양상 뮤 논리를 위한 속성 명세 패턴

전 승 수　권 기 현

경기대학교 정보과학부 소프트웨어 공학 연구실

Property Specification Patterns for Modal μ-Calculus

Seungsu Jun　Gihwon Kwon

Software Engineering Laboratory, Department of Computer Science, Kyonggi University

요 약

본 논문에서는 양상 뮤 논리를 위한 속성 명세 패턴 연구를 통해 시제 논리에 대한 패턴 기반의 단일한 프레임워크를 제시한다. 본 연구에서는 Dwyer 의 속성 명세 패턴 분류를 상태(S)와 행동(A)으로 세분화하고 이를 다시 강함(A)과 약함(E)으로 다시 세분했다. 이러한 의미 기반의 계층적 패턴 분류 체계를 통해 양상 뮤 논리의 속성 명세 패턴을 분석했으며 실제 모형 검사기에서 사용된 예제들의 패턴 분류에 적용했다. 그 결과 기존의 분류 체계보다 더 정확한 분류가 가능했을 뿐만 아니라, 속성 명세의 작성 및 이해가 용이하였다.

1. 연구 배경

모형 검사는 시스템이 바람직한 속성을 만족하는지를 논리적으로 검증하는 방법이다 [1]. 모형 검사를 하기 위해서 사용자는 검사할 속성을 표현해야 한다. 이것을 속성 명세라고 부른다. 명세 수단으로 시제 논리, 오토마타, 시각 언어 등 다양한 표현들이 사용되고 있는데, 이들 중에서 가장 많이 사용되는 것이 시제 논리이다 [2]. 왜냐하면 시제 논리는 표현력이 우수해서 다양한 속성들을 명세할 수 있기 때문이다. 그런데 시제 논리로 작성된 속성 명세를 사용자들이 쉽게 이해할 수 있을까 하는 의문이 있다. 거꾸로, 주어진 속성을 시제 논리로 정확히 명세할 수 있을까 하는 의문도 있다. 예를 들어 양상 뮤 논리(*modal μ-calculus*) [2]로 작성된 아래의 속성 명세를 살펴보자.

νZ. [emergency](μX. [alarm](μY. [alarm](νZ. [alarm]ff ∧ [–]Z) <–>tt ∧ [–alarm]Y) <–>tt ∧ [–alarm]X) ∧ [–]Z

위 명세를 해석하기 위해서는 논리에 사용된 기호와 의미, 그리고 배경 지식에 대한 이해가 요구된다. 특별히 양상 뮤 논리의 경우 최대 고정점과 최소 고정점에 대한 이해가 필수적이다. 위 명세의 의미는 "응급 상황이 발생하면 연속해서 두 번 경고음이 반드시 울려야 한다" 이다. 시제 논리인 CTL(Computation Tree Logic)과 LTL(Linear Temporal Logic)이 고급 명세 언어라고 한다면, 양상 뮤 논리는 어셈블리 언어와 같다. 따라서 명세의 작성과 해석이 쉽지않다. 본 연구는 양상 뮤 논리의 명세 작성 및 이해를 도와주기 위하여 연구되었다.

속성 명세에 관련된 문제들을 해결하기 위하여 명세 전용 언어[3], 패턴 이용[4]등에 관한 연구가 있었다. 기존 연구의 조사 결과 Dwyer 의 속성 명세 패턴이 우리의 연구 방향에 가장 부합된 것으로 밝혀졌다. 그러나 패턴을 CTL, LTL, GIL(Graphical Interval Logic)로 매핑하고 있을 뿐, 본 연구에서 대상으로 하는 양상 뮤 논리로의 매핑은 다루지 않았다. 또한 CTL 과 LTL 은 각각 상태와 경로 만을 다루며 GIL 은 이벤트 만을 다루고 있는 반면에 양상 논리는 상태와 행동의 두 가지 개념을 모두 수용하고 있기 때문에, 그의 분류 체계를 양상 뮤 논리에 그대로 적용할 수 없다.

