



# SAT/SMT Summer School

2011년 6월 12일 ~ 2011년 6월 17일

서울대학교 프로그래밍 연구실  
이승중

## 여름학교를 다녀와서

2011년 6월 12일부터 6월 17일까지 미국 MIT에서 열렸던 SAT/SMT 여름학교에 다녀왔습니다. SAT/SMT에 대한 전반적인 소개와 실제로 어떻게 쓰이는 지, 지금은 SAT/SMT solver가 어떤 방식으로 발전하고 있는 지에 대해 전문가들의 이야기를 들을 수 있는 자리였습니다.

또한 SAT/SMT의 전문가뿐만 아니라 다양한 분야의 사람들을 만날 수 있었던 기회도 되었습니다. 특히 SAT/SMT를 각자의 분야에 접목시킬 목적으로 오는 사람들을 많이 볼 수 있었습니다. 많은 이야기를 나누지는 못했지만, SAT/SMT가 이론적인 분야뿐만이 아닌, 여러 분야에서 실용적으로 관심을 갖고 있는 기술이라는 것을 느낄 수 있었습니다.

## 수업

수업은 6일에 걸쳐 총 35개의 강의로 진행되었습니다.(첫날 5개, 그 외 6개씩) 크게 SAT/SMT solver의 기초, 적용, SAT/SMT solver를 사용한 도구, 세 개의 종목으로 나눌 수 있었습니다. 어떻게 SAT/SMT solver를 사용했을까에 대해서 궁금해 하던 터라 적용 쪽에 보다 관심을 갖고 수업을 들었습니다.

많은 강의가 있었지만 저에게 인상 깊었던 강의 몇 가지만 소개합니다. 강의 전체에 대한 설명은 SAT/SMT Lecture Notes<sup>1)</sup>에서 보실 수 있습니다.

### SMT-LIB initiative

---

SMT solver의 입력으로 쓰이는 표준 언어 규격인 SMT-LIB에 대해서 설명하는 시간이었습니다. 제작자가 직접와서 강의를 하였는데, 왜 SMT-LIB를 구상하게 되었는지, 어떻게 사용할 수 있는 지에 대해서 알 수 있었습니다.

여러 SMT-LIB를 만든 목적은 여러 SMT solver들이 나오는 가운데 보다 손쉽게 다른 solver를 쓸 수 있게 하기 위해서라고 합니다. SMT solver를 사용하는 입장에서는 SMT-LIB를 쓰면 쉽게 다른 solver로 교체해서 사용해볼 수 있을 것이고, SMT solver 제작자 입장에서는 여러 theory들의 이름이나 필요한 형식들을 규격화 해서 지금 만들어진 solver의 분류를 쉽게 할 수 있습니다.

더불어 SMT-LIB을 지원하는 엔진들끼리 성능 벤치마크를 쉽게 할 수 있다는 이점이 있습니다. 따라서 매년 SMT-COMP가 열리고 많은 엔진들이 여기에 도전한다고 합니다.

현재 SMT-LIB을 지원하는 SMT solver는 CVC3, STP, OpenSMT, MathSAT, Z3 Yices 등 유명한 SMT solver들은 모두 지원한다고 합니다.

SMT-LIB의 실제 문법을 보니 약간 흥미로웠는데 Lisp의 S-Expression을 사용하여 괄호로 동작을 표현하고 있습니다.

```
; This example illustrates basic arithmetic and
; uninterpreted functions
(declare-funs ((x Int) (y Int) (z Int)))
(assert (>= (* 2 x) (+ y z)))
(declare-funs ((f Int Int) (g Int Int Int)))
(assert (< (f x) (g x x)))
(assert (> (f y) (g x x)))
check-sat
(get-info model)
push
(assert (= x y))
```

---

1) <http://people.csail.mit.edu/vganesh/summerschool/lectures.html>

```
check-sat
pop
quit
```

