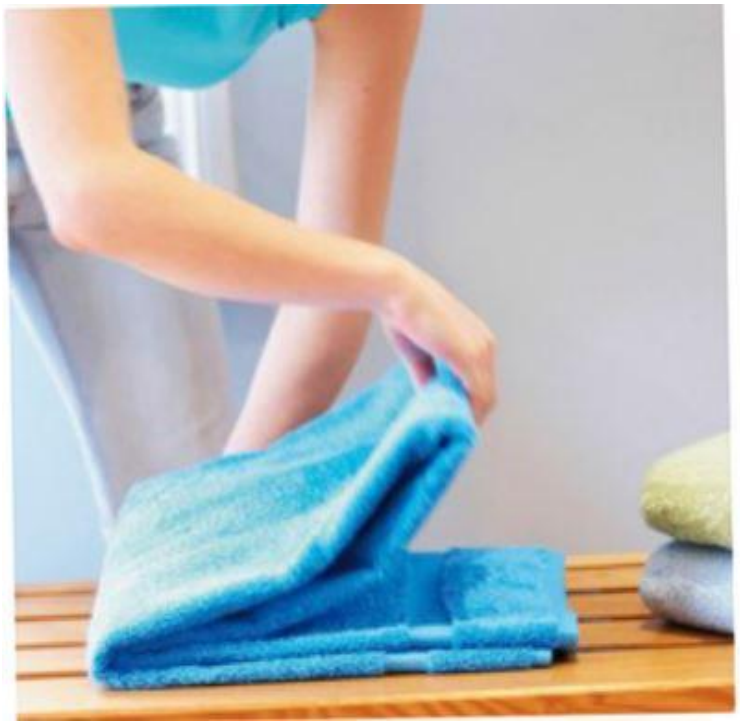
Spreadsheet Programming using Examples



Sumit Gulwani

Microsoft

Keynote at SEMS

July 2016

Motivation

99% of computer users cannot program!

They struggle with simple repetitive tasks.

Programming by examples (PBE) can revolutionize this landscape!

Spreadsheet help forums

Excel Forum
Download Excel forum android application
MS Office Application Help
Forum | What's New? | Members List | Calendar | Forum Rules | Winner Board
Today's Posts FAQ Calendar Community Forum Actions Quick Links

ExcelExperts.com
Excel Consultancy, VBA Consultancy, Training and Tips Call:+442081234832

MrExcel.com
Your One Stop for Excel Tips & Solutions
Forum | Search New Posts | Zero Reply Posts | Subscribed Threads
FAQ Forum Actions Quick Links

Typical help-forum interaction

300_w30_aniSh_c1_b → w30

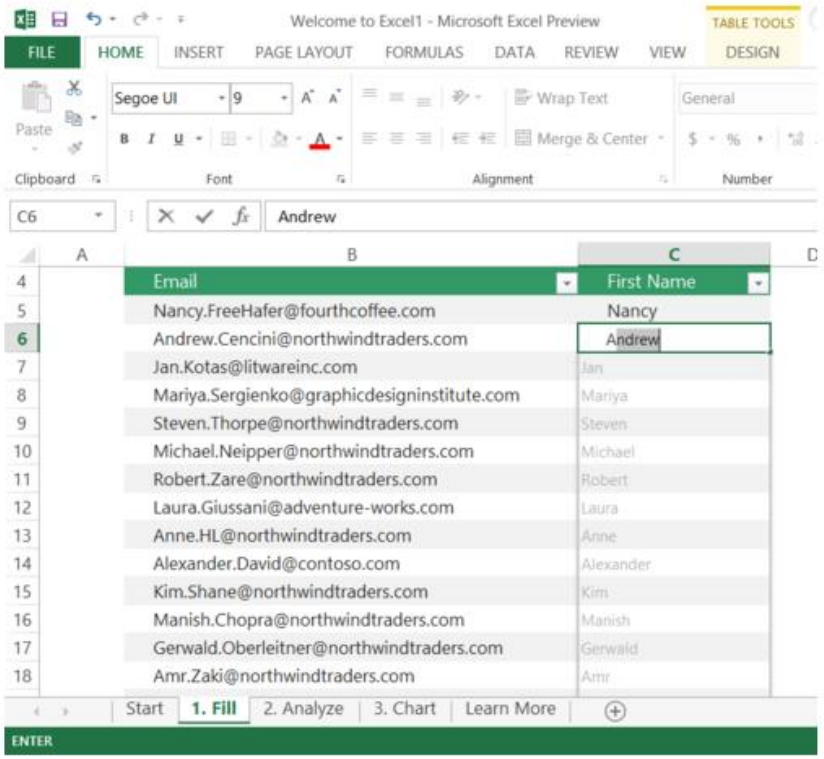
300_w5_aniSh_c1_b → w5

=MID(B1,FIND(“_”,$B:$B)+1,
FIND(“_”,REPLACE($B:$B,1,FIND(“_”,$B:$B), “”))-1)

