

A Tale of

Synthesis

Diverse Coffee

Théo MATRICON

Hobbies

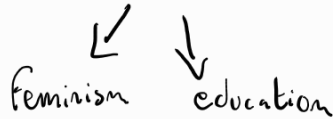
climbing



geopolitics



philosophy
(societal)



reserved

~~gluten~~ ~~lactose~~ kind of "kills me"
(I have Crohn'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 Fijałkow

1 How to synthesize a program matching executable tests?

tests

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

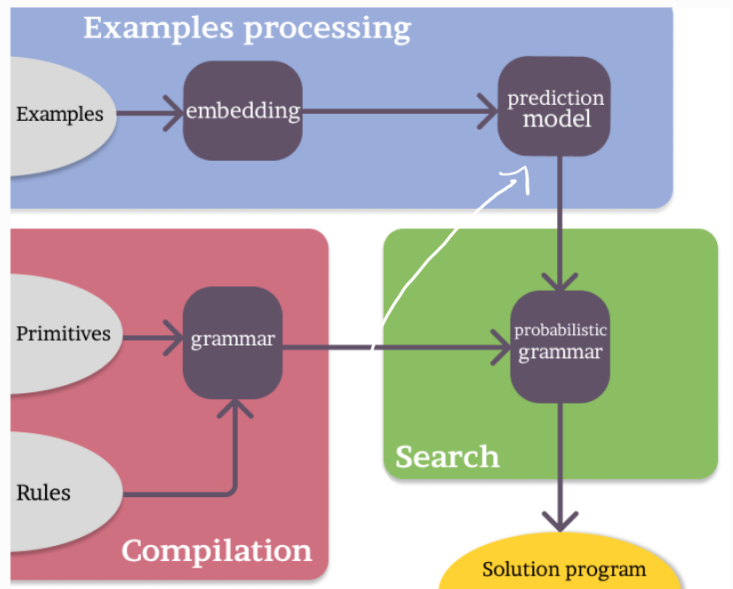
DSL
syntax + semantic

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

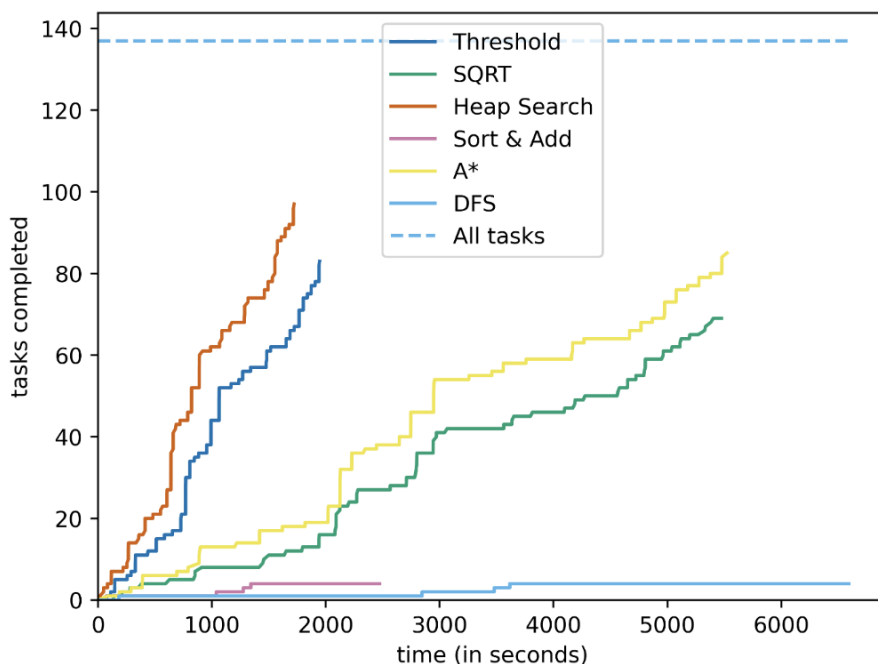
foo P = (sort (map
 $\lambda x. \text{ite}(\text{isEven } x) (+ x 1) x)$
P
))



