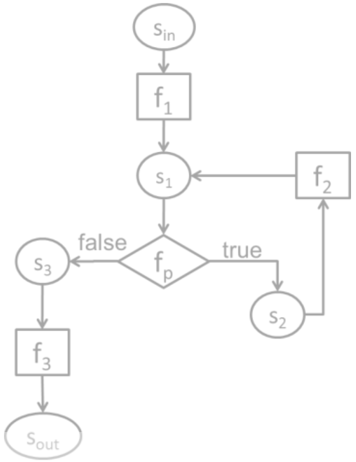
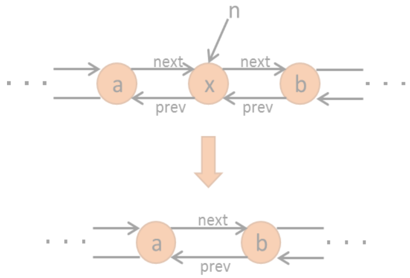
$$\exists c \forall in\ Q(c, in)$$

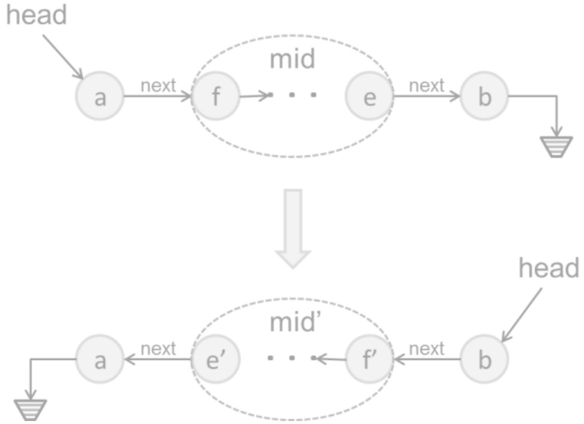
```
/* Average of x and y without using x+y (avoid overflow)*/
int avg(int x, int y){
  int t = expr({x/2, y/2, x%2, y%2, 2 }, {PLUS, DIV});
  assert t == (x+y)/2;
  return t;
}
```

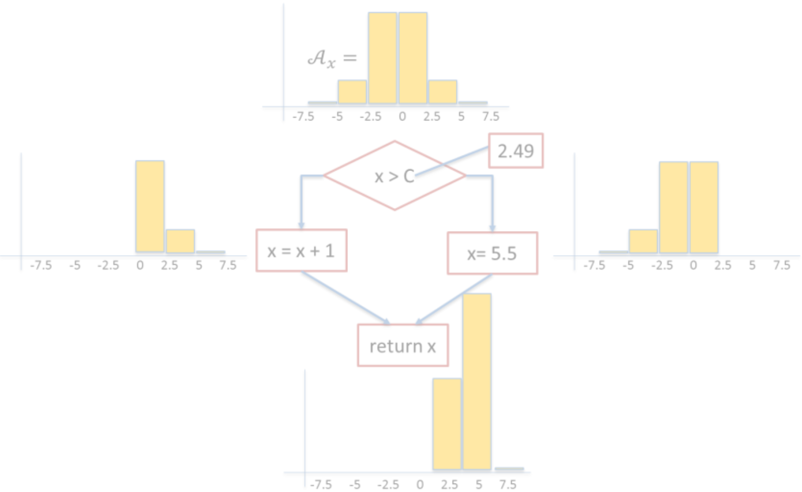





```
{
   s  = n.succ;
   p = n.pred;
   p.succ = s;
   s.pred = p;
}
```

Program Synthesis





$$\varphi(p)$$

$$Sk[c](in)$$

Lecture 1
Course Overview and Introduction to Synthesis

Nadia Polikarpova

Instructor

Nadia Polikarpova

- Assistant Professor
- Member of the ProgSys group since last year
- Before that: postdoc at MIT with Armando Solar-Lezama
- Research areas: program synthesis and program verification

