

상태 감소를 위한 추상화 기법

박지연^{*0}, 이정희^{*}, 이문근^{**}

전북대학교 컴퓨터학과^{*}, 전북대학교 전자정보공학부^{**}

{jypark, jehlee, mklee}@cs.chonbuk.ac.kr

A Abstraction Method for State Minimization

Ji-yeon Park^{*0}, Jeong-hee Lee^{*}, Moon-kun Lee^{**}

^{*} Dept. of Computer Science, Chonbuk National Univ.

^{**}Division of Electronics and Information Engineering, Chonbuk National Univ.

요약

상태 기반 정형 기법이 가진 문제점 중 하나인 상태 폭발 문제를 해결하기 위해, 본 논문에서는 명세 모델과 실행 모델을 정의하고 각 모델에 적용되는 추상화를 정의한다. 명세 모델은 구문을 바탕으로 추상화로 비논리 내부연산 추상화와 구조 추상화를, 실행 모델은 시간, 자료, 위치가 가진 의미를 통해 추상화하는 시간, 자료, 위치 공간 추상화를 수행한다. 예제에 추상화를 적용하여 상태 감소와 계층성 생성, 복잡도 감소의 과정을 보인다.

1. 서론

상태 기반 정형 기법에서 발생하는 상태 폭발 문제[1, 2]는 상태 기계로 표현된 시스템의 행위에 대한 이해와 분석을 어렵게 한다. 특히, 실시간 시스템에서는 실시간 속성으로 인하여 비(非)실시간 시스템에 비해 상태 공간의 복잡도가 높으며 이에 따라 실시간 시스템의 이해와 분석도 더욱 어렵게 된다.

상태 폭발 문제를 해결하기 위해 많은 연구[1, 2, 3, 4, 5, 6, 7, 8, 9]가 진행되고 있으며 시스템의 실행 시 발생하는 상태 또는 상태 공간의 크기 감소를 주요 목적으로 하고 있다. 그러나 이러한 연구들은 상태감소가 발생한 원인과 내용이 상세히 기술되지 않은 단점과 감소되지 않은 상태들 중에 존재하는 상태 감소의 가능성을 해결하지 않은 제약점들을 가지고 있다.

본 논문은 이러한 단점과 제약점들을 효과적으로 해결하기 위해 가능한 모든 경우의 상태감소를 보장하고 각 상태감소의 근거와 내용을 표현할 수 있는 새로운 상태감소 방법을 제안한다.

상태감소 방법으로 명세 모델 기반 구문 추상화인 비조건 내부 연산(UIC: *Unconditionally Internalized Computational*) 추상화와 구조(*Structural*) 추상화, 실행 모델 기반 의미 추상화인 위치(*Locational*) 공간 추상화와 자료/시간(*Data/Temporal*) 공간 추상화를 정의한다. 정의한 추상화의 장점은 명세 모델에서 구문적 추상화를, 실행 모델에서 의미적 추상화를 수행하여 실행 모델이 가진 추상화 정보를 명세 모델의 추상화된 정보로부터 설명할 수 있다. 또한, 추상화 정보는 모두 계층적으로 표현되기 때문에 계층간의 상태 변화도를 이해할 수 있다 그리고, 개층적으로 상위 단계에서는 낮은 복잡도를 가지고 시스템의 개략적 분석을 수행할 수 있으며, top-down 또는 bottom-up 방식을 통하여 원하는 계층 단계에서의 시스템을 이해할 수 있게 한다.

2. 관련 연구

실시간 시스템을 명세하고 상태 폭발 문제 해결에 초점을 두고 시스템의 속성을 검증하기 위해 다양한 연구가 있다.

[2]는 구조적 계층성을 이용하여 서브시스템의 세부 사항을 숨김으로써 다수의 시스템의 조합에 의해 발생 가능한 상태 폭발을 방지한다. 그러나, 제어 결정이 데이터 값과 독립적으로 수행된다는 가정에 의해 실제 시스템에 적용하기 어렵다.

[3]은 데이터 도메인을 다수의 동치 클래스로 클러스터링하는 상태 감소 방법을 제안한다. 그러나, 실시간 시스템과 같이 많은 변수가 사용되는 시스템에는 적용이 어렵다. 또한, 시스템의 시간 속성을 정의하지 않기 때문에 실시간 시스템에는 적용될 수 없으며 시스템의 모든 변수를 고려하기 때문에 불필요한 데이터 도메인을 생성할 수 있다.

