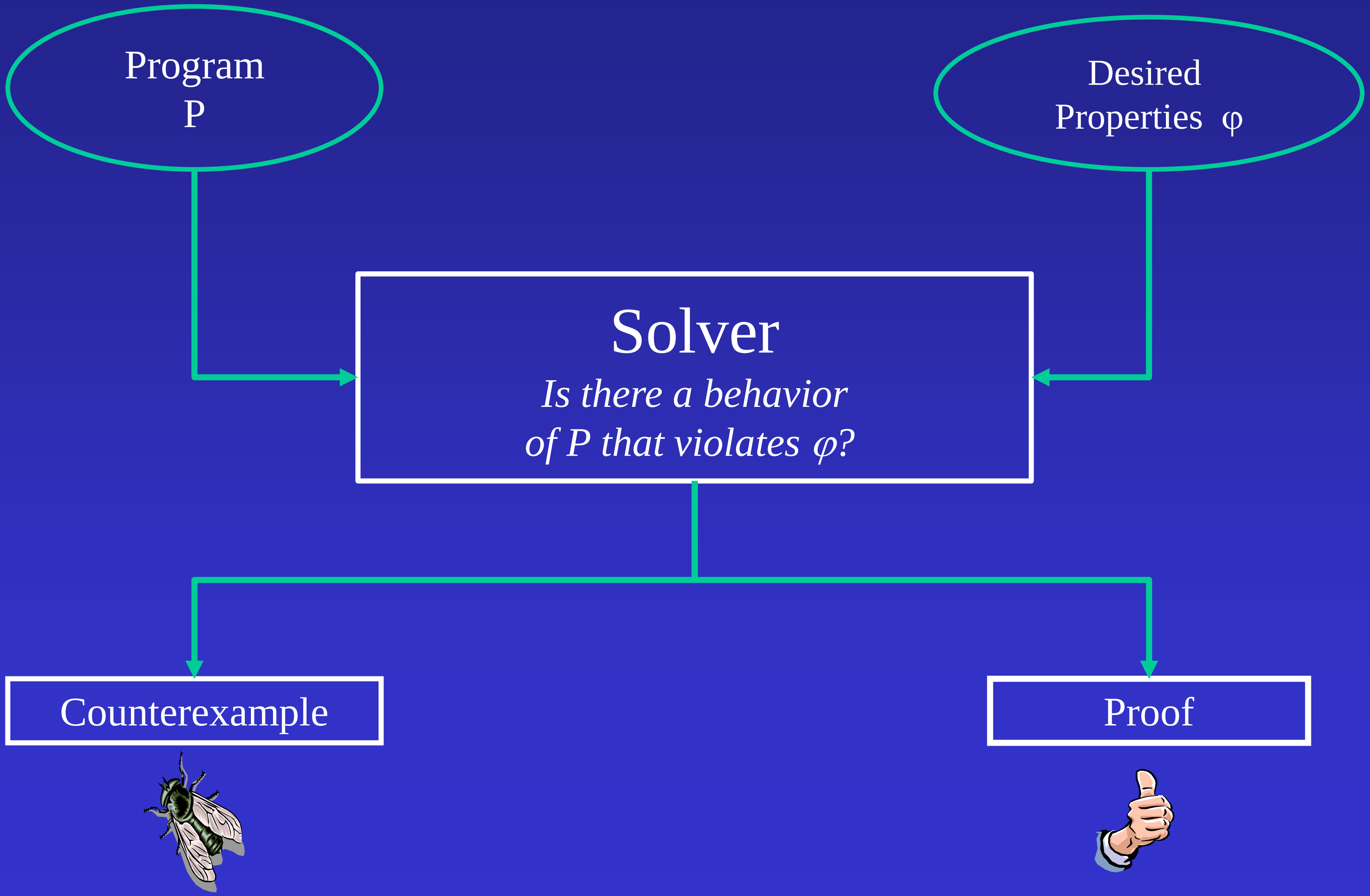
Introduction to Software Synthesis

Mooly Sagiv
Ideas and Slides taken from Sumit Gulwani, Armando *Solar-Lezama*
Eran Yahav, Emain Torlak,

Recap

Problem	Tools
Propositional SAT solving	MiniSat, Z3
First order solving with theories (SMT)	Z3, CVC3
Bounded Model Checking	CBMC, JBMC
Concolic Execution	DART, KLEE, SAGE, Cloud9, Mayhem
Static analysis	SLAM(SDV), Astrée, TVLA, CSSV
Testing	PITTEST, AFL
Program Synthesis	SKETCH(MIT), Rosettee(UWASH)

Verification vs. Synthesis



Verification vs. Synthesis

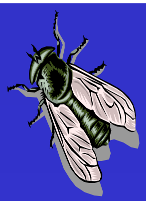
Set of programs $\prod$

Desired Properties φ

Solver

Is there a program $P \in \prod$ that satisfies φ?

Counterexample



Efficient Program P

Potential Applications

- Low level programming
  - Configuration
  - Bit manipulation
- Programming for end-users
  - Excel
  - Spreadsheet
  - Spark

What is software synthesis?