Logistics

Lecture

- When: Tue/Thu 3:30-4:50
- Where: CSE 4140

Office Hours

- When: Tue 5:00-6:00
- Where: my office (CSE 3102)

Course Website

- https://github.com/nadia-polikarpova/cse291-program-synthesis
- To ask questions: use the issue tracker (you'll need a github account)

Goals and activities

1. Understand what program synthesis can do and how

lectures
read and discuss research papers

2. Use existing synthesis tools

3. Contribute to synthesis techniques and tools towards a publication in an academic conference

project

Evaluation

2 credits

Class Participation: 5%

- answer questions in class
- answer your peers' questions on the issue tracker

Paper reviews: 45%

- 9 papers, 5% each

Final Project: 50%

- Team formed by deadline: 5%
- 1-page project proposal: 15%
- Project presentation: 15%
- Final report: 15%

4 credits

Papers reviews

Due on Wed of weeks 2-10, by the end of the day

- First review due next week

Posted on the syllabus page at least a week before due date

Reviews submitted through EasyChair: see wiki

- You will get an email invitation to join the “Program Committee” for CSE291-2018
- Create an EasyChair account **using the email from the invitation** (@uscd instead of @eng!)
- Login as PC Member, not Author
- You will be able to see reviews by others once you submit your own

Review content: see wiki

Project

Kinds of projects:

- re-implement a technique from a paper
- apply existing synthesis framework to a new domain
- extend/improve existing synthesis algorithm or tool
- develop a new synthesis algorithm or tool
- ...

Judged in terms of

- quality of execution
- originality
- scope

Project

Team forming

Teams of 2/3

Pick a project:

- List of suggested projects on the wiki (but feel free to propose your own)
- Talk to me!

Proposal

One page: explain what you plan to do and give some evidence that you've started to work on it

Presentation

During the exam week

- ~5-10 min per project

Report

3-8 pages, structured like a research paper

And now the good stuff

What is program synthesis?

The FORTRAN Automatic Coding System

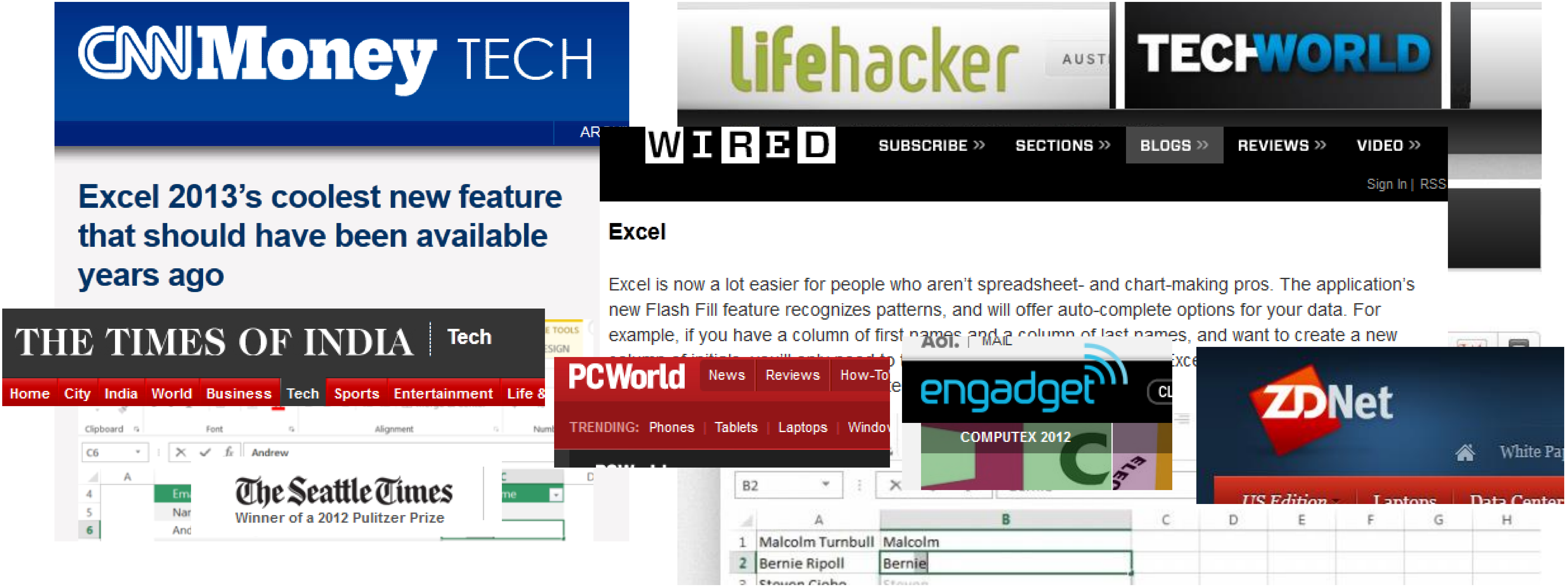
J. W. BACKUS†, R. J. BEEBER†, S. BEST‡, R. GOLDBERG†, L. M. HAIBT†, H. L. HERRICK†, R. A. NELSON†, D. SAYRE†, P. B. SHERIDAN†, H. STERN†, I. ZILLER†, R. A. HUGHES§, AND R. NUTT||