[4]는 시간 상태 공간의 처리를 통해 무한한 도달성 그래프를 유한한 도달성 그래프로 변환한다. 데이터와 제어 공간은 고려하지 않아 실제 적용하기에는 어려움이 있다.

[5]은 TA(*Timed Automata*)로 모델링된 시스템의 상태 공간을 표현하는 *Region Graph*의 크기를 감소하기 위해 TA가 가지는 시간에 대한 동치 관계를 적용하여 시간 공간을 여러 파티션으로 분할한다. 상태는 시간과 위치 정보로 표현되어 변수의 값이 실행과 검증에 영향을 주는 소프트웨어의 정형적 모델링에는 적용하기 어렵다.

[6]은 TA에 대한 상태 최소화 방법으로 *history equivalence*와 *transition bisimulation* 개념을 사용하여 시간 개념이 존재하는 상태를 시간 개념이 없는 오토마타에 의해 생성된 상태로 만든다. 그 결과 데이터 공간이나 이벤트에 대한 도달성 분석은 가능하다 시간 공간에 대한 분석은 고려할 수 없다.

[7]은 상태 집합에 대해 특정한 함수 관계를 적용하여 사상된 상태의 집합에 의해 이루어지는 추상화 모델을 제안한다.

[8]의 연구는 '*largest equivalence relation*' 개념을 사용하여 도달성 그래프 생성과 상태 감소를 동시에 수행하고자 하였다.

[9]에서의 디지털 하드웨어에 대한 상태 감소 방법은 머신의 집합을 임출력의 값에 대한 *bisimulation* 관계를 통해 동치 클래스를 생성하여 상태 감소를 한다. 이 방법은 임출력이 다양한 소프트웨어의 상태 감소에 적용하기 어렵다.

3. 명세 모델 기반 추상화

명세 모델을 정의하면 다음과 같다.

정의 3.1: 시스템은 실시간 상태 기계인 RSM $M = \langle \Sigma, N, n_0, F, T, V, C \rangle$ 으로 명세된다. 여기에서 Σ 는 유한한 포트의 집합을, N 은 노드의 유한 집합을, $n_0 \in N$ 은 시작 노드를 나타낸다. $F \subseteq N$ 는 최종 노드의 유한 집합이며, T 는 전이 조건과 시간 제약을 가진 노드 간 전이의 유한 집합, V 는 데이터 변수의 유한 집합을, C 는 지역 시계(*local clock*)를 나타낸다. 하나의 RSM은 다른 RSM과 병렬적으로 수행될 수 있으며, 계층적인 관계를 가질 수 있다.

명세 모델 기반 추상화는 명세 모델이 가진 특정 구문을 정의하여 이 구문을 만족하는 노드와 전이를 추상화한다. 구문 추상화는 비조건 내부연산 추상화와 구조적 추상화로 정의한다. 구문 추상화는 시스템 구조가 가진 계층성뿐만 아니라 구문적 계층성을 생성하므로 명세의 단순성(*simplicity*)을 제공하고, 단순화된 명세 모델들을 조합하므로 병렬 실행되는 시스템의 분석 복잡도를 낮출 수 있다.

3.1 비조건 내부연산 추상화

UIC 추상화는 UIC 관계(UIC-*relation*)에 있는 노드와 전이들을 추상화하는 방법으로, 추상화하는 단계에서 잘못 생성될 수 있는 전이 관계를 방지하기 위해 추상화 과정에 제약을 둔다.