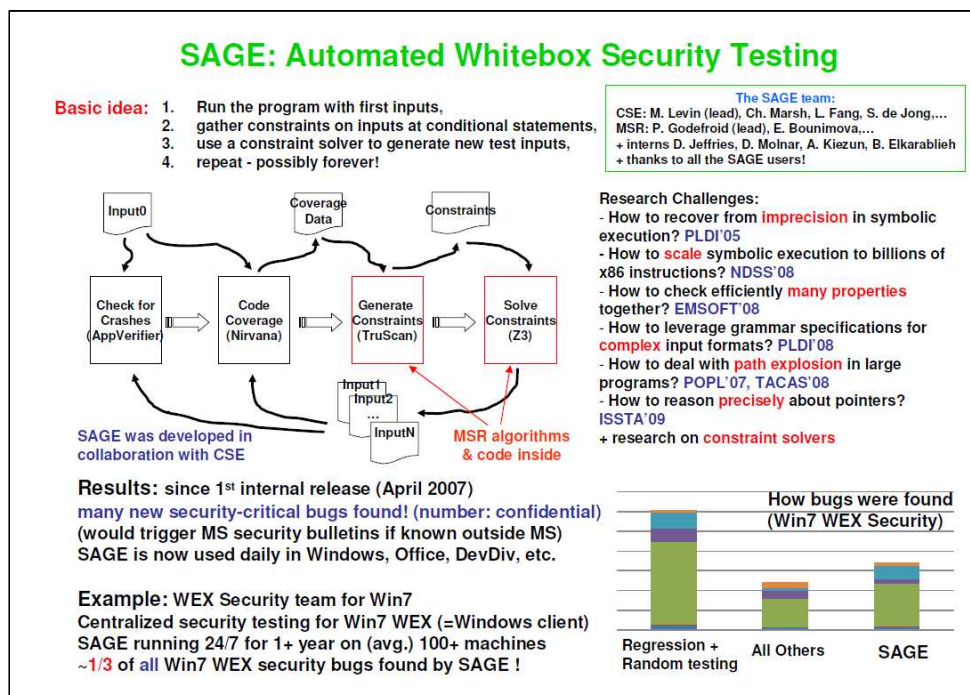
이것을 보면서 요약 실행기를 보다 간편하게 만들기 위해 전에 고안하고 있었던 기술 언어가 생각났습니다. 사람들이 언어를 많이 쓰도록 하려면 무엇보다 편리하고 직관 적이어야 한다는 생각이 들었습니다.

## SAGE: Automated Whitebox Fuzzing using SMT solvers

SAGE는 프로그램에서 보안문제를 찾아주는 툴입니다. 이 강의에서는 SMT solver 를 사용해서 어떻게 보안문제를 찾는 지에 대해 이야기 하고 있습니다. Fuzzing은 랜덤하게 입력 값을 집어넣어 보안문제를 찾는 방식을 말합니다.

기존의 보안문제를 찾는 프로그램들은 말 그대로 랜덤하게 아무 입력 값을 집어넣는 방식이었습니다. 하지만 보통 프로그램들은 특정한 입력이 들어올 때만 반응을 하기 때문에, Fuzzing으로 효과적으로 문제를 찾아내기가 힘듭니다. 따라서 프로그램을 실행시켜 입력 값을 얻어내고, 프로그램이 도달하는 데까지 거쳐간 분기문들 만을 모아 이 분기문들을 SMT solver의 입력으로 집어넣어 프로그램의 여러 지점에 도달 가능 하도록 다른 입력을 만들어 내어 보안버그를 찾는 방식입니다.

이 방식은 많은 양의 코드를 어떻게 처리할 수 있었는지에 대해 중점적으로 설명했 습니다. 특히 이 SAGE를 병렬적으로 만들어서 100개 이상의 머신에 1년 내내 돌려 Windows 7 코드의 보안버그를 찾았고, 계속 찾고 있다고 하는 점이 인상깊었습니다.