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-5, NO. 4, JULY 1979

Synthesis: Dreams $\Longrightarrow$ Programs

ZOHAR MANNA AND RICHARD WALDINGER

: techniques are presented for deriving programs
iven specifications. The specifications express the
:d program without giving any hint of the algo-
l. The basic approach is to transform the specifi-
cording to certain rules, until a satisfactory pro-

INTRODUCTION

IN RECENT years there has been increasing activity in the field of program verification. The goal of these efforts is to construct computer systems for determining whether a

Zohar Manna

Richard Waldinger

Synthesis Methods

- Deductive Synthesis
  - Derive the low level implementation from high level implementation from high level specificaion
- Inductive Synthesis
  - Synthesize a program whose behavior satisfies a set of input/output examples

Why now?

- Computer programs are every ware
- Small programs are tricky
- Maturity of underlying technology
  - Machine learning algorithms
  - SMT
  - Powerful hardware

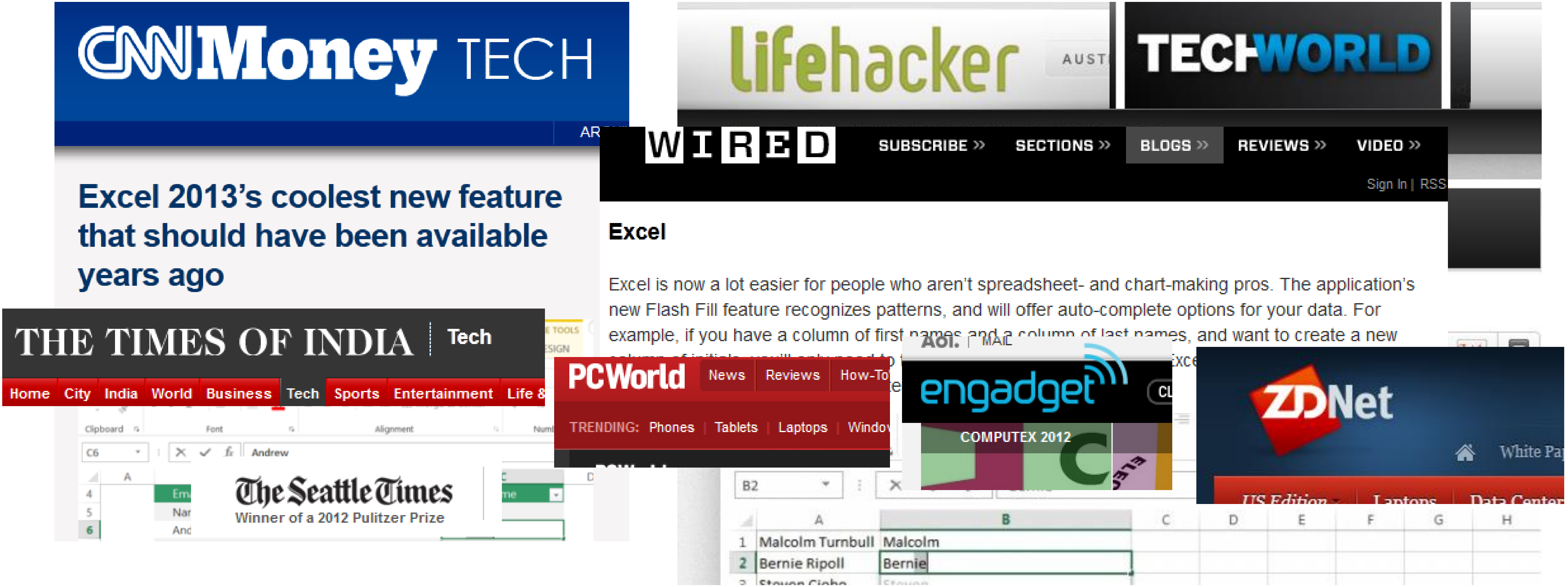
Motivation

99% of computer users cannot program!

