Concurrent Views Framework

Paper by Thomas Dinsdale-Young, Lars Birkedal, Phillipa Gardner, Matthew Parkinson & Hongseok Yang

Presentation by Ahmad Salim Al-Sibahi
Disposition

• Introduction
• Views
  – Definition
  – Soundness
• Example
• Conclusions
INTRODUCTION

Objective and Achievements
Objective

A sound generalized framework for description of compositional reasoning systems.
## Motivation

<table>
<thead>
<tr>
<th>Year</th>
<th>Authors/Institutions</th>
<th>Method/Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>1969</td>
<td>C.A.R. Hoare</td>
<td>Hoare Logic</td>
</tr>
<tr>
<td>1976</td>
<td>S. Owicki and D. Gries</td>
<td>Owicki-Gries Methods</td>
</tr>
<tr>
<td>1983</td>
<td>C. B. Jones</td>
<td>Rely-Guarantee Method</td>
</tr>
<tr>
<td>1987</td>
<td>J. Y. Girard</td>
<td>Linear Logic</td>
</tr>
<tr>
<td>1990</td>
<td>P. Wadler</td>
<td>Linear Types</td>
</tr>
<tr>
<td>2002</td>
<td>J. C. Reynlods</td>
<td>Separation Logic</td>
</tr>
<tr>
<td>2004</td>
<td>S. Brookes</td>
<td>Concurrent Separation Logic</td>
</tr>
<tr>
<td>2005</td>
<td>M. J. Parkinson and G. M. Bierman</td>
<td>Abstract Predicates</td>
</tr>
<tr>
<td>2007</td>
<td>X. Feng, R. Ferreira and Z. Shao</td>
<td>SAGL</td>
</tr>
<tr>
<td></td>
<td>V. Vafeiadiadis and M. J. Parkinson</td>
<td>RGSep</td>
</tr>
<tr>
<td>2009</td>
<td>X. Feng</td>
<td>Local Rely-Guarantee Method</td>
</tr>
<tr>
<td></td>
<td>M. Dodds et al.</td>
<td>Deny-Guarantee Method</td>
</tr>
<tr>
<td>2010</td>
<td>T. Dinsdale-Young et al.</td>
<td>Concurrent Abstract Predicates</td>
</tr>
</tbody>
</table>
Key Achievements

• Presented a simple but highly applicable method of abstraction

• Instantiated the framework with key examples
  – Rely-guarantee method
  – Concurrent separation logic
  – Concurrent abstract predicates
  – Recursive reference and unique pointer type systems
  – Adapted Owicki-Gries methods

• Proved soundness of framework using Coq
Composable Concurrent Programs

CONCURRENT VIEWS FRAMEWORK
Definition

- Semantics

- Composition

\[ \ast : \text{View} \times \text{View} \rightarrow \text{View} \]

- Unit view \( (I) \)

\[ \forall V : \text{View}. I \ast V = V \ast I = V \]
Composition

• Partial-correctness Triple

\[ \{ P \} C \{ Q \} \]

• Compositionality

\[ \begin{align*}
\{ P_1 \} & \quad C_1 & \{ Q_1 \} \\
\{ P_2 \} & \quad C_2 & \{ Q_2 \}
\end{align*} \]

\[ \begin{array}{c}
\{ P_1 * P_2 \} & \quad C_1 \parallel C_2 & \quad \{ Q_2 * Q_2 \}
\end{array} \]
Soundness

• Reification Function

\[ \cdot : \text{View} \rightarrow \mathcal{P}(S) \]

• Theorem
Application of Views

EXAMPLE
Rely-Guarantee

• Definition

\[ \{P\} \vdash R, G \models \{Q\} \]

• View

\[(P, R, G) \quad (Q, R, G)\]

• Composition

\[
\begin{cases}
(P_1 \cap P_2, R_1 \cap R_2, G_1 \cup G_2) & \text{if } G_1 \subseteq R_2 \land G_2 \subseteq R_1 \\
\bot & \text{otherwise}
\end{cases}
\]

• Reification

\[\lbrack (P, R, G) \rbrack = P\]
Impact and Future Work

CONCLUSION
Impact

• Annotation-based extension for safe parallelism in C# (C. Gordon, et al.; University of Washington and Microsoft)

• Structural Separation Logic of POSIX filesystems (A. Wright; Imperial College)
Future Work

- Formalize practical-oriented approaches such as STM and SCOOP using Views
- Incorporate Views-based reasoning logic into tools for static analysis
Discussion and Reflection

QUESTIONS