

Automating string processing in spreadsheets using input-output examples

Junliang Yan

Spreadsheets

	A	B	C
1	Name	City	Phone
2	A	Nanjing	12345678910
3	B	Yangzhou	10987654321
4	C	Zhenjiang	11111111111
5	D	Suzhou	22222222222
6	E	Changzhou	33333333333
7	F	Wuxi	66666666666
8	G	Nantong	99999999999

Collect cities of residence

Name	Adress	City	
A	Jiangsu, nanjing, nju		
B	Hubei, wuhan, hust		
C	Shandong, jinan, sdu		
D	Hunan, changsha, csu		

Copy-Paste one by one?

Collect cities of residence

Name	Adress	City	
A	Jiangsu, nanjing, nju		
B	Hubei, wuhan, hust		
C	Shandong, jinan, sdu		
D	Hunan, changsha, csu		

✗ Copy-Paste one by one?

✓ Flashfill using input-output example!

Flashfill using input-output examples

Flashfill using input-output examples

- String Manipulation Language

```
SubStr2(  
    Input(2),  
    TokenSeq(AlphaToken),  
    2  
)
```

Program

Flashfill using input-output examples

- String Manipulation Language
- **Synthesize** a program with **input-output examples**

1	Input(2)	Output
A	Jiangsu, nanjing, nju	nanjing

Input-output examples

Synthesize
➔

```
SubStr2(  
    Input(2),  
    TokenSeq(AlphaToken),  
    2  
)
```

Program

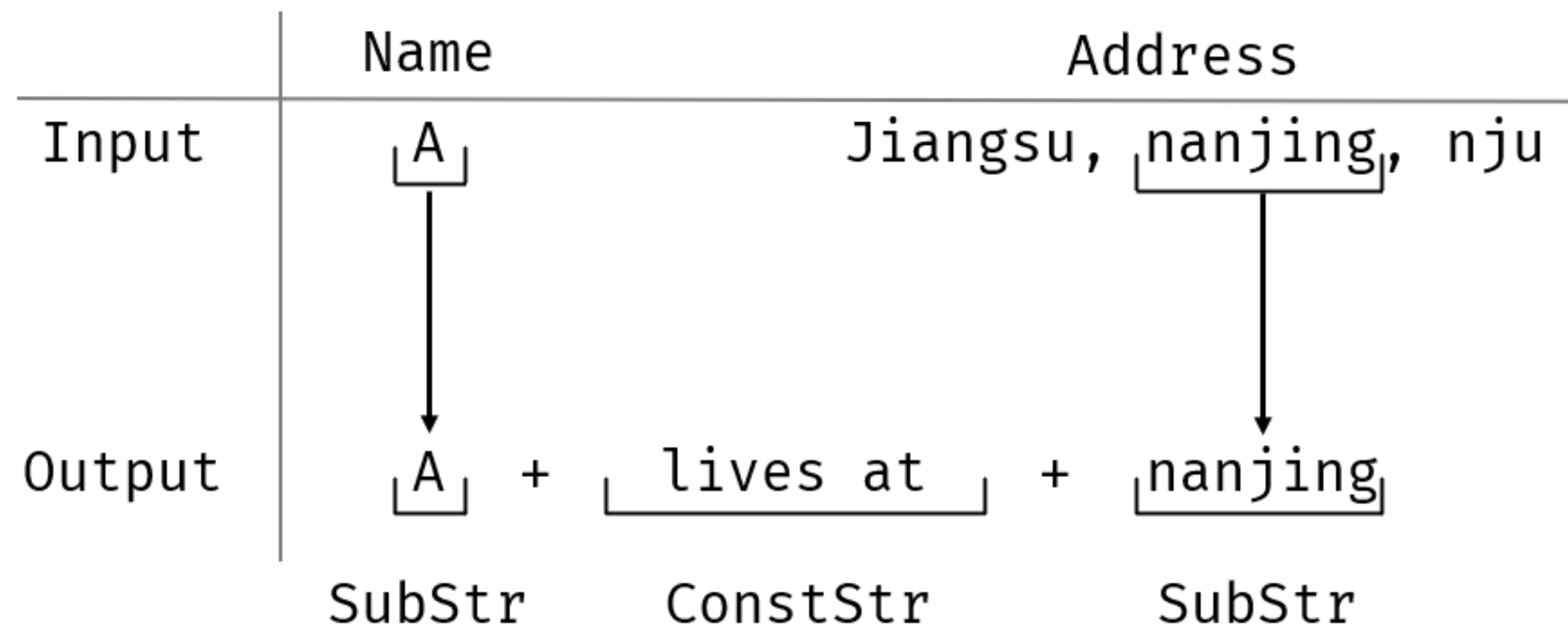
Constructing Output Strings Language

Trace

Trace Expr is a concatenation of the substrings of inputs or a constant string.

Trace

Trace Expr is a concatenation of the substrings of inputs or a constant string.



Output: A lives at nanjing

SubString

SubStr (,)

SubString

SubStr (Input, ,)

Input : **Index**, which input string is used.

SubString

SubStr(Input, Left, Right)

Input : **Index**, which input string is used.

Left, Right: **Position Expressions**, the range of substring.

Name	Address	City
A	Jiangsu, nanjing , nju	nanjing

SubStr(, ,)

SubString

SubStr(Input, Left, Right)

Input : **Index**, which input string is used.

Left, Right: **Position Expressions**, the range of substring.

Name	Address	City
A	Jiangsu, nanjing , nju	nanjing

SubStr(Address, ,)

SubString

`SubStr(Input, Left, Right)`

`Input` : **Index**, which input string is used.