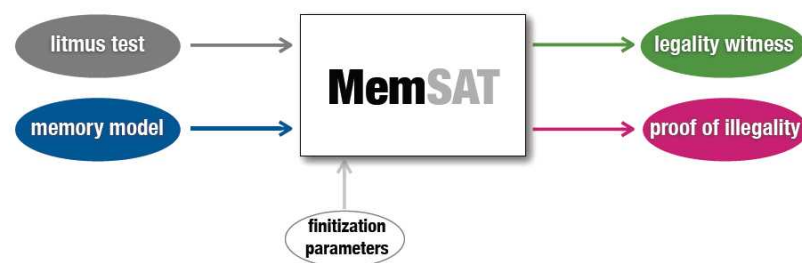


## Alloy/Kodkod and Application

---

Kodkod는 SAT 기반의 제약식을 풀어주는 도구입니다. 이번 강의에서는 Kodkod를 사용하여 언어의 메모리 모델의 명세가 제대로 되었는지를 확인하는 MemSAT이란 도구에 대해서 설명했습니다.

메모리 모델이란 병렬 프로그램에서 프로그래머와 프로그래밍 환경(자바, C 같은)사이의 계약으로 시스템의 메모리를 동시에 읽고 쓸 때 발생할 수 있는 상황을 일관성 있게 정해놓은 것입니다. 보통 메모리 모델은 모델을 표현하는 공리와 정규화 되지 않은 리트머스 테스트(보통 assert를 포함한 소스코드)들로 구성이 되어있습니다. 공리가 제대로 지켜지도록 프로그래밍 환경을 제어하는 것이 어렵기 때문에 이 도구를 만들었다고 합니다. 특히 자바는 VM 위에서 돌아가기 때문에 하드웨어에 관계없이 메모리 행동이 동일해야 합니다.



MemSAT은 이런 공리들과 리트머스 테스트를 입력으로 받아서 리트머스 테스트의 assertion 조건이 충족되는 출력 값을 내보내거나, 만약 출력 값을 정할 수 없으면 만족되지 않는 최소 제약의 집합을 돌려줍니다.

MemSAT은 자바의 메모리 모델에 대해서 공리가 제대로 되어있는 지를 확인하였고, 잘못된 결과를 발견하여 자바언어의 새로운 버전에서 메모리 모델을 수정하였다고 합니다.

## Calculus of Data Structure for Verification and Synthesis

---

이 강의에서는 제약만으로 코드를 생성하는 Software Synthesis에 대해서 설명했습니다. 프로그래머는 코드를 직접 작성할 필요 없이, 이 코드의 입력과 출력의 조건이 무엇인지만 서술하면 그에 맞는 코드와 코드에 대한 Precondition을 만들어 주는 방식입니다.

조건을 기술할때는 특별한 표현을 쓰는데 choose라는 구문을 사용하여 조건을 만들어냅니다. 예를 들어 어떤 bigSet이 미리 정의되어 있고, setA, setB에 bigSet을 받

씩 나누어서 가지고 있게 하는 코드를 만든다고 하면 조건은 이런 식으로 작성할 수 있고,

```
val bigSet = ...
val (setA, setB) = choose((a: Set, b: Set) =>
  ( a.size == b.size && a union b == bigSet && a intersect b == empty))
```

위의 조건을 사용하여 생성된 코드는 다음과 같습니다.

```
assert (bigSet.size % 2 == 0)
val n = bigSet.size/2
val setA = take(n, bigSet)
val setB = bigSet -- setA
```

먼 미래에는 코드를 작성할 때 실제로 코드를 작성하는 것이 아닌, 내가 하고 싶은 일들만 적어주면 알아서 코드를 생성하는 그런 시대가 올 수도 있을 것 같습니다.

## Liquid Types

---

Liquid Types에 관해서 간단하게나마 알 수 있는 강의였습니다. 보통 프로그램이 어떤 성질을 만족하는 지를 알아보려면, 프로그램을 나타내는 Invariant와 프로그램이 받아들일 수 있는 성질인 Property를 통해 Invariant => Property를 만족하는 지에 대해서 SMT solver로 풀면 됩니다. 간단해 보이지만 복잡한 구조를 가진 데이터들은 한정자를 가진 Invariants를 필요로 하고, 이런 한정자들은 상태를 기하급수적으로 증가시켜서 SMT solver가 문제를 풀지 못하게 합니다.

