# SEMINAR

## FGI 3: Logik und Semantik von Programmen

Mi 12–14 Uhr, C–221, Beginn: 21.10.09

Rüdiger Valk

# Ziel des Seminars

In dieser Veranstaltung werden Schlüsselqualifikationen durch

- selbstständiges Recherchieren,
- Strukturieren,
- Präsentieren und
- Moderieren erworben.

Die Zulassung zur Modulprüfung setzt die regelmäßige Teilnahme an dem Seminar, der Vortrag eines Referats und die Abgabe einer Hausarbeit (Hausarbeit) und eines statt.

# Terminplan

- 21.10. Einführung und Themenvergabe
- 28.10. ---
- 4.11. /11.11./18.11./25.11/2.12/9.12/16.12/
6.1./13.1./20.1./27.1/3.2.
(12 Vortragstermine)

# Hausarbeit

- LaTex
- mindestens 20 bis 30 Seiten
- Layout: Springer
- Aufbau

# LaTex

## Modelling Concurrency with Quotient Monoids

Ryszard Janicki[⋆] Dai Tri Man Lê[⋆⋆]

Department of Computing and Software,
McMaster University,
Hamilton, ON, L8S 4K1 Canada
{janicki,ledt}@mcmaster.ca

**Abstract.** Four quotient monoids over step sequences and one with compound generators are introduced and discussed. They all can be regarded as extensions (of various degrees) of Mazurkiewicz traces [14] and comtraces of [10].

**Keywords:** quotient monoids, traces, comtraces, step sequences, stratified partial orders, stratified order structures, canonical representations.

### 1 Introduction

Mazurkiewicz traces or partially commutative monoids [1, 5] are quotient monoids over sequences (or words). They have been used to model various aspects of concurrency theory since the late seventies and their theory is substantially developed [5]. As a language representation of partial orders, they can nicely model "true concurrency."

For Mazurkiewicz traces, the basic monoid (whose elements are used in the equations that define the trace congruence) is just a free monoid of sequences. It is assumed that generators, i.e. elements of trace alphabet, have no visible internal structure, so they could be interpreted as just names, symbols, letters, etc. This can be a limitation, as when the generators have some internal structure, for example if they are sets, this internal structure may be used when defining the set of equations that generate the quotient monoid. In this paper we will assume that the monoid generators have some internal structure. We refer to such generators as 'compound', and we will use the properties of that internal structure to define an appropriate quotient congruence.

One of the limitations of traces and the partial orders they generate is that neither traces nor partial orders can model the "not later than" relationship [9]. If an event $a$ is performed "not later than" an event $b$, and let the step $\{a, b\}$ model the simultaneous performance of $a$ and $b$, then this "not later than" relationship can be modelled by the following set of two step sequences $s = \{\{a\}\{b\}, \{a, b\}\}$. But the set $s$ cannot be represented by any trace. The problem is that the trace independency relation is symmetric, while the "not later than" relationship is not, in general, symmetric.

[⋆] Partially supported by NSERC grant of Canada.
[⋆⋆] Partially supported by Ontario Graduate Scholarship.

Latex und den article style benutzen!

Frei verfügbare Versionen erhaltet Ihr z.B unter
http://www.texshop.org
http://www.latex-project.org/
http://miktex.org/
http://www.tug.org/texlive/

**Empfehlung:** Tetex (oder Livetex)
Enthalten in jeder Distribution oder für Nicht-Distributionen als integrierte Systeme erhältlich:
Windows: http://www.cygwin.com
Mac OSX: http://fink.sf.net
(Jeweils packages: tetex, (x)emacs, auctex)

Latex ist auch an der Uni installiert und sollte dort sofort laufen.
bibtex ermöglicht die Verwendung von separaten Literaturdateien.
(In dem Text selbst sind natürlich die entsprechenden Literatureinträge möglichst mit Seitenangaben (bei allem
was länger als 20 Seiten ist) anzugeben.

Nur Webseiten als Referenzen reichen nicht aus.

Unter
http://www.springer.com/comp/lncs/authors.htm ...
findet Ihr eine Vorlage für das Erstellen von wissenschaftlichen Beiträgen mit Hilfe von Latex.