그래서 본 연구에서는 Dwyer 의 분류 체계를 양상 논리에 맞게 세분화했고, 모형 검사기에서 사용되었던 예제들의 분류에 적용해 보았다. 그 결과 기존의 분류 체계보다 더 정확히 분류하였다. 이것은 속성 명세 패턴의 활용이 명세 작성과 이해에 도움이 된다는 사실을 입증한다.

2. 속성 명세 패턴

2.1 패턴 분류

본 연구에서는 다음과 같은 이유로 양상 뮤 논리를 택했다. 첫째, 표현력이 가장 높다. 둘째, 상태와 행동에 대한 표현이 모두 가능하다. 셋째, CTL, LTL, CTL* 를 양상 뮤 논리로 변환할 수 있다. 즉 양상 뮤 논리는 시제 논리에 대한 단일 프레임워크를 제공하기 때문이다.

양상 뮤 논리를 위한 속성 명세 패턴의 분류는 그림 1 과 같다. 대분류로는 Occurrence 와 Order 가 있으며, 중 분류에는 8 개의 패턴들이 있다: Absence, Universality, Bonded Exsitence, Existence, Response, Precedence, Chain Response, Chain Precedence. 양상 뮤 논리는 상태와 행동에 대한 표현이 모두 가능하기 때문에 이들을 상태(S)와 행동(A)으로 세분화 한다. 속성이 모든 경로에서 만족된다면 강하다고 하며, 속성을 만족하는 경로가 존재한다면 약하다고 한다. 따라

서 상태와 행동 식은 강함(A) 과 약함(E) 으로 다시 세분된다.

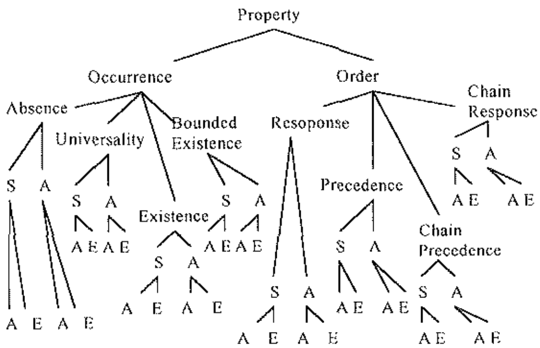

그림 1 패턴 분류

예를 들어 안전성 속성을 나타내는 Universality 패턴을 살펴보자. 이 패턴의 의미는 " 속성은 항상 만족 한다" 이다. 상태와 행동에 따라서 다음과 같이 세분화 된다:

- 상태 중심(S): $\nu Z.\Phi\wedge[-]Z$
- 행동 중심(A): $\nu Z.[K]ff\wedge[-]Z$

상태와 행동 중심의 속성이 모든 경로에서 만족되는지, 아니면 하나 이상의 경로에서 만족되는지에 따라서 아래와 같이 더 세분화된다:

- 상태 중심, 강함(SA): $\nu Z.\Phi\wedge[-]Z$
- 상태 중심, 약함(SE): $\nu Z.\Phi\wedge\langle-\rangle Z$
- 행동 중심, 강함(AA): $\nu Z.[K]ff\wedge[-]Z$
- 행동 중심, 약함(AE): $\nu Z.[K]tt\wedge\langle-\rangle Z$

이러한 의미 기반의 세부적 패턴 분류는 가장 정확한 의미의 표현을 가능하게 한다.

