Subterm Sharing

Avoid **multiple copies** of one expression

Make equal things **identical** to enforce equality axioms
Equivalence Classes

How to represent equivalence classes in code?

\[ f(a) = b \quad f(b) = a \]
for (f, args), name in this.names.items():
    arg_classes = map(this.classof, args)
    name2 = this.add(f, arg_classes)
    this.eqclass[name] = name2
Class Progress

Logical reasoning

Program logics

Static analysis

First-order Logic

Decision Procedures

Mixing Theories

Equality

Integers

Arrays
Integers

Solving conjunctions of **integer equalities**
   Back to high school with elimination and substitution

**Variable elimination** for integer inequalities
   Combining two inequalities into one

**How fast** can integer reasoning be?
   SAT, multiplication, and rational numbers
Systems of Equations

Flashback to high school math
The Solver Query

**Statement** $p$
\[(\neg a_1 \lor a_2) \land (\neg b_1 \lor b_2) \land \ldots\]

**Assignment** $\Gamma$
\[a_1 \land \neg b_2 \land \ldots\]

Today: a solver for **integer queries**

No negations!

$p := x < y \mid x = y$

$x, y := x + y \mid x \times n \mid v \mid n$

\[\Gamma = x < 1 \land 0 < x\]
\[\Gamma = x < y \land y < z \land x + z < y\]
\[\Gamma = 2 \times x + 3 \times y = 5 \land x + 4 \times y = 0\]
Standard Form

Convenient to use ≤ (in a consistent direction)

\[ x < y \iff x \leq y - 1 \quad \text{and} \quad x = y \iff x \leq y \land y \leq x \]

Convenient to distribute multiplications

\[ (a + b) \times n \iff a \times n + b \times n \]

Convenient to separate constants and variables

\[ 1 + 2 \times (x + 3 \times y + 4) \leq y \iff 2 \times x + 6 \times y \leq -9 \]

Result: matrix formula \( Mx \leq C \)
Examples

\[ \Gamma = x < 1 \land 0 < x \]  
\[ \Gamma = x < y \land y < z \land x + z < y \]

\[ \begin{array}{c|c|c|c}
1 & x & \leq & 0 \\
-1 & & \leq & -1 \\
\end{array} \]

\[ \begin{array}{c|c|c|c|c}
1 & -1 & & x & \leq & -1 \\
1 & -1 & & y & \leq & -1 \\
1 & -1 & 1 & Z & \leq & -1 \\
\end{array} \]

\[ \Gamma = 2 \times x + 3 \times y = 5 \land x + 4 \times y = 0 \]

\[ \begin{array}{c|c|c|c|c}
2 & 3 & & x & \leq & 5 \\
-2 & -3 & & y & \leq & -5 \\
1 & 4 & & y & \leq & 0 \\
-1 & -4 & & & \leq & 0 \\
\end{array} \]
Restrict $Mx \leq C$ to $Mx = C$ (for now)

\[
\begin{bmatrix}
1 & -1 & \leq & -1 \\
1 & -1 & & -1 \\
1 & -1 & 1 & -1
\end{bmatrix}
\]
Equations

Restrict $Mx \leq C$ to $Mx = C$ (for now)

\[
\begin{align*}
    x & \quad 1 & -1 & -1 \\
    y & \quad 1 & -1 & -1 \\
    z & \quad 1 & -1 & 1
\end{align*}
\]

Matrix inverses only work for square matrices

Let’s look at two other techniques for solving linear equations
Substitution

Solve linear equations by **eliminating variables**

\[ \Gamma = 2 \times x + 3 \times y = 5 \land x + 4 \times y = 0 \]

\[ x = -4 \times y \]

\[ 2 \times (-4 \times y) + 3 \times y = 5 \]

\[ -5 \times y = 5 \]

\[ y = -1 \]
Substitution

Solve linear equations by **eliminating variables**

\[ \Gamma = 2 \times x + 3 \times y = 5 \land x + 4 \times y = 0 \]

\[ 2 \times x = 5 - 3 \times y \]

\[ 2 \times x + 8 \times y = 0 \]

\[ (5 - 3 \times y) + 8 \times y = 0 \]

\[ 5 + 5 \times y = 0 \]

\[ y = -1 \]