Flash Fill (Excel 2013 feature) demo

CNN Money

Excel 2013's coolest new feature that should have been available years ago



"Automating string processing in spreadsheets using input-output examples"; POPL 2011; Sumit Gulwani

Number Transformations

Input	Output (Round to 2 decimal places)
123.4567	123.46
123.4	123.40
78.234	78.23

Excel/C#: #.00

Python/C: .2f

Java: #.##

Input	Output (Nearest lower half hour)
0d 5h 26m	5:00
0d 4h 57m	4:30
0d 4h 27m	4:00
0d 3h 57m	3:30

Synthesizing Number Transformations from Input-Output Examples; CAV 2012; Singh, Gulwani

Semantic String Transformations

MarkupRec Table

Id	Name	Markup
S33	Stroller	30%
B56	Bib	45%
D32	Diapers	35%
W98	Wipes	40%
A46	Aspirator	30%

CostRec Table

Id	Date	Price
S33	12/2010	$145.67
S33	11/2010	$142.38
B56	12/2010	$3.56
D32	1/2011	$21.45
W98	4/2009	$5.12

Input v_1	Input v_2	Output (Price + Markup*Price)
Stroller	10/12/2010	$145.67 + 0.30*145.67
Bib	23/12/2010	$3.56 + 0.45*3.56
Diapers	21/1/2011	
Wipes	2/4/2009	
Aspirator	23/2/2010	

Learning Semantic String Transformations from Examples;
VLDB 2012; Singh, Gulwani

Data Science Class Assignment


```
| style="text-align: center;"| {{Sort|01|[[Super Bowl I|I]]}}
|{{Dts|1967|January|15}}
| style="background:#d0e7ff;"|{{Sort|Green Bay Packers 01|[[1966 Green Bay Packers season|Green Bay Packers]]<sup>‡</sup>}}
| style="text-align: center;"| {{Sort|3510|35–10}}
| style="background:#fcc;"|{{Sort|Kansas City Chiefs 01|[[1966 Kansas City Chiefs season|Kansas City Chiefs]]<sup>^</sup>}}
|{{Sort|Los Angeles Memorial Coliseum 01|[[Los Angeles Memorial Coliseum]]}}
|{{Sort|Pasadena, California 01|[[Los Angeles]], [[California]]{{#tag:ref|Both [[Los Angeles, California|Los An
| style="text-align: center;"| {{Sort|061946|61,946}}
| style="text-align: center;"|<ref>{{Cite journal |last=Maule |first=Tex |url=http://sportsillustrated.cnn.com/vau
|-
| style="text-align: center;"| {{Sort|02|[[Super Bowl II|II]]}}
|{{Dts|1968|January|14}}
| style="background:#d0e7ff;"|{{Sort|Green Bay Packers 02|[[1967 Green Bay Packers season|Green Bay P
| style="text-align: center;"| {{Sort|3314|33–14}}
| style="background:#fcc;"|{{Sort|Oakland Raiders 01|[[1967 Oakland Raiders season|Oakland Raiders]]<su
|{{Sort|Orange Bowl 01|[[Miami Orange Bowl|Orange Bowl]]}}
|{{Sort|Miami, Florida 01|[[Miami]], [[Florida]]{{#tag:ref|[[Miami Gardens, Florida|Miami Gardens]] was incor
| style="text-align: center;"| {{Sort|075546|75,546}}
| style="text-align: center;"|<ref>{{Cite journal |url=http://aol.sportingnews.com/nfl/story/2008-01-15/super-b
|-
| style="text-align: center;"| {{Sort|03|[[Super Bowl III|III]]}}<!---During the AFL-NFL merger, As the Colts mo
|{{Dts|1969|January|12}}
| style="bac                                                    Jets season|New York Jets]]<sup>^</s
| style="text
| style="bac                                                    imore Colts season|Baltimore Colts]]<
|{{Sort|Ora
|{{Sort|Mia
| style="tex
| style="tex
<includeon
|-
| style="tex
|{{Dts|1970|January|11}}
| style="background:#fcc;"|{{Sort|Kansas City Chiefs 02|[[1969 Kansas City Chiefs season|Kansas City Chiefs]]<sup>^</sup> (2)}}
| style="text-align: center;"| {{Sort|2307|23–7&amp;nbsp;}}
| style="background:#d0e7ff;"|{{Sort|Minnesota Vikings 01|[[1969 Minnesota Vikings season|Minnesota Vikings]]<sup>‡</sup>}}
|{{Sort|Tulane Stadium 01|[[Tulane Stadium]]}}
|{{Sort|New Orleans, Louisiana|[[New Orleans]], [[Louisiana]]}}
| style="text-align: center;"| {{Sort|080562|80,562}}
| style="text-align: center;"|<ref>{{Cite web |url=http://www.cbsnews.com/htdocs/sports/football/history/superbowl_04.html |title=Super Bowl History: Super E
```

```
I,1967,Green Bay Packers 01,35-10,Kansas City Chiefs 01,Los Angeles Memorial Coliseum
III,1969,New York Jets 01,16-7,Indianapolis Colts 01,Orange Bowl 02 IV,1970,Kansas Ci
V,1971,Indianapolis Colts 02,16-13,Dallas Cowboys 01,Orange Bowl 03 VI,1972,Dallas Co
VII,1973,Miami Dolphins 02,14-7,Washington Redskins 01,Los Angeles Memorial Coliseum 0
IX,1975,Pittsburgh Steelers 01,16-6,Minnesota Vikings 03,Tulane Stadium 03 X,1976,Pit
XI,1977,Oakland Raiders 02,32-14,Minnesota Vikings 04,Rose Bowl 01 XII,1978,Dallas Co
XIII,1979,Pittsburgh Steelers 03,35-31,Dallas Cowboys 05,Orange Bowl 05 XIV,1980,Pitt
981,Oakland Raiders 03,27-10,Philadelphia Eagles 01,Louisiana Superdome 02 XVI,19
VII,1983,Washington Redskins 02,27-17,Miami Dolphins 04,Rose Bowl 03 XVIII,1984,Oakl
XIX,1985,San Francisco 49ers 02,38-16,Miami Dolphins 05,Stanford Stadium 01 XX,1986,C
XXI,1987,New York Giants 01,39-20,Denver Broncos 02,Rose Bowl 04 XXII,1988,Washington
XXIII,1989,San Francisco 49ers 03,20-16,Cincinnati Bengals 02,Joe Robbie Stadium 01 X
```