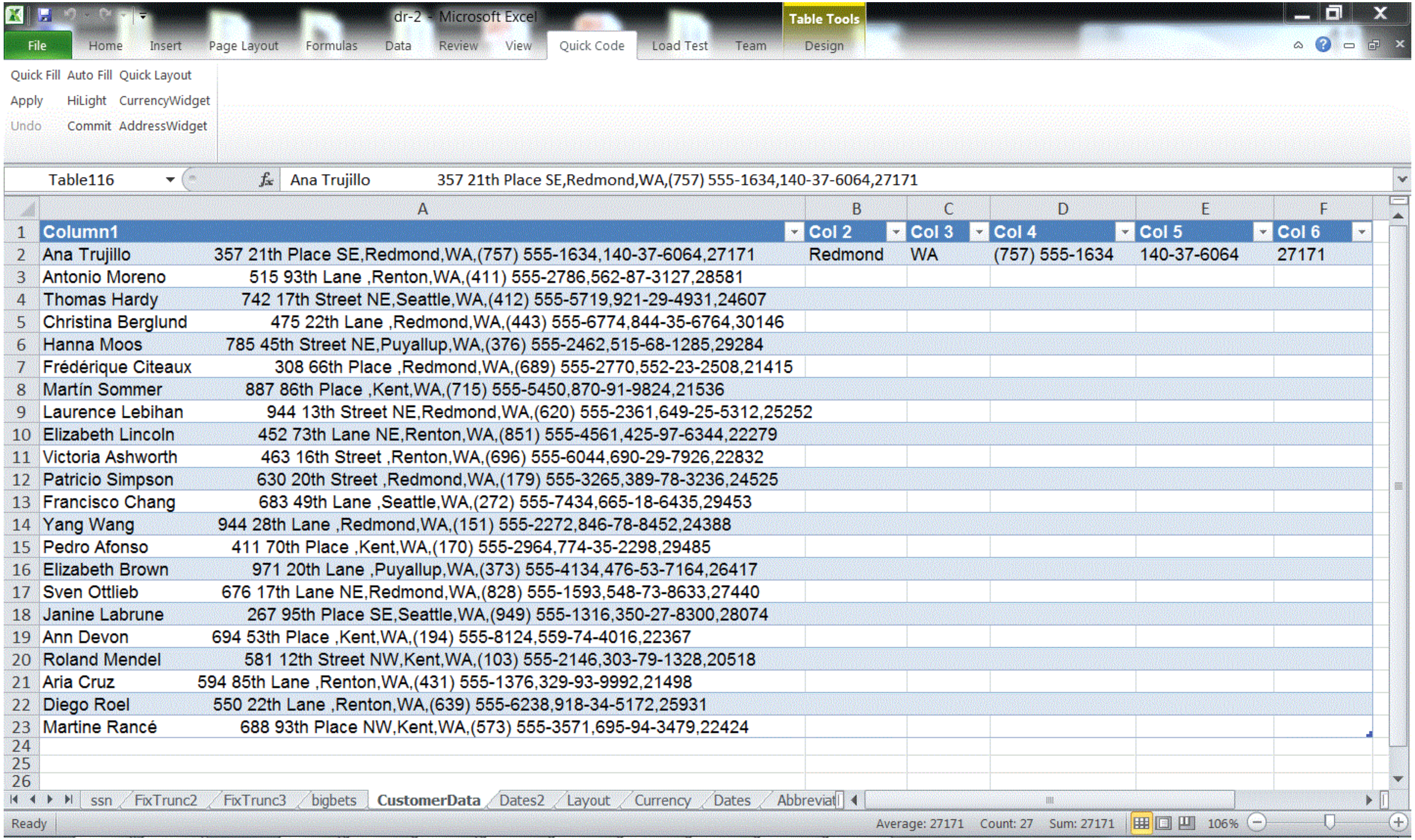
They struggle with simple repetitive tasks.

Spreadsheet help forums

Real world application of synthesis



FlashFill: a feature of Excel 2013 (Sumit Gulwani POPL'11)



	A	B	C	D	E	F
1	Column1	Col 2	Col 3	Col 4	Col 5	Col 6
2	Ana Trujillo 357 21th Place SE,Redmond,WA,(757) 555-1634,140-37-6064,27171	Redmond	WA	(757) 555-1634	140-37-6064	27171
3	Antonio Moreno 515 93th Lane ,Renton,WA,(411) 555-2786,562-87-3127,28581					
4	Thomas Hardy 742 17th Street NE,Seattle,WA,(412) 555-5719,921-29-4931,24607					
5	Christina Berglund 475 22th Lane ,Redmond,WA,(443) 555-6774,844-35-6764,30146					
6	Hanna Moos 785 45th Street NE,Puyallup,WA,(376) 555-2462,515-68-1285,29284					
7	Frédérique Citeaux 308 66th Place ,Redmond,WA,(689) 555-2770,552-23-2508,21415					
8	Martín Sommer 887 86th Place ,Kent,WA,(715) 555-5450,870-91-9824,21536					
9	Laurence Lebihan 944 13th Street NE,Redmond,WA,(620) 555-2361,649-25-5312,25252					
10	Elizabeth Lincoln 452 73th Lane NE,Renton,WA,(851) 555-4561,425-97-6344,22279					
11	Victoria Ashworth 463 16th Street ,Renton,WA,(696) 555-6044,690-29-7926,22832					
12	Patricio Simpson 630 20th Street ,Redmond,WA,(179) 555-3265,389-78-3236,24525					
13	Francisco Chang 683 49th Lane ,Seattle,WA,(272) 555-7434,665-18-6435,29453					
14	Yang Wang 944 28th Lane ,Redmond,WA,(151) 555-2272,846-78-8452,24388					
15	Pedro Afonso 411 70th Place ,Kent,WA,(170) 555-2964,774-35-2298,29485					
16	Elizabeth Brown 971 20th Lane ,Puyallup,WA,(373) 555-4134,476-53-7164,26417					
17	Sven Ottlieb 676 17th Lane NE,Redmond,WA,(828) 555-1593,548-73-8633,27440					
18	Janine Labrune 267 95th Place SE,Seattle,WA,(949) 555-1316,350-27-8300,28074					
19	Ann Devon 694 53th Place ,Kent,WA,(194) 555-8124,559-74-4016,22367					
20	Roland Mendel 581 12th Street NW,Kent,WA,(103) 555-2146,303-79-1328,20518					
21	Aria Cruz 594 85th Lane ,Renton,WA,(431) 555-1376,329-93-9992,21498					
22	Diego Roel 550 22th Lane ,Renton,WA,(639) 555-6238,918-34-5172,25931					
23	Martine Rancé 688 93th Place NW,Kent,WA,(573) 555-3571,695-94-3479,22424					
24						
25						
26						