**Variable elimination** is a standard logic technique

Recall **proof by resolution** in boolean logic
Elimination

Solve linear equations by **simplifying equations**

\[
\Gamma = 2 \times x + 3 \times y = 5 \land x + 4 \times y = 0
\]

Create **row combinations** with 0 elements

Gaussian elimination algorithm does this **systematically**
Elimination

Systematic combinations of equations

\[
\begin{align*}
  a & \quad ? & \quad ? & \quad x & \quad w \\
  b & \quad ? & \quad ? & \quad y & \quad u \\
  c & \quad ? & \quad ? & \quad z & \quad v \\
  d & \quad ? & \quad ? & \quad & \quad q
\end{align*}
\]
Elimination

Systematic combinations of equations

\[
\begin{align*}
\text{a} & \quad \text{?} & \quad \text{?} & \quad \text{x} & \quad \text{w} \\
\text{0} & \quad \text{?} & \quad \text{?} & \quad \text{y} & \quad \text{a} - \text{b} \cdot \text{w} \\
\text{c} & \quad \text{?} & \quad \text{?} & \quad \text{z} & \quad \text{v} \\
\text{d} & \quad \text{?} & \quad \text{?} & & \quad \text{q}
\end{align*}
\]
Elimination

Systematic combinations of equations

\[
\begin{align*}
\begin{array}{ccc}
\text{-c} & \text{a} & ? \\
\text{a} & ? & ? \\
0 & b & ? \\
0 & c & ? \\
0 & d & ? \\
\end{array}
& \begin{array}{ccc}
0 & u - b w \\
0 & v - c w \\
0 & q - d w \\
w
\end{array}
\end{align*}
\]
# Elimination

Systematic combinations of equations

\[
\begin{bmatrix}
\text{a} & ? & \text{c} \\
0 & \text{a2} & \text{b} \\
0 & 0 & \text{a3} \\
0 & 0 & 0
\end{bmatrix}
\quad \begin{align*}
x & = w \\
y & = ? \\
z & = ?
\end{align*}
\]

\[
\begin{bmatrix}
\text{-c} \\
\text{b} \\
\vdots
\end{bmatrix}
\]

\[
\begin{bmatrix}
\text{a} \\
0 \\
0 \\
0
\end{bmatrix}
\]

\[
\begin{bmatrix}
? \\
\text{b} \\
\text{a2} \\
\text{a3}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\text{c} \\
\text{b} \\
\text{a2} \\
\text{a3}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\text{x} \\
\text{y} \\
\text{z}
\end{bmatrix}
\]

\[
\begin{bmatrix}
w \\
? \\
?
\end{bmatrix}
\]
Elimination

Systematic combinations of equations

$$\begin{array}{ccc}
\ldots & a & 0 & 0 \\
0 & a_2 & 0 \\
0 & 0 & a_3 \\
0 & 0 & 0
\end{array} \begin{array}{c}
x \\
y \\
z
\end{array} = \begin{array}{c}
? \\
? \\
? \\
?
\end{array}$$

Overall algorithm is $O(n^3)$ to solve $n$ equations

Beware of a special case, pivoting, for 0 coefficients
Inequalities

Variable elimination generalizes substitution
Core Idea

Pick a variable ($z$) to eliminate from the inequalities

\[
\begin{align*}
2x + 3y & \leq 7 \\
3x - y + z & \leq 5 \quad \rightarrow \quad z \leq 5 - 3x + y \\
x - 2y - z & \leq 3 \quad \rightarrow \quad x - 2y - 3 \leq z \\
4x - 3y & \leq 8 \quad \leftarrow \quad x - 2y - 3 \leq 5 - 3x + y
\end{align*}
\]

New inequalities with one fewer variable
Example

$$\begin{array}{c|c|c|c}
1 & -1 & x & -1 \\
1 & -1 & y & \leq -1 \\
1 & -1 & 1 & z & -1
\end{array}$$

Eliminate the $z$ variable from these inequalities

$$\begin{array}{c|c|c|c}
1 & -1 & x & -1 \\
1 & 0 & y & \leq -2
\end{array}$$

Now eliminate the $y$ variable

$$\begin{array}{c|c|c|c}
1 & x & \leq -2 & \text{Satisfiable}
\end{array}$$
Multiple Variables

