동과 비슷하지만 다음과 같이 두 가지 차이점이 존재한다. 첫 번째 차이점은 본 논문에서의 웹서비스는 일정 시간 간격 동안 실행된다는 것이며, 두 번째 차이점은 Moore가 만약 에이전트가 행동을 구성하는 행동들을 안다면 행동이 인식적으로 실행 가능하다고 정의한 반면에, 본 논문에서는 웹서비스 내부의 변화와 상관 없이 단지 웹서비

스의 전제조건과 효과만 고려한다는 것이다. 이러한 차이점은 위에서 언급한대로 웹서비스를 캡슐화된 객체로 간주하고 내부 변화를 무시할 수 있기 때문이다. 예를 들어 3.2절에서 두 장의 비행기 표를 예약하는 웹서비스가 인식적으로 실행 가능하기 위해서는 에이전트가 두 장의 티켓이 아직 예약이 안되어 있다는 있는 것을 알아야 하며(전제조건) 또한 웹서비스를 실행한 후에 티켓이 예매되었다는 것을 알 수 있어야 한다(효과).

그 동안 많은 사람들이 이러한 인식적 실행 가능성에 대한 지식 전제조건 문제에 대해서 연구해 왔지만, 이들 대부분은 물리적 실행 가능성에 대해서 고려하지 않았다. 사실 이 문제는 행동 또는 계획이 실제 물리적으로 수행 가능한지를 밝히는 것으로 비교적 간단한 문제이다. 하지만 본 연구에서는 웹서비스의 성공과 실패에 상관 없이 웹서비스가 수행되는 상태를 표현하기 위해 물리적 실행 가능성에 대한 정의를 명확히 할 필요가 있다. 예를 들어 3.2절에서 두 장의 비행기 표를 예약하는 웹서비스가 물리적으로 실행 가능하기 위해서는 에이전트가 웹서비스의 성공 또는 실패에 상관 없이 인터넷을 통해 비행기 예약 사이트에 접속할 수 있어야 하며, 또한 타임아웃이 발생하거나 사용자가 취소 버튼을 클릭하기 전에 예약 작업을 수행할 수 있어야 한다.

정의 6. 웹서비스 WS는 만약 (WS의 효과) ϵ (WS가 실제 수행된 후에 WS의 월드 상태) 이면 성공했다고 하며, 만약 성공이 아니면 실패했다고 한다.

정의 7. 만약 에이전트 A가 시간 t = [t1,t2] 동안 성공 또는 실패에 상관 없이 물리적으로 웹서비스 WS를 실행할 수 있다면 WS는 시간 t1에 에이전트 A에게 물리적으로 실행 가능하다.

이와 같은 정의는 웹서비스의 실행이 일정 시간 간격에서만 수행된다고 정의하였기 때문에 가능하다.

5. 웹서비스 조합

5.1 웹서비스 조합의 실행 가능성

Davis는 수행 가능성이라는 용어를 사용하여 결정적 계획과 비결정적 계획의 인식적 실행 가능성을 정형화 하였다. 하지만 이 수행 가능성에 대한 정의에서는 서론에서 설명한 바와 같이 웹서비스 조합이 실행하는 동안 다른 에이전트에 의해서 월드 상태들이 바뀔 수 있는 실행 환경에 대해서는 고려하지 않았다. 따라서 이 절에서는 Davis의 수행 가능성에 대한 정의를 확장하고, TL-ALCFK_{NF}언어를 이용하여 웹서비스 조합의 원자성을 보존하기 위한 방법을 정형화한다.

7) 실패는 논리적으로 불가능한 웹서비스가 아니며 실행가능하지 않은 (infeasible) 웹서비스의 실행으로 여러 상태를 가지게 된 것이다. 또한 실제 실패한 웹서비스는 항상 실행가능하다.

정의 8. 만약 웹서비스 조합 WSC를 구성하는 모든 웹서비스가 시간 t 에 물리적으로 실행 가능하고 무한 루프로 들어가지 않는다면 WSC는 시간 t 에 *유한 시간 내에 수행 가능하다*.

정의 8은 웹서비스 조합의 실행과 종료에 대한 특성을 정의한 것으로써 이는 정의 7로부터 유추될 수 있다. 다음은 웹서비스 조합에서 웹서비스의 가능한 다음 단계(possible next step)과 다음 단계(next step)에 대한 정의이다.

