모델 추상화를 이용한 네모라이즈 게임 풀이*

이정림 권기현
경기대학교 정보과학부
{jrim, khkwon}@kyonggi.ac.kr

Solving Nemorize Games via Model Abstraction

Jungrim Lee , Gihwon Kwon
Computer Science Department, Kyonggi University

요 약

모델 체킹에서 모델의 크기가 커질수록 검사해야 할 상태 공간이 지수적으로 증가하는 것을 상태폭발문제라고 부르며, 이 문제를 해결하기 위해 추상화 기법이 사용된다. 본 논문에서는 네모라이즈 게임을 대상으로 추상화 기법을 적용하여 게임 모델의 상태 공간을 줄여 기존 방법으로 풀지 못했던 게임을 풀었다. 이 게임은 한붓그리기처럼 출발지부터 이동가능한 모든 지점을 한번만 거쳐, 목적지까지 가는 경로를 찾아내는 도달성 게임이다. 이 게임은 Esterel 언어로 모델링 되었다. Esterel은 동기적 언어로써 게임을 유한상태모델로 모델링하고 관련 모델 체커인 Xeve를 사용하여 모델을 검사한다. Xeve는 모델 체킹 후 특정 출력신호를 방출하기 위한 입력신호들의 시퀀스를 생성해준다. 이 시퀀스가 게임의 해답인 경로가 되는데 Xes라는 시뮬레이터를 통해 실제 정확한 해답인지를 확인한다.

1. 서 론

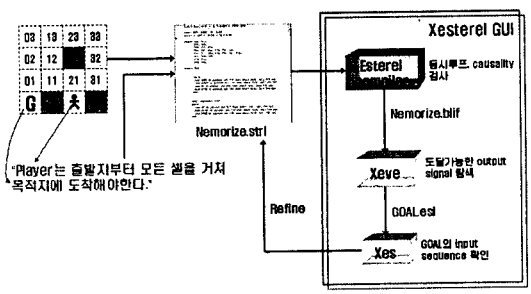
네모라이즈는 핸드폰에 탑재되는 보드게임의 일종으로 한붓그리기처럼 플레이어가 이동가능한 모든 지점을 한번만 거쳐 목적지까지 가는 도달성 게임이다. 도달성 게임이란, 게임 규칙을 준수하면서 초기 상태에서 목표 상태로 가는 경로를 찾는 것을 말한다. 우리는 이런 도달성 게임을 모델 체킹에 적용하여 자동으로 게임을 푸는데 관심이 많다. 게임을 풀기위해 게임과 관련된 사항을 유한상태모델(예를 들어 초기상태, 게임규칙, 상태 공간 등)로 표현하고 목표 상태에 도달가능한지를 속성으로 표현하여 모델체커에 입력한다. 도달성 게임에서 이 속성은 "Safe until Goal"의 형태로 표현된다. 그리고 목표 상태로 가는 경로가 존재한다면 모델체커는 그 경로를 출력해준다.[1]

본 논문에서는 Esterel 환경을 사용하여 네모라이즈 게임을 푼다. Esterel은 임베디드 시스템, 통신 프로토콜, 제어 회로와 같은 실시간 반응형 시스템을 위한 동기적 프로그래밍 언어이다. 또한 관련도구인 Xeve, Xes는 각각 Esterel의 속성검증과 시뮬레이션을 담당하는 모델체커와 시뮬레이터이다. Esterel 프로그램은 현재 상태와 입력신호에 따라 출력신호를 방출하는 Mealy모델로 표현되고 속성은 출력신호들의 조합으로 표현된다. 따라서 목표 상태까지 도달 가능함을 출력신호로 표현해서 목표 상태까지 가는 경로가 존재하면 Xeve는 그 출력신호가 방출가능하다는 것과 함께 목표 상태로 도달되는 입력신호의 시퀀스를 증거로 출력한다. 증거로 출력된 이 시퀀스가 게임의 해답인 경로가 되는 것이다[2][3].