정의 3.2: RSM의 노드 $s_i, s_{i+1} \in N$ 와 전이 $t_i \in T$ 에 대해서 t_i 의 원시 노드가 s_i 이고 목적 노드가 s_{i+1} 일 때 s_i 를 원시 노드로 하는 전이가 t_i 뿐이고, t_i 의 전이의 내용이 다른 RSM과의 상호 작용과 논리 연산을 포함하지 않을 때 s_i 와 s_{i+1} 의 관계를 t_i 에 의한 UIC 관계(UIC-*relation*)라 정의하고, 추상화되는 상태들을 UIC 모드, 추상화된 RSM을 UIC-RSM이라 한다. □

정의 3.3: UIC 추상화는 노드와 전이 사이의 UIC 관계를 기반으로 다음과 같은 경우에 따라 이루어진다:

본 연구는 한국과학재단 특정기초연구(1999-2-303-003-3) 지원으로 수행되었음

경우 1) 노드 s_i, s_{i+1}, s_{i+2} 에 대해 s_i, s_{i+1} 가 전이 t_i 에 의해 UIC 관계에 있고 s_{i+1} 과 s_{i+2} 가 UIC 관계가 아닐 때 s_{i+1} 이 s_i 로부터가 아닌 다른 노드로부터의 전이를 가진 경우: UIC 모드는 t_i 까지만을 추상화하고 추상화된 UIC 모드로부터 s_{i+1} 로의 레이블이 없는 전이를 생성하며 추상화를 수행한다.

경우 2) 경우 1이 아닌 경우: UIC 관계에 있는 노드와 전이를 UIC 모드로 추상화한다.

UIC 추상화는 논리적 조건과 상호 작용, 시간에 따른 제약과 관련된 전이에만 초점을 둔 명세 모델을 생성한다.

3.3 구조적 추상화

명세 모델의 구조를 추상화하기 위해 정규적 선택(Regular Selection, RS) 구조와 정규적 반복(Regular Repetition, RR) 구조, 혼합(Merged, MG) 구조를 정의하여 시스템의 정규적인 구조를 추상화하여 명세 모델에 계층성을 제공한다.

RS 구조는 RSM의 노드(또는 UIC-RSM의 모드) $s_i, s_{i+1}, s_{i+2}, \dots, s_{i+m}, \dots, s_{k-1}, s_k, s_{k+1}, s_{k+2}, \dots, s_{k+n}$ 에 대해 s_i 에서 $s_{i+1}, s_{i+2}, \dots, s_{i+m}$ 에서 s_{i+m} 로의 전이가 각각 $t_i, t_{i+1}, \dots, t_{i+m} \in T$ (또는 T_U)일 때 $t_i, t_{i+1}, \dots, t_{i+m}$ 의 전이가 논리 연산자를 사용한 조건문을 가지며 각각의 전이에 대한 노드와 전이의 경로가 노드 s_k 로 도달하고, s_k 에서 발생하는 전이가 $s_i, s_{i+1}, s_{i+2}, \dots, s_{i+m}, \dots, s_{k-1}, s_k, s_{k+1}$ 중 어느 하나로도 연결되지 않은 구조로, 구조 내부의 전이는 실행 모델을 생성하기 위한 상대 RSM과의 동기를 나타내는 전이를 포함하지 않는다.

RR 구조는 RSM(또는 UIC-RSM)의 노드 s_i, \dots, s_k 에 대해 s_i 에서 s_k 로의 전이가 존재하고 s_k 의에서 발생하는 전이 중 한 개의 목적 노드가 s_i 인 반복 구조로, 구조 내부의 전이는 실행 모델을 생성하기 위한 상대 RSM과의 동기를 나타내는 전이를 포함하지 않는다. 또한, 두 개 이상의 정규 구조가 일부 노드들을 공유하여 발생하는 구조를 혼합 구조라 한다.

정규 구조는 명세 모델이나 실행 모델에서 다중으로 포함되어 발생할 수 있다. 구조들 간의 포함 관계가 없이 RSM 또는 UIC-RSM이 하나의 계층 구조만을 가지는 단순 구조(Flat Structure)와 하나의 구조가 다른 구조를 포함하는 중첩 구조(Nested Structure)로 정의한다. 중첩구조에서 포함된 구조를 하위 구조라 하면 하위 구조의 모든 노드는 상위 구조에 속한다. 정규적 구조가 명세 모델의 계층성을 제공할 때 중첩 구조는 포함 관계에 따른 추상화를 통하여 다중 계층을 제공한다.

RSM 또는 UIC-RSM이 RR, RS, MG 구조를 가졌을 때 이 구조를 추상화하는 것을 구조적 추상화라 하며, 구조적 추상화된 노드와 RSM을 S 모드(S-mode)와 S-RSM이라 한다.

