Graphtransformationstag 2019

Freitag, den 12. April 2019

Berthold Hoffmann

Fachbereich 3 – Informatik, Universität Bremen
Büro: Mehrzweck-Hochhaus, Raum MZH 3250
Bibliotheksstraße 1, 28359 Bremen

Vorträge

(Die Präsentationen verbergen sich hinter den Links.)

- Annegret Habel, Christian Sandmann, Oldenburg:
  Rule-Based Repair for Railroad Control
- Berthold Hoffmann, Mark Minas, München:
  Efficient Generalized Predictive Shift-Reduce Parsing for Hyperedge Replacement Graph Grammars
- Berthold Hoffmann, Bremen:
  Preisfragen(Siehe unten)
- Hans-Jörg Kreowski, Sabine Kuske, Aaron Lye, Bremen:
  Fusion Grammars: A Novel Approach to the Generation of Hypergraph Languages
- Michael Löwe, Hannover:
  Double-Pushout Rewriting in Context (or how to tame the AGREE-tiger)
- Graham Campbell, Brian Courtehoute, Detlef Plump, York:
  Linear-time Graph Algorithms in GP 2
- Arend Rensink, Enschede:
  What is Software Change?
- Hans-Jörg Kreowski, Bremen:
  Zwei Gedichte (Siehe unten)

Hans-Jörg Kreowski, Bremen:
Zwei Gedichte

Attributes in Berthold's Scientific Work

predictive efficient cleaned predictive
deterministic contextual predictive
contextual contextual contextual adaptive
adaptive conditional adaptive conditional adaptive adaptive
shaped generic adaptive structural adaptive
rule-based parallel hierarchical context-exploiting nested
context-exploiting nested hierarchical hierarchical
hierarchical nested visual hierarchical
simple rule-based efficient abstract
graph-theoretical mathematical automatic extended

studentisch generisch visuell erweitert

Berthold's Coauthors

FMMFMFMFFME CCCFFMMFDM JYTMFADFADCT
MLFDMNMDJT ERDDMNDWDRMA DMDMDMDPGMMDP
RPSMMMMMGA HSDAGBDDPHM RUBAIGHIIHHIHI

Berthold Hoffmann, Bremen:
Preisfragen

- Wo werden *Graphsprachen* benutzt?
- Wo werden *Graph-Grammatiken* benutzt?
- Wo wird *Graph-Parsieren* benutzt?
- Wo könnte GRAPPA benutzt werden?

Preise

– Jedem, der uns eine interessante *Graphsprache* und/oder eine interessante *Graph-Grammatik* nennt, spendieren wir eine Grappa![1]
– