

A Take of

Synthesis

DiverSe Coffee

Théo MATRICON

Hobbies

Climbing



geopolitics



philosophy
(societal)

Feminism ↓
 education

reserved

~~gluten~~ ~~Packose~~
kind of "kills me"
(I have Cohn's disease)

- Past year: financial officer of the PhD association
- actively participating in psychosocial risks working group (RPS)
- RPS referent for PhD students

LaBRI

3rd year PhD
with



Bordeaux



Nathanaël Fijalkow

1 How to synthesize a program matching executable tests?

tests

$$\begin{cases} \text{foo } [1, 3, 2] = [1, 3, 3] \\ \text{foo } [4, 2, 3] = [3, 3, 4] \\ \text{foo } [7, 15, 8, 3, 10] = [4, 7, 9, 11, 15] \end{cases}$$

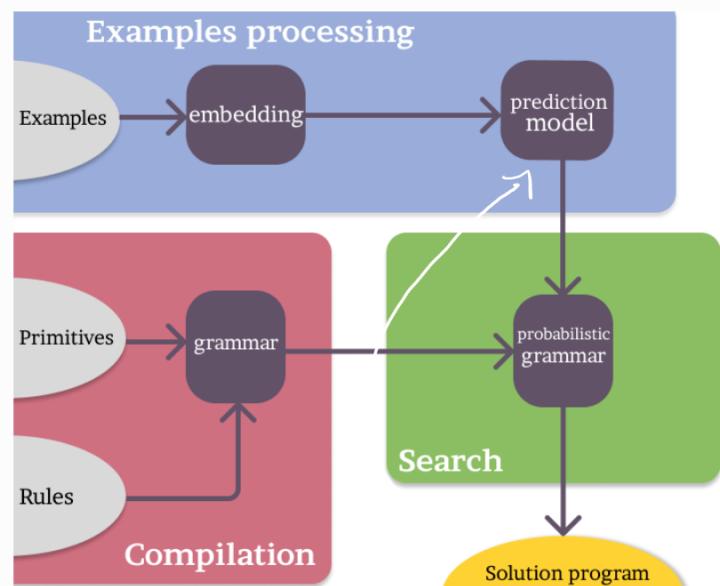
DSL
+
syntax
semantic

$$\begin{cases} \text{map : } ('a \rightarrow 'b) \rightarrow 'a \text{ list} \rightarrow 'b \text{ list} \\ \text{len : } 'a \text{ list} \rightarrow \text{int} \\ \text{filter : } ('a \rightarrow \text{bool}) \rightarrow 'a \text{ list} \rightarrow 'a \text{ list} \\ \vdots \end{cases}$$

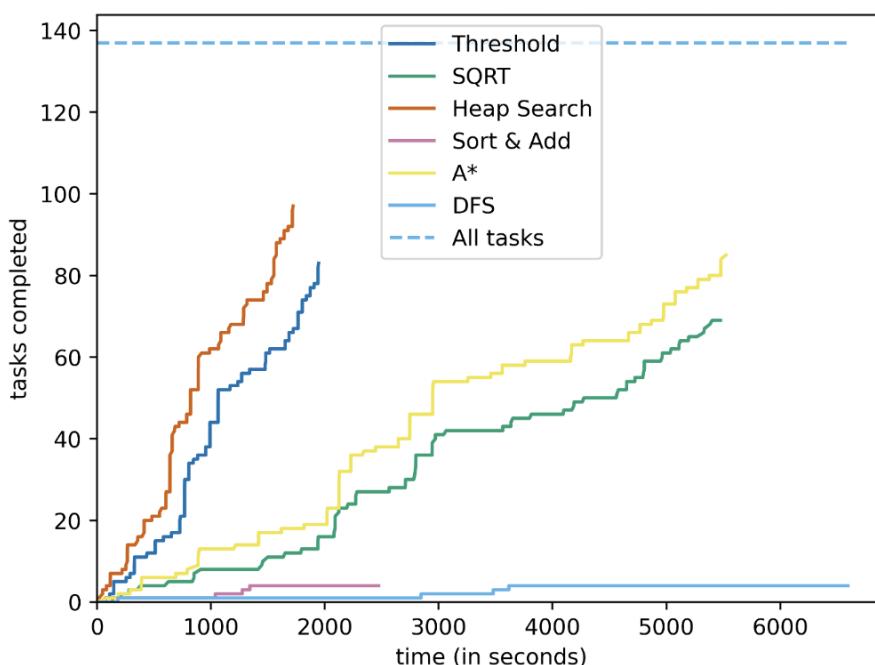
```
foo P = (sort (map
                  λx. lite (isEven x) (+ x 1) x)
                  P
                ))
```