정의 9. 웹서비스 WS'는 만약 다음 두 가지 조건을 만족하면 일정 시간 $t = [t1, t2]$ 동안 수행된 웹서비스 WS의 가능한 다음단계가 된다.

- (1) WS'가 일정 시간 $t' = [t3, t4]$ 동안 수행된다고 할 때, WS'는 시간 $t3$ 에 인식적으로 실행 가능하고 물리적으로 실행 가능하다.
- (2) WS'는 WS를 수행한 후에 실행된다.

정의 10. 만약 에이전트가 웹서비스 WS의 가능한 다음 단계들 중에서 특정 웹서비스 WS'를 결정할 수 있다면 WS'는 WS의 다음 단계가 된다.

명제 3. 단순 TL-ALCFK_{NF} 지식 베이스에서 시간 $t = [t1, t2]$ 동안 웹서비스 WS가 수행된 후에 에이전트가 시간 $t' = [t3, t4]$ 동안 수행되는 웹서비스 WS'를 시간 $t3$ 에 WS의 (가능한) 다음 단계라고 알기 위해서는 WS'와 WS가 ($t2 \text{ m } t3$), ($t3 \text{ mi } t2$), ($t2 \text{ b } t3$), ($t3 \text{ a } t2$)들 중 하나의 시간 관계를 가지는 단순 TL-ALCFK_{NF} 개념과 어서션이어야 한다.

기존의 상황 계산법(situation calculus)에 기반을 둔 계획 시스템에서는 계획이 하나의 상태에서 다른 상태로의 함수들로 정의되는 반면에, TL-ALCFK_{NF}에서 웹서비스 조합은 Allen의 시간 관계에 따라 웹서비스와 각각의 상태들이 이루는 시간적인 관계에 의해서 정의된다. 더군다나 모든 웹서비스는 원자성을 가지고 수행되기 때문에 (가능한) 다음 단계의 정의에서는 m, mi, b, a를 제외한 다른 시간 관계를 가지지 않는다.

정의 11. 실패한 웹서비스 WS의 다음 단계로써 시간 $t = [t1, t2]$ 동안 수행되는 보정 웹서비스 CWS는 만약 에이전트 A가 시간 $t1$ 에 CWS의 ((실패한 WS의 효과) \sqsupset (CWS의 전제조건))와 같은 전제조건을 알고, ((CWS의 효과) \sqsupset (만약 WS가 성공했다면 WS의 가능한 다음 단계의 전제조건 \sqcup WS의 전제조건))와 같은 효과를 안다면, 시간 $t1$ 에 에이전트 A에게 인식적으로 실행 가능하다.

정의 11은 웹서비스 조합이 트랜잭션적으로 수행가능하기 위해 웹서비스 조합의 원자성을 보존할 수 있도록 하는 보정 웹서비스(Compensating Web Services)를 정의한 것이다. 즉, 웹서비스 조합을 구성하는 웹서비스

WS가 실패하는 경우, 만약 에이전트가 WS에 대한 보정 웹서비스를 알고 WS를 수행한 후에 보정 웹서비스를 수행할 수 있다면 웹서비스 조합은 트랜잭션적으로 수행 가능하게 된다.

정의 12. 만약 웹서비스 WS가 실행된 후에 다른 웹서비스에 의해 WS의 전제조건을 시간 t 에 가지게 된다면, WS는 시간 t 에 취소되었다고 한다.

보정 웹서비스는 특정 웹서비스들의 집합으로써, 어떤 웹서비스가 실패했을 때 보정 웹서비스의 실행을 통해 웹서비스 또는 웹서비스 조합을 실행한 후의 월드 상태를 실행하기 전의 월드 상태로 바꾸거나(취소) 또는 다른 대체 가능한 웹서비스를 수행하도록 한다. 하지만 만약 보정 웹서비스마저도 실패한다면 웹서비스 조합은 보정 웹서비스를 수행한다고 하더라도 트랜잭션적으로 실행 가능하지 않게 된다. 따라서 어떤 웹서비스 조합이 트랜잭션적으로 수행 가능하기 위해서는 에이전트가 모든 웹서비스가 취소될 수 있다는 것을 알아야 한다. 이를 위해 정의 5의 인식적으로 실행 가능한 웹서비스에 대한 정의를 기반으로 다음과 같이 인식적으로 수행 가능한(epistemically executable) 웹서비스를 정의한다.