FlashFill: a feature of Excel 2013

Column1	Col 2	Col 3	Col 4	Col 5	Col 6
Ana Trujillo 357 21th Place SE,Redmond,WA,(757) 555-1634,140-37-6064,27171	Redmond	WA	(757) 555-1634	140-37-6064	27171
Antonio Moreno 515 93th Lane ,Renton,WA,(411) 555-2786,562-87-3127,28581	Renton	WA	(411) 555-2786	562-87-3127	28581
Thomas Hardy 742 17th Street NE,Seattle,WA,(412) 555-5719,921-29-4931,24607	Seattle	WA	(412) 555-5719	921-29-4931	24607
Christina Berglund 475 22th Lane ,Redmond,WA,(443) 555-6774,844-35-6764,30146	Redmond	WA	(443) 555-6774	844-35-6764	30146
Hanna Moos 785 45th Street NE,Puyallup,WA,(376) 555-2462,515-68-1285,29284	Puyallup	WA	(376) 555-2462	515-68-1285	29284
Frédérique Citeaux 308 66th Place ,Redmond,WA,(689) 555-2770,552-23-2508,21415	Redmond	WA	(689) 555-2770	552-23-2508	21415
Martín Sommer 887 86th Place ,Kent,WA,(715) 555-5450,870-91-9824,21536	Kent	WA	(715) 555-5450	870-91-9824	21536
Laurence Lebihan 944 13th Street NE,Redmond,WA,(620) 555-2361,649-25-5312,2525	Redmond	WA	(620) 555-2361	649-25-5312	25252
Elizabeth Lincoln 452 73th Lane NE,Renton,WA,(851) 555-4561,425-97-6344,22279	Renton	WA	(851) 555-4561	425-97-6344	22279
Victoria Ashworth 463 16th Street ,Renton,WA,(696) 555-6044,690-29-7926,22832	Renton	WA	(696) 555-6044	690-29-7926	22832
Patricio Simpson 630 20th Street ,Redmond,WA,(179) 555-3265,389-78-3236,24525	Redmond	WA	(179) 555-3265	389-78-3236	24525
Francisco Chang 683 49th Lane ,Seattle,WA,(272) 555-7434,665-18-6435,29453	Seattle	WA	(272) 555-7434	665-18-6435	29453
Yang Wang 944 28th Lane ,Redmond,WA,(151) 555-2272,846-78-8452,24388	Redmond	WA	(151) 555-2272	846-78-8452	24388
Pedro Afonso 411 70th Place ,Kent,WA,(170) 555-2964,774-35-2298,29485	Kent	WA	(170) 555-2964	774-35-2298	29485
Elizabeth Brown 971 20th Lane ,Puyallup,WA,(373) 555-4134,476-53-7164,26417	Puyallup	WA	(373) 555-4134	476-53-7164	26417
Sven Ottlieb 676 17th Lane NE,Redmond,WA,(828) 555-1593,548-73-8633,27440	Redmond	WA	(828) 555-1593	548-73-8633	27440
Janine Labrune 267 95th Place SE,Seattle,WA,(949) 555-1316,350-27-8300,28074	Seattle	WA	(949) 555-1316	350-27-8300	28074
Ann Devon 694 53th Place ,Kent,WA,(194) 555-8124,559-74-4016,22367	Kent	WA	(194) 555-8124	559-74-4016	22367
Roland Mendel 581 12th Street NW,Kent,WA,(103) 555-2146,303-79-1328,20518	Kent	WA	(103) 555-2146	303-79-1328	20518
Aria Cruz 594 85th Lane ,Renton,WA,(431) 555-1376,329-93-9992,21498	Renton	WA	(431) 555-1376	329-93-9992	21498
Diego Roel 550 22th Lane ,Renton,WA,(639) 555-6238,918-34-5172,25931	Renton	WA	(639) 555-6238	918-34-5172	25931
Martine Rancé 688 93th Place NW,Kent,WA,(573) 555-3571,695-94-3479,22424	Kent	WA	(573) 555-3571	695-94-3479	22424

Inductive Synthesis

Synthesize a program whose behavior satisfies a set of examples

Doesn't machine learning do that?

Traditional Bayesian Machine Learning

- Learn a function from a set of examples
- Scalability is very important, algorithms must scale to millions of data points
- Data is assumed to be noisy;
  - need to avoid overfitting
- Space of possible functions is highly stylized
- Background knowledge incorporated as preprocessing and feature selection

Inductive Synthesis

- Learn a function from a set of examples
- Scalability is not so important, usually we are dealing with small numbers of examples
- Data is assumed to be clean
  - It's annoying when user says f(x)=y and the system assumes the user is wrong and decides that f(x)=z
- Space of possible functions can be arbitrary
- Background knowledge encoded in the description of the space and in the search itself

Functional Synthesis

Goal: Synthesize a function that satisfies a specification

- How do we know the specification has been satisfied? Isn’t verification itself already quite hard?
- Can we leverage inductive synthesis machinery for this problem?
- What is the relevant space of functions?
- How do we explore this space efficiently?

The Sketch Synthesis System

Armando Solar-Lezama
bit.ly/iptutorial2015

CSAIL

Language Design Strategy

Extend base language with one construct

Constant hole: ??

```
int bar (int x)
{
    int t = x * ??;
    assert t == x + x;
    return t;
}
```

→

```
int bar (int x)
{
    int t = x * 2;
    assert t == x + x;
    return t;
}
```

Synthesizer replaces ?? with a constant

High-level constructs defined in terms of ??