To get Started!

```
cat superbowl.txt | awk '$1=$1' ORS=' ' | sed 's/|- |/\n|/g' | grep "^| style=\"t
ext-align: center;\"" | grep -v "Championship"
```

FlashExtract Demo

Ships inside two Microsoft products:

Powershell 

ConvertFrom-String cmdlet

Custom Log,
Custom Field

"*FlashExtract: A Framework for data extraction by examples*";
PLDI 2014; Vu Le, Sumit Gulwani

Layout Transformations

Input Table

Bureau of I.A.	
Regional Director	Numbers
Niles C.	Tel: (800)645-8397
	Fax: (907)586-7252
Jean H.	Tel: (918)781-4600
	Fax: (918)781-4604
Frank K.	Tel: (615)564-6500
	Fax: (615)564-6701

Output Table

	Tel	Fax
Niles C.	(800)645-8397	(907)586-7252
Jean H.	(918)781-4600	(918)781-4604
Frank K.	(615)564-6500	(615)564-6701

	A	B	C	D	E
1		value	year	value	year
2	Albania	1000	1950	930	1981
3	Austria	3139	1951	3177	1955
4	Belgium	541	1947	601	1950
5	Bulgaria	2964	1947	1959	1958
6	Czech . . .	2416	1950	2503	1960

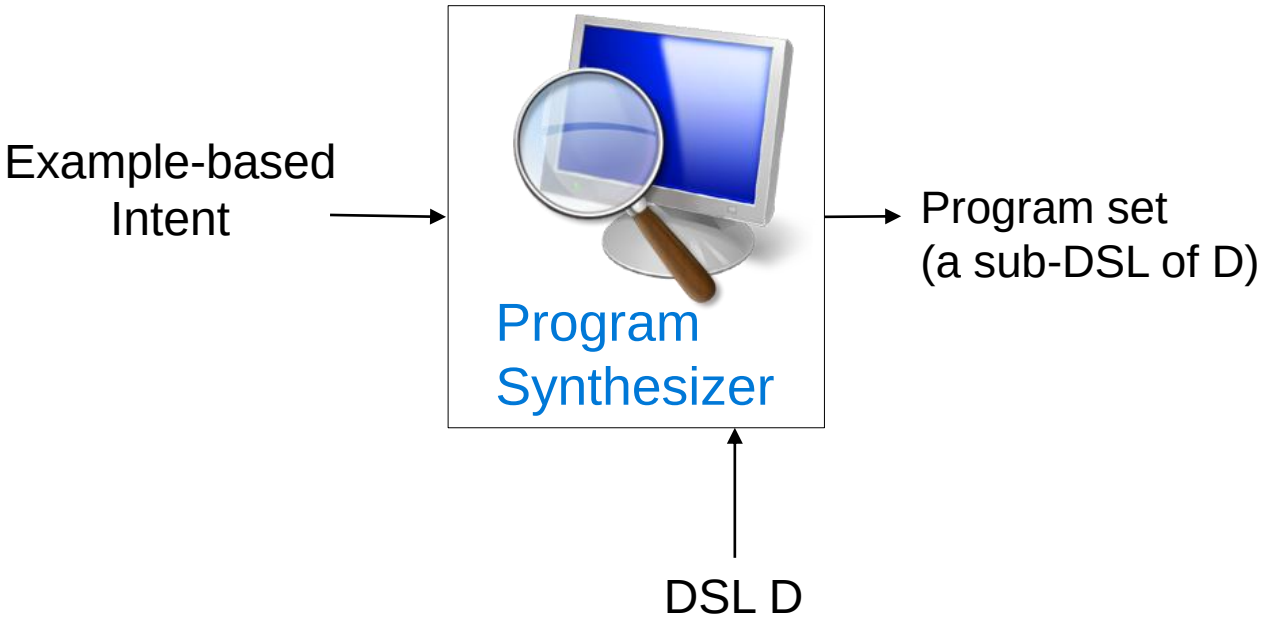
. . .