하지만 게임의 크기가 커질 경우 상태폭발문제가 발생해서 해답을 못 찾아내거나 오랜 시간 상태공간을 탐색해야하는 어려움이 있었다. 따라서 본 논문에선 상태공간을 줄이기 위해 추상화 기법을 적용하여 풀지 못했던 게임을 풀어낸다. 2장에서는 네모라이즈 게임풀이를 위한 프레임웍과 게임 모델을 정의하고 3장에선 상태공간을 줄이기 위해 게임 모델에 추상화를 수행한다. 4장에서는 실험을 통한 결과를 분석하고 5장에서 결론을 맺는다.

2. 게임풀이

우리는 게임을 풀기위해 Xesterel이라는 Esterel 통합개발환경을 사용한다. 게임풀이 프레임웍은 아래 그림 1과 같다.



(그림 1) 게임풀이 프레임웍

게임을 Esterel로 모델링하고 Esterel compiler를 통해 유한상태모델로 변환한다. 이때 Esterel은 외부 입력에 대한 처리시간 없이 출력이 동시에 발생되기 때문에 모델이 동시루프에 빠지지 않는지, 인과관계(causality)가 올바른지를 검사한다. 그 다음, 변환된 모델을 Esterel 모델체커 Xeve에 입력한 후 목표상태의 출력신호(GOAL)가 방출 가능한지 안전성 검사를 한다. 이 출력신호는 목표 상태에 도달 가능해야만 방출되는 것이다. 이 출력신호가 방출가능하다면 출력신호를 방출하기위한 입력신호의 시퀀스를 생성해준다(GOAL.esi). 마지막으로 시뮬레이터 Xes를 통해 시퀀스가 올바르면 게임

* 본 연구는 한국과학재단 특정기초연구 (R01-2005-000-11120-0) 지원으로 수행되었음.

의 해답인 경로가 되고 아니면 모델을 다시 정제한다. Esterel로 명세할 게임 모델 G는 다음과 같다.

$$G=\langle P, I, O, C, E, goal\rangle$$

- P : 플레이어의 행위에 대한 유한상태모델이다.
- $I=\{left, right, up, down\}$ 플레이어의 이동방향을 나타내는 입력신호($i\in I$)의 집합이다.
  - $\tilde{I}=I^{+}$ 의 시퀀스이다. $(\forall \tilde{i}\in\tilde{I}.\ |\tilde{i}|\geq 2)$
- $O=\{OUT_{00}, \ldots\ldots, OUT_{mn}, GOAL\}$ 이동 가능한 위치에서 내보내는 출력신호의 집합이다.
- $C:(x,y)\rightarrow 2^{I}/\varnothing$ 임의의 위치를 입력받아 그 위치에서 벽이 아닌 이동 가능한 방향을 돌려주는 함수이다.
- $E\subseteq O$ 현재까지 방출된 출력신호들의 집합이다.
- $goal$: 목적지의 위치를 (x_g, y_g)의 순서쌍으로 표현한다.

$P=\langle S, s_0, \delta\rangle$는 다음과 같다.

- $S=\{(x,y)|0\leq x<n, 0\leq y<m\}$는 $n\times m$의 플레이어의 이동 가능한 위치집합이다.
- $s_0\in S_O$ 플레이어의 초기위치이다.
- $\delta: S\times I\rightarrow S\times O$ 플레이어의 전이함수이다. 전이 함수 $\delta((x,y),d)=((x',y'),OUT_{x'y'})$는 아래와 같이 정의한다.

$$\delta\equiv\left[\begin{array}{l}((x>0)\wedge(i=left)\wedge(left\in C(x,y))\wedge\\(OUT_{x'y'}\notin E)\wedge(x'=x-1)\wedge(y'=y)\wedge OUT_{x'y'})\\\vee\\((x<n-1)\wedge(i=right)\wedge(right\in C(x,y))\wedge\\(OUT_{x'y'}\notin E)\wedge(x'=x+1)\wedge(y'=y)\wedge OUT_{x'y'})\\\vee\\((y<m-1)\wedge(i=up)\wedge(up\in C(x,y))\wedge\\(OUT_{x'y'}\notin E)\wedge(x'=x)\wedge(y'=y+1)\wedge OUT_{x'y'})\\\vee\\((y>0)\wedge(i=down)\wedge(down\in C(x,y))\wedge\\(OUT_{x'y'}\notin E)\wedge(x'=x)\wedge(y'=y-1)\wedge OUT_{x'y'})\end{array}\right]$$