Integer Generator→ Sets of Expressions

Expressions with ?? == sets of expressions

- linear expressions — `x*?? + y*??`
- polynomials — `x*x*?? + x*?? + ??`
- sets of variables — `?? ? x : y`

Example: Registerless Swap

Swap two words without an extra temporary

```
int W = 32;

void swap(ref bit[W] x, ref bit[W] y){
    if(??){ x = x ^ y;}else{ y = x ^ y; }
    if(??){ x = x ^ y;}else{ y = x ^ y; }
    if(??){ x = x ^ y;}else{ y = x ^ y; }
}

harness void main(bit[W] x, bit[W] y){
    bit[W] tx = x; bit[W] ty = y;
    swap(x, y);
    assert x==ty && y == tx;
}
```

From simple to complex holes

We need to compose ?? to form complex holes

Borrow ideas from generative programming

- Define generators to produce families of functions
- Use partial evaluation aggressively

Generators

Look like a function

- but are partially evaluated into their calling context

Key feature:

- Different invocations → Different code
- Can recursively define arbitrary families of programs

Example: Least Significant Zero Bit

- 0010 0101 → 0000 0010

```
int W = 32;
bit[W] isolate0 (bit[W] x) {        // W: word size
        bit[W] ret = 0;
        for (int i = 0; i < W; i++)
                if (!x[i]) { ret[i] = 1; return ret;   }
}
```

Trick:

- Adding 1 to a string of ones turns the next zero to a 1
- i.e. 000111 + 1 = 001000

Sample Generator

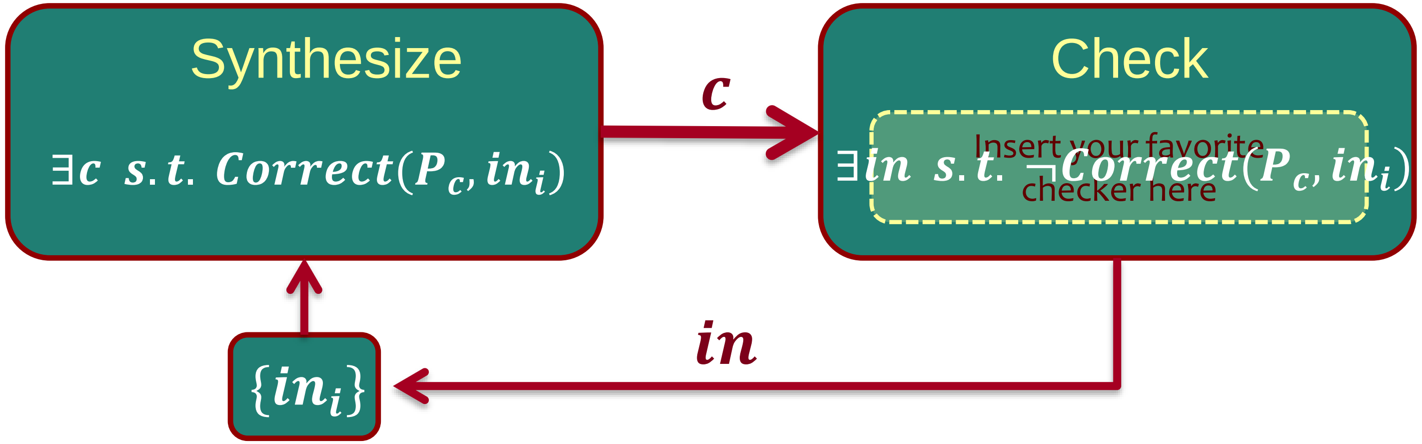
```
/**
 * Generate the set of all bit-vector expressions
 * involving +, &, xor and bitwise negation (~).
 * the bnd param limits the size of the generated expression.
 */

generator bit[W] gen(bit[W] x, int bnd){
    assert bnd > 0;
    if(??) return x;
    if(??) return ??;
    if(??) return ~gen(x, bnd-1);
    if(??){
        return {| gen(x, bnd-1) (+ | & | ^) gen(x, bnd-1) |};
    }
}
```