한편, Universality 패턴은 Z=f(Z)의 최대 고정점으로 표현되는데 이는 항상 참(True)의 값을 갖는 전체 집합 S 로부터 하향식(Top-down)방식으로 펼쳐진다. 경로 중심의 강함(AA)를 반복계산법으로 풀어보면 다음과 같다.

$$\nu Z.\ [K]ff\wedge[-]Z = true$$
$$\wedge\ [K]ff\wedge[-]true$$
$$\wedge\ [K]ff\wedge[-]\ ([K]ff\wedge[-]true)$$
$$\ldots$$

위의 식을 해석해 보면" 모든 경로에서 이벤트 K 에 의한 이동의 무한 연속은 가능하다" 이며 시스템의 안전성(Safety) 속성에 해당된다. 또한 최소 고정점의 경우 시스템의 궁극성(Liveness) 속성을 표현하며 이는 Exisetence 패턴으로 분류된다. 시스템이 갖는 모든 속성은 표 1 과 같이 양상 뮤 논리로 표현된다.

2.2 적용

명세 패턴으로 시스템의 공평성(Fairness) 속성을 검사한다고 하자. 속성 요구는 "자동차가 접근하면 인제가 자동차는 건너가야 한다" 그리고 "자동차 혹은 기차의 건너가는 것이 무한 발생해야 한다"이다. Bounded Existence 패턴을 적용하면 아래와 같다.

표 1 양상 뮤 논리를 위한 속성 명세 패턴

속성	패턴	의미	중심		식
	Abs-	AG(!Φ)	S	A	νZ.¬Φ∧[-]Z
		EG(!Φ)		E	νZ.¬Φ∧<->Z
		AG(!K)	A	A	νZ.[K]ff∧[-]Z
		EG(!K)		E	νZ.[K]ff ∧<->Z
	Uni-	AG(Φ)	S	A	νZ.Φ∧[-]Z
		EG(Φ)		E	νZ.Φ∧<->Z
		AG(K)	A	A	νZ.[K]tt ∧[-]Z
		EG(K)		E	νZ.[K]tt ∧<->Z
C	Uni-ver-	AG[Φ→AX(AF(ψ))]	S	A	νZ.Φ([-](ψ∨<->tt∧[-]Z))∧[-]Z
		EG[Φ→EX(EF(ψ))]		E	νZ.Φ(<->(ψ∨<->tt∧<->Z)) ∧ <->Z
		AG[K→AX(AF(L))]	A	A	νZ.[K]([-](<->tt∧[-L]Z))∧[-]Z
		EG[K→EX(EF(L))]		E	νZ.[K](<->(<-> tt∧<-L> Z)) ∧ <->Z
Live	Exist	AFΦ	S	A	μZ.Φ∨(<->tt∧[-]Z)
		EFΦ		E	μZ.Φ∨<->Z
		AF K	A	A	μZ.(<->tt∧[-K]Z)
		EF K		E	μZ.(<->tt∧<-K>Z)
	Boun	AGFΦ	S	A	νX.μY.<->Y∨(<->X∧Φ)
		EGFΦ		E	νX.μY.<->Y∨([-]X∨Φ)
		AGF K	A	A	νX.μY.[-]([<->tt∧[-K] Y]∨[-]X
		EGF K		E	νX.μY.<->([<->tt∧<-K>Y] ∨<->X
		A[ΦUψ]	S	A	μZ.ψ∨(Φ∧<->tt∧[-]Z)
		A[ΦWψ]		E	μZ.ψ∨(Φ∧<->tt∧<->Z)
		E[ΦUψ]		A	νZ.ψ∨(Φ∧[-]Z)
		E[ΦWψ]		E	νZ.ψ∨(Φ∧<->Z)
		A[KUL]	A	A	μZ.[-(K∪L)]ff∧<->tt∧[-L]Z)
		A[KWL]		E	νZ.[-(K∪L)]ff∧[-L]Z)
		E[KUL]		A	μZ.[-(K∪L)]ff∧<->tt∧<-L> Z)
		E[KWL]		E	νZ.[-(K∪L)]ff∧<-L>Z)
	Prec-	A[!ΦWψ]	S	A	μZ.ψ∨(!Φ∧ [-]Z)
		E[!ΦUψ]		E	νZ.ψ∨(!Φ∧<->tt∧[-]Z)
		A[!KWL]	A	A	νZ.[-(-K∪L)]ff∧[-L]Z)
		E[!KUL]		E	μZ.[-(-K∪L)]ff∧<->tt∧<-L > Z)
	Chain Prec-	!E[!ΦU(Ψ&!Φ &EX(EF(γ))	S	A	μZ.(ψ∧!P∧<->(Y∨<-> Z)∨ (! Φ ∧<->tt∧[-]Z)
		!E[!ΦW(Ψ&! Φ&EX(EF(γ))		E	μZ.(ψ∧!P∧<->(Y∨<-> Z) ∨ (! Φ∧ [-]Z)
		!E[!KU(L&!K &EX(EF(J))	A	A	μZ.([L∧!K]tt∧<->(<->tt∨<-J>Z)∨(<->tt∧[-K]Z)
		!E[!KW(L&!K &EX(EF(Jγ))		E	μZ.([L∧!K]tt∧<->(<->tt∨<-J>Z)∨(<->tt∧<-K>Z)
	Resp-onse	AG(Φ→AF(ψ))	S	A	νX.Φ(μY.Ψ∨(<->tt∧[-]Y)∧[-]X
		EG(Φ→EF(ψ))		E	νX.Φ(μY.Ψ∨<->Y)∧<->X
		AG(K→AF(L))	A	A	νX.[K](μY. (<->tt∧[-]Y)∧[-]X
		EG(K→EF(L))		E	νX.[K](μY.<-L>Y)∧<->X
	Chain Resp-	AG(Φ→AF(ψ &AX(AF(γ))	S	A	νX.Φ(μY.(Ψ∧[-]γ)∨<->tt∧[-]Y)∧[-]X
		EG(Φ→EF(Ψ &EX(EF(γ))		E	νX.Φ(μY.(Ψ∧<->γ)∨<->Y)∧<->X
		AG(K→AF(L &AX(AF(J))	A	A	νX.[K](μY.([-L]ff∧[-]][-J]ff)∨<->tt∧[-]Y)∧[-]X
		EG(K→EF(L &EX(EF(J))		E	νX.<K>(μY.([-L]ff∧[-]][-J]ff)∨<->Y)∧<->X

```
vX.[car](μY.<->tt∧[-ccross]Y)∧[-]X)
```