Group by negative, positive, or zero coefficient

\[-z + 2y \leq 12\]
\[-z + 3y - x \leq 1\]

\[z + 2x \leq 3\]
\[z + y - x \leq 4\]
\[2z - y \leq 6\]
\[-y \leq 0\]

Rewrite to isolate variable

\[2y - 12 \leq z\]
\[-1 + 3y - x \leq z\]

\[z \leq 3 - 2x\]
\[z \leq 4 - y + x\]
\[2z \leq 6 - y\]
\[-y \leq 0\]
Multiple Variables

\[
\begin{align*}
2y - 12 & \leq z \\
-1 + 3y - x & \leq z \\
z & \leq 3 - 2x \\
z & \leq 4 - y + x \\
2z & \leq 6 - y \\
\end{align*}
\]

Make left and right equal with common multiple:

\[
\begin{align*}
4y - 24 & \leq 2z \\
-2 + 6y - 2x & \leq 2z \\
2z & \leq 6 - 4x \\
2z & \leq 8 - 2y + 2x \\
2z & \leq 6 - y \\
- y & \leq 0 \\
\end{align*}
\]
Multiple Variables

\begin{align*}
4y - 24 & \leq 2z & 2z & \leq 6 - 4x & -y & \leq 0 \\
-2 + 6y - 2x & \leq 2z & 2z & \leq 8 - 2y + 2x \\
2z & \leq 6 - y
\end{align*}

Equivalent to one large equation:

\[
\max(4y - 24, -2 + 6y - 2x) \leq 2z \leq \min(6 - 4y, 8 - 2y + 2x, 6 - y)
\]
Multiple Variables

$$\max(4y - 24, -2 + 6y - 2x) \leq \min(6 - 4y, 8 - 2y + 2x, 6 - y)$$

 Equivalent to pairwise inequalities:

- $$4y - 24 \leq 6 - 4y$$
- $$4y - 24 \leq 8 - 2y + 2x$$
- $$4y - 24 \leq 6 - y$$
- $$-2 + 6y - 2x \leq 6 - 4y$$
- $$-2 + 6y - 2x \leq 8 - 2y + 2x$$
- $$-2 + 6y - 2x \leq 6 - y$$

Add back equations without $$z$$:

- $$-y \leq 0$$
Eventually, we eliminate down to 1 variable

\[ \begin{align*}
x & \leq 2 \\
2x & \leq 3 \\
3x & \leq 4
\end{align*} \]

\[ 1 \leq x \quad 0 \leq 1 \]

Resulting equations are easy to check:

\[ \begin{align*}
1 & \leq 2 \\
0 & \leq 1 \\
2 & \leq 3 \\
3 & \leq 4
\end{align*} \]
Course Updates

Project Proposals
Project Ideas

Some **memorable projects** from past years

— Verifying a neural network using SMT

— Verifying Rust code with a model checker

— SMT for type checking (“Liquid Types”)

— Finding math counterexamples with SMT
Project Proposals

1 page Project Proposals due in a week

You’ve been assigned a group, unless you wanted solo project

Make sure to check the rubrik for what to include:

**Group Project**
- User need
- Goal
- Technical approach
- Per-milestone plan

**Solo Assignment**
- Language design
- Example code
- Likely complications
- Per-milestone plan
Alternative Assignment

The solo assignment is **verifying quicksort**:

- Implement a new programming language
- Add annotations for verification
- Generate verification conditions to SMT
- Verify and test a quicksort implementation
Complexity
Why integer equations are hard
Speed

Algorithm works but produces a lot of inequalities

Group $n$ inequalities into $n_+ + n_- + n_0$ inequalities

Form $n_+n_-$ new inequalities, plus $n_0$ old ones

If $n_+ = n_- = n/2$ and $n_0 = 0$, takes $n$ to $n^2/4$

After all $k$ variables eliminated, $n^{2^k}/4^k$ equations

There are tricks to skip redundant equations…

In practice, variants of linear programming are used
Linear optimization

Finds $x_1, \ldots, x_n$ that form a vector $x$:

\[
\begin{align*}
\text{Maximize} & \quad c^T x \\
\text{Given} & \quad M x \leq b \\
\end{align*}
\]