Country	Harvest	Date
Albania	1000	1950
Albania	930	1981
. . .		
Austria	3139	1951
Austria	3177	1955
. . .		
Belgium	541	1947
Belgium	601	1950
. . .		

PBE allows creation of output table from couple of example tuples.

Flashrelate: extracting relational data from semi-structured spreadsheets using examples; PLDI 2014; Barowy, Gulwani, Hart, Zorn

Programming-by-Examples Architecture



Domain-specific Language (DSL)

- Balanced Expressiveness
  - Expressive enough to cover wide range of tasks
  - Restricted enough to enable efficient search
    - Restricted set of operators
      - those with small inverse sets
    - Restricted syntactic composition of those operators

- Natural computation patterns
  - Increased user understanding/confidence
  - Enables selection between programs, editing

Flash Fill DSL (String Transformations)

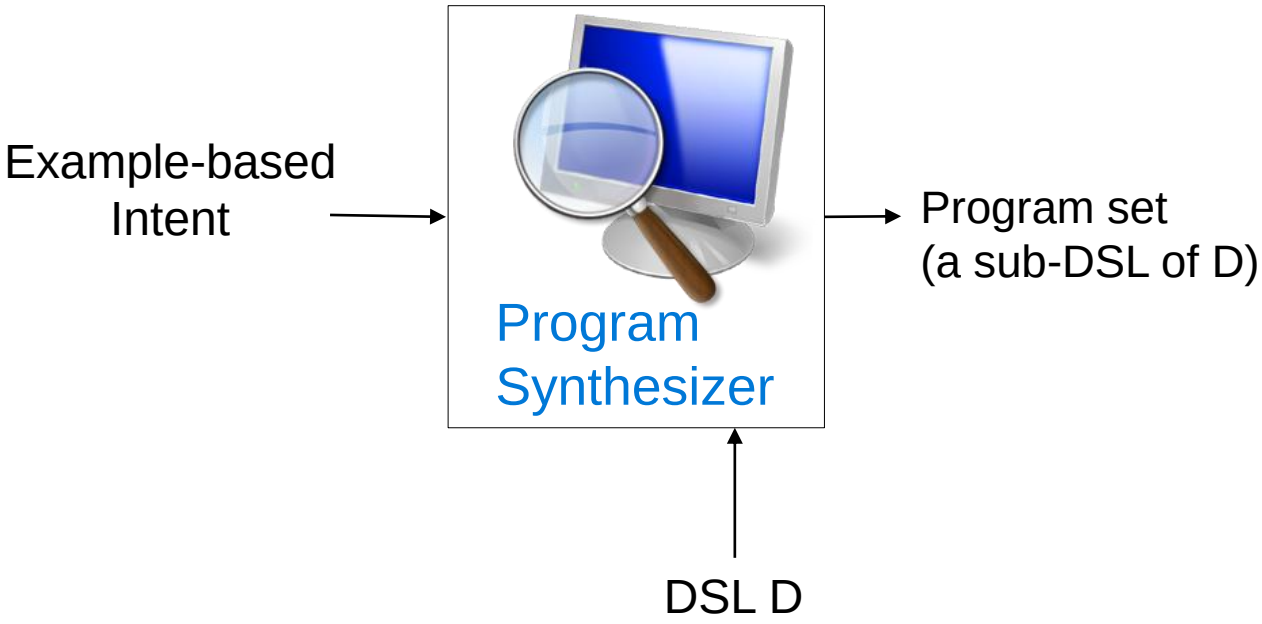
$$Tuple(String\ x_1, \ldots, String\ x_n) \rightarrow String$$

```
top-level expr T := if-then-else(B,C,T)
                  |  C

condition-free expr C := Concatenate(A,C)
                       |  A

atomic expression A := SubStr(X,P,P)
                     | ConstantString

input string X := x1 | x2 | …

position expression P := K
                       | Pos(X, R1, R2, K)
```

K^{th} position in X whose left/right side matches with R_1/R_2.

```
Boolean expression B := …
```

Programming-by-Examples Architecture



Search Methodology

Goal: Set of program expr of kind e that satisfies spec ϕ

[denoted $[e \vDash \phi]$]

e: DSL (top-level) expression

ϕ: Conjunction of (input state σ ⇝ output value v)

Methodology: Based on divide-and-conquer style problem decomposition.

- $[e \vDash \phi]$ is reduced to simpler problems (over sub-expressions of e or sub-constraints of ϕ).
- Top-down (as opposed to bottom-up enumerative search).