INTRODUCTION

THE FORTRAN project was begun in the summer of 1954. Its purpose was to reduce by a large factor the task of preparing scientific problems for IBM's next large computer, the 704. If it were possible for the 704 to code problems for itself and produce as

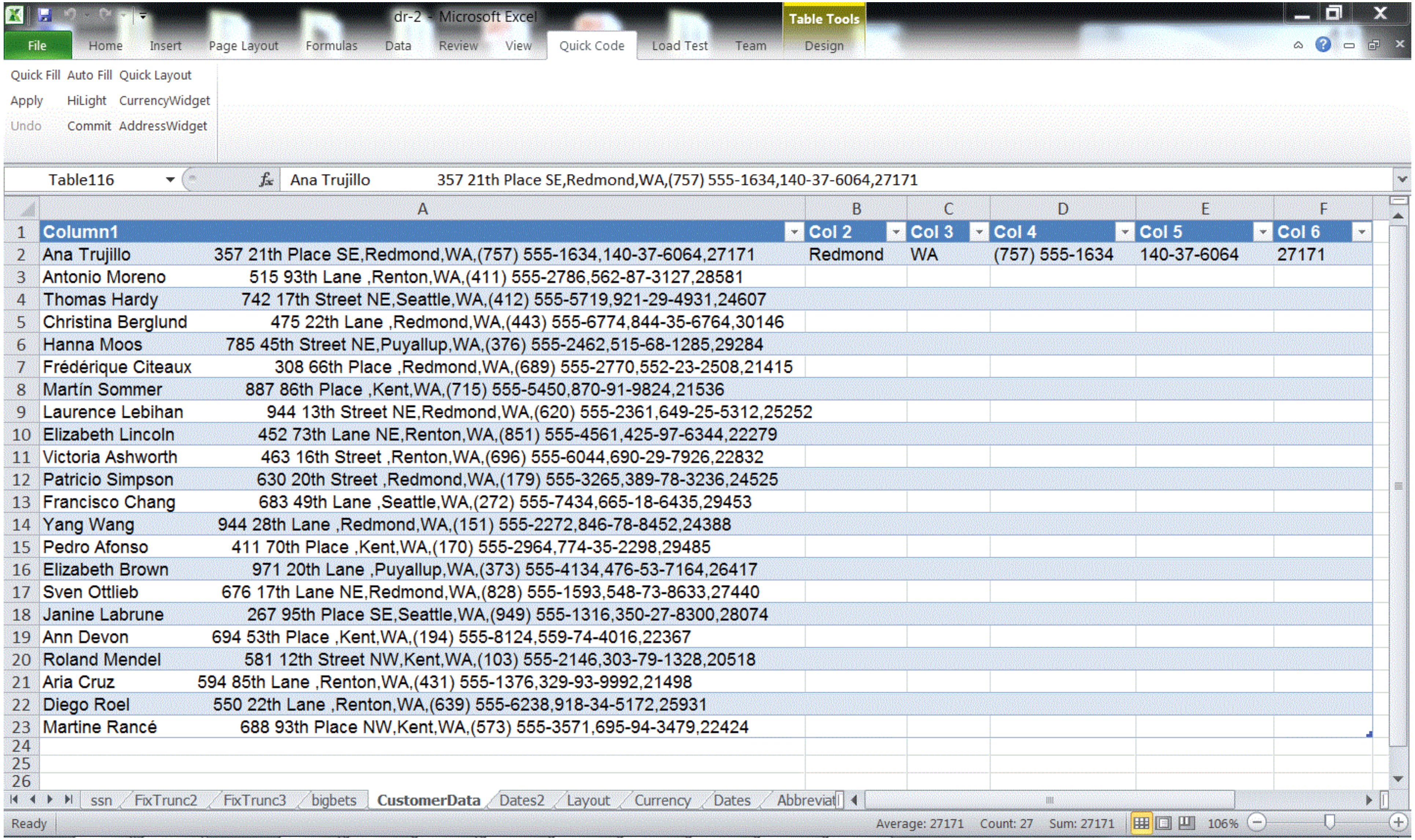
system is now complete. It has two components: the FORTRAN language, in which programs are written, and the translator or executive routine for the 704 which effects the translation of FORTRAN language programs into 704 programs. Descriptions of the FORTRAN language and the translator form the principal

