

# Overview of E0 205: Mathematical Logic and Theorem Proving

Deepak D'Souza and Kamal Lodaya

Department of Computer Science and Automation  
Indian Institute of Science, Bangalore.

February 24, 2021

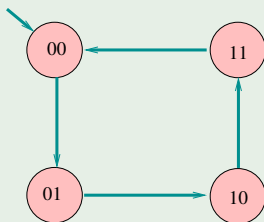


# Course Contents

- Mathematical Logic
  - Propositional and First-Order Logic
  - Sound and complete proof systems
  - Compactness
- Decision Procedures
  - Equality and Uninterpreted Functions (EUF)
  - Real and Integer Linear Arithmetic
  - Quantifiers
  - Array logic
  - Nelson-Oppen combination
  - Constrained Horn Clauses

# Boolean SAT solving

Does the system satisfy the temporal logic formula  
 $G(b \implies X(\neg b))$ ?



In bounded model-checking we could ask for a path of length 2 that violates the specification: Is

$$\neg a_0 \wedge \neg b_0 \wedge T(a_0, b_0, a_1, b_1) \wedge T(a_1, b_1, a_2, b_2) \wedge b_1 \wedge b_2,$$

where  $T(a, b, a', b') = (\neg a \wedge a' \wedge b \iff b') \vee (a \wedge \neg a' \wedge b \iff \neg b')$ ,  
 satisfiable?

# Linear Arithmetic

Bounded model-checking for programs:

```
int x = 19;
int y = 15;
while (x >= 10) {
  int z = -1;
  x = x + z;
}
assert(y >= x);
```

Does there exist zero-iteration  
execution violating the assertion: Is

$$x_1 = 19 \wedge y_1 = 15 \wedge x_1 < 10 \wedge y_1 < x_1$$

satisfiable?

# Linear Arithmetic

Floyd-Hoare style verification of programs:

Are the constraints:  $\forall x, y, z, x' :$

```
int x = 19;
int y = 15;
// inv: x <= 20
while (x >= 10) {
    int z = -1;
    x = x + z;
}
assert(y >= x);
```

$$x = 19 \implies x \leq 20$$

$$x \leq 20 \wedge x \geq 10 \wedge z = -1 \wedge x' = x + z \implies x' \leq 20$$

$$x \leq 20 \wedge y = 15 \wedge \neg(x \geq 10) \implies y \geq x'$$

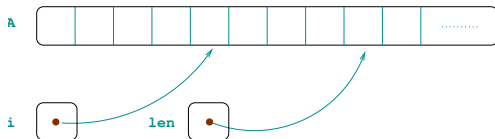
valid?

# Array Logic

```

ainit(int A[], int len) {
// Pre: 0 <= len
  int i = 0;
  while (i < len) {
    A[i] = 0;
    i = i + 1;
  }
// Post:
forall k: ((0 <= k < len)
           => A[k] = 0)

```



Loop invariant:

$$(0 \leq i \leq len) \wedge \forall k ((0 \leq k < i) \implies A[k] = 0)$$

Verification condition:

$$[(0 \leq i \leq len) \wedge \forall k ((0 \leq k < i) \implies A[k] = 0) \wedge \neg(i < len)] \implies$$

$$\forall k : ((0 \leq k < len) \implies A[k] = 0).$$

# Uninterpreted Functions with Equality (EUF)

Is this formula valid?

$$g(g(g(x))) = x \wedge g(g(g(g(g(x)))))) = x \implies g(x) = x.$$

Congruence closure algorithm.

# Nelson-Oppen Combination

Example: Is this sentence satisfiable?

$$f(f(x) - f(y)) \neq f(z) \wedge x \leq y \wedge y + z \leq x \wedge z \geq 0$$



# Nelson-Oppen Combination

Example: Is this sentence satisfiable?

$$f(f(x) - f(y)) \neq f(z) \wedge x \leq y \wedge y + z \leq x \wedge z \geq 0$$

No, because the arithmetic constraints imply that  $x = y$  and  $z = 0$ ; and the functional constraints must then imply that  $f(f(x) - f(y)) = f(0) = f(z)$ .

# Nelson-Oppen Combination

Shows how we can combine decision procedures for two theories into a decision procedure for their union.

“Equality Sharing” Procedure:

Is this sentence satisfiable?

$$f(f(x) - f(y)) \neq f(z) \wedge x \leq y \wedge y + z \leq x \wedge z \geq 0$$

Arithmetic Constraints

$$\begin{aligned} x &\leq y \\ y + z &\leq x \\ z &\geq 0 \\ g_2 - g_3 &= g_1 \end{aligned}$$

Function Constraints

$$\begin{aligned} f(g_1) &\neq f(z) \\ f(x) &= g_2 \\ f(y) &= g_3 \end{aligned}$$

# Nelson-Oppen Combination

Does this procedure work for integer arithmetic and functions?

Is this sentence satisfiable? (int  $x$ )

$$1 \leq x \wedge x \leq 2 \wedge f(x) \neq f(1) \wedge f(x) \neq f(2)$$

Arithmetic Constraints

$$1 \leq x$$

$$x \leq 2$$

$$a = 1$$

$$b = 2$$

Function Constraints

$$f(x) \neq f(a)$$

$$f(x) \neq f(b)$$

Need “convex” theories.

# Constrained Horn Clauses

Find unary relations  $f$ ,  $g$  and  $inv$  such that:

```
int x = 19;
while (*) {
  int z = f();
  x = x + z;
}
int y = g();
assert(y >= x);
```

$$x = 19 \implies inv(x)$$

$$inv(x) \wedge f(z) \wedge x' = x + z \implies inv(x')$$

$$inv(x) \wedge g(y) \implies y \geq x$$

# Constrained Horn Clauses

```

int x = 19;
while (*) {
  int z = f();
  x = x + z;
}
int y = g();
assert(y >= x);

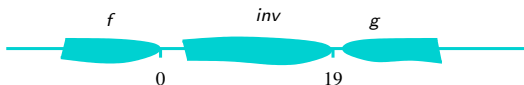
```

Find unary relations  $f$ ,  $g$  and  $inv$  such that:

$$x = 19 \implies inv(x)$$

$$inv(x) \wedge f(z) \wedge x' = x + z \implies inv(x')$$

$$inv(x) \wedge g(y) \implies y \geq x$$





## Course details

- Weightage: 40% assignments + seminar, 20% midsem exam, 40% final exam.
- Assignments to be done on your own.
- Dishonesty Policy: Any instance of copying in an assignment will fetch you a 0 in that assignment + one grade reduction + report to DCC.
- Seminar (in pairs) can be chosen from list on course webpage or your own topic.
- Course webpage:  
`www.csa.iisc.ac.in/~deepakd/logic-2021`
- Teaching assistants for the course: P. Habeeb and Stanly Samuel
- Those interested in crediting/auditing please send me an email so that I can add you to the course mailing list.