따라서 논리적으로 한정된 타입을 나타내기 위해 Invariant를 Logic과 Type으로 나타냅니다. Logic에서는 데이터가 어떤 성질을 가졌는지를 기술하며, Type은 그 데이터를 가지고 있는 자료구조에 대한 제약을 나타냅니다. 이렇게 하면 자료구조의 성질에 의한 한정자가 SMT solver로 넘어가는 것을 줄일 수 있고, 그 데이터 타입에 맞는 정확한 성질을 따로 관리하여 도출해 낼 수 있습니다.

수업에서는 BaseTypes, Collections, Closures, Generics에 대해 각각 Liquid Type을 어떻게 도출했는지에 대해서 설명했습니다. 공통된 사항으로는 각각 type에 대한 템플릿이 존재해서 그 템플릿과 다른 곳의 제약 정보를 바탕으로 SMT를 사용하여 타입 추론하듯이 Liquid Type을 계산해냈습니다.

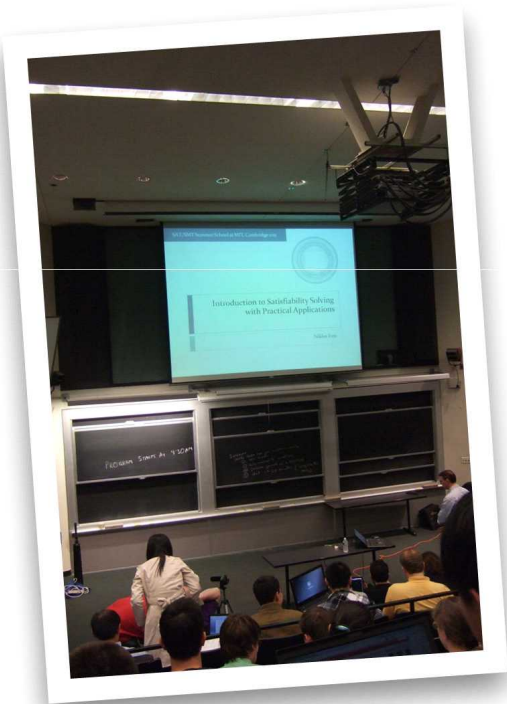
이런 방식으로 ML언어에서 List-based sorting과 Finite Map, Red-Black Tree의 몇 가지 invariant를 증명하였다고 합니다.



## 수업을 들으면서 느낀 점

### 수업만큼 중요한 Coffee Break

매 수업이 끝나고 나면 15~30분 정도의 쉬는 시간이 있었습니다. 많은 사람들이 이 시간에 수업을 했던 연사를 찾아가 질문을 하고 같은 분야의 지인들을 만나서 관심분야에 대해서 대화를 하였습니다. 이런 시간들을 통해서 수업 때 궁금했거나 이해가 가지 않았던 것들을 보다 명확히 할 수도 있고 더 나은 아이디어를 도출해 낼 수 있을 것 같았습니다. 이런 대화에 적극적으로 나서지 못했던 것이 아쉽습니다.



### 수업에서 아쉬웠던 점

수업을 들으면서 아쉬웠던 점은, 연사가 청중의 수준을 SAT에 관해 이미 알고 있는 사람들라고 생각했다는 점입니다. 많은 이 분야 전문가들이 각자 자신이 연구하는 내용, 응용했던 사례들을 기본부터 설명하는 과정 없이 바로 본론으로 들어갔습니다.

따라서 SAT/SMT Solver에 익숙하지 않았던 사람들에게는 무엇이 실제 사용하는데 문제가 되는지, 어떻게 Solver에 입력을 넣었는지 이해하는데 어려움이 있었습니다.

### 동기 부여

많은 수업 참가자들, 이론적인 내용을 막힘없이 발표하는 강사들, 관심 있어 하는 주제에 당당히 자기 의견을 개진하는 사람들을 보면서 세상에 열정을 갖고 활발히 연구하는 사람이 엄청 많구나 하는 것을 느꼈고, 좀 더 열심히 노력해야겠다고 다짐하는 계기가 되었습니다.

## 볼거리

6일간 아침 8시 30분부터 저녁 6시까지 계속되는 강행군에 근처를 돌아다닐 기력도 없었지만, 틈틈이 그리고 수업이 끝난 다음 날 하루 동안 MIT 주변을 구경할 수 있었습니다.

### MIT

여름학교가 열린 MIT 주변을 돌아다니는 것도 재미있었습니다. 학생회관이라던가 본 부건물, 잡동사니를 마구 얹어놓은 듯한 CSAIL 건물도 인상 깊었습니다.



### Harvard University

MIT에서 지하철로 두 정거장 정도 더 가면 하버드 대학교가 나옵니다. 고풍스러운 기숙사 건물과 간간히 들어선 술집의 생기발랄한 사람들을 보니, 세련되고 한적했던 MIT와는 정반대의 느낌이 들었습니다.



하버드 도서관



## Charles River

메사추세츠 주에는 보스턴과 캠브리지를 가르는 강이 있습니다. 페이스북 창시자 마크 주커버그의 이야기를 다룬 영화 ‘소셜네트워크’에도 하버드 학생들이 카누를 타는 장면이 나왔던 강입니다.

여름학교 기간에는 날씨가 좋지 않아 강에 배 구경을 많이 못했지만, 종강을 하고나니 날씨가 좋아져서 요트를 볼 수 있었습니다.

아침에 두어 번 숙소에서 학교까지 운동삼아 걸어가 보았는데 생각보다 가까워서 40분정도밖에 걸리지 않고, 강변 풍경을 즐길 수 있어서 좋았습니다.

## 보스턴 항구

보스턴은 바다와 맞닿아 있어서 보스턴 시청에서부터 조금만 걸어가면 항구가 나옵니다. 너른 바다를 구경할 수도 있고, 수족관, 섬으로 가는 배를 탈 수도 있습니다.





## 생활정보

**화장실** 지하철 역 안에 화장실이 없어서 애를 먹었습니다. 밤 9시쯤에 화장실이 급한데 아무데도 없어서 역 하나만큼 화장실 찾으러 걸은 적도 있습니다. 스타벅스같이 화장실을 제공하는 곳을 이용할 경우 반드시 이용하고 나오는 것이 좋을 것 같습니다.

**지하철, 버스** 지하철은 Charlie Ticket을 끊어서 타게 됩니다. 한번 타는데 \$2.00 인데, \$15.00 짜리 7일 Link-pass를 끊으면 버스, 지하철을 7일 동안 무제한으로 탈 수 있습니다. 지하철 표지판에는 inbound, outbound라는 방향이 적혀져 있는데 보스톤 시내를 중심으로, 중심으로 가는 방향을 inbound, 중심에서 나가는 방향을 outbound라고 부릅니다.



**숙소** 숙소를 예약할 때는, private bath인지 확인 하는 것이 좋을 것 같습니다. 이번에 미처 확인을 못했더니 그 층에 있는 사람들과 전부 같은 화장실을 쓰게 되었습니다. 샤워하고 나서 샴푸와 비누를 샤워실에 두고 왔다가 잃어버리는 등 매우 불편하므로 꼭 확인 하는 것이 좋겠습니다.

**팁** 약 15%로 잡으면 되고, 스타벅스나 맥도날드 같이 카운터에 주문하는 데서는 팁을 낼 필요가 없습니다. 신용카드로도 낼 수 있는데, 일단 계산서와 신용카드를 웨이터에게 주고 싸인을 받으려 오면, 그때 팁을 얼마 주겠다고 적습니다. 더불어 팁을 포함한 총 가격을 계산해서 적은 다음 싸인을 해서 주면 결제가 완료됩니다. 계산서에 팁이 포함되어있는 경우는 따로 내지 않아도 됩니다.

## 맺음말

이번 6일 동안의 여름학교는 흥미롭고 새로운 경험이었습니다. 몰랐던 SAT에 대해 조금이나마 맛을 볼 수 있었고, 말로만 듣던 하버드나 MIT를 구경해볼 수 있었던 기회였습니다.

학교 근처에서 헤매지 않도록 초행길에 도움을 주신 MIT 박사과정 덕환이 형님께 감사드리고, 여름학교에서 뵈어 이런저런 이야기 나눌 수 있었던 임정희 박사님, 박용수 교수님도 앞으로 계속 인연 맺을 수 있으면 좋겠습니다. 이런 좋은 경험 갖게 해주신 이광근 교수님께도 감사드립니다.