Summer School on Software Synthesis
Schloss Dagstuhl
August 8-12 2011

Ras Bodik, Sumit Gulwani, Viktor Kuncak, Eran Yahav
Welcome to Schloss Dagstuhl

A “computer science monastery” with its own ghost

Seclusion facilitating communication

Logistics of your stay: talk to me or the reception

- get familiar with facilities (bikes, table tennis, billiard)
- think of a trip for Wed afternoon
- pay when checking out
History of this summer school

Dagstuhl Seminar on Software Synthesis, Dec 2009
- brought together several communities
- they gave tutorials to each other
- very well received seminar ==> what to do next?

Idea: give these tutorial talks to PhD students
- after all, they do all the work
- and hence will advance the field
- again, a spectrum of approaches to be presented
## Schedule

<table>
<thead>
<tr>
<th>Time</th>
<th>Location</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:30 - 8:45</td>
<td>restaurant</td>
<td>breakfast</td>
<td>breakfast</td>
<td>breakfast</td>
<td>breakfast</td>
<td>breakfast</td>
</tr>
<tr>
<td>8:45 - 10:00</td>
<td></td>
<td>Introduction (Ras)</td>
<td>Eran</td>
<td>Eran</td>
<td>Vijay</td>
<td>Vijay</td>
</tr>
<tr>
<td>10:00 - 10:30</td>
<td>classroom</td>
<td>coffee break</td>
<td>coffee break</td>
<td>coffee break</td>
<td>coffee break</td>
<td>coffee break</td>
</tr>
<tr>
<td>10:30 - 12:00</td>
<td></td>
<td></td>
<td>Barbara</td>
<td>Johann</td>
<td>Johann</td>
<td>Discussion (Ras)</td>
</tr>
<tr>
<td>12:15 - 2:00</td>
<td>restaurant</td>
<td>lunch</td>
<td>lunch</td>
<td>lunch</td>
<td>lunch</td>
<td>lunch</td>
</tr>
<tr>
<td>2:00 - 3:30</td>
<td></td>
<td>Armando</td>
<td>Armando</td>
<td>afternoon</td>
<td>Armando and Johann</td>
<td>departure</td>
</tr>
<tr>
<td>3:30 - 4:00</td>
<td>restaurant</td>
<td>coffee and cake</td>
<td>coffee and cake</td>
<td>outing to nearby</td>
<td>coffee and cake</td>
<td></td>
</tr>
<tr>
<td>4:00 - 5:30</td>
<td></td>
<td>Viktor</td>
<td>Viktor</td>
<td>attraction</td>
<td>Ras</td>
<td></td>
</tr>
<tr>
<td>6:00 - 7:30</td>
<td>restaurant</td>
<td>dinner</td>
<td>dinner</td>
<td>dinner</td>
<td>dinner</td>
<td></td>
</tr>
<tr>
<td>8:00 - tbd</td>
<td>wine cellar</td>
<td>discussions; AutoBayes install help</td>
<td>Vernissage at 7:30; discuss.</td>
<td>discussions, hands-on, etc</td>
<td>discussions, hands-on, etc</td>
<td></td>
</tr>
</tbody>
</table>
Introductions

Name
school
research interests
thesis topic or still looking?

Hobbies:
what you’d be doing this week if you were not here?
Why are we here?

Synthesis 2.0:
Resurgence of synthesis in several communities

- embedded comp.
- HPC, clusters of multicores
- HW synth
- query optimization in DB
- robotics
- compiler optimization
- numerical SW
- design from src
- model from traces
- exec. monitors
- reverse engineering

Matlab → executable

C → RTL

HCI - end-user programs / from demos

- test case gen
- spec. synth.
- spec → mockup
Why are we here?

Synthesis 2.0: Resurgence of synthesis in several communities

- Deductive Synthesis
- Synthesis from demonstration
- Controller synthesis
- Transformational synthesis with performance exploration
- Synthesis of loop invariants for verification
- Partial programs for intelligent agent programming
- Bug repair
- Efficient program space exploration
What is synthesis?

Wikipedia:

Program synthesis comprises a range of technologies for the automatic generation of executable computer programs from high-level specifications of their behaviour. In contrast to compilation, the specifications are usually non-algorithmic.
What is synthesis?

is synthesis

-----------------------------

PBD
Compiler that does not work

a rewrite-based compiler

is not synthesis

-----------------------------

manual coding
What is synthesis?

is synthesis

search/constraint solving
semantics-aware
resolves freedom in specification
choose order of transformations
specs are declarative

is not synthesis

optimizing compiler
(deterministic xformations)
syntax-directed
Successes

• Academic
  Quick Code
  Smart Edit
  SpecWare

• Industrial
  Auto Bayes
  Autofilter
  FFTW (Spiral)
  HW synth
  UML -> code

(POD for Excel)
Academic successes

- Constable: programs from constructive proofs
- Play-In-Play-Out
- SmartEdit: programming by demonstration
  - genetic programming
  - SKETCHING
  - DSLs (SRL, LabView, Matlab ?)
  - super optimizers
  - inductive synthesis of regular languages
Industrial successes

• KIDS
• FFTW, SPiRAL
• autotuning (linear algebra libraries)
• program refinement (B method)
  • hw synthesis
  • compiler synthesis
  • controller synthesis
Why synthesis now?

Needs (challenge problems)

- multicore programming
- end-user programming
- safety critical
Why synthesis now?

Opportunities (why became possible to do now)

cycles, not smarts
-search, constraint solving

algorithms, abstractions, DSL/libs

funding available

lessons of verification failures
Some themes

• How to obtain a specification?
• How to develop and debug domain theories?
• How to exploit recent advances in verification and decision procedures?
• What lessons can be learnt from success/failure stories?
• What problems could lead to great dissertations?
Another round of introductions

• What is synthesis?
• What drew you to synthesis?
• What artifacts do you want to synthesize?
• What is the input to your "synthesizer"?
• What techniques do you use?
• A modest success:
• A spectacular failure:
• In Dagstuhl, I want to understand X
Logistics

After dinner cheese and wine in the cellar

Wed afternoon / evening outing
What would you like from this school