- DSL agnostic
- Plug and play



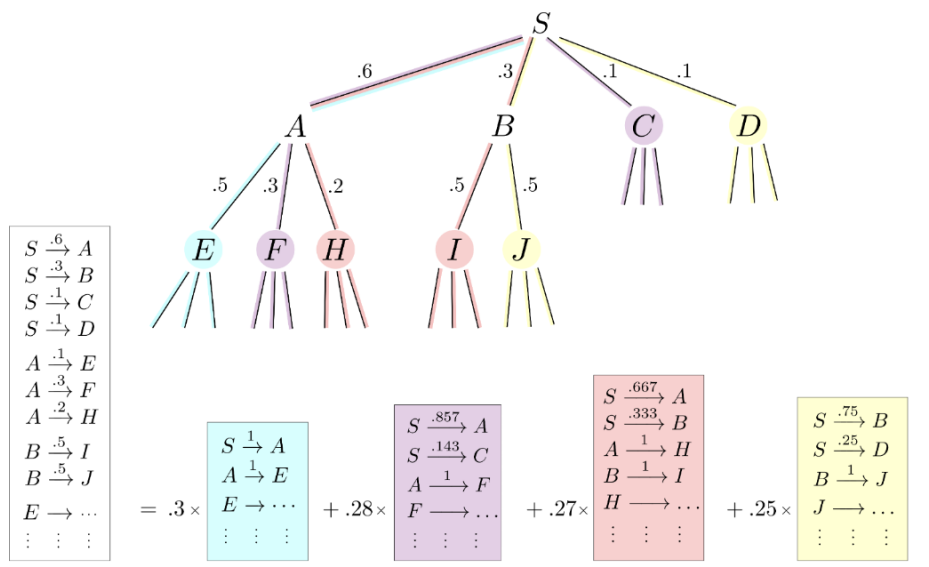
1.A. Can we have better enumerative algorithms?



SPLITTING

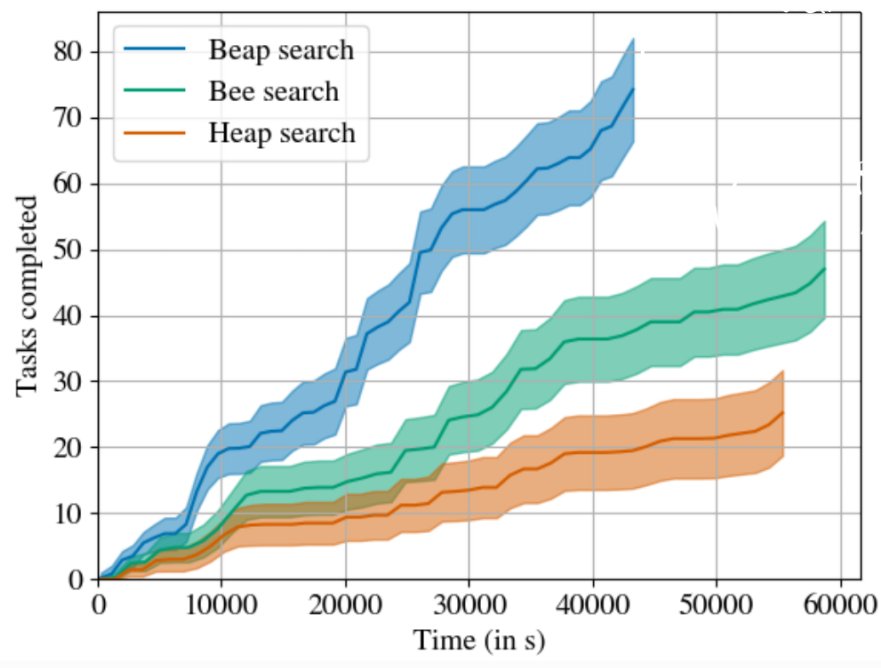


PARALLELISATION



EXCLUSIVE

PREVIEW



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

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

Observe $ite\ C\ A\ B \equiv ite\ (not\ C)\ B\ A$

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?

