

A. Identificação

A.1 César de Jesus Pereira da Cunha Rodrigues
cjr@di.uminho.pt

A.2 Software Refinement by Calculation

A.3 José Nuno Fonseca Oliveira
Universidade do Minho
Departamento de Informática
jno@di.uminho.pt

A.3 Janeiro de 2003

A.4 2006

B. Resumo

B.1 I&D

Refinamento de software

B.2 Resumo

Abordamos os fundamentos do desenvolvimento de “programas” por cálculo, recorrendo no âmbito do cálculo de relações binárias dito *pointfree*.

Desenvolvemos a transposição funcional, uma técnica que tem por objectivo converter relações para funções, de modo a promover a álgebra de relações através da álgebra de funções.

Algumas leis novas de SETS são apresentadas e provadas envolvendo o cálculo de isomorfismos e a Maybe transposta, o tipo de dados das relações e a abordagem genérica (politipismo) através de uma monade genérica especificada depois de considerada a transposição. Também levamos em conta o tópico do invariant refinement.

É apresentada uma aproximação construtiva, calculacional à teoria da normalização de bases de dados relacionais. Apresentamos um novo conjunto de leis de SETS que evitam a violação das formas normais, causada por dependências parciais, dependências transitivas e dependências multivalor. Elas eliminam um invariante “semântico” no processo de refinamento correspondente à tarefa de normalização que tem lugar.

Relacionamos o refinamento algorítmico com o refinamento relacional com dois tipos de refinamento coalgébrico.

O refinamento de coálgebras envolve uma mudança do espaço de discurso, correspondente às dicotomias: refinamento coalgébrico versus refinamento relacional e algorítmico, coálgebras versus relações, cf. transformada de Laplace.

No nosso caso, realizamos refinamento coalgérico por recurso ao refinamento relacional e algorítmico das relações de transição correspondentes.

Como tal, reduzimos o refinamento de uma coálgebra ao refinamento algébrico.

B.3 Objectivos Estrategicos

- Estudar os fundamentos do desenvolvimento formal de programas por cálculo.
- Provar as vantagens do cálculo de relações binárias, em estilo dito *pointfree*, enquanto ferramenta formal de cálculo.

C. Contribuições

- C.1**
- Tranposição de relações e sua aplicação no contexto do refinamento, i.e. dos items seguintes.
 - desenvolvimento de um cálculo de refinamento de dados, com aplicações à caracterização formal da teoria da normalização de bases de dados relacionais.
 - Desenvolvimento de um cálculo de refinamento de algoritmos, previamente conhecido na literatura. A novidade consiste na aplicação de métodos relacionais.
 - Redução de resultados recentes, no contexto do refinamento coalgérico, a resultados clássicos, e.g. o cálculo de algoritmos mencionado anteriormente e o refinamento relacional (também conhecido como *data refinement*, na literatura anglo-saxónica).

C.2

- [OR04]

Functional transposition is a technique for converting relations into functions aimed at developing the relational algebra *via* the algebra of functions.

This paper attempts to develop a basis for *generic transposition*. Two instances of this construction are considered, one applicable to any relation and the other applicable to simple relations only.

Our illustration of the usefulness of the generic transpose takes advantage of the *free theorem* of a polymorphic function. We show how to derive laws of relational combinators as free theorems of their transposes. Finally, we relate the topic of functional transposition with the *hashing* technique for efficient data representation.

D. Enquadramento

D.1 Enquadramento Científico

Os objectivos do doutoramento enquadram-se no contexto da workshop internacional The BCS FACS Refinement Workshop e da cooperação internacional RelMiCS (Métodos Relacionais nas Ciências da Computação).

D.2 Motivação

Utilizar um cálculo mais que centenário na resolução das dificuldades sentidas no desenvolvimento de software.

D.3 Objectivos Detalhados

Conseguir alcançar resultados originais listados nas contribuições, que permitiram alcançar os objectivos.

D.4 Trabalhos Alternativos

Surgiram trabalhos alternativos na área do refinamento algoritmico [BEM92] e [Gro02]. A primeira referência utiliza um cálculo relacional em estilo pointwise e a segunda surge no contexto de Z e do cálculo de predicados.

D.5 Bibliografia Principal

As referências principais são as associadas à *Algebra of Programming*, [ABH⁺92, BdM97, Bac04]

E. Avaliação

E.1 Análise Comparativa

O cálculo de relações binárias em estilo pointfree permite um cálculo mais efectivo do que as aproximações da concorrência, [BEM92, Gro02],

E.2 Auto-avaliação do Trabalho Realizado

Conseguir alcançar resultados novos, no contexto dos objectivos propostos.

E.3 Auto-avaliação da Documentação Produzida

A publicação [OR04] generalizou resultados existentes e foi aplicada na descoberta de novas leis de refinamento de dados e na introdução de nova notação para falar de coálgebras com uma determinada relação de transição.

References

- [ABH⁺92] Chritiene Aarts, Roland Backhouse, Paul Hoogendijk, Ed Voermans, and Jaap van der Woude. A relational theory of datatypes, December 1992.
- [Bac04] R.C. Backhouse. Mathematics of program construction, June 2004. with help and contributions by Marcel Bijsterveld and Henk Doornbos and Rik van Geldrop and Diethard Michaelis and Jaap van der Woude.
- [BdM97] R. Bird and O. de Moor. *Algebra of Programming*. Series in Computer Science. Prentice-Hall International, 1997. C. A. R. Hoare, series editor.
- [BEM92] Noureddine Boudriga, Fathi Elloumi, and Ali Mili. On the lattice of specifications: Applications to a specification methodology. *Formal Asp. Comput.*, 4(6):544–571, 1992.
- [Gro02] Lindsay Groves. Refinement and the z schema calculus. *ENTCS*, 70(3), 2002. The BCS FACS Refinement Workshop, Denmark July 2002.
- [OR04] J.N. Oliveira and C.J. Rodrigues. Transposing relations: from *Maybe* functions to hash tables. In D. Kozen, editor, *MPC'07* <http://www.cs.cornell.edu/Projects/MPC2004> : *Seventh International Conference on Mathematics of Program Construction, 12-14 July, 2004, Stirling, Scotland, UK (Organized in conjunction with AMAST'04)*, volume 3125 of *Lecture Notes in Computer Science*, pages 334–356. Springer, July 2004.