3.4 명세 모델의 조합

추상화된 명세 모델들을 조합하여 병렬 실행되는 시스템에서 발생하는 모든 상태를 생성한다. 조합되는 명세 모델의 집합 $S = \{M_1, M_2, \dots, M_n\}$ 에 대해 각 명세 모델의 노드의 집합이 N_1, N_2, \dots, N_n 이고 전이의 집합이 T_1, T_2, \dots, T_n 이라면 명세 모델의 조합 결과 발생하는 노드의 집합은 $N_1 \times N_2 \times \dots \times N_n$ 이고, 전

이의 집합은 $\cup_{i=1}^n \cup_{j=1}^m (1 \leq i \leq n, 1 \leq j \leq m)$ 로 정의한다. 단, 명세 모델의 전이의 집합 중 동기적인 전이에 대한 조합의 전이는 동시에 만족할 때만 생성된다.

4. 실행 모델 기반 추상화

실행 모델은 실행 정보를 모델링한 것으로 도달성 그래프의 형태로 표현되며 명세 모델의 조합으로부터 생성된다.

정의 4.1: 주어진 일련의 RSM의 집합 $S = \{M_1, M_2, \dots, M_n\}$, $n > 1$ 의 실행 모델은 그래프 $G = \langle N, n_0, E \rangle$ 이다. N 은 도달성 그래프의 노드의 집합으로 RSM들의 발생 노드(위치)와 자료, 시간이다. $n_0 \in N$ 은 그래프의 시작 노드이며, $E = N \times T_1 \times N$ 는 에지의 집합으로 T_1 는 N 에서 발생 가능한 전이들의 집합이다.

4.1 의미 추상화

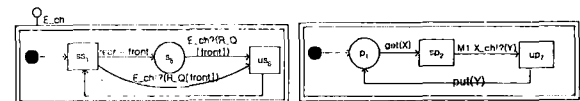
의미 추상화는 실행 모델에 적용되는 추상화로, 실행 모델의 위치, 자료, 시간이 가진 의미를 파악하여 추상화하는 방법이다. 의미 추상화를 통하여 실행 정보를 복잡도가 낮은 상위 계층에서 개략적으로 제공할 수 있다. 의미 추상화는 위치 공간 추상화와 자료, 시간 공간 추상화로 구성된다.

위치 추상화는 실행 모델의 노드가 RSM의 일정한 전이 관계에 의해 반복적으로 생성될 때 실행 모델의 반복되는 노드와 전이의 경로를 추상화한다. 추상화된 실행 노드와 에지의 반복 횟수와 반복에 의한 실행 정보는 개략적인 예측 값 또는 실행 결과 값으로 추상화되어 표현한다.

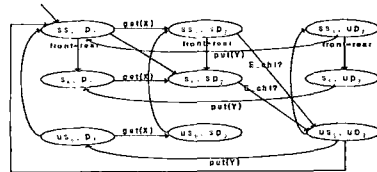
자료/시간 공간 추상화는 위치 공간 추상화와 동시에, 또는 독립적으로 이루어진다. 자료 공간 추상화는 노드를 정의하는 값 중 자료 값이 일정한 함수 관계로 변화하거나, 특정 범위에 존재하는 경우와 같이 자료 값이 특정한 의미로 표현될 때 그와 관련한 실행 모델의 노드와 전이를 추상화한다. 시간 공간 추상화는 노드의 시간 값이 가진 변화 관계나 범위의 의미에 따라 자료 공간 추상화와 같은 방법으로 추상화한다.

5. 예제

추상화의 예로 사용자로부터 입력받은 수를 원형 큐에 보관하여 다른 RSM으로부터 요청이 있을 때 입력받은 수를 순서대로 전달해 주는 작업을 반복하는 RSM M_1 과 사용자로부터 입력받은 수와 다른 RSM으로부터 요청하여 전달받은 수를 곱해 출력하는 동작을 반복하는 RSM M_2 를 사용한다. <그림 1>은 예제에 구문 추상화가 모두 적용된 S-RSM M_1, M_2 이다. 두 RSM은 각각 7개와 9개의 상태를 가졌으나, 3개로 감소하였다.