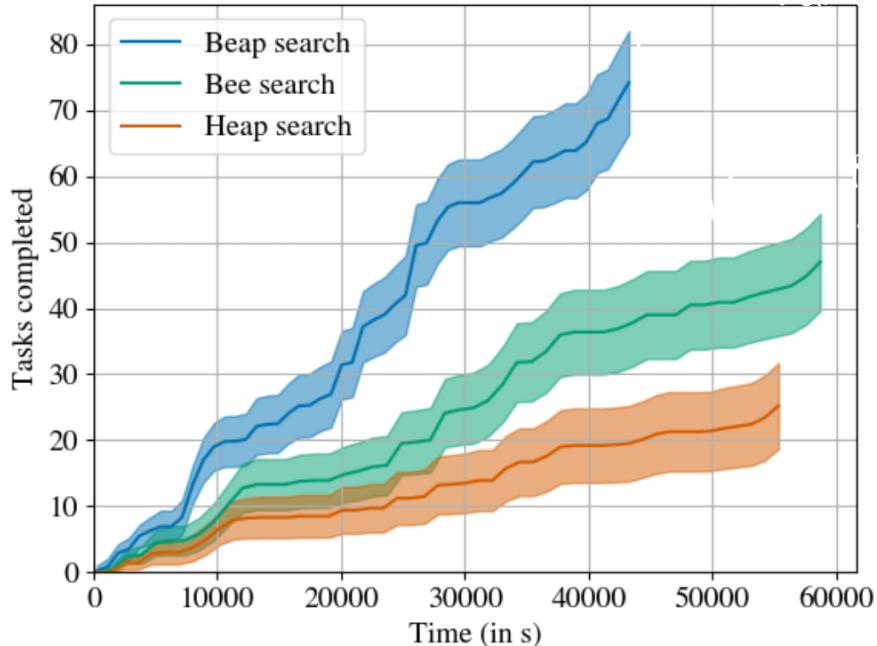
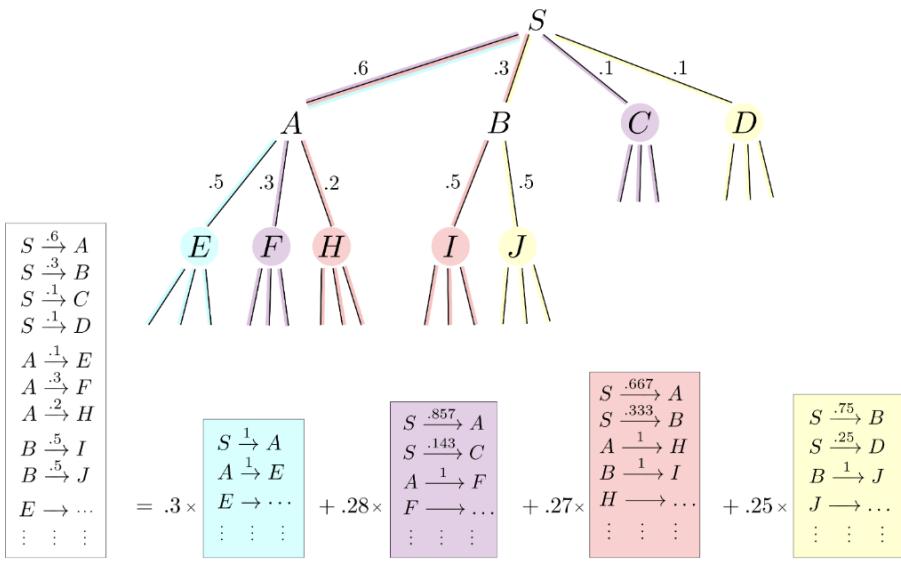


- DSL agnostic
- Prog and play



1.A. Can we have better enumerative algorithms?





1. B. Can we Learn from the programs we tested so far?

1. C. Can we Pervage DSL specific knowledge to reduce the search space?

Observe $\text{ite } C \text{ A B} \equiv \text{ite } (\text{not } C) \text{ B A}$

enumerating both is useless!

remove one from the search space

Can we find these equations automatically?

Yes everything is executable the only constraints are:

time and being able to generate inputs

2. Can we synthesize a program that talks about Knowledge?

Summit Attendees

	A	B	C	D	E	F
1	Attendees	Office Address	Hi _____	E-mail	Zip Code	
2	Gerald Parker	320 N Morgan St #600, Chicago, IL 60607	Hi Gerald			
3	Phillip Davis	111 8th Ave. New York, NY 10011				
4	Ralph Gonzales	2930 Pearl Street, Boulder, CO. 80301				
5	Shawn Jenkins	1160 Bordeaux Drive, Sunnyvale, CA 94089				
6	Thomas Murphy	1160 Bordeaux Drive, Sunnyvale, CA 94089				
7	Norma Long	2930 Pearl Street, Boulder, CO. 80301				
8	Beverly Harris	320 N Morgan St #600, Chicago, IL 60607				
9	Melissa Torres	1160 Bordeaux Drive, Sunnyvale, CA 94089				
10	Joshua Watson	111 8th Ave. New York, NY 10011				
11	Larry Wright	500 W 2nd St. Austin, TX 78701				
12	Lois Brooks	2930 Pearl Street, Boulder, CO. 80301				
13	Steve James	1160 Bordeaux Drive, Sunnyvale, CA 94089				
14	Helen Rivera	2930 Pearl Street, Boulder, CO. 80301				
15	Samuel Bailey	111 8th Ave. New York, NY 10011				
16	Kenneth Gonzalez	2930 Pearl Street, Boulder, CO. 80301				
17	Sharon Washington	1160 Bordeaux Drive, Sunnyvale, CA 94089				
18	Mildred Evans	1160 Bordeaux Drive, Sunnyvale, CA 94089				
19						
20						
21						
??						