그리고 "자동차가 넘어갔다(Q)면 항상 다음으로 기차가 넘어가는 것(R)이 발생한다"는 Precedence 패턴을 적용하면

```
μZ.([Q]ff∨([-R]ff∧[-]Z∧<->tt))
```

이다. 그러므로 인과 관계의 무한 반복 속성을 갖는 공평성 속성 명세는

```
μX.vY1.([Q]ff∨([-ccross](vY2.(R∧X)∧[-ccross]Y2))
∧[-ccross]Y1
```

이다.

2.3 활용

복잡한 속성 명세일수록 패턴 기반 접근은 더욱 효율적이다. 다음은 CWB-NC(The concurrency Work Bench of the New Century)[5]에서 "기차길 신호기 시스템'에 대한 실제 요구 속성 명세의 사례이다. "언젠가 silent가 발생한다"는 의미의 속성을 양상 뮤 논리로 표현하고 이를 패턴의 포함 관계로 해석하면 다음과 같다.

```
prop eventually_silent = not ( min A1 = ( not (
 ['det:0] ( not ( max D = ( min E1 = ( not ( not ( min
F(not(['comm_out:2,'stat_out:2]ff)\/<'recovered:0>F)))/
\ ( <'tick:4>D \/ <-'tick:4>E1 ) ) ) ) ) ) \/ <->A1 ))
```

위의 복잡한 명세 내부에는 다음과 같은 패턴 포함관계가 있다.

Existence~Universally(~Response(~BoundedExistence ~(~Existence `recovered:0 or (comm_out:2 or ~statout:2) and tick:4-ff or tick:4)))))

본 연구에서는 ECW(Edinburgh Concurrency Workbench)와 CWB-NC의 실제 요구 속성 명세에 대한 분석을 진행했으며 그 결과는 표 2와 같다.

