

Корректность ПО

Валерий Исаев

05 мая 2011 г.

Существующие подходы

Тестирование

Тестирование

Достоинства:

- ▶ Простота
- ▶ Распространенность

Недостатки:

- ▶ Ненадежность
- ▶ "Низкоуровневость"

Model checking [1]

Model checking [1]

Достоинства:

- ▶ Автоматизированность
- ▶ Возможность верифицировать параллельные системы

Недостатки:

- ▶ Комбинаторный взрыв
- ▶ Узкая область применения

Логический вывод

- ▶ Hoare Logic [2]
- ▶ Separation Logic [3,4,5]
- ▶ Type theory [6,7,8]

Hoare Logic

- ▶ Тройки Хоара $\{P\}S\{Q\}$
- ▶ Простой императивный язык
- ▶ Расширения для указателей, переходов, подпрограмм, конкурентных вычислений

Пример

```
{ array_length(t) >= 1 and
    sorted_array(t,1,array_length(t)) }
l := 1; u := array_length(t); p := 0;
while l <= u do
    m := (l + u) / 2;
    if t[m] < v then l := m + 1
    else if t[m] > v then u := m - 1
    else begin p := m; l := u + 1 end
done
{ (1 <= p <= array_length(t) and t[p]=v) or
  (p = 0 and not In(t,1,array_length(t))) }
```

Пример

```
{ array_length(t) >= 1 and
    sorted_array(t,1,array_length(t)) }
l := 1; u := array_length(t); p := 0;
while l <= u do
{ invariant 1 <= l and u <= array_length(t)
  and 0 <= p <= array_length(t)
  and (p = 0 -> In(t,1,array_length(t)) ->
        In(t,l,u)) and (p > 0 -> t[p]=v)
  variant u - l }
m := (l + u) / 2;
if t[m] < v then l := m + 1
else if t[m] > v then u := m - 1
else begin p := m; l := u + 1 end
done
{ (1 <= p <= array_length(t) and t[p]=v) or
  (p = 0 and not In(t,1,array_length(t))) }
```

Применение

- ▶ Hardware verification
- ▶ Software verification
 - ▶ Why [9]
 - ▶ VCC [10]
 - ▶ Frama-C [11]

Логический вывод

Достоинства:

- ▶ Формальное доказательство корректности
- ▶ Гибкость

Недостатки:

- ▶ Сложность

Теория Типов

История[12,49]

- ▶ 1900-ые: Типы (Bertrand Russell)
- ▶ 1920-ые: Комбинаторная логика (Haskell Curry)
- ▶ 1930-ые: Лямбда исчисление (Alonzo Church)
- ▶ 1950-ые: Соответствие "Теоремы-как-типы"(Haskell Curry)
- ▶ 1960-ые: Обобщение для лямбда исчисления и прочих систем (William Howard)
- ▶ 1960-ые,1970-ые: Язык Automath для автоматической проверки доказательств (Nicolaas de Bruijn)
- ▶ 1970-ые: Теория типов (Martin-Lof)
- ▶ 1980-ые: Разработка NuPRL (Robert Constable)
- ▶ 1980-ые: Calculus of Constructions (Thierry Coquand, Gerard Huet)

Системы

- ▶ Theorem provers:
 - ▶ Coq [13,14]
 - ▶ HOL [15]
 - ▶ NuPRL [16]
- ▶ Языки:
 - ▶ Agda2 [17]
 - ▶ Epigram 2 [18,19,20,21,22]

Теория Типов

- ▶ Типы в языках программирования [23,24]
- ▶ Легковесный подход [25,26,27,28]
- ▶ Изоморфизм Карри-Говарда [29]
- ▶ Зависимые типы [24,30,31,32,33]
- ▶ Теория типов и другие подходы к верификации [34,35]

Примеры

Другие примеры

- ▶ Обобщенное программирование [22,36]
- ▶ Описание протоколов, безопасные каналы [25,27,28]
- ▶ Описание форматов данных/языков, безопасные парсеры [32]
- ▶ Описание схемы базы данных, безопасные SQL запросы [32]
- ▶ Безопасная конкурентность (мутексы, STM) [25,37,38,39]

Другие примеры

- ▶ Использование типов в низкоуровневом программировании [24,25,26]
- ▶ Безопасная работа с памятью [24,34]
- ▶ Параметризованные вычисления [35,40]
- ▶ Спецификация эффектов [40,41]
- ▶ Спецификация использования ресурсов [24,42,43,44,45,46]
- ▶ Спецификация завершаемости [47,48]
- ▶ ...