"FlashMeta: A Framework for Inductive Program Synthesis";
OOPSLA 2015; Alex Polozov, Sumit Gulwani

Problem Reduction Rules

$$[e \vDash \phi_1 \wedge \phi_2] = Intersect([e \vDash \phi_1], [e \vDash \phi_2])$$

An alternative strategy:

$$[e \vDash \phi_1 \wedge \phi_2] = Filter([e \vDash \phi_1], \phi_2)$$

Let e be a non-terminal defined as $e ::= e_1 \mid e_2$

$$[e \vDash \phi] = Union([e_1 \vDash \phi], [e_2 \vDash \phi])$$

Problem Reduction Rules

$$[F(e_1, \ldots, e_n) \vDash \sigma \rightsquigarrow v] =$$
$$Union(\{\mathrm{F}([e_1 \vDash \sigma \rightsquigarrow u_1], \ldots, [e_n \vDash \sigma \rightsquigarrow u_n]) \mid (u_1, \ldots, u_n) \in F^{-1}(v)\})$$

$F(S_1, \ldots, S_n)$ denotes $\{F(e_1, \ldots, e_n) \mid e_1 \in S_1, \ldots, e_n \in S_n\}$

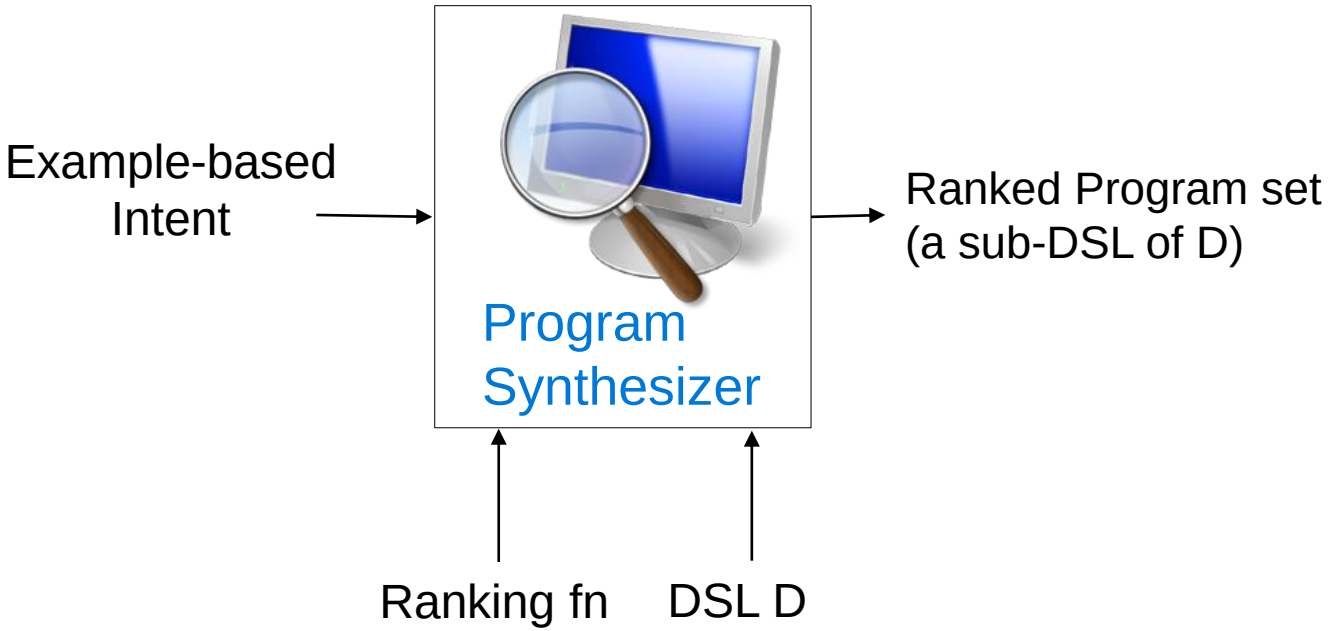
Inverse Set: Let F be an n-ary operator.

$$F^{-1}(v) = \{(u_1, \ldots, u_n) \mid F(u_1, \ldots, u_n) = v\}$$

$$Concat^{-1}(\text{"Abc"}) = \{(\text{"Abc"}, \epsilon), (\text{"}Ab\text{"}, \text{"c"}), (\text{"A"}, \text{"bc"}), (\epsilon, \text{"Abc"})\}$$

$$[Concat(X, Y) \vDash (\sigma \rightsquigarrow \text{"Abc"})] = \mathrm{Union}(\{$$
$$Concat([X \vDash (\sigma \rightsquigarrow \text{"Abc"})], [Y \vDash (\sigma \rightsquigarrow \epsilon)]),$$
$$Concat([X \vDash (\sigma \rightsquigarrow \text{"Ab"})], [Y \vDash (\sigma \rightsquigarrow \text{"}c\text{"})]),$$
$$Concat([X \vDash (\sigma \rightsquigarrow \text{"A"})], [Y \vDash (\sigma \rightsquigarrow \text{"}bc\text{"})]),$$
$$Concat([X \vDash (\sigma \rightsquigarrow \epsilon)], [Y \vDash (\sigma \rightsquigarrow \text{"}Abc\text{"})])\})$$

Programming-by-Examples Architecture



Ranking scheme: Program features

Prefer simpler programs

- Fewer constants.
- Smaller constants.

