L* 기반의 Assume Guarantee Reasoning 의 개선

이정재*, 최진영*
*고려대학교 컴퓨터학과
e-mail : jjlee@formal.korea.ac.kr

Improvement of L*-based Assume Guarantee Reasoning

Jungjae Lee*, Jin-Young Choi*
*Dept. of Computer Science, Korea University

요 약

L* 기반의 Assume-Guarantee Reasoning[1]은 시스템 검증 시의 상태 폭발(state explosion)을 줄이는 데 크게 기여하였다. 그러나 시스템 검증에 소모되는 시간은 일반적인 모델 체킹 도구를 사용할 때보다 크게 증가시킨다. 본 논문에서는 시스템의 검증에 소모되는 시간을 줄이기 위하여 L* 기반의 Assume-Guarantee Reasoning 의 개선안을 제안하였다.

1. 서론

[1]의 연구는 Angluin 의 L* algorithm[3]을 사용하여 Assume Guarantee rule(AG rule)[2]에 사용되는 적절한 가정 A 을 찾아내는 완전 자동화된 방법론을 제안하였다. L*는 알려지지 않은 정규 언어 U 에 대한 포함관계를 알 수 있는 문자열들을 이용하여 Minimally Adequate Teacher (MAT)와 반복적으로 상호작용을 통하여 U 를 인식하는 Deterministic Finite Automaton(DFA)를 생성하는 알고리즘이다. [1]의 연구에서는 AG rule 을 만족하는 가정 A 를 찾기 위해서 먼저 모든 문자열을 인식하는 후보자를 MAT 에게 제공하여 A 인지 확인한다. A 가 아닐 때에는 MAT 가 제공하는 반례를 이용하여 새로운 후보자를 생성하고, 다시 MAT 에게 A 인지 확인하는 일을 반복하여, 두 조건을 모두 만족시키는 A 를 자동으로 찾는다.

[1]은 시스템의 검증을 위해 사용되는 상태의 수를 크게 줄여주어 기존의 모델 체킹 도구에서 상태 폭발로 인하여 검증할 수 없었던 시스템들을 일부 검증 가능하게 해주었지만 검증에 소모되는 시간은 기존의 모델 체킹 도구에서보다 크게 증가시킨다. 이러한 가장 큰 이유는 L* algorithm 이 새로운 후보자를 생성하는 과정에서 특정 문자열들이 U 에 속하는지를 MAT 에게 질문하게 되는데, 이 때에 MAT 는 이를 확인하기 위하여 매번 주어진 문자열을 모델에서 시뮬레이션해야하고, 새로운 후보자가 생성되었을 때에도 후보자가 A 인지를 확인하는 과정에서 매번 모델 체킹을 해야하기 때문이다.

본 논문에서는 [1]을 이용하여 시스템을 검증할 때에 시스템의 검증에 소모되는 시간을 줄이기 위한 새로운 개선안을 제안한다.

- 본 연구는 교육과학기술부/한국과학재단 우수연구센터 육성사업 (R11-2008-007-03002-0)과 한국소프트웨어진흥원의 SW 공학 요소기술 개발과 전문인력 양성사업의 지원으로 수행되었음

2. L* 기반의 Assume Guarantee Reasoning 의 소개

2.1 L* algorithm

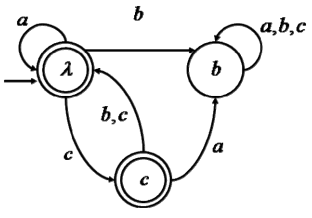
L*는 U 를 학습을 위하여 아래의 두 가지 종류의 질문에 'True' 또는 'False' 대답을 해주는 MAT를 필요로 한다.

1. 포함관계 질문(Membership query):
   $\omega \in \sum^*$에 대하여, $\omega \in U$ 인가?
2. 후보자 질문(Candidate query):
   후보자 DFA C에 대하여, $L(C) = U$ 인가?

본 논문에서는 개선안과 직접적으로 관련이 있는 포함관계 질문만을 설명한다. L*는 U 를 학습하기 위하여 필요한 정보(ω 에 대한 MAT 의 포함관계 질문의 대답)들을 기록하는 관찰표(Observation Table)를 생성하고 관찰표가 닫힐(close)때까지 MAT 를 통하여 관찰표의 문자열들이 U 에 포함되어있는지를 확인 후, 이를 이용하여 대응하는 후보자 DFA 를 생성한다. 아래의 그림 1 의 (a)는 닫혀있는 관찰표를 나타내고, (b)는 (a)에 대응하는 후보자 DFA 이다.

		E	
		λ	a
S	λ	True	True
	b	False	False
	c	True	False
S·Σ	a	True	True
	b	False	False
	c	True	False
	ba	False	False
	bb	False	False
	bc	False	False
	ca	False	False
	cb	True	True
	cc	True	True

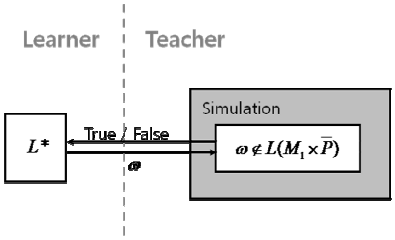
(a)



(b)

(그림 1)닫혀있는 관찰표와 대응하는 DFA 의 예시

2.2 MAT의 포함관계 질문 확인



(그림 2) MAT의 포함관계 질문 구현

MAT는 L*가 질문하는 w에 대하여 $\omega \in U$ 인지 아닌지를 확인하기 위하여 $w \notin L(M_1 \times \overline{P})$ 인지를 시뮬레이션해본다. 만약 $w \notin L(M_1 \times \overline{P}\downarrow_{\Sigma})$ 라면 'True'라 대답을 하고, $w \in L(M_1 \times \overline{P})$ 라면 'False'라고 대답하는데 L*는 이를 관찰표에 기록한다. 위의 (그림 2)는 이러한 과정을 나타낸다.