Example: Least Significant Zero Bit

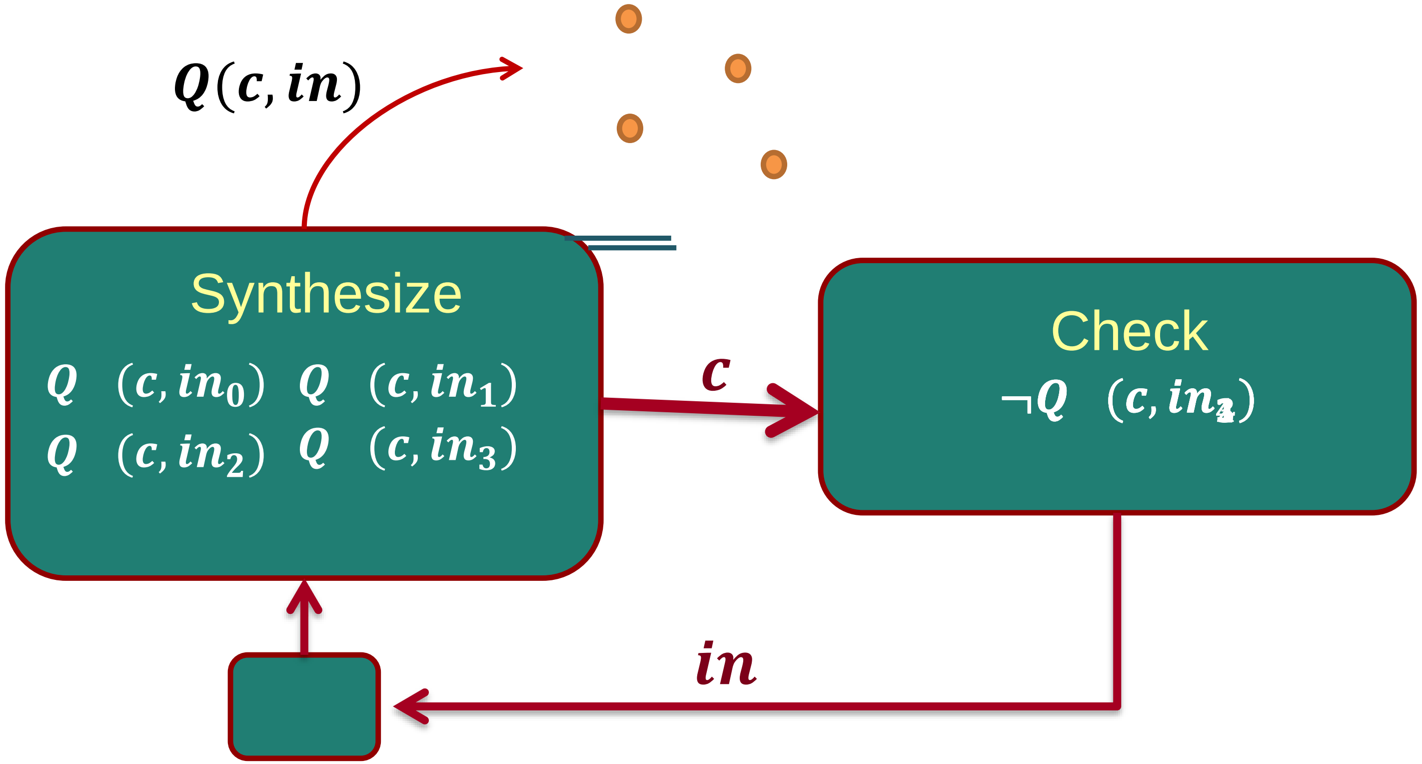
```
generator bit[W] gen(bit[W] x, int bnd){
    assert bnd > 0;
    if(??) return x;
    if(??) return ??;
    if(??) return ~gen(x, bnd-1);
    if(??){
        return {| gen(x, bnd-1) (+ | & | ^) gen(x, bnd-1) |};
    }
}

bit[W] isolate0sk (bit[W] x)  implements isolate0 {
     return gen(x, 3);
}
```

HOW DOES IT WORK?

CEGIS Synthesis algorithm



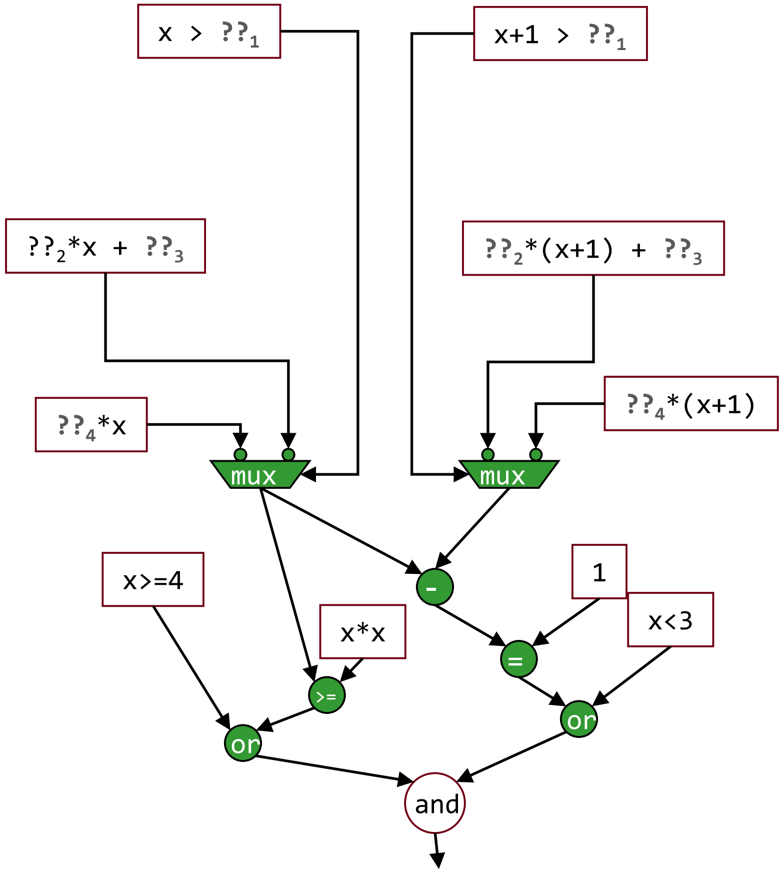
CEGIS



$Q(c, in)$
Synthesize
$Q(c, in_0)$ $Q(c, in_1)$
$Q(c, in_2)$ $Q(c, in_3)$
c
Check
$\neg Q(c, in_4)$
in