Input	Output
Rishabh Singh	Rishabh
Ben Zorn	Ben

- 1st Word
- If (input = “Rishabh Singh”) then “Rishabh” else “Ben”
- “Rishabh”

“*Predicting a correct program in Programming by Example*”;
[CAV 2015] Rishabh Singh, Sumit Gulwani

Ranking scheme: Data features

Prefer simpler programs

- Fewer constants.
- Smaller constants.

Input	Output
Missing page numbers, 1993	1993
64-67, 1995	1995

- 1st Number from the beginning
- 1st Number from the end

How to select between programs with same number of same-sized constants?

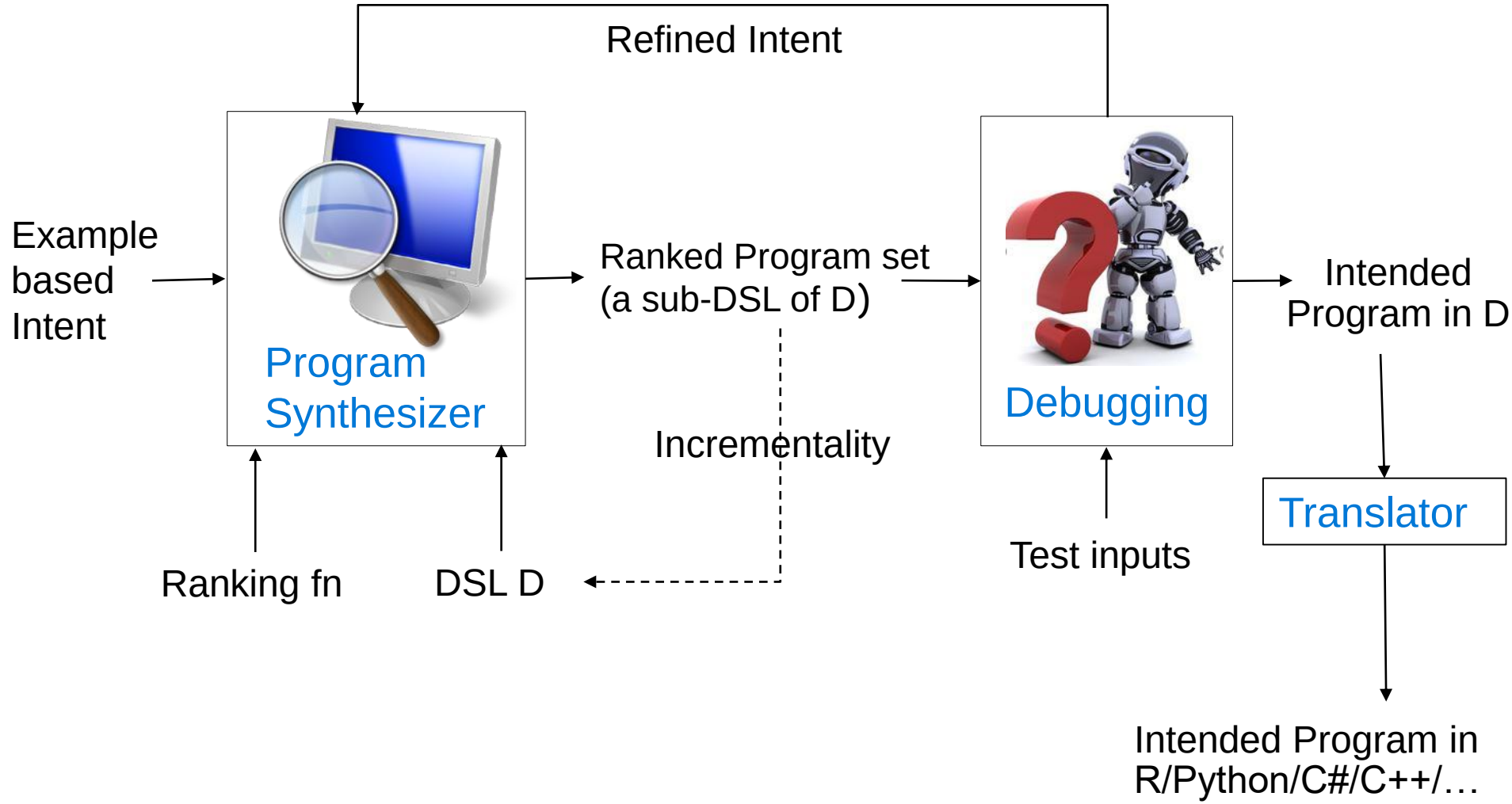
Prefer programs that generate more uniform output.