Modern program synthesis: FlashFill



FlashFill: a feature of Excel 2013

[Gulwani 2011]



	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
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	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	Seattle	WA	(412) 555-5719	921-29-4931	24607
5	Christina Berglund 475 22th Lane ,Redmond,WA,(443) 555-6774,844-35-6764,30146	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	Puyallup	WA	(376) 555-2462	515-68-1285	29284
7	Frédérique Citeaux 308 66th Place ,Redmond,WA,(689) 555-2770,552-23-2508,21415	Redmond	WA	(689) 555-2770	552-23-2508	21415
8	Martín Sommer 887 86th Place ,Kent,WA,(715) 555-5450,870-91-9824,21536	Kent	WA	(715) 555-5450	870-91-9824	21536
9	Laurence Lebihan 944 13th Street NE,Redmond,WA,(620) 555-2361,649-25-5312,2525	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	Renton	WA	(851) 555-4561	425-97-6344	22279
11	Victoria Ashworth 463 16th Street ,Renton,WA,(696) 555-6044,690-29-7926,22832	Renton	WA	(696) 555-6044	690-29-7926	22832
12	Patricio Simpson 630 20th Street ,Redmond,WA,(179) 555-3265,389-78-3236,24525	Redmond	WA	(179) 555-3265	389-78-3236	24525
13	Francisco Chang 683 49th Lane ,Seattle,WA,(272) 555-7434,665-18-6435,29453	Seattle	WA	(272) 555-7434	665-18-6435	29453
14	Yang Wang 944 28th Lane ,Redmond,WA,(151) 555-2272,846-78-8452,24388	Redmond	WA	(151) 555-2272	846-78-8452	24388
15	Pedro Afonso 411 70th Place ,Kent,WA,(170) 555-2964,774-35-2298,29485	Kent	WA	(170) 555-2964	774-35-2298	29485
16	Elizabeth Brown 971 20th Lane ,Puyallup,WA,(373) 555-4134,476-53-7164,26417	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	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	Seattle	WA	(949) 555-1316	350-27-8300	28074
19	Ann Devon 694 53th Place ,Kent,WA,(194) 555-8124,559-74-4016,22367	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	Kent	WA	(103) 555-2146	303-79-1328	20518
21	Aria Cruz 594 85th Lane ,Renton,WA,(431) 555-1376,329-93-9992,21498	Renton	WA	(431) 555-1376	329-93-9992	21498
22	Diego Roel 550 22th Lane ,Renton,WA,(639) 555-6238,918-34-5172,25931	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	Kent	WA	(573) 555-3571	695-94-3479	22424

Modern program synthesis: Sketch

[Solar-Lezama 2013]

Problem: isolate the least significant zero bit in a word

- example: 0010 0101 → 0000 0010

Easy to implement with a loop

```
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; }
}
```

Can this be done more efficiently with bit manipulation?