정의 13. 시간 $t = [t1, t2]$ 동안 수행하는 웹서비스 WS가 있을 때, 만약 에이전트 A가 시간 $t1$ 에 WS의 전제조건과 효과 그리고 WS가 취소될 수 있는지를 안다면, WS는 시간 $t1$ 에 에이전트 A에게 인식적으로 수행 가능하다.

정의 14. 웹서비스 WS, WS의 가능한 다음 단계 WS', WS의 보정 웹서비스 CWS, 이렇게 세 개의 웹서비스가 있을 때, 만약 WS가 실패하면 CWS는 WS의 다음 단계가 되며 WS'는 CWS의 가능한 다음 단계가 된다.

만약 웹서비스 조합의 모든 웹서비스가 성공했다면 CWS를 실행할 필요 없지만, 만약 웹서비스 조합의 특정 웹서비스 WS가 실패했다면 에이전트는 WS의 다음 단계로써 CWS를 선택할 수 있다. 또한 WS가 성공하면 WS'는 WS의 가능한 다음 단계가 되며, WS가 실패하면 CWS는 WS의 다음 단계가 되고, WS'는 CWS의 가능한 다음 단계가 된다.

정의 15. 웹서비스 조합 WSC는 다음과 같은 조건들을 만족하면, 시간 $t1$ 에 에이전트 A에게 트랜잭션적으로 수행 가능(transactionally executable)하다.

- (1) WSC는 시간 $t1$ 에 유한 시간 내에 수행 가능하다.
- (2) 에이전트 A는 모든 웹서비스 또는 보정 웹서비스가 (가능한) 다음 단계인지를 안다.
- (3) WSC의 모든 다음 단계의 웹서비스는 일정시간 $t = [t1, t2]$ 동안 인식적으로 수행 가능하다.
- (4) WSC의 모든 다음 단계의 보정 웹서비스는 일정 시

간 $t = [t1, t2]$ 동안 인식적으로 실행 가능하다.

정의 15는 위에서 정의한 것들을 기반으로 트랜잭션적으로 수행 가능한 웹서비스 조합을 정의한 것이다. 일반적으로 웹서비스 조합의 특성은 웹서비스 조합이 가지고 있는 보정 웹서비스가 어떤 특성을 가지고 있으며 또한 어떻게 수행되는지에 따라 달라진다. 따라서 본 논문에서는 웹서비스 조합의 특성을 원자성, 시간, 에이전트의 지식에 따라 분류하였다. 비록 이러한 분류는 정형화된 방법은 아니지만 웹서비스 조합에서 보정 웹서비스가 어떤 특성을 가지고 있는지, 그리고 본 논문에서 제시한 TL-ALCFK_{NF} 언어를 이용해서 트랜잭션적으로 수행 가능한 웹서비스 조합이 어떻게 수행되는지를 보이기 위한 것이다.

- 비원자(non-atomic) 웹서비스 조합 : 어떤 웹서비스 조합은 사용자의 요구에 따라서 원자적으로 처리되지 않을 수 있으며[9], 이러한 웹서비스 조합에서는 에이전트가 웹서비스 조합을 수행하다가 실패하게 되면 더 이상의 보정이나 작업을 수행하지 않고 단순히 에러를 일으키며 종료하게 된다. 하지만 이는 원자성을 보장하려는 본 논문의 목적과 맞지 않기 때문에 본 연구에서는 이에 대해 고려하지 않는다.
 - 원자 웹서비스 조합 : 본 논문에서 설명한 특성이다.
 - 즉시(immediate) 웹서비스 조합 : 특정 웹서비스가 실패했을 때 에이전트는 대체 가능한 웹서비스들을 찾게 되며 만약 존재하지 않는다면, 이미 완료된 웹서비스를 보정 웹서비스를 이용해서 블랙시커야 한다. 이는 전통적인 분산 트랜잭션의 방법과 비슷한 방법이다.
 - 지연(deferred) 웹서비스 조합 : 만약 웹서비스 조합이 오랫동안 수행될 수 있다면, 실패한 웹서비스 또는 보정 웹서비스는 웹서비스 조합의 타임아웃 시간 내에 여러 번 재 시도될 수 있다.
 - 박식한(omniscient) 에이전트 : 박식한 에이전트는 웹서비스가 실패한 경우 실패한 웹서비스 대신에 수행될 수 있는 다른 서비스를 알고 웹사이트 등을 통해 검색하여 수행할 수 있기 때문에, 이 경우 웹서비스 조합이 물리적으로 실행 가능하기만 하면 트랜잭션적으로 수행 가능할 수 있다.
 - 박식하지 않은(non-omniscient) 에이전트 : 박식하지 않은 에이전트는 자신이 알지 못하는 다른 웹서비스를 찾아 수행할 수 없기 때문에, 웹서비스가 실패할 경우 웹서비스 조합을 취소하거나 대신 사람이 개입해서 웹서비스 조합을 수행해주어야 한다. 이와 유사한 연구로 중립적 실행(Middle-Ground execution)[1] 방법이 있다.
- 다음 절에서는 위와 같은 분류에 따르는 예제를 통해