Outline

- Core Synthesis Architecture
  - Domain-specific Language
  - Search methodology
  - Ranking function

➢ Next generation Synthesis
  - Interactive
  - Predictive
  - Adaptive

Programming-by-Examples Architecture



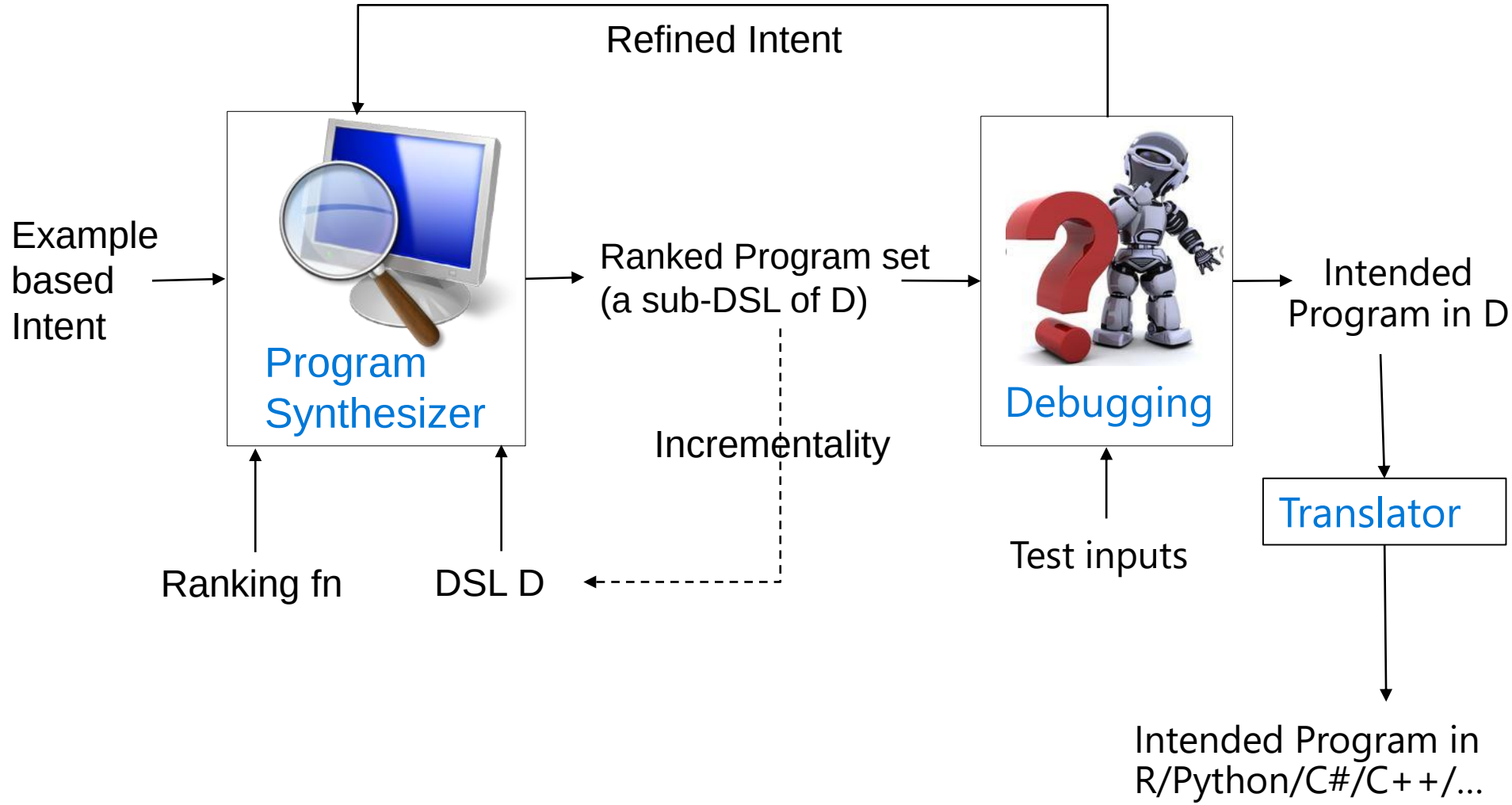
Interactive Debugging

- Sampling inputs
- Asking questions
- Visual explanations of the synthesized program

Predictive

- Intended programs can sometimes be synthesized from just the input.
  - Tabular data extraction, Sort, Join
- Can save large amount of user effort.
  - User need not provide examples for each of tens of columns.