이 게임을 종료하기 위한 조건은

$$((x=x_g)\wedge(y=y_g)\wedge(E=\{OUT_{00}, \ldots\ldots, OUT_{mn}\})\Rightarrow GOAL$$

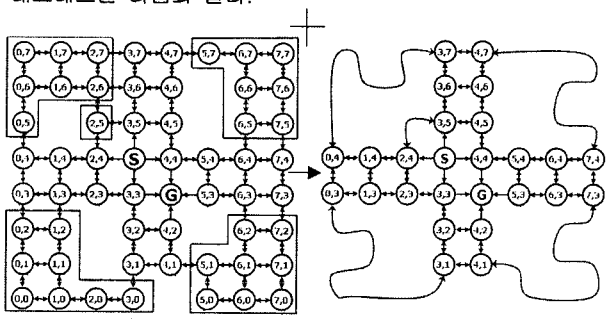
이다. 만약 $GOAL$이 방출 가능하다면 이 $GOAL$을 방출하기 위한 입력신호(i)의 시퀀스($\tilde{i}$)를 생성해준다. 이것이 게임의 해답이 된다. 하지만 게임의 단계가 올라갈수록 모델의 크기가 커져 검사해야할 상태수가 증가하는 상태폭발문제가 발생하였다[4]. 따라서 이 상태폭발 문제를 해결하기 위해 우리는 모델에 추상화를 수행한다.

3. 게임풀이를 위한 추상화

추상화는 원래의 시스템의 행위를 가지고 있는 좀 더 적은 크기의 모델을 생성한다. 시스템의 크기는 작지만 동일한 행위를 하는 모델을 동치 관계에 있다고 한다. 추상화된 모델이 원래 모델보다 좀 더 많은 행위를 가지고 있다면 원래 모델과 상위 근사화(over approximation) 관계에 있는 추상화된 모델이 생성 된 것이다. 만일 원래 모델보다 작은 행위를 가지고 있는 추상화된 모델을 생성했다면 하위 근사화(under approximation) 관계에 있는 모델을 생성한 것이다. 이 관계를 가지고 있는 추상화된 모델을 생성했다면 상위 근사화 관계에 있는 모델은 원래 모델의 행위를 모두 따라할 수 있다. 그리고 추가적인 행위를 가지고 있게 된다. 비록 행위의 수는 많게 되지만 일반적으로 추상화 된 모델의 크기는 원래 모델의 크기보다 적은 모델을 생성할 수 있다. 이 경우 추상화된 시스템에서 속성을 만족한다면 적은 행위를 가지는 원래 시스템에서도 속성을 만족한다. 하지만 속성을 만족하지 않는다면 그것이 원래 모델에 있는 행위 때문에 속성을 만족하지 않는 것인지, 추가된 행위 때문에 속성을 만족하지 못하는 것인지 모르게 된다.

한편, 하위 근사화 관계에 있는 추상화된 모델은 원래 시스템에 비해 추상화된 시스템이 적은 행위를 가지고 있다. 따라서 적은 행위에서 속성을 만족한다고 해서 원래 시스템에서 속성이 만족된다고 할 수는 없다. 추상화된 모델에서 속성이 만족되지 않는다면 추상화된 모델 보다 더욱 많은 행위를 가지고 있는 원래 모델에서는 당연히 속성을 만족하지 않게 된다. 우리는 추상화된 모델에서 속성이 만족하지 않는지에 관심이 있다. 만약 목표 상태에 도달가능하지 않으면 모델을 다시 정제하고 만약 목표 상태에 도달가능하면 추상화된 부분을 펼쳐서 원래 모델에서도 만족하게 됨을 보인다. 따라서 게임의 상태공간을 줄이기 위해 원래 모델과 하위 근사화 관계를 갖는 추상화를 수행한다.