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

Sketch: space of possible implementations

```
/**
 * 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) |};
    }
}
```

Sketch: synthesis goal

```
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);
}
```

Modern program synthesis: Synquid

[Polikarpova et al. 2016]

Problem: intersection of strictly sorted lists

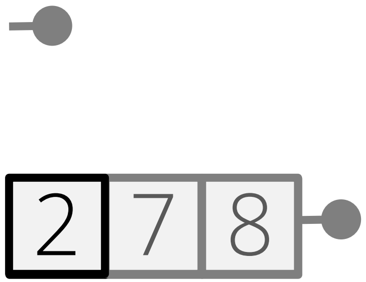
- example: intersect [4, 8, 15, 16, 23, 42] [8, 16, 32, 64] → [8, 16]

Also: we want a guarantee that it's correct on all inputs!

Synquid: datatypes

Step 1: define a data type for sorted lists

```
data List a where
  Nil :: List a
  Cons :: h:a →
          t:List a →
          List a
```



Synquid: components and synthesis goal

Step 2: define a set of components

- Which primitive operations is our function likely to use?
- Here: {`Nil`, `Cons`, `<`}

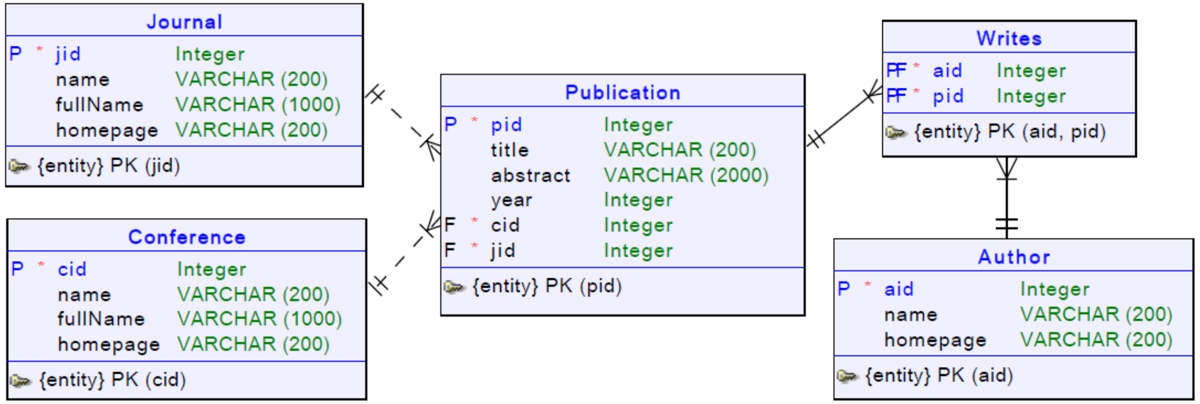
Step 3: define synthesis goal as a type

```
intersect :: xs:SList a  →  ys:SList a  →
             {ν:SList a | elems ν = elems xs ∩ elems ys}
```

Modern program synthesis: SQLizer

[Yaghmazadeh et al. 2017]

Problem: “Find the number of papers in OOPSLA 2010”



Output:

```sql
SELECT count(Publication.pid)
FROM Publication JOIN Conference ON Publication.cid = Conference.cid
WHERE Conference.name = "OOPSLA" AND Publication.year = 2010
```

What is program synthesis?

Automatic programming?

- but I have to tell the computer what I want...

level of abstraction

???

Python, Haskell, ...