단순 TL-ALCFK_{NF} 지식 베이스에서 트랜잭션적으로 수행 가능한 웹서비스 조합이 어떤 특징을 가지는지를 보인다. 특히 본 연구에서는 보정 웹서비스의 특징을 가장 잘 표현하는 즉시 웹서비스 조합과 박식한 에이전트의 경우를 고려한다.

5.2 트랜잭션적으로 수행 가능한 웹서비스 조합 예제

5.2.1 즉시 웹서비스 조합

이 절에서는 여행을 위해 비행기, 호텔, 렌터카를 예약하는 즉시 웹서비스 조합 시나리오를 이용하여 트랜잭션적으로 수행 가능한 웹서비스 조합이 어떻게 수행되는지를 보인다.

아래의 그림은 단순 TL-ALCFK_{NF} 지식 베이스에서 이러한 웹서비스 조합 시나리오를 어떻게 표현하는지를 보인 것으로써, 웹서비스 조합을 수행하는 에이전트가 비행기, 호텔, 렌터카를 예약하는 세 개의 웹서비스와 이 웹서비스에 대해 보정을 수행할 수 있는 보정 웹서비스를 알고 있다는 것을 표현하고 있으며, 또한 이 웹서비스들이 어떤 시간 순서로 수행되는지를 나타내고 있다. 특히 이 예제에서 Reserve-Flight는 3.2절에서 설명한 것과 같으며 다른 웹서비스도 비슷한 방법으로 표현될 수 있다.

$$K_{Reserve-Trip} \doteq \diamond(x y z u v w) (x b y)(x b u)(y b z)(y b v)(z b w). \\ (A_{Reserve-Flight} @ x \sqcap A_{Reserve-Hotel} @ y \sqcap A_{Reserve-Car} @ z \sqcap \\ A_{Compensate-RF} @ u \sqcap A_{Compensate-RH} @ v \sqcap A_{Compensate-RC} @ w)$$

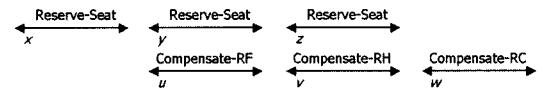


그림 5 Reserve-Trip 개념에서 각 개념들의 시간적 의존 관계

우선 이 웹서비스 조합이 물리적으로 실행 가능하다고 가정하면 웹서비스 조합은 무한 루프를 가지지 않기 때문에 유한 시간 내에 수행 가능하게 된다. 또한 에이전트가 Reserve-Flight를 알고 있다면 Reserve-Flight는 인식적으로 실행 가능하게 되고 결국 성공적으로 수행할 수 있게 되며, 다음에 에이전트는 Reserve-Flight의 가능한 다음 단계가 Reserve-Hotel이라는 것을 알기 때문에 Reserve-Hotel을 수행하려고 할 것이다. 하지만 여기에서 Reserve-Flight를 수행하는 동안 모든 호텔의 방(individuals)이 예약되었다는 것을 가정하자. 이와 같은 상황에서 에이전트는 Reserve-Hotel을 실제 수행하기 전에 실행 가능한지를 검사함으로써 예약할 방이 없다는 사실을 알 수 있기 때문에 Reserve-Hotel 대신에 Compensate-RH를 다음 단계로써 수행하게 된다. 여기에서 Compensate-RH는 Reserve-Hotel과 똑같은 효과를 가지는 다른 웹서비스가 될 수 있으며 만

약 이러한 웹서비스가 없다면 Reserve-Flight를 취소시킴으로써 비록 웹서비스 조합은 실패했지만 웹서비스 조합의 원자성은 보존할 수 있게 된다. 반면에 에이전트가 다른 웹서비스를 찾아서 성공적으로 수행했다면 Compensate-RH의 가능한 다음 단계가 Reserve-Car인 것을 알기 때문에 이를 수행함으로써 웹서비스 조합을 성공적으로 마칠 수 있게 된다.