Achtung: ftp://ftp.springer.de/pub/tex/latex/llncs/lat ...
ist der direkte Zugriff auf die llncs-Version.

Beispiele für einen guten Aufbau findet Ihr unter den tausenden von Artikeln, die Ihr bei Springer findet!

Die Formatierungsvorlage liefert aber einen guten Rahmen für das Layout, Schriftgröße, etc.
Insbesondere ergibt sich aber so eine Art Maß für den Umfang der Ausarbeitung.

Bitte eine Rechtschreibprüfung verwenden.
Texte mit offensichtlichen Fehlern in Ausarbeitungen werden zurückgegeben.

Tipp bei der Latex-Benutzung:
latex bibtex latex latex, erst dann sind auch die Literatureinträge auch korrekt eingebunden.

# Themen

1. Ersetzungskalküle
   (Teile aus Baader/Nipkow: Term Rewriting and All That)
   z.B. Termination

# Themen

2. Petrinetze und algebraische Spezifikation
   (Reisig und weitere)

```
@Article{Reisig1991_tcs,
author = {Wolfgang Reisig},
title = {{Petri nets and algebraic specifications}},
journal = {Theoretical Computer Science},
year = 1991,
volume = 80,
number = 1,
pages = {1-34},
month = mar
}
```

```
@InProceedings{ KindlerR1997_lncs1337,
author = {Ekkard Kindler and Wolfgang Reisig},
title = {{Verification of Distributed Algorithms with Algebraic Petri Nets}},
editor = {Christian Freksa and Matthias Jantzen and Rüdiger Valk},
```

# Themen

3. Fixpunkttheorie

# Themen

4. Semiotik

# Themen

5. Temporale Logik
===============

Die Paper, in denen dies das erste Mal wirklich dargelegt wird:

* Pnueli, A. "The Temporal Logic of Programs." 18th Ann. Symp. Foundations of Computer Science, 1977, IEEE, pp. 46-57.
* Pnueli, A. "The Temporal Semantics of Concurrent Programs." Theoretical Computer Science 13 (1981), pp. 45-60 (Bzw. LNCS 70, 1979)

Literatur dazu in Buchform:

* Kr\"oger, F. and Merz, S. "Temporal Logic and State Systems", Springer-Verlag, Berlin, 2008
* Huth, M. and Ryan, M. D. "Logic in Computer Science: Modelling and Reasoning about Systems", 2nd ed. Cambridge University Press, Cambridge, UK, 2004.

# Themen

6. Model Checking
==============

Die Paper, in denen die Idee das erste Mal ausgefuehrt wurde:
* Clarke, E. M. and Emerson, E. A. "Design and synthesis of synchronization skeletons for branching time temporal logic." In Workshop Logic of Programs (Yorktown Heights N.Y., 1981), D. Kozen, ed., vol. 131 of Lecture Notes in Computer Science, Springer, 1982, pp. 52-71
* Queille, J. P. and Sifakis, J. "Specification and verification of concurrent systems in Cesar." In 5th Intl. Symp. Programming (Torino, Italy, 1981), vol. 137 of Lecture Notes in Computer Science, Springer, 1982, pp. 337-351.

Zwei gute Einfuehrungen (und Rueckblicke) dazu:
* Clarke, E. M.: "The Birth of Model Checking." In "25 Years of Model Checking", Grumberg, O. and Veith, H. (Eds.), vol. 5000 of Lecture Notes in Computer Science, Springer, 2008, pp. 1-26
* Emerson, E. A.: "The Beginning of Model Checking: A Personal Perspective." In "25 Years of Model Checking", Grumberg, O. and Veith, H. (Eds.), vol. 5000 of Lecture Notes in Computer Science, Springer, 2008, pp. 27-45

Literatur dazu in Buchform:
* Clarke, E. M., Grumberg, O., and Peled, D. "Model Checking", MIT Press, Cambridge, Mass., 1999
* Huth, M. and Ryan, M. D. "Logic in Computer Science: Modelling and Reasoning about Systems", 2nd ed. Cambridge University Press, Cambridge, UK, 2004.
* Baier, C. and Katoen J.P. "Principles of Model Checking", MIT Press, Cambridge, Mass., 2008