`Left, Right`: **Position Expressions**, the range of substring.

Name	Address	City
A	Jiangsu, nanjing , nju	nanjing

`SubStr(Address, CPos(9), CPos(-6))`

SubString

`SubStr(Input, Left, Right)`

`Input` : **Index**, which input string is used.

`Left, Right`: **Position Expressions**, the range of substring.

Name	Address	City
A	Jiangsu, nanjing , nju	nanjing

✗ `SubStr(Address, CPos(9), CPos(-6))`

✓ `SubStr(Address, Pos(ϵ , RE, 2), Pos(RE, ϵ , 2))`

where RE = LowercaseTokens

Regular Expressions

Only use a **small subset** of regular expressions.

Regular Expression is **a sequence of tokens**.

Regular Expressions

Only use a **small subset** of regular expressions.

Regular Expression is **a sequence of tokens**.

```
R = TokenSequence(LowercaseTokens, NumericTokens)
```

```
R = [a-z]+ [0-9]+
```

Regular Expressions

Only use a **small subset** of regular expressions.

Regular Expression is **a sequence of tokens**.

```
R = TokenSequence(LowercaseTokens, NumericTokens)
```

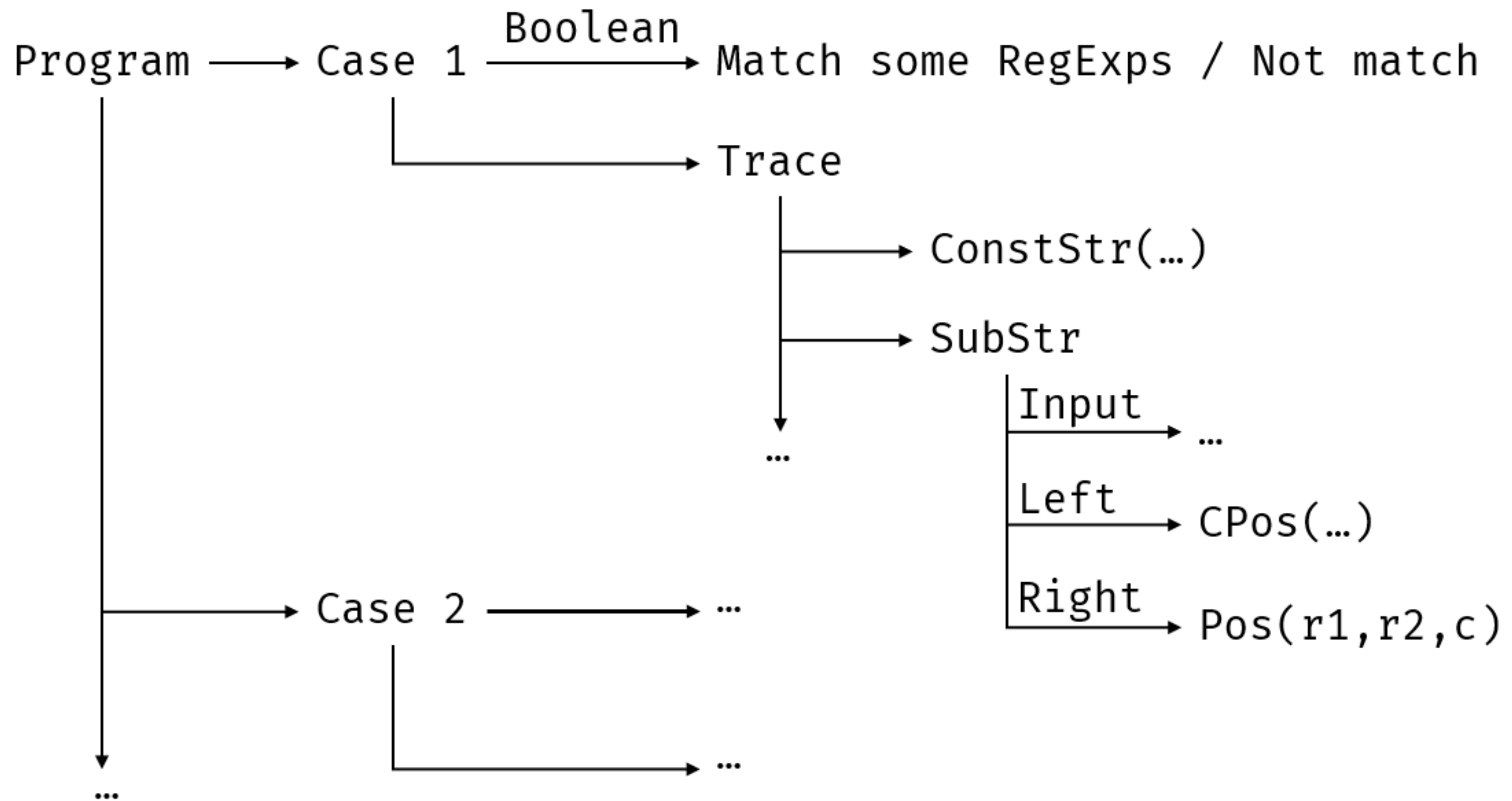
```
R = [a-z]+ [0-9]+
```

No kleen star ($[a-z]^*$).


No disjunct operation ($[a-z] \mid [0-9]$).

➔ **Efficient Algorithm**

Conditionals



Example

Num	Rev	Code 
123	321	case 123: return 321
456	654	case 456: return 654
789	987	
147	741	
258	852	
369	963	

Alogrithm

END

Q & A