5.2.2 박식한 에이전트

웹서비스 조합의 또 다른 예제로써, 박식한 에이전트가 영화를 예약하는 시나리오를 통해 트랜잭션적으로 수행 가능한 웹서비스 조합이 어떻게 수행되는지를 보인다. 특히 이 절에서는 어떤 사람이 친구와 함께 가능한 시내로부터 가까운 영화관에서 어떤 영화를 보기 위해 두 개의 서로 붙어 있는 좌석을 예매하길 원하며 박식한 에이전트가 이 예약을 수행해 주는 상황을 가정한다.

박식한 에이전트는 우선 현재 알고 있는 단순 TL-ALCFK_{NF} 지식 베이스에서 알려진 영화관들 중 시내에서 가능한 가장 가까운 곳을 찾아 영화를 예약하려고 할 것이다. 하지만 만약 실패한다면 영화를 예약할 때까지 다른 영화관을 찾아 예약을 시도할 것이다. 다음 개념은 이러한 시나리오를 나타낸 것이다.

$$\begin{aligned}
 KReserve-Movie &= \diamond(x\ y)(z\ b\ y). \\
 &\quad ((AReserve-Seat) \otimes x \sqcap (ACompensating-RS) \otimes y) \\
 KReserve-Seat &= \diamond(x\ y\ z\ u\ v)(\# f\ z)(\# m\ y)(\# d\ z)(\# f\ u)(\# m\ v). \\
 &\quad ((A*SEAT1 : AVacant) \otimes z \sqcap (A*SEAT2 : AVacant) \otimes u \sqcap \\
 &\quad (A*SEAT1 : ANEXTTO \downarrow A*SEAT2) \otimes z \sqcap \\
 &\quad (A*SEAT1 : AReserved) \otimes y) \sqcap (A*SEAT2 : AReserved) \otimes v)
 \end{aligned}$$

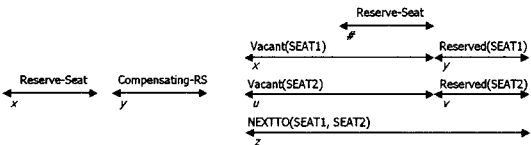


그림 6 Reserve-Movie 개념에서 각 개념들의 시간적 의존 관계

위의 Reserve-Movie 개념은 Reserve-Seat와 Compensating-RS로 구성된다. Reserve-Seat는 현재 단순 TL-ALCFK_{NF} 지식 베이스에서 알려진 어떤 영화관에서 예약하려는 영화에 대해 남아 있는 좌석이 있는 것을 안다면 이를 예약하는 웹서비스이며, Compensating-RS는 Reserve-Movie를 수행할 수 있을 때까지 다른 영화관을 찾아 예약을 수행하는 보정 웹서비스이다.

이 때 다른 영화관은 현재 지식 베이스에 존재하는 것이 아니라 다른 웹사이트 또는 다른 지식 베이스에 존재할 수 있으며, 이와 같이 영화관에 대한 다른 지식을 찾는 작업은 박식한 에이전트이기 때문에 가능하다. 하지만 만약 에이전트가 끝내 이 예약을 성공하지 못했다면 이 웹서비스 조합은 이전 절에서 설명했던 즉시

웹서비스 조합과 같이 취소를 해야 한다. 이는 에이전트가 Reserve-Movie 웹서비스 조합을 알지 못해서 수행할 수 없었던 것이 아니라 다른 에이전트들 때문에 모든 영화관에서 모든 좌석이 예약이 되었기 때문이다

6. 관련 연구

자동화된 계획 시스템에 대한 연구는 그 동안 여러 분야에서 다양하게 이루어져 왔으며, 특히 인공지능 분야에서는 최근 기존의 전통적인 계획 시스템을 확장한 이론적 결정 계획(DTP : Decision-theoretic planning) [10]에 대한 연구가 제안되고 있다. 이러한 연구에서 다루고 있는 문제는 계획을 구성하는 각각의 행동들이 불확실한 효과를 가지며, 계획을 실행하는 에이전트도 완전하지 않은 정보를 가지기 못하기 때문에, 결국 계획의 최종 목적을 이룰 수 없게 되는 것이다. 이러한 계획의 불확실성(uncertainty)은 계획을 자동으로 결정하고 실행하는데 해결해야 하는 문제로써, 이를 해결하기 위한 대표적인 방법으로 MDPs(Markov decision processes) 방법, 조건적 계획 또는 불확정 계획 방법등이 있다.

