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

Objective and Achievments

INTRODUCTION

Objective

A sound generalized framework for description of compositional reasoning systems.

Motivation

1969	C.A.R. Hoare	Hoare Logic
1976	S. Owicki and D. Gries	Owicki-Gries Methods
1983	C. B. Jones	Rely-Guarantee Method
1987	J. Y. Girard	Linear Logic
1990	P. Wadler	Linear Types
2002	J. C. Reynlods	Separation Logic
2004	S. Brookes	Concurrent Separation Logic
2005	M. J. Parkinson and G. M. Bierman	Abstract Predicates
2007	X. Feng, R. Ferreira and Z. Shao	SAGL
	V. Vafeiadis and M. J. Parkinson	RGSep
2009	X. Feng	Local Rely-Guarantee Method
	M. Dodds et al.	Deny-Guarantee Method
2010	T. Dinsdale-Young et al.	Concurrent Abstract Predicates

Key Achievments

- 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

$$* : View \times View \to View$$

- Unit view (*I*)

$$\forall V : View. I * V = V * I = V$$

Composition

- Partial-correctness Triple

$$\{P\}\,C\,\{Q\}$$

- Compositionality

$$\frac{\begin{array}{c}\{P_1\}\,C_1\,\{Q_1\}\\ \{P_2\}\,C_2\,\{Q_2\}\end{array}}{\{P_1 * P_2\}\,C_1 \| C_2\,\{Q_2 * Q_2\}}$$

Soundness

- Reification Function

$$\lfloor \cdot \rfloor : View \to \mathcal{P}\left(S\right)$$

- Theorem

Application of Views

EXAMPLE

Rely-Guarantee

- Definition

$$R, G \vdash \{P\}\, C\, \{Q\}$$

- View

$$(P, R, G) \qquad (Q, R, G)$$

- Composition

$$\begin{cases} (P_1 \cap P_2, R_1 \cap R_2, G_1 \cup G_2) & \textbf{if } G_1 \subseteq R_2 \wedge G_2 \subseteq R_1 \\ \bot & \textbf{otherwise} \end{cases}$$

- Reification

$$\lfloor (P, R, G) \rfloor = P$$

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