

**Templates of slides for P4**

**Experiments with your synthesizer**

# State your synthesis problem

---

Refresh your problem with sample code. Ideally, show

- the spec,
- the sketch (the structure of the sketch may be important)
- the result of synthesis

To save space,

- you can omit the spec, if obvious
- you can overlay the result on the sketch

Define the type of holes

- basic types of holes: numbers, strings, anything else?
- higher level: expressions, sequences of instructions?
- implementation status: which holes are future work

[45 seconds]

# Your synthesis algorithm

---

For most of you, this slide will say:

1) I translate the program to a formula

- the formula insist that the desired program is correct on a set  $S$  of inputs
- state how you obtained  $S$ , and the size of  $S$

2) I solve the formula with a solver Foo

- and then I map the model back to code
- say how you feed the model back to the sketch

If you are using a different algorithm, give more details than suggested above.

[30 seconds]

# Your encoding

---

Describe encoding of program semantics into logic.

Pick a representative fragment of your program.

with the important data types and key operations

Show the resulting logical encoding

- say how ints, etc, are translated (Ints, bitvector, unary?)
- explain impact of data encoding on encoding of ops

A figure really helps here

it could show the encoding as a circuit

Implementation of your synthesizer (important):

tell us how you generate the formula

If you don't use logic encoding, you must explain how you prune your search space

ie, how you eliminate many candidates at once

[30 seconds]

# Demo and scalability

---

Screenshot of your system synthesizing  
use your judgment what to illustrate here

# Lessons and limitations

---

## Unique encoding challenges

Edward: impact of the spec

incorporation of negative examples (P must not output  $v$ )

Peter: impact of the program semantics

encoding of concurrency (instruction interleaving)

What is symbolic in your language and what is not

impact this choice has on verification

[45 secs]

# Checklist

---

- demo (screenshot)
- description of your encoding
- description of your synthesizer implementation
- scalability chart

# Next project presentation

---

In P5, you will have a compiler that generates scalable encoding