\{ For real $x$ \}

Efficient algorithms exist for linear optimization

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Year</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplex Algorithm</td>
<td>1947</td>
<td>Average $O(n^3)$, worst-case $O(2^n)$</td>
</tr>
<tr>
<td>Ellipsoid Algorithm</td>
<td>1979</td>
<td>Worst-case $O(n^4)$</td>
</tr>
<tr>
<td>Karmarkar’s Algorithm</td>
<td>1984</td>
<td>Worst-case $O(n^{3.5})$</td>
</tr>
<tr>
<td>Path-following</td>
<td>2018</td>
<td>Worst-case $O(n^{2.372})$, practical sizes $O(n^3)$</td>
</tr>
</tbody>
</table>
Non-integers

Rationals and reals are less complex than integers

Linear optimization efficiently solves linear real inequalities
Tarsi’s algorithm eliminates variables even with multiplication
Multiplication queries relatively efficient (doubly exponential)

Luckily, real-world integer queries tend to be simple
Multiplication and divisibility reasoning rare
Real solutions can guide search for integer solutions
SAT in Integers

SAT problems can be rephrased as integer problems

So solving integer inequalities is as hard as SAT.
Quantifiers

What about quantified linear equations?

Quantifier Elimination \( \exists x, P(x) \leftrightarrow Q \)

Is it always possible?

\( \exists x, x < y \land -y < x \) \quad \exists x, 2x = y

\( 0 < y \) \quad ?

Adding \( n \mid x \) relation makes elimination possible
Quantifiers

What about quantified linear equations?

\[
cx < E_1 \quad E_2 < cx \quad n \mid cx + E_3 \quad n \nmid cx + E_4
\]

- Rewrite \(cx\) as quotient, remainder of \(ns\)
- Test all possible remainders
- Combine remaining inequalities
- Result is formula without \(x\)

Details in Chapter 7 of textbook
Multiplication

For any program $P$, consider $\exists x, P(x) = y$.

There is a polynomial $Q(y, x_1, \ldots, x_{11})$ so that

$$\exists x, P(x) = y \iff \exists x_1 \cdots \exists x_{11}, Q(y, x_1, \ldots, x_{11}) = 0$$

Facts about multiplication as hard as any fact at all!

That said, heuristics can handle some simple cases…
Examples

This is true **only** when \( n \) is prime:

\[
\exists a \exists b \forall (i \leq \bar{n}) \exists s \exists w \exists p \exists q \forall j \forall v \exists e \exists g
\{(s + w)^2 + 3w + s = 2i \land ([j = w \land v = q] \lor [j = 3i \land v = p + q] \\
\lor [j = s \land (v = p \lor (i = \bar{n} \land v = q + \bar{n}))] \lor [j = 3i + 1 \land v = pq] \\
\rightarrow a = v + e + e jb \land v + g = jb)\}. 
\]

There’s an \( n \) for which this **cannot** be proven or disproven:

\[
\exists a \forall i \leq \bar{n} \exists s \exists w \exists p \exists q \forall j \forall v \exists e \exists g 
\{(s + w)^2 + 3w + s = 2i \land ([j = w \land v = q] \lor [j = 3i \land v = p + q] \\
\lor [j = s \land (v = p \lor (i = \bar{n} \land v = q + \bar{n}))] \lor [j = 3i + 1 \land v = pq] \rightarrow a = v + e + e jb \\
\land v + g = jb)\}. 
\]
Next class: Integers

To do:

- Course feedback
- Read Chapter 8
- Assignment 2
Integers

Solving conjunctions of integer equalities
  Back to high school with elimination and substitution

Variable elimination for integer inequalities
  Combining two inequalities into one

How fast can integer reasoning be?
  SAT, multiplication, and rational numbers
DECOMPOSITION

READ OVER

WRITE
FRAGMENT

ARRAY

PROPERTIES
To do:

☐ Course feedback
☐ Read Chapter 8
☐ Assignment 2

Next class:
Integers