우리는 네모라이즈 게임 10단계에서 상태폭발을 경험했다. 네모라이즈 10단계의 원래 모델과 추상화된 모델에 대한 상태그래프는 다음과 같다.



(그림 2) 원래 모델과 추상화된 모델

그림 2와 같이 게임을 상태 그래프로 표현해보면 일반화된 추상화 규칙들을 발견할 수 있다. 먼저, 시작상태와 도착상태를 제외한 모든 상태들 간의 전이는 양방향성을 갖는다.

$\forall s, s', s''\in S/\{s_0, goal\}$일 때,
$s''\leftrightarrow s\leftrightarrow s'$ ($s_0\not\rightarrow s$, s', s''는 s와 인접한 상태)

임의의 상태 s에서 이동가능한 방향의 개수 $n\in N$를 돌려주는 함수는 $f: S\rightarrow N$ 이다. 이때 $n=2$인 경우, s는

$s''\rightarrow s\rightarrow s'$ 또는 $s''\leftarrow s\leftarrow s'$

의 두개의 방향성만을 갖는다. 따라서 s'과 s''의 이동은 transitive하기 때문에 s를 제거하고 s'과 s''은 아래와 같이 직접 연결가능하다.

$s'' \leftrightarrow_a s'$ ($\leftrightarrow_a$는 추상화된 전이)

추상화된 전이 $\leftrightarrow_a$는 두 개 이상의 기본전이로 이루어지기 때문에 입력신호의 시퀀스인 $\tilde{i} \in \tilde{I}$를 갖는다. 또한 $\tilde{i}$의 역방향 시퀀스는 $\tilde{i}^{-1} \in \tilde{I}$로 나타낸다. 또한 추상화된 전이 $\leftrightarrow_a$는 결정적이기 때문에 기본전이보다 먼저 선점되어야 교착상태에 빠지지 않는다. 본 게임에서 교착상태란 목적지가 아닌 곳에서 더 이상 이동 가능하지 않은 상태를 말한다.
정리하면 아래와 같은 4개의 추상화 규칙을 얻을 수 있다.

1) $s'' \leftrightarrow s \leftrightarrow s' \Rightarrow s'' \leftrightarrow_a s' \ (n=2)$
2) $s'' \leftrightarrow_a s \leftrightarrow s' \Rightarrow s'' \leftrightarrow_a s' \ (n=2)$
3) $s'' \leftrightarrow_a s \leftrightarrow_a s' \Rightarrow s'' \leftrightarrow_a s' \ (2 \le n \le 4)$
4) $s \leftrightarrow_a s' \wedge s \leftrightarrow s' \Rightarrow s \leftrightarrow_a s'$

3)의 규칙에서 s는 추상화된 전이 2개 외에 기본 전이도 가질 수 있지만 추상화하면서 s와 함께 제거된다. 1),2),3)규칙에 의한 전이함수 $\delta(s,\tilde{i})$는 다음과 같다.

$\delta(s'',\tilde{i}) = s', \ \delta(s',\tilde{i}^{-1}) = s'' \ (\tilde{i}, \tilde{i}^{-1} \in \tilde{I}, \ \tilde{i} \neq \tilde{i}^{-1})$

예를 들어, 그림 2의 상태 $(2,0)$에서 $f(2,0)) = 2$ 이다.

$\delta((1,0),\{right,right\}) = (3,0), \ \delta((3,0),\{left,left\}) = (1,0)$

또한, $s'' \leftrightarrow_a s \leftrightarrow s' (n>2)$에서 $s'' \rightarrow_a s \rightarrow s'$으로 전이가 일어나는 경우 s는 다음 상태 s'을 결정할 수 있다. 교착상태에 빠지지 않기 위해 이동가능한 방향의 개수가 1인 s'을 결정한다. 만약 그런 s'이 존재하지 않으면 s'은 비결정이 된다. (여기서 $s'' \leftarrow_a s \leftarrow s'$의 경우는 고려하지 않는다.)
현재 상태 s로부터 다음 상태 s'에서의 이동가능한 방향의 개수를 돌려주는 함수는 $f' : (\delta(s,i)) \rightarrow N$ 이다.
따라서 5번째 추상화 규칙은 다음과 같다.