이 중 MDP 방법은 최근 위와 같은 불확실성을 해결하기 위해 제안된 이론적인 프레임워크로써, 이 프레임워크에서는 특정 상황에서 수행 가능한 모든 행동을 명시한 최적화 정책(optimal policy)을 구하여 이를 이용해 계획을 수행하도록 하고 있다. 특히 이 정책이 유한한 경우 조건적 계획 또는 불확정 계획이라고 하며, 계획이 실행되는 동안 발생하는 각각의 상황에 대해서 가지(branch)를 가지고 결정하도록 한다.

하지만 이러한 방법은 서론에서 언급했듯이 계획이 부분적으로 성공한다 하더라도 가장 최선의 또는 가장 높은 확률을 가지는 행동을 선택한다는 점에서 트랜잭션을 다루는 본 연구와 차이가 있다.

웹서비스 조합에 대한 연구는 대부분 기존 인공지능 분야의 계획 시스템에 기반을 두고 있으며, 최근에는 이러한 연구들을 시멘틱 웹서비스에 적용하기 위한 논문들이 제안되었다.

Wu et. al[11]는 기존에 알려진 작업 분해(task decomposition)를 통해 계획을 만들고 수행하는 인공지능의 계획 기술을 이용한 HTN(Hierarchical Task Network) 계획 시스템을 기반으로 SHOP2(Simple Hierarchical Ordered Planner) 시스템을 구현하였으며, DAML-S의 프로세스 모델을 SHOP2의 모델로 변환하는 방법을 제안함으로써 자동화된 웹서비스 조합을 수행할 수 있음을 보였다. 하지만 이 논문에서는 SHOP2에서 계획을 수행하는 시간 동안 다른 에이전트의 간섭을 받지 않는다는 것을 가정했기 때문에 웹서비스 조합의 원자성을 보장하기 힘들다.

또한 Mandell과 McIlraith[2]는 현재 웹서비스 산업 분야에서 웹서비스 프로세스 언어로 가장 많이 사용되고 있는 BPEL을 시맨틱 웹에 적용하기 위한 시도를 수행하였다. 이를 위해 BPEL에 OWL을 적용한 SDS (Semantic Discovery Service)를 이용해 동적인 서비스 바인딩과 시맨틱 변환을 수행할 수 있음을 보였으며, BPEL이 본래 인터페이스 중심(interface-oriented) 언어일 뿐만 아니라 프로세스의 재구성에 대한 부작용 때문에 현재의 BPEL 모델로는 자동화된 웹서비스 조합이 힘들다는 문제점을 지적하였다. 이와 같이 BPEL, WS-CAF[9]와 같은 웹서비스 프로세스와 트랜잭션 표준들은 온톨로지 구축과 추론을 위한 시맨틱이 부족하기 때문에 OWL을 비롯한 로직 기반의 지식 표현과 추론 방법이 추가적으로 필요하다.

이와 같은 연구들과 달리 본 연구에서는 특히 시맨틱 웹에서 웹서비스 조합에 대한 지식 전제조건을 다루었으며, 이와 관련되어서는 기존에 다음과 같은 연구들이 제안되었다.

Morgenstern[12]은 본 연구에서 소개한 Davis의 연구가 하나의 에이전트만 고려하고 있는 점을 개선하기 위해서 구문(syntactic) 이론을 고안하였으며, 이 이론을 이용해 여러 에이전트들이 계획을 수행하기 위해 서로 통신할 수 있는 방법을 제안하였다. 즉, 어떤 에이전트가 특정 행동을 혼자 수행할 수 없을 때 다른 에이전트들에게 이 행동에 대한 수행을 위임함으로써 계획을 완료할 수 있도록 하고 있다. 실제 Morgenstern의 연구는 계획이 실패했음에도 불구하고 다른 방법을 이용해 계획을 완료할 수 있게 하는 점에서 본 연구와 비슷한 점이 있다. 하지만 Morgenstern의 연구는 여러 에이전트들이 서로 상호작용을 하는 환경을 고려하고 있는 반면에, 본 연구에서는 인터넷에 존재하는 웹서비스뿐만 아니라 웹서비스들을 수행하는 에이전트 프로그램이 분산성과 자치성을 가지기 때문에, 여러 에이전트간의 상호작용이 아니라 하나의 에이전트가 인터넷 분산 환경에서 어떻게 웹서비스 조합을 수행해야 하는지에 대해 고려하였다. 또한 Morgenstern은 본 연구에서 다루고 있는 웹서비스의 원자성을 고려하지 않았다.