표 2 패턴 합계

패턴 유형	수
Absence	6
Universally	16
Existence	20
Bounded Existence	1
Response	6
Precedence	0
Response Chain	0
Precedence Chain	0
UNKNOWN	0
Total	43

3. 기존 연구와의 비교 및 기여도

Lamport 는 속성을 크게 안전성과 궁극성으로 분류하였다[6]. 그러나 이 분류는 너무 포괄적이어서 보다 세분화될 필요가 있다. Mana 는 선형 시제 논리를 위해서 조금 더 상세한 분류 구조를 제안했다[7]. 그러나 이 방법은 시제 논리 식의 구문을 중심으로 한 분류법이다. 식의 크기가 커져갈 때 구문보다는 의미적인 분류가 보다 바람직하다. 그래서 Dewyer 는 위의 두 분류를 보완해서 상세하고도 의미적인 기반의 분류법을 제시하였다. 그의 분류는 CTL, LTL 에는 잘 적용되었으나 우리가 대상으로 하는 양상 뮤 논리에는 부적절하다. 한편 Stirling 은 양상 뮤 논리의 속성을 크게 안전성, 궁극성, 공평성, 순환, 출현 회수로 분류했다[8]. 그러나 우리의 분류 구조는 다섯가지 속성을 모두 커버한다. 따라서 제시된 분류 구조를 사용한다면 양상 뮤 논리의 작성과 이해에 도움이 될 것이다.

4. 결론 및 향후 연구

본 논문에서는 양상 뮤 논리를 위한 속성 명세 패턴 연구를 통해 모든 명세 논리를 포괄하는 단일한 프레임워크를 제공하였다. 결과적으로 양상 뮤 논리의 복잡한 실제 명세의 분석 및 이해와 표현 용이했으며 상태와 행동 중심의 속성 및 경로 중심의 세부적 분류의 확장을 통해 보다 복잡한 양상 뮤 논리의 분석 및 표현을 가능하게 했다. 또한 새로운 고급 명세 언어 및 접근법의 연구를 가능하게 한다. 향후 연구 과제는 다음과 같다. 첫째, 명세 패턴의 안전성 보장을 위한 실제 명세의 수집, 분류 및 통계 작업을 진행할 것이다. 둘째, 명세 패턴의 확장을 위해 속성, 패턴, 의미, 표현 간의 규칙성을 증명하며 이를 컴포넌트 기반 방법론에 적용할 것이다. 셋째, 양상 뮤 논리를 위한 명세 패턴 기반의 속성 명세 자동 시스템을 구현하고자 한다.

참고문헌

[1] E.M. Clarke et.al, Model Checking, MIT Press, 1999
[2] E.A. Emerson, "Model Checking and the Mu-Calculus," Proceedings of the DIMACS Symposium on Descriptive Complexity and Finite Models, pp. 185--214, 1997
[3] J.C. Corbett, et.al, "A Language Framework For Expressing Checkable Properties of Dynamic Software," Proceedings of the SPIN Software Model Checking Workshop, LNCS 1885, 2000.
[4] M.B. Dwyer, et.al, "Property Specification Patterns for Finite-State Verification," Proceedings of the Workshop on Formal Methods in Software Practice, 1998
[5] R. Cleaveland, The Concurrency Workbench of the New Century, http://www.cs.sunysb.edu/~cwb/
[6] L. Lamport, "Proving the correctness of multiprocess Programs", IEEE Transactions on Software Engineering, SE-3(2):125-143,1977
[7] Z. Manna, et.al, The Temporal Logic of Reactive and Concurrent Systems, SpringerVerlag, 1992
[8] C. Stirling, Modal and Temporal Logics for Processes. Lecture Notes in Computer Science 1043, pp.149-237, 1996