C

assembly

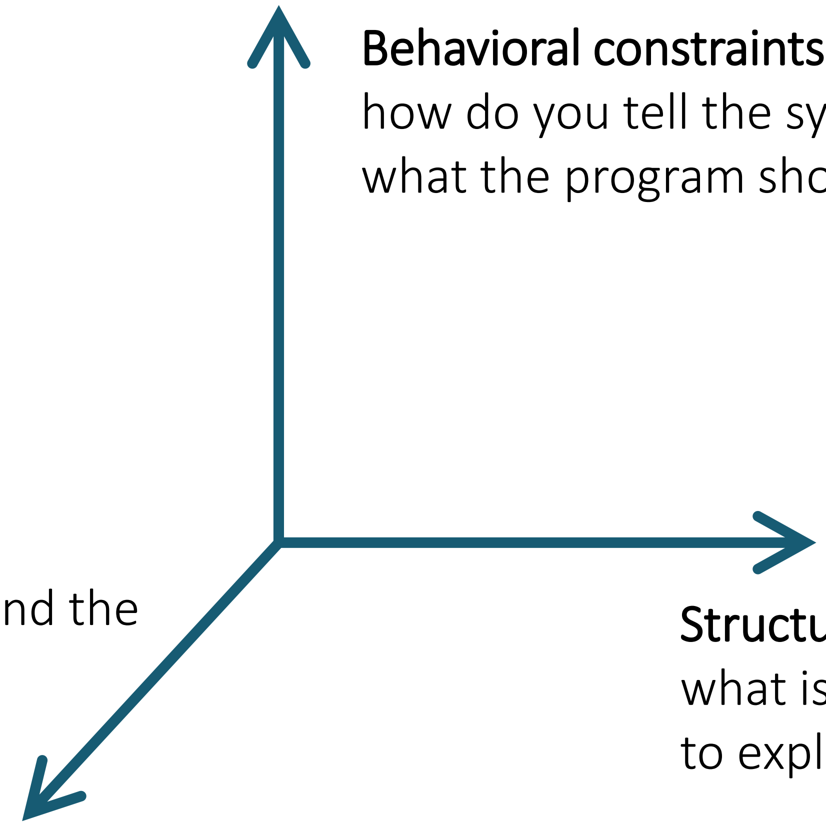
machine code

Synthesis
=
an unusually concise / intuitive programming language
+
a compiler based on search

Dimensions in program synthesis

[Gulwani 2010]

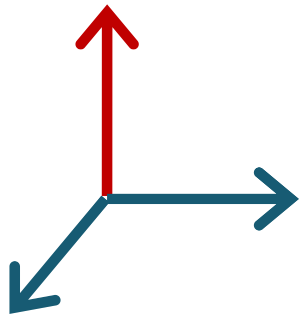
Behavioral constraints:
how do you tell the system
what the program should do?



Search strategy:
How does the system find the
program you want?

Structural constraints:
what is the space of programs
to explore?

Behavioral constraints

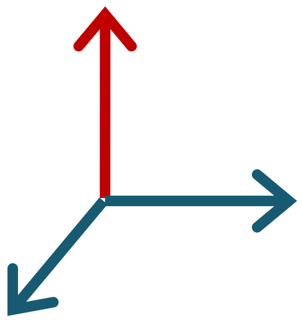


How do you tell the system what the program should do?

- What is the input language / format?
- What is the interaction model?
- What happens when the intent is ambiguous?

Q: What did behavioral constraints look like in FlashFill / Sketch / Synquid / SQLizer?

Behavioral constraints: examples



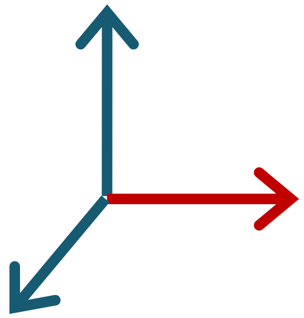
Input/output examples

Equivalent program

Formal specifications (pre/post conditions, types, ...)