5) $s'' \leftrightarrow_a s \leftrightarrow s' \Rightarrow s'' \leftrightarrow_a s' \ (n>2, \ f'(\delta(s,i)) = 1)$

그림 2의 추상화한 하위 근사화 모델에서 목표 상태까지 도달 가능하더라도 원래 모델에선 만족하지 않는다. 하지만 추상화 규칙에 의해 줄인 부분을 펼쳐서 추상화 수행 시 얻어낸 경로$(\tilde{i}, \tilde{i}^{-1})$를 적용하면 원래 모델에서도 도달가능하게 된다.

4. 실험

앞에서 정의한 추상화 규칙이 적용된 네모라이즈 10단계의 상태그래프는 위의 그림 2에서 볼 수 있다.
네모라이즈 게임 10단계에 추상화규칙 1)을 한번 적용하여 추상화한 모델을 1단계, 이어서 추상화규칙 3)으로 정제한 모델을 2단계, 그리고 더 이상 추상화가 안 될 때까지 각 규칙을 적용하여 그림 2와 같은 3단계 모델을 얻었다. 추상화 모델의 수행결과는 아래 표 1과 같다.

(표 1) 네모라이즈 게임 10단계 수행결과

추상화	1단계	2단계	3단계
Step	94	86	50
Reachable States	412,443,191	33,761,977	156,390
Size of BDD of Reachable	81,864	45,676	13,998
time	26:53:49	02:58:52	00:04:59
최대 Memory 사용량	831,812KB	296,856KB	40,248KB

*시스템 사양: Windows XP, CPU-2.6Ghz, RAM-1GB

표 1은 게임의 복잡도를 점진적으로 줄이는 추상화 단계를 보인다. 상태의 수가 감소함에 따라 상태공간은 지수적으로 감소하며 상태 공간을 탐색하는 시간도 상당히 줄어드는 것을 알 수 있다. (참고로, 1단계 추상화 모델을 범용 모델 체커 NuSMV[5]로 검사한 경우에는 상태폭발문제가 발생했다.)
따라서 우리는 각 단계별로 추상화를 수행하여 게임의 복잡도를 낮추고 Esterel을 이용해 빠른 해답을 얻어내었다.

5. 결론

핸드폰에 탑재되는 네모라이즈 게임은 게임규칙을 준수하면서 초기상태에서 목적 상태까지 가는 경로를 찾아내는 도달성 게임이다. 이 게임은 Esterel 환경에서 모델 체킹을 이용해 경로를 찾아낸다. 하지만 모델의 크기가 커질수록 상태폭발문제가 발생하였다. 따라서 본 논문에서 상태공간을 줄이기 위한 추상화기법을 제시하였다. 세 가지 추상화 규칙을 이용하여 원래 모델과 하위 근사화 관계에 있는 추상화 모델을 만들어서 풀지 못했던 단계들을 풀어냈다. 하지만 이 추상화 규칙만으로는 아직 풀지 못한 단계들이 있기 때문에, 현재 네모라이즈 게임의 모든 단계를 다루기 위한 추상화 연구가 진행 중에 있다.

참 고 문 헌

[1] 권기현, "모델 검증을 이용한 게임 풀이", 정보과학회학회지, 제21권 제1호, pp.7-14, 2003.
[2] G. Berry, "The Esterel v5 Language Primer," Version v5_91, July 27, 2000.
[3] Amar Bouali, "Xeve: an Esterel Verification Enviornment," Version v1_3, December, 1997.
[4] E. M..Clarke, O. Grumberg, S. Jha, Y. Lu and H. Veith, "Progress on the State Explosion Problem in Model Checking,"in Proceedings of 10 Years Dagstuhl, LNCS 2000, pp.154-169, 2000.
[5] Y. Lu, Automatic Abstraction in Model Checking, Ph.D. thesis, Carnegie Mellon University, Department of Electrical and Computer Engineering, 2000.