McIlraith와 Son[1]은 자동화된 웹서비스 조합을 수행하기 위해 로직 프로그래밍 언어 중 하나인 Golog을 적용한 방법을 제안하였다. 또한 McIlraith와 Son은 Davis의 논문과 비슷한 결과로써 에이전트가 웹서비스 조합을 알고 혼자 수행할 수 있는지에 대한 지식 자가충분 프로그램(knowledge self-sufficient program)과 물리적으로 실제 수행될 수 있는지에 대한 물리적 자가충분 프로그램(physically self-sufficient program)을 정의하였다. 하지만 이러한 정의는 본 논문의 모델과 달

리 1차 로직(first-order logic)과 상황 계산법에 기반을 두고 있으며 본 논문에서 사용한 자동인식 연산자와 같은 표현을 할 수 없다. 또한 에이전트가 웹서비스를 수행하지 못했을 경우 사람이 개입해서 수행하도록 하는 중립적 실행 방법만을 사용하고 있지만, 본 논문에서는 원자성을 고려하기 위해 다양한 방법을 고려하였다.

이 외에도 시맨틱 웹에서 자동화된 웹서비스 조합을 수행하기 위한 여러 연구들이 소개되었지만, 위와 같이 이러한 연구들은 각각 정형화된 시맨틱을 가진 모델을 정의하고 이에 따라 동작하는 방법을 제안하였다. 위의 연구들의 경우, Wu et. al과 McIlraith와 Son은 1차 로직과 상황 계산법 모델을 이용하였으며, 특히 McIlraith와 Son은 프롤로그를 이용해 시스템을 구현하였다. 또한 Mandell과 McIlraith는 DQL(DAML Query Language)과 1차 로직 모델 기반의 JTP(Java Thorem Prover) 추론기를 이용하였으며, Morgenstern은 구문 모델을 적용하였다. 본 연구에서도 TL-ALCFK_{NF}의 시맨틱 모델에 따라 자동화된 웹서비스 조합을 보였으며, 다른 연구들과 달리 시맨틱 웹에서 웹서비스 조합의 원자성을 보존하도록 하였다.

7. 결론

본 연구에서는 인공 지능 분야의 계획 시스템에 대해 연구되어 왔던 지식 전제조건 문제를 시맨틱 웹서비스 조합에 대해서 해석하고 이를 정형화하였다. 일반적으로 웹서비스는 계획 시스템의 행동들과 달리 원자성이 보장되는 트랜잭션 관리가 중요하다. 하지만 기존의 지식 전제조건 문제와 계획 시스템들에서는 이에 대해 고려되지 않았다. 따라서 본 연구에서는 기존 분산 트랜잭션의 개념을 이용하여 트랜잭션적으로 수행 가능한 웹서비스 조합을 정의하였으며, 이를 통해 시맨틱 웹에서 에이전트가 웹서비스 조합을 원자적으로 수행할 수 있는 조건을 명시하였다.

또한 이러한 개념을 정형화하기 위해 시간, 웹서비스, 웹서비스 조합들을 표현하는 언어인 TL-ALCF에 모달 연산자 **K**와 **A**를 추가한 TL-ALCFK_{NF} 언어를 소개하였으며, 실제 이 언어를 이용해 인식 추론을 수행할 수 있도록 테이블로우 계산법을 제안하였다. 이와 같이 본 연구에서는 원자성이 보장될 수 있는 시맨틱 웹서비스 조합에 대한 정형화를 수행함으로써, 실제 OWL-S과 같은 온톨로지를 이용한 다른 시스템들에서 에이전트가 자동화된 웹서비스 조합을 수행할 수 있는 이론적 기반을 제시하였다.