Natural language

Structural constraints

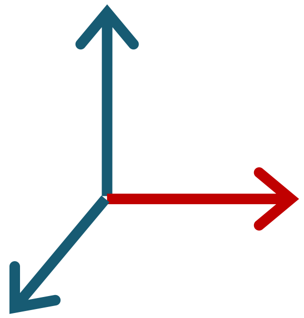


What is the space of programs to explore?

- Large enough to contain interesting programs, yet small enough to exclude garbage and enable efficient search
- Built-in or user defined?
- Can we extract domain knowledge from existing code?

Q: What did structural constraints look like in FlashFill / Sketch / Synquid / SQLizer?

Structural constraints: examples



Built-in DSL

User-defined DSL (grammar)

- + statistical models

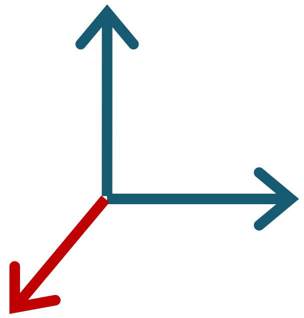
User-provided components

- within straight-line code
- within recursive functional programs

Languages with synthesis constructs

- e.g. generators in Sketch

Search strategies



Synthesis is search:

- Find a program in the space defined by *structural constraints* that satisfies *behavioral constraints*

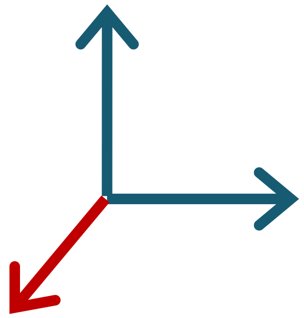
Challenge: the space is astronomically large

- The search algorithm is the heart of a synthesis technique

How does the system find the program you want?

- How does it know it's the program you want?
- How can it leverage structural constraints to guide the search?
- How can it leverage behavioral constraints to guide the search?

Search strategies: examples



Enumerative (explicit) search

- exhaustively enumerate all programs in the language in the order of increasing size

Stochastic search

- random exploration of the search space guided by a fitness function

Representation-based search

- use a data structure to represent a large set of programs

Constraint-based search

- translate to constraints and use a solver

Structure of the Course

Module 1: Synthesis from Examples

Module 2: Synthesis from Specifications

Module 3: Applications of Synthesis

Module 1: Synthesis from examples

Synthesize a program whose behavior satisfies a set of examples

Doesn't machine learning do that?

Traditional Machine Learning

- Learn a function from a set of examples
- Millions of data points
  - Scalability is a challenge
- Data is noisy
  - Need to avoid overfitting, but also approximate solutions are good enough
- Search space is parametrized

Optimization-based search (fast)

Inductive Synthesis

- Learn a function from a set of examples
- Small numbers of examples
  - Ambiguity is a challenge
- Data is clean
  - It's annoying when user says f(x)=5 and the system assumes the user is wrong and decides that f(x)=6
- Search space has complex structure

Combinatorial search (slow)

Module 2: Synthesis from specifications

Sometimes examples are not enough

- inputs and outputs might be large / complex
- a complex problem might need many examples
- providing the output might require knowing the details of the algorithm (think Red-Black Tree insertion)
- we might want correctness guarantees on all inputs

What kinds of other specifications are there?

How do we validate solutions against these specifications?

How do we use automated reasoning to guide the search?

Module 3: Applications of synthesis

Data science
- Data wrangling by example

Security
- Synthesis of cryptographic schemes, synthesis of access control checks

Databases
- Query synthesis, schema matching

Graphics
- “Prodirect” manipulation of vector graphics, lifting kernels

Machine learning
- Concept learning from few examples

Education
- Automatic feedback generation for programming assignments

Weeks 1-2

Topic: Enumerative synthesis form examples

Paper: Alur, Radhakrishna, Udupa. Scaling Enumerative Program Synthesis via Divide and Conquer

- Review due Wednesday
- Link to PDF on the course wiki
- Submit through EasyChair (check email for invite)

Project:

- Teams due next Friday
- Submit through a Google Sheet (check email for invite and instructions)