

Templates of slides for P3

A brief update on your problem (important)

Describe in English

- what artifacts (programs, etc) will you synthesize,
- what is your behavioral spec? (i/o examples? a ref impl?)
- list the benchmarks/tests you created to drive your design

[30 seconds]

Demo of a program in your language (YL)

Show us an example of a typical program in YL

Walk us through the code

Include a screenshot of running example

Revisions to YL (brief slide)

Describe how you simplified YL

- what observations allowed you to drop a construct
- or reduce it to a smaller language core?

Did you have to add a construct?

- what use case necessitated that?

Argue YL is the smallest language for your task

Holes in YL (the key slide)

Describe your candidate space (CS)

what set of programs will your synthesizer search

CS can be described by defining a “hole” construct

holes are what your synthesizer will complete

A definition of “hole” has two parts:

a) where can holes appear in YL programs

eg: loop bounds; lock placement; Xpath expressions

b) with what expressions are holes completed?

eg: linear over induction vars; lock placement freedom; equality between node ids and string literals

You should try to explain these points entirely on the example from the next slide.

Example and size of Candidate Space

Show example of a sketch

- reuse example from slide 3; add holes to it

Size of the candidate space you need to search

- may be larger than you expect (unless you found tiny YL)
- determines what algorithm you need to use
- likely will show you can't just gen-and-test candidates

Back-of-the-envelope space estimates

- think of holes as grammars (add holes to YL grammar, G_H)
- what is the size of largest program you need to find, s
- how many candidate programs of up to size s are in $L(G_H)$?

Checklist

- demo of screenshot
- revisions to your language
- sketches: what are YL holes and where can they go?
- size of candidate space that needs to be searched

Next project presentation

In P4, you will have a compiler from YL to formulas

Or to some other scalable synthesis algorithm