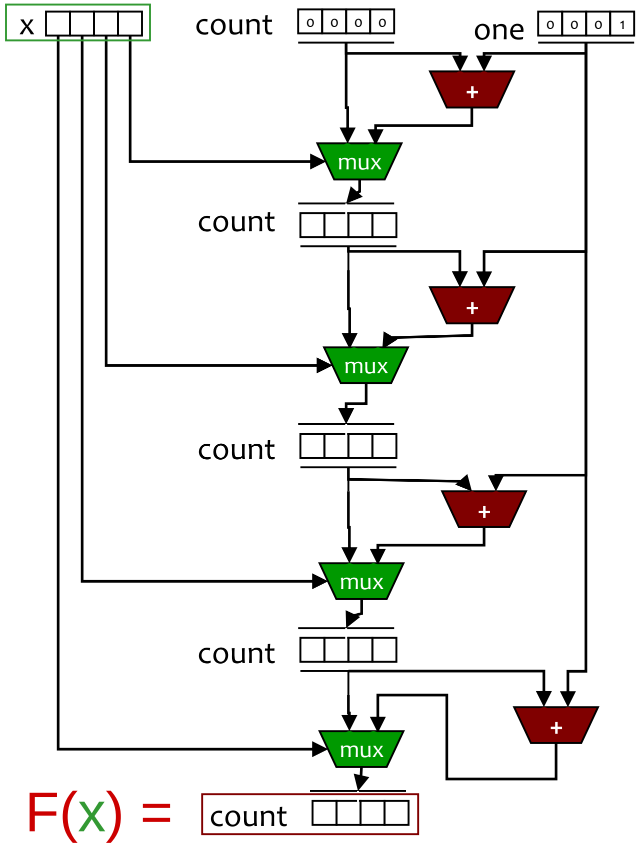
A sketch as a constraint system

```
int lin(int x){
   if(x > ??1)
      return ??2*x + ??3;
   else
      return ??4*x;
}

void main(int x){
   int t1 = lin(x);
   int t2 = lin(x+1);

   if(x<4) assert t1 >=  x*x;

   if(x>=3) assert t2-t1 == 1;
}
```

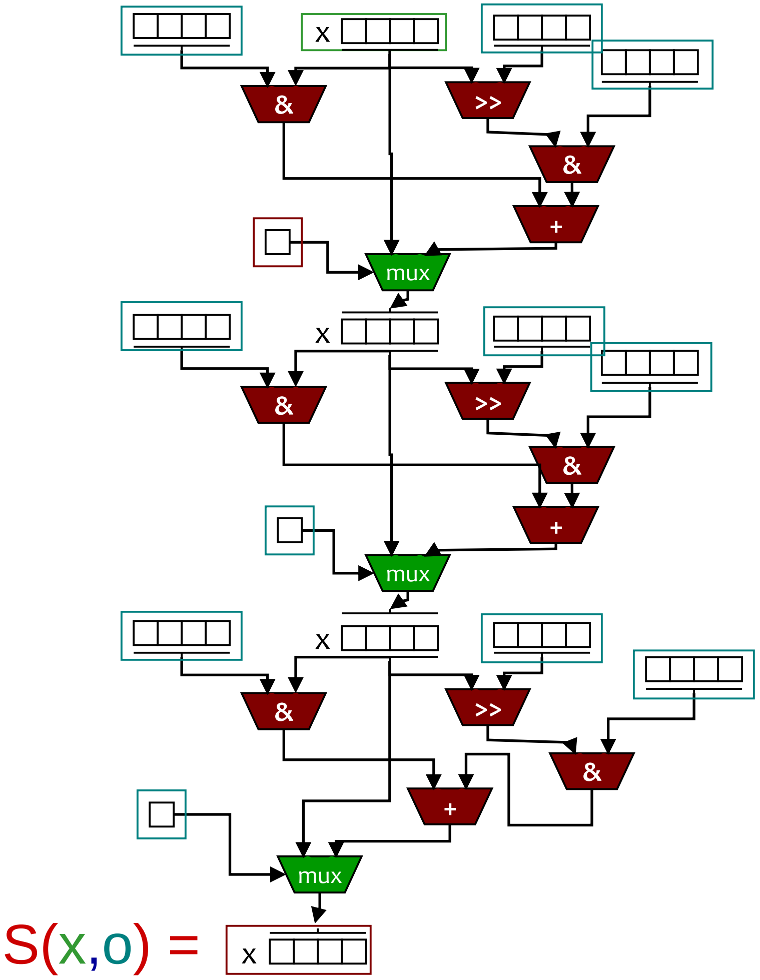


Ex : Population count. 0010 0110 → 3

```
int pop (bit[W] x)
{
  int count = 0;
  for (int i = 0; i < W; i++) {
    if (x[i]) count++;
  }
  return count;
}
```



```
int popSketched (bit[W] x)
    implements pop {
    repeat(??) {
            x = (x & ??)
            + ((x >> ??) & ??);
    }
    return x;
}
```



Synthesizing Locks and Fences

Input:

- A concurrent program

Assertions

Output:

- Locks that guarantee that the assertions hold

```
x = 1 ;            ||  x = x + 1 ;
x = x + 1 ;
assert x  == 3
```