City (input)

Bordeaux

Kraków

Dagstuhl

Buenos Aires

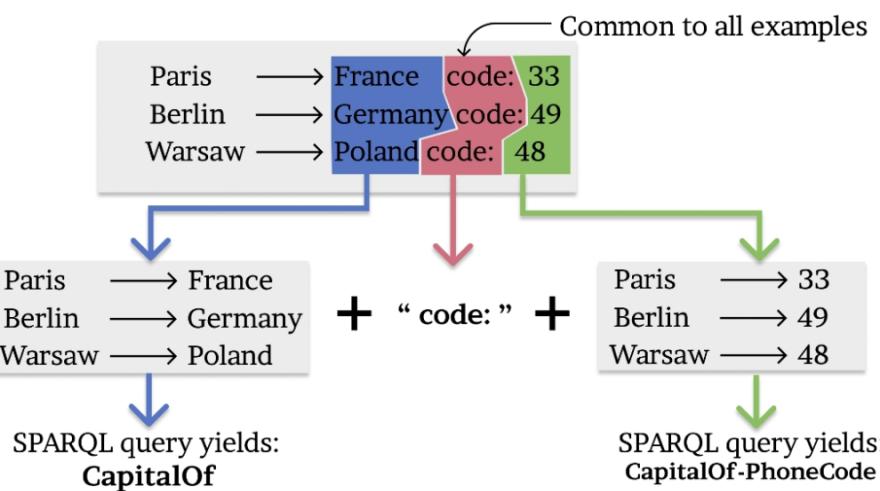
Phone Code (output)

3 3

4 8

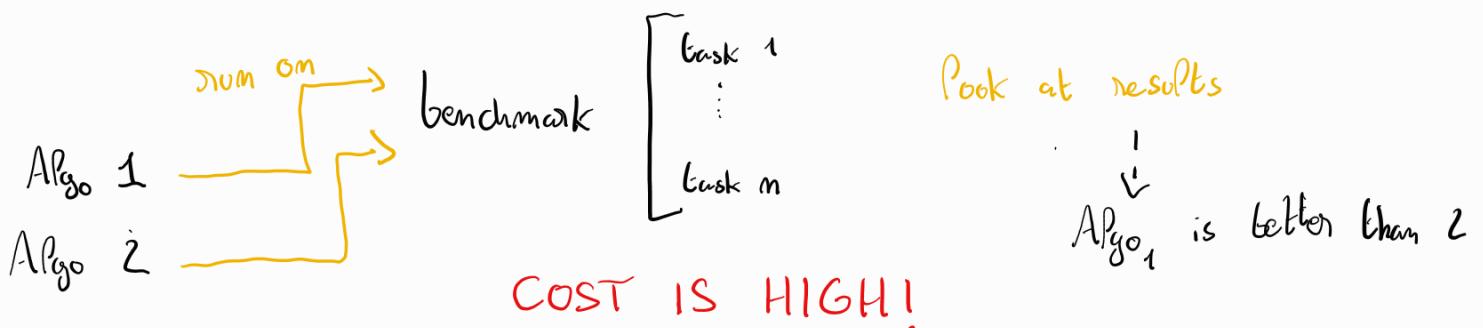
4 9

5 4 ←



3. How to compare algorithms effectively?

pipeline:



IDEA:

while not confident:

$t = \text{select_task}()$

$\text{run_algos_on_task}([A_1, A_2], t)$

$\text{confident} = \text{update_confidence}()$

RESULTS

if a lot of tasks are correlated: 100% acc. / 5% time

if not:

- if variants are "far enough": ~ 95% acc. / 10-20% time

- if not: ~ 95% acc. / 95% time

3.A. Can we improve on these results?

3.B Can we use this to determine the relevance of benchmarks?

3.C Can we extend to any number of algorithms?

- Can we solve reinforcement learning problems by synthesizing programs ?
- Can we have faster synthesis of formulas on the GPU:
 - Mixed Boolean Arithmetic
 - Boolean Formula
- Can we generate bottom-up code with LLMs to give execution feedback ?

Conclusion

- I like
- making tools that can be used by others.
 - making new/better algorithms