Programming-by-Examples Architecture

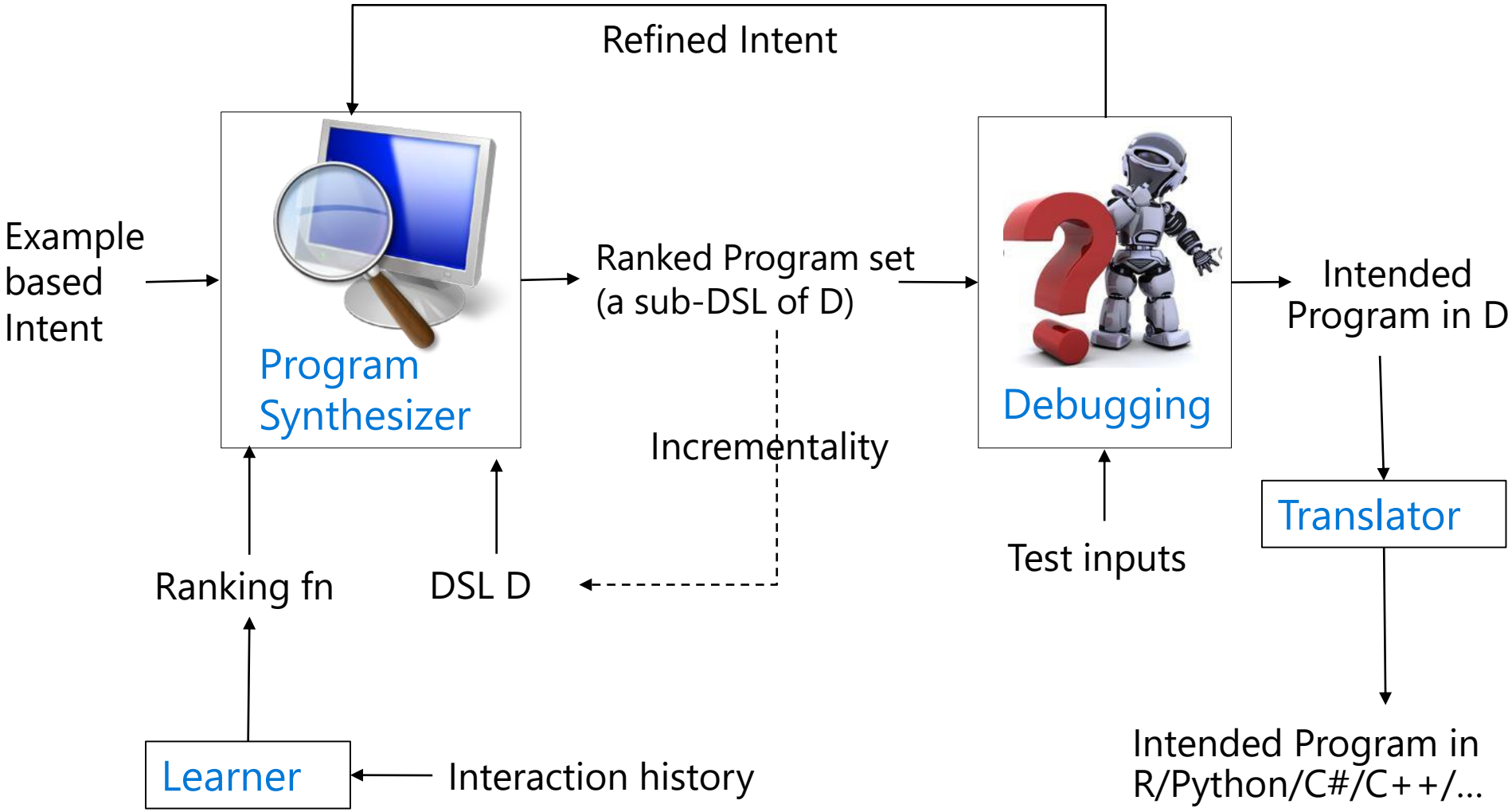


Adaptive

- Learn from past interactions
  - of the same user (personalized experience).
  - of other users in the enterprise/cloud.

- The synthesis sessions now require less interaction.

Programming-by-Examples Architecture



PROSE Framework

https://microsoft.github.io/prose

- Efficient implementation of the generic search methodology.
- Provides a library of reduction rules.

Role of synthesis designer

- Implement a DSL and provide reduction rules for new operators.
- Provide ranking strategy.
- Can specify tactics to resolve non-determinism in search.

The PROSE Team

Allen Cypher

Sumit Gulwani

Ranvijay Kumar

Vu Le

Daniel Perelman

We are hiring interns/full-time!

Alex Polozov

Mohammad Raza

Danny Simmons

Adam Smith

Abhishek Udupa

Future Directions

- Learn from usage data
- Probabilistic noise handling
- Programming using natural language
- Application to robotics

Conclusion

- PBE can enable easier & faster data wrangling.
  - 99% of computer users are non-programmers.
  - Data scientists spend 80% time cleaning data.

- Algorithmic search
  - Domain-specific language
  - Deductive methodology based on back-propagation
- Ambiguity resolution
  - Ranking
  - Interactivity

Reference: "*Programming by Examples (and its applications in Data Wrangling)*", In *Verification and Synthesis of Correct and Secure Systems*; IOS Press; 2016
[based on Marktoberdorf Summer School 2015 Lecture Notes]