3. 포함관계 질문 수의 감소를 위한 개선안

L* 알고리즘은 알려지지 않은 정규언어를 인식하는 DFA 를 생성하는 알고리즘이다. 때문에 [1]에서는 LTS 로 변환된 모델과 속성을 DFA 로 변환하여 Assume Guarantee rule 을 만족하는 적절한 *L(A)*를 학습하고 이를 인식하는 DFA A 를 생성한다. 이 때에, LTS 로부터 변환된 DFA 는 다음과 같은 속성을 가진다.

속성 1. LTS 로부터 변환된 DFA 의 집합을 DFA_{LTS} 라 할 때, $\forall \omega \in \Sigma^*, \forall e \in \Sigma^*, \forall M \in DFA_{LTS}$ 에 대하여 $\omega \in L(\overline{M}) \Rightarrow \omega \cdot e \in L(\overline{M})$ 이다.

증명 1. LTS $M_{LTS} = (s_0, S, \Sigma, \delta)$ 로부터 변환된 DFA $M_{DFA} = (s_0', S', \Sigma', \delta', F)$ 은 아래의 조건을 만족한다.

1. $s_0' = s_0$
2. $S' = S \cup \{s_{trip}\}$
3. $\Sigma' = \Sigma$
4. $F = S$
5. $\delta' = \delta \cup \{(s_{trip}, \alpha, s_{trip}) \mid \alpha \in \Sigma\} \cup \{(s, \alpha, s_{trip}) \mid \forall s \in S. \nexists s' \in S. \forall \alpha \in \Sigma. (s, a, s') \in \delta\}$

또한 $\overline{M_{DFA}} = (S_0'', S'', \Sigma'', \delta'', F')$ 은 아래의 조건을 만족한다.

6. $s_0'' = s_0'$
7. $S'' = S'$
8. $\Sigma'' = \Sigma'$
9. $F' = S' - F$
10. $\delta'' = \delta'$

위의 조건 9 에서 결국 $F' = \{s_{trip}\}$ 의 하나의 상태가 된다. 또한 조건 5 에서 $\forall \alpha \in \Sigma. \delta(s_{trip}, \alpha) = s_{trip}$ 이므로 결국 한번 s_{trip} 으로 전이하면 다시는 다른 상태로 전이할 수 없게 된다. 즉. 임의의 문자열 ω 에 대하여 $\omega \in L(\overline{M})$ 이면 현재 상태는 s_{trip} 이므로 임의의 문자 $\alpha \in \Sigma$ 에의 하여 전이 하더라도 현재 상태는 항상 s_{trip} 에 머물게 된다. 따라서 $\omega \cdot e \in L(\overline{M})$ 이다.

□

위의 속성 1 을 직관적으로 설명하면, 모든 LTS 로부터 변환된 DFA M 에 대하여 $\overline{M}$ 이 승인하는 모든 문자열 w 는 어떠한 문자열 e 가 접미사로 붙더라도 $\overline{M}$ 이 승인한다는 의미이다. 이러한 특성을 L*에 적용하면 관찰표를 업데이트 하는 과정에서 MAT 에 요청하는 w 의 포함관계 질문의 수를 줄일 수 있다.

MAT 의 포함관계 질문의 구현에서 MAT 는 L*가 질문하는 w 에 대하여 $w \in L(M_1 \times \overline{P})$ 라면 'False'라는 대답을 해주어야 한다. 속성 1 을 이용하여, 만약 어떠한 w 에 대해서 MAT 가 이미 'False'라는 대답을 해주었다면 그 이후에 포함관계 질문을 하는 w 를 접두사로 가지는 모든 w' 에 대하여 $w' \in L(M_1 \times \overline{P})$ 이 성립할 것이다. 이러한 모든 w' 에 대한 포함관계 질문은 MAT 가 시뮬레이션을 통하여 포함관계를 확인해 볼 필요가 없이 L*가 자신의 관찰표를 보고 스스로 확인할 수 있다. 기존의 MAT 에서 시뮬레이션을 통하여 $w' \in L(M_1 \times \overline{P}\downarrow_{\Sigma})$ 인지를 판별한 것에 비해서 본 개선 방안은 MAT 에 대한 포함관계 질문을 줄임으로써 관찰표를 갱신하는데 걸리는 시간을 줄일 수 있다.

4. 결론

본 논문에서는 [1]의 개선안으로 LTS 로부터 변환된 DFA 의 특성을 이용한 포함관계 질문 수 감소 방안을 제시하였다. 이를 통하여 새로운 후보자를 생성하는데 소모되는 시간이 줄어들 것이다.

향후 과제로 본 논문에서 제안한 개선 방안을 직접 구현하여 개선 정도를 직접 실험할 예정이다.

참고문헌

[1] Cobleigh, J.M., Giannakopoulou, D., Pasareanu, C.S. "Learning assumptions for compositional verification". In Proc. of Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Volume 2619 ,pp.331-346, 2003.

[2] Thomas A. Henzinger, Shaz Qadeer, Sriram K. Rajamani, Serdar TasiranO, "An assume-guarantee rule for checking simulation" ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 24, Issue 1, pp.51 - 64, 2002

[3] D. Angluin, "Learning regular sets from queries and counterexamples.", Information and Computation, Volume 75, Issue 2, pp.87–106, 1987.