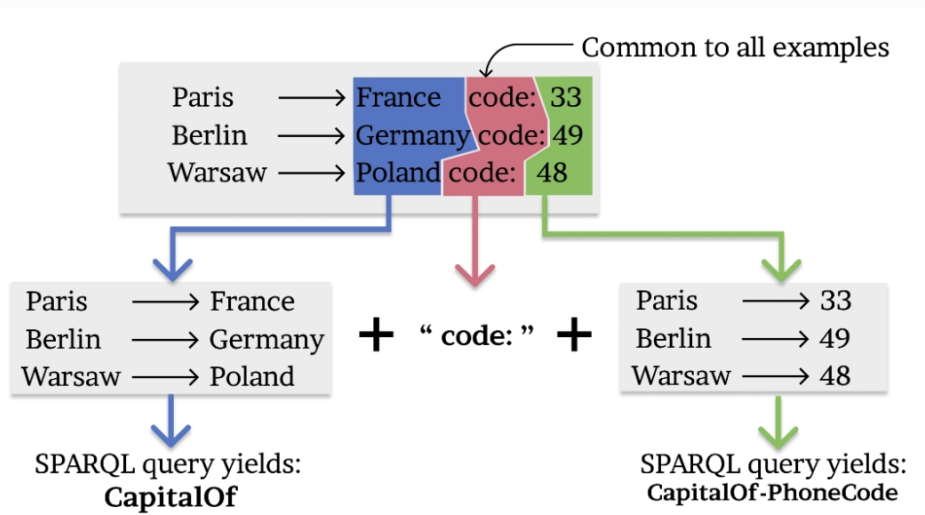
Summit Attendees ☆ ■

File Edit View Insert Format Data Tools Add-ons Help

100% % \$ 123 Arial 8 B I A

Attendees	Office Address	Hi _____	E-mail	Zip Code
Gerald Parker	320 N Morgan St #600, Chicago, IL 60607	Hi Gerald		
Phillip Davis	111 8th Ave. New York, NY 10011			
Ralph Gonzales	2930 Pearl Street, Boulder, CO. 80301			
Shawn Jenkins	1160 Bordeaux Drive, Sunnyvale, CA 94089			
Thomas Murphy	1160 Bordeaux Drive, Sunnyvale, CA 94089			
Norma Long	2930 Pearl Street, Boulder, CO. 80301			
Beverly Harris	320 N Morgan St #600, Chicago, IL 60607			
Melissa Torres	1160 Bordeaux Drive, Sunnyvale, CA 94089			
Joshua Watson	111 8th Ave. New York, NY 10011			
Larry Wright	500 W 2nd St. Austin, TX 78701			
Lois Brooks	2930 Pearl Street, Boulder, CO. 80301			
Steve James	1160 Bordeaux Drive, Sunnyvale, CA 94089			
Helen Rivera	2930 Pearl Street, Boulder, CO. 80301			
Samuel Bailey	111 8th Ave. New York, NY 10011			
Kenneth Gonzalez	2930 Pearl Street, Boulder, CO. 80301			
Sharon Washington	1160 Bordeaux Drive, Sunnyvale, CA 94089			
Mildred Evans	1160 Bordeaux Drive, Sunnyvale, CA 94089			

City (input)	Phone Code (output)
Bordeaux	3 3
Kraków	4 8
Dagstuhl	4 9
Buenos Aires	5 4 ←



3. How to compare algorithms effectively? variants

pipeline:



IDEA:

while not confident:

$t = \text{select_task}()$

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

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 loops that can be used by others.
- making new/better algorithms

