Chairs’ Reports

POPL 2009
General Chair’s Report
PC Chair’s Report
Submission Topics

abstract interpretation
algebraic / categorical methods
bio-computing
compilers
concurrency
contracts
database programming
debugging
distributed systems
domain-specific languages
dynamic analysis
functional programming
logic
low-level languages
memory management
module systems

multi-core programming
object-oriented programming
partial evaluation / multi-stage prog.
process calculi
program transformation
program verification
scientific computing
security
semantics
static analysis
testing
tools
type inference
types
web programming
# Topic Popularity

<table>
<thead>
<tr>
<th>Count</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>static analysis</td>
</tr>
<tr>
<td>60</td>
<td>program verification</td>
</tr>
<tr>
<td>57</td>
<td>semantics</td>
</tr>
<tr>
<td>47</td>
<td>types</td>
</tr>
<tr>
<td>40</td>
<td>logic</td>
</tr>
<tr>
<td>32</td>
<td>concurrency</td>
</tr>
<tr>
<td>31</td>
<td>program transformation</td>
</tr>
<tr>
<td>28</td>
<td>compilers</td>
</tr>
<tr>
<td>24</td>
<td>object-oriented programming</td>
</tr>
<tr>
<td>17</td>
<td>tools</td>
</tr>
<tr>
<td>16</td>
<td>multi-core programming</td>
</tr>
<tr>
<td>16</td>
<td>functional programming</td>
</tr>
<tr>
<td>14</td>
<td>type inference</td>
</tr>
<tr>
<td>13</td>
<td>domain-specific languages</td>
</tr>
<tr>
<td>13</td>
<td>distributed systems</td>
</tr>
<tr>
<td>11</td>
<td>security</td>
</tr>
<tr>
<td>11</td>
<td>algebraic / categorical methods</td>
</tr>
<tr>
<td>10</td>
<td>process calculi</td>
</tr>
<tr>
<td>10</td>
<td>abstract interpretation</td>
</tr>
<tr>
<td>8</td>
<td>low-level languages</td>
</tr>
<tr>
<td>8</td>
<td>dynamic analysis</td>
</tr>
<tr>
<td>8</td>
<td>contracts</td>
</tr>
<tr>
<td>7</td>
<td>partial evaluation / multi-stage prog.</td>
</tr>
<tr>
<td>5</td>
<td>testing</td>
</tr>
<tr>
<td>5</td>
<td>module systems</td>
</tr>
<tr>
<td>4</td>
<td>web programming</td>
</tr>
<tr>
<td>4</td>
<td>debugging</td>
</tr>
<tr>
<td>3</td>
<td>scientific computing</td>
</tr>
<tr>
<td>3</td>
<td>memory management</td>
</tr>
<tr>
<td>3</td>
<td>database programming</td>
</tr>
<tr>
<td>0</td>
<td>bio-computing</td>
</tr>
</tbody>
</table>
A Little Survey

• Some members of the POPL community would like to see a world where POPL submissions would routinely be accompanied by mechanically checked proofs.

• Question: How many POPL submissions are already using proof assistants?
The Survey

☐ Check this box if you have used a proof assistant in some way in developing the results in your paper -- e.g., for formalizing and sanity-checking definitions. (Your responses to these questions will be used only for informational purposes; they will not affect your chances of acceptance.)

☐ Check this box if the proofs of your main results have been fully mechanically checked.

☐ Check this box if you may or may not have used a proof assistant in some way but prefer not to say which.
<table>
<thead>
<tr>
<th>Category</th>
<th>Submissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine assisted</td>
<td>30</td>
</tr>
<tr>
<td>Fully verified</td>
<td>12</td>
</tr>
<tr>
<td>Declined to specify</td>
<td>5</td>
</tr>
<tr>
<td>No response</td>
<td>112</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>159</strong></td>
</tr>
<tr>
<td>Category</td>
<td>Submissions</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Machine assisted</td>
<td>30</td>
</tr>
<tr>
<td>Fully verified</td>
<td>12</td>
</tr>
<tr>
<td>Declined to specify</td>
<td>5</td>
</tr>
<tr>
<td>No response</td>
<td>112</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>159</strong></td>
</tr>
</tbody>
</table>
Most Influential POPL Paper from 1999
Andrew Myers
Practical Mostly-Static Information Flow
Andrew Myers' 1999 POPL Paper "Practical Mostly-Static Information Flow Control" demonstrated the practicality of using static information flow analysis to protect privacy and preserve integrity by giving an efficient information flow type checker for an extension of the widely-used Java language. The work has had a significant impact both within and beyond the programming language community. In particular, subsequent work for other languages has largely followed the path laid out in this paper, and the compiler infrastructure developed for JFlow (now called Jif) is widely used as a research platform. Furthermore, using the JFlow work as a basis, several major research initiatives are investigating the challenges of building complex, real-world systems with confidentiality guarantees.
POPL Logo Competition
Competition Organizers

• Chair: Swarat Chaudhuri

• Selection committee:
  – Luca Cardelli
  – Swarat Chaudhuri
  – Shriram Krishnamurthi
  – Benjamin Pierce
And the winner is...

Jan Christiansen

Christian-Albrechts University, Kiel
POPL’s new logo!
http://www.cafepress.com/popl

(2009 shirts coming soon!)
POPL 2010