본래 OWL과 OWL-S는 기술 로직의 시맨틱 모델을 기반으로 온톨로지를 표현하기 위해 만들어진 언어이다. 즉, 시맨틱 웹에서 특정 도메인의 지식 베이스를 표현하

기 위한 언어로 OWL이 이용되며, 이러한 OWL을 기반으로 웹서비스에 대한 시맨틱을 표현하기 위한 언어로 OWL-S가 사용된다. 따라서 실제 OWL과 OWL-S를 이용해 표현된 온톨로지를 지식베이스에서 구성하고 추론하는 것은 기술 로직의 시맨틱에 따라 수행된다. 또한 본 연구에서 제안한 TL-ALCFK_{NF}에서 ALCF는 OWL-S의 온톨로지 로 매핑[14]이 가능하기 때문에 이에 추가하여 서론에서 언급한 바와 같이 트랜잭션적으로 수행 가능한 웹서비스 조합을 위해 필요한 시간(TL)과 지식(K_{NF})에 대한 기술 로직 시맨틱 모델을 이용한다면 본 논문을 OWL-S 기반의 실제 시스템에 적용 가능할 것이다.

참고 문헌

- [1] S. McIlraith and T. C. Son. Adapting Golog for Composition of Semantic Web Services. In Proc. of the 8th International Conference on Principles and Knowledge Representation and Reasoning (KR'02), pp.482-496 (2002).
- [2] D.J. Mandell and S.A.McIlraith. Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation. In Proc. Of the 2nd International Semantic Web Conference, pp.227-241 (2003).
- [3] E. Davis. Knowledge Preconditions for Plans. Journal of Logic and Computation, 4(5):721-766 (1994).
- [4] Artale and E. Franconi. A Temporal Description Logic for Reasoning about Actions and Plans, Journal of Artificial Intelligence Research, 9:463-506 (1998).
- [5] J. F. Allen. Temporal Reasoning and Planning. In J. F. Allen, H. A. Kautz, R. N. Pelavin and J. D. Tenenber, editors, Reasoning about Plans, Chap. 1, pp.2-68, Morgan Kaufmann (1991).
- [6] F. M. Donini, D. Nardi, R. Rosati. Description Logics of Minimal Knowledge and Negation as Failure. ACM Transactions on Computational Logics, 3(2):177-225 (2002).
- [7] J. McCarthy and P.J. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In B. Meltzer and D. Michie, editors, Machine Intelligence 4, pp.463-502, Edinburgh University Press (1969).
- [8] R. Moore. A Formal Theory of Knowledge and Action. In J. Hobbs and R. Moore, editors, Formal Theories of the Commonsense World, pp.319-358, ALEX Publishing, Norwood, New Jersey (1985).
- [9] OASIS WS-CAF TC Draft. Web Services Composite Application Framework. In http://www.oasis-open.org/committees/documents.php?wg_abbr=ws-caf (2005).
- [10] C. Boutilier, T. Dean, and S. Hanks. Decision-Theoretic Planning: Structural Assumptions and Computational Leverage. Journal of Artificial Intelligence Research 11, pp.1-94 (1999).
- [11] D. Wu, E. Sirin, B. Parsia, J. Hendler, and D. Nau. Automatic Web Services Composition using SHOP2. In Workshop on Planning for Web Services (2003).
- [12] L. Morgenstern. Foundations of a Logic of Knowledge, Action, and Communication. PhD thesis, New York University (1987).
- [13] B. Hollunder and W. Nutt. Subsumption Algorithms for Concept Languages. Technical Research Report RR-90-04, DFKI, Germany (1990).
- [14] I. Horrocks, D.L. McGuinness, C.A. Welty. Digital Libraries and Web-Based Information Systems. In the Description Logic Handbook, edited by F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider, pp.436-459, Cambridge University Press (2002).



김 상 군

1999년 충남대학교 공과대학 정보통신공학과(공학사). 2001년 충남대학교 공과대학 컴퓨터공학과(공학석사). 2001년~현재 충남대학교 공과대학 컴퓨터공학과 박사과정. 2003년~현재 (주)K4M 연구원. 관심분야는 시맨틱웹, e-비즈니스



이 규 철

1984년 2월 서울대학교 공과대학 컴퓨터공학과(공학사). 1986년 2월 서울대학교 공과대학 컴퓨터공학과(공학석사). 1990년 8월 서울대학교 공과대학 컴퓨터공학과(공학박사). 1994년 미국 IBM Almaden Research Center 초빙 연구원. 1995년~1996년 미국 Syracuse University 초빙 교수. 2001년~현재 전자상거래 표준화 통합 포럼 전자거래 기반 기술위원회 위원장. 2003년~현재 한국전자거래학회 편집이사. 2003년~현재 웹 코리아 포럼 부위원장. 현재 충남대학교 공과대학 컴퓨터공학과 교수. 관심분야는 데이터베이스, XML, 정보 통합, e-비즈니스