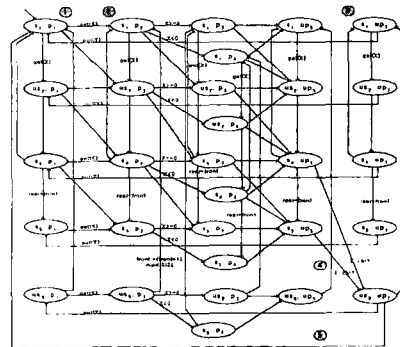
<그림 1> S-RSM M_1, M_2



<그림 2> S-RSM M_1, M_2 의 조합 결과

<그림 2>는 <그림 1>의 S-RSM을 조합한 것으로, 노드들의 전이 관계에 영향을 주는 조건 전이, 동기화 전이, IO에 대한 분석이 가능하다. 복잡도는 노드의 수가 가장 적은 S-RSM을 조합한 것이 가장 낮다.

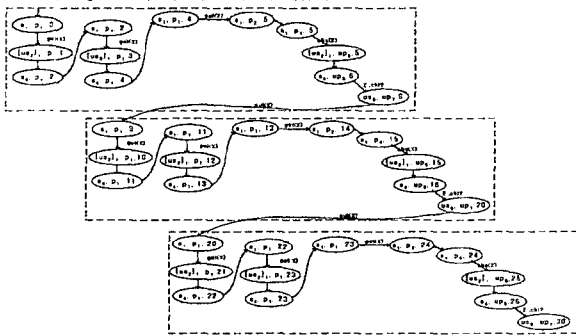
<그림 3>은 UIC-RSM의 조합 결과로, S-RSM의 실행 모델보다 하위 계층 단계에 있으며, 높은 복잡도를 가지며, 구조가 포함하던 정규적 반복 구조와 선택 구조에 대한 노드와 전이 관계를 볼 수 있다.



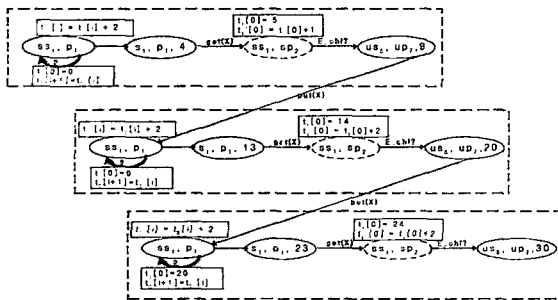
<그림 3> UIC-RSM M_1, M_2 의 조합 결과

각 RSM의 위치 값을 중심으로 조합된 노드와 전이로부터 실행했을 때 발생하는 결과를 시간과 자료를 포함한 실행 모델로 나타낸다. RSM의 실행 모델은 무한하거나 유한한 노드와 에지를 생성할 수 있으나 실행 모델을 위치, 자료, 시간 공간

추상화를 통해 노드와 에지의 수를 감소시킴으로써 상위 단계에서 실행에 대한 대략적 정보를 분석할 수 있다. 예를 들어, UIC-RSM의 실행 모델은 <그림 3>으로부터 <그림 4>와 같이 생성된다. 각 노드의 수는 전이에 따른 시간 값이며, S-RSM의 실행 모델에 대해 계층적으로 하위에 존재한다. <그림 4>가 가진 위치 정보의 의미와 시간 정보의 의미, 즉, 의미 추상화를 수행하면 1차적으로 <그림 5>와 같이 추상화된다. <그림 5>는 "ss1, p1" 노드가 2회 반복되는 것에 대해 추상화되었으며, 추상화에 해당하는 자료 정보 또한 추상화한다. 또한, S-RSM 정보로부터 추상화된 "ss1, sp2"의 자료 정보도 추상화된 형태로 표현된다. "ss1, sp2"는 RS, RR 구조가 혼합된 노드의 추상화 형태로 RS, RR 구조가 가진 모든 노드를 실행하지 않으므로, 점선 형태로 추상화한다. 이와 같이 실행 모델의 의미 추상화는 구문 정보로부터 유도될 수 있다.

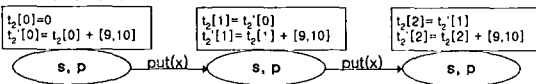


<그림 4> UIC-RSM 조합으로부터 생성된 실행 모델



<그림 5> <그림 4>에 위치 공간 추상화를 적용한 실행 모델

실행 모델이 가진 의미에 대해 반복적으로 의미 추상화를 적용한다. <그림 6>은 <그림 5>가 가진 3회 반복되는 위치 의미를 추상화하기 위해 자료/시간 정보를 추상화한 모습이다. 각 노드는 수행 시 변화되는 시간 값의 변화 폭과 초기 값으로 의미가 추상화되었다.