Преимущества

- ▶ Безопасность
- ▶ Отсутствие проверок во время исполнения
- ▶ Масштабируемость
- ▶ Документация
- ▶ Проектирование

Библиография

- 1 Model Checking, Edmund M. Clarke, Jr., Orna Grumberg, Doron A. Peled, MIT Press, 1999
- 2 An Axiomatic Basis for Computer Programming, C. A. R. Hoare, 1969
- 3 Local Reasoning about Programs that Alter Data Structures, Peter O'Hearn, John Reynolds, Hongseok Yang, 2001
- 4 Separation Logic: A Logic for Shared Mutable Data Structures, John Reynolds, 2002
- 5 Separation Logic and Abstraction, Matthew Parkinson, Gavin Bierman, 2005
- 6 Type theory and Functional programming, Simon Thompson, 1999

- [7 Programming in Martin-Lof's type theory. An introduction, Bengt Nordstrom, Kent Petersson, Jan M. Smith, Oxford University Press, 1990](#)
- [8 Computation and reasoning. A type theory for computer science, Zhaohui Luo, Clarendon Press, 1994](#)
- [9 <http://why3.gforge.inria.fr/>](#)
- [10 <http://research.microsoft.com/en-us/projects/vcc/>](#)
- [11 <http://frama-c.com/index.html>](#)
- [12 History of Lambda-calculus and Combinatory Logic, Felice Cardone, J. Roger Hindley, 2006](#)

[13](http://coq.inria.fr/) <http://coq.inria.fr/>

[14](#) Certified programming with dependent types, Adam Chlipala,
2011

[15](#) <http://hol.sourceforge.net/>

[16](#) <http://www.cs.cornell.edu/info/projects/nuprl/>

[17](#) <http://wiki.portal.chalmers.se/agda/pmwiki.php>

[18](#) <http://www.e-pig.org/darcs/Pig09/web/>

- 19 Observational Equality, Now!, Thorsten Altenkirch, Conor McBride, Wouter Swierstra, 2007
- 20 Why Dependent Types Matter, Thorsten Altenkirch, Conor McBride, James McKinna, 2005
- 21 The view from the left, Conor McBride, James McKinna, 2004
- 22 The Gentle Art of Levitation, James Chapman, Pierre-Evariste Dagand, Conor McBride, Peter Morris, 2010
- 23 Types and Programming Languages, Benjamin C. Pierce, MIT Press, 2002
- 24 Advanced Topics in Types and Programming Languages, ed. Benjamin C. Pierce, MIT Press, 2005

- 25 Fun with Type Functions, Oleg Kiselyov, Simon Peyton Jones, Chung-Chieh Shan, 2010
- 26 Position: Lightweight static resources. Sexy types for embedded and system programming, Oleg Kiselyov, Chung-Chieh Shan, 2007
- 27 Haskell Session Types with (Almost) No Class, Riccardo Pucella, Jesse A. Tov, 2008
- 28 Session Types in Haskell, Matthew Sackman, Susan Eisenbach, 2008
- 29 Lectures on the Curry-Howard Isomorphism, Morten Heine B. Sorensen, Pawel Urzyczyn, 1998
- 30 A type-correct, stack-safe, provably correct expression compiler in Epigram, James McKinna, Joel Wright, 2006

- 31 Safe Arrays via Regions and Dependent Types, Christian Grothoff, Jens Palsberg, Vijay Saraswat, 2006
- 32 The Power of Pi, Nicolas Oury, Wouter Swierstra, 2008
- 33 Total Parser Combinators, Nils Anders Danielsson, 2010
- 34 Hoare Type Theory, Polymorphism and Separation, Aleksandar Nanevski, Greg Morrisett, 2006
- 35 Parameterised Notions of Computation, Robert Atkey, 2008
- 36 Constructing Universes for Generic Programming, Peter W. J. Morris, 2007

- 37 A Poor Man's Concurrency Monad, Koen Claessen, 1993
- 38 Composable Memory Transactions, Tim Harris, Simon Marlow, Simon Peyton Jones, Maurice Herlihy, 2006
- 39 Lock Free Data Structures using STM in Haskell, Anthony Discolo, Tim Harris, Simon Marlow, Simon Peyton Jones, Satnam Singh, 2006
- 40 A Functional Specification of Effects, Wouter Swierstra, 2008
- 41 Notions of Computation and Monads, Eugenio Moggi, 1989
- 42 Equality-Based Uniqueness Typing, Edsko de Vries, Rinus Plasmeijer, David M Abrahamson, 2007

- 43 Uniqueness Typing Simplified, Edsko de Vries, Rinus Plasmeijer, David M Abrahamson, 2008
- 44 L3: A Linear Language with Locations, Amal Ahmed, Matthew Fluet, Greg Morrisett, 2004
- 45 Linear types can change the world! Philip Wadler, 1990
- 46 Is there a use for linear logic? Philip Wadler, 1991
- 47 Total Functional Programming, D.A.Turner, 2004
- 48 General Recursion via Coinductive Types, Venanzio Capretta, 2005
- 49 A Modern Perspective on Type Theory, Fairouz Kamareddine, Twan Laan, Rob Nederpelt, 2005