Veselin Raychev, Martin T. Vechev, Eran Yahav:
Automatic Synthesis of Deterministic Concurrency. SAS 2013: 283-303

Michael Kuperstein, Martin T. Vechev, Eran Yahav:
Automatic inference of memory fences. SIGACT News 43(2): 108-123 (2012)

Synthesizing API calls

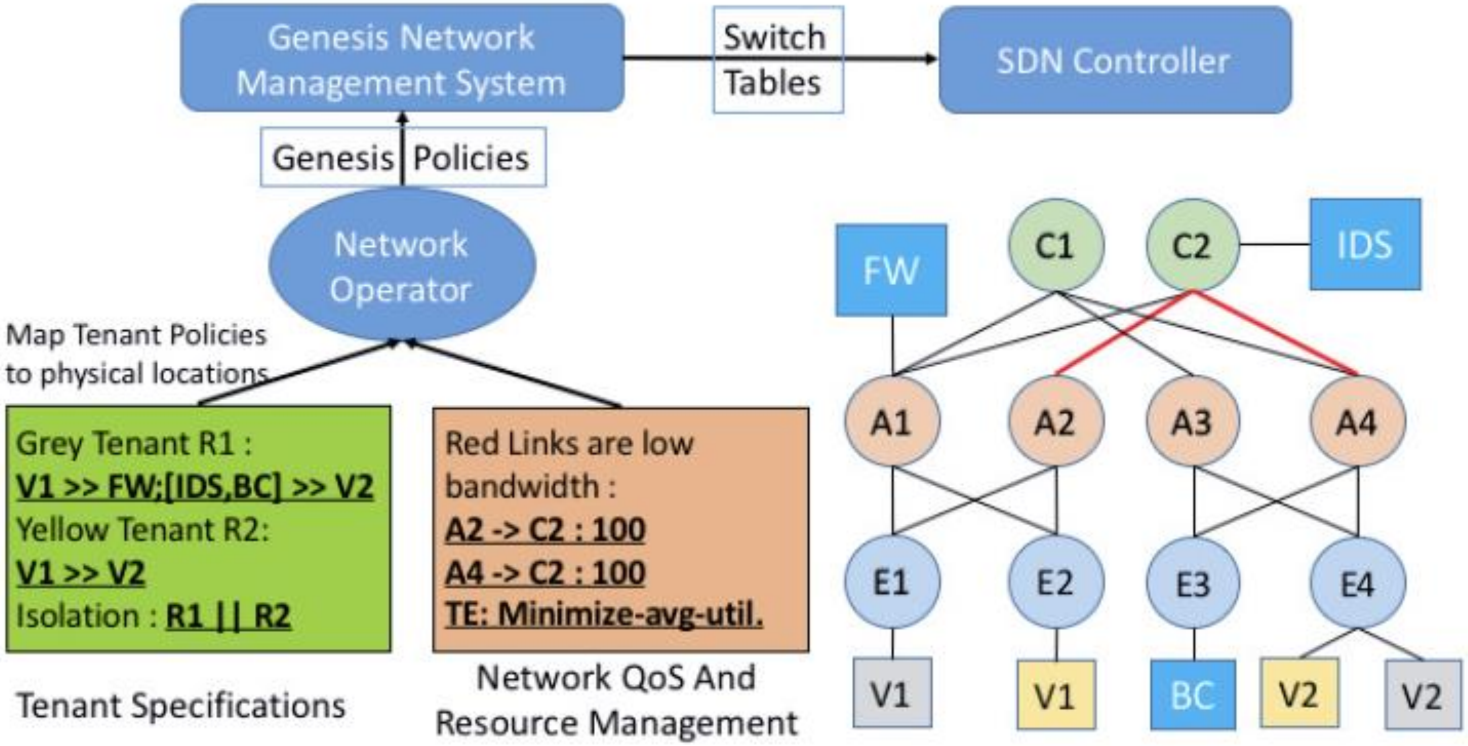
- Modern libraries provide a lot of functionality
- But hard to figure out the right sequence of calls
  - A lot of boilerplate code

```
public void test1()
{ Area a1 = new Area(new Rectangle(0, 0, 10, 2));
  Area a2 = new Area(new Rectangle(-2, 0, 2,
10));
 Point2D p = new Point2D.Double(0, 0);
assertTrue(a2.equals(rotate(a1, p, Math.PI/2))); }
```

```
Area rotate(Area obj, Point2D pt, double angle)
{ AffineTransform at = new AffineTransform();
double x = pt.getX();
double y = pt.getY();
 at.setToRotation(angle, x, y);
Area obj2 = obj.createTransformedArea(at);
return obj2;
}
```

Yu Feng, Ruben Martins, Yuepeng Wang, Isil Dillig, Thomas W. Reps:
Component-based synthesis for complex APIs. POPL 2017: 599-612
John K. Feser, Swarat Chaudhuri, Isil Dillig:
Synthesizing data structure transformations from input-output examples. PLDI 2015: 229-239

Synthesizing Network Policies



Kausik Subramanian, Loris D'Antoni, Aditya Akella:
Genesis: synthesizing forwarding tables in multi-tenant networks. POPL 2017: 572-585

Synthesizing Cloud Configurations

- Number of machines
- Network
- Maximize throughputs
- Minimize cost