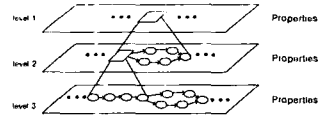
<그림 6> 반복되는 실행 경로를 추상화한 모델

최상위 단계의 실행 모델에서는 반복 정보와 반복에 따른 시간 정보를 개략적인 형태로 파악할 수 있다. 즉, 정보의 정확성은 부족하지만 개략적인 형태로 시스템 실행에 대한 쉽고 빠른 정보를 제공할 수 있다.

6. 결론 및 향후 연구

본 논문에서는 추상화를 적용함으로써 시스템의 명세 정보, 실행 정보를 최상위 계층에서는 개략적이지만 낮은 복잡도의 형태로, 최하위 계층에서는 정확한 정보를 높은 복잡도의 형태

로 제공하여 시스템의 이해를 다양한 각도에서 제공할 수 있다.



<그림 7> 모델의 계층성과 속성

향후 연구로는 추상화된 명세 모델을 통해 추상화된 계층에서 적용할 수 있는 모델 검사(Model Checking) 기법을 정의, 적용하는 것이다. 추상화를 통한 모델 검사 연구[6, 7]는 계층성이 없는 모델로부터 생성된 실행 모델이 특정 속성들을 만족하는지의 여부를 결정한다. 본 연구에서는 <그림 7>과 같이 계층성에 따라 검증 가능한 속성을 분류하여, 시스템간의 상호작용 여부와 같이 복잡도가 낮은 상위 계층에서 검증가능한 속성은 높은 복잡도를 가진 하위 계층의 정보를 사용하지 않고서도 검증할 수 있도록 하여 시스템의 이해도뿐만 아니라 검증의 복잡도 또한 감소시키는 연구를 지속적으로 수행할 것이다. 이를 위해 CTL[1]과 같은 추상화 정보와 속성을 표현, 검증하는 논리의 정의가 필요하다.

[참고문헌]

- [1]E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic Verification of Finite-State Concurrent Systems using Temporal Logic Specifications. *ACM Transactions on Programming Languages and Systems*, pp. 244-263, April 1986, 8(2).
- [2]W. J. Yeh and M. Young. Compositional Reachability Analysis using Process Algebra. In *Proc. of Conf. on Testing, Analysis and Verification*, pp. 49-59, August 1992.
- [3]I. Kang and I. Lee. State Minimization for Concurrent System Analysis Based on State Space Exploration. In *Proc. of Conf. on Computer Assurance*, pp. 123-134, 1994.
- [4]S. Raju. An Automatic Verification Technique for Communicating Real-Time State Machines. Technical Report 93-04-08, Dept. of Computer Science and Engineering, Univ. of Washington, April, 1993.
- [5]R. Alur, C. Courcoubetis, N. Halbwachs, D.Dill, H. Wong-Toi, Minimization of Timed Transition Systems, In W. R. Cleave and, editor, *CONCUR92: 3rd Intl. Conf. on Concurrency Theory*. Lecture Notes in Computer Science vol. 630, Springer, pp. 340-354, 1992.
- [6] I. Kang, I. Lee and Y. Kim. An Efficient State Space Generation for the Analysis of Real-Time Systems. *IEEE Transaction on Software Engineering*, Vol. 26, No. 5, pp. 453-477, May 2000.
- [7]Edmund M. Clarke, Orna Grumberg, David E. Long, Model Checking and Abstraction, In *Proc. of the 19th Annual ACM Symposium on Principles of Programming Languages*, pp. 343-354, January. 1992.
- [8]David Lee, Mihalis Yannakakis, Online Minimization of Transition Systems(Extended Abstract), In *Proc. 24th ACM Symposium on Theory of Computing*, pp. 264-274, May 1992.
- [9]Adan Aziz, Vigyan Singhal, Gitanjali M. Swamy, Minimizing Interacting Finite State Machines: A Compositional Approach to Language Containment, In *Proc. of Intl. Conf. on Computer Design*, pp.255-263, October. 1994.