# Themen

7. Model Checking (SPIN)
========================
Paper zu einem der zur Zeit meist genutzten Model Checker:
* Holzmann, G. J. "The model checker SPIN", IEEE Transactions on
  Software Engineering, 23(5):279-295, 1997.
* Holzmann, G. J., Najm, E., Serhrouchini, A. "SPIN model checking:
  an introduction. International Journal on Software Tools for
  Technology Transfer, 2(4):321-327, 2000

Literatur dazu in Buchform:
* Ben-Ari, M. "Principles of the Spin Model Checker", Springer, 2008.
* Holzmann, G. J. "The SPIN Model Checker: Primer and Reference
  Manual", Addison-Wesley, 2003.
* s.o. bei Model Checking

# Themen

8. Theorem Proving
===============

 * Paulson, L. C., "Designing a Theorem Prover", Informal Publication, 1990.
   (Ist auch als Kapitel im 'Handbook of Logic in Computer Science (Vol. 2)'
   zu finden.)

# Themen

9. Theorem Proving (Isabell)
============================
 * Meikle, L. I., Fleuriot, J. D. "Formalizing Hilbert's Grundlagen
   in Isabelle/Isar", In Theorem Proving in Higher Order Logics, 16th
   International Conference, TPHOLs 2003, Rom, Italy, September 8-12,
   2003, Proceedings, LNCS 2578, Springer, 2003, pp. 319-334.
 * Manual und Diss zu Isabell im Ornder Theorem_Proving/Isabell

# Themen

10. Algebraic Specification
=======================

* Ehrig, H., Mahr, B., Classen, I., Orejas, F. "Introduction to Algebraic Specification. Part 1: Formal Methods for Software Development", The Computer Journal, 35(5), 1992, pp. 460-467.
* Ehrig, H., Mahr, B., Orejas, F. "Introduction to Algebraic Specification. Part 2: From Classical View to Foundations of System Specifications", The Computer Journal, 35(5), 1992, pp. 468-477.

Literatur dazu in Buchform:

* Ehrich, H.-D., Gogolla, M., Lipeck, U.W. "Algebraische Spezifikation abstrakter Datentypen", Teubner 1989
* Ehrig, H., Mahr, B., Cornelius, F., Gro{\ss}e-Rhode, M., Zeitz, P. "Mathematisch-strukturelle Grundlagen der Informatik", Springer, 2001.
* Ehrig, H., Mahr, B. "Fundamentals of Algebraic Specification 1: Equations and Initial Semantics, volume 6 of EATCS Monographs on Theoretical Computer Science, Springer, 1985.
* Ehrig, H., Mahr, B. "Fundamentals of Algebraic Specification 2: Module Specifications and Constrants, volume 21 of EATCS Monographs on Theoretical Computer Science, Springer, 1990.

# Themen

13. Algebraic Graph Transformation
==================================

* Baresi, L., Heckel, R. "Tutorial Introduction to Graph Transformation: A Software Engineering Perspective", In "Graph Transformation. First International Conference, ICGT 2002, Barcelona, Spain, October 7-12, 2002, Proceedings", LNCS 2505, Corradini, A., Ehrig, H., Kreowski, H.-J., Rozenberg, G. (Eds.), Springer, 2002.

Literatur dazu in Buchform:

* Ehrig, H., Ehrig, K., Prange, U., Taentzer, G. "Fundamentals of Algebraic Graph Transformation", Springer, 2006.

# Themen

13. Linear Logic
=============
 * Lincoln, P. "Linear Logic", SIGACT 1992.
 * Girard, J.-Y. "Linear logic : Its syntax and semantics." Advances in Linear Logic, edited by J.-Y. Girard, Y. Lafont & L. Regnier, London Mathematical Society Lecture Note Series 222, Cambridge University Press, 1--42. 1995.
 * Girard, J.-Y. "Linear Logic", Theoretical Computer Science, 50:1-102, 1987

14. Petri Nets are Monoids  / MAUDE
=============
 * J. Meseguer, U. Montanari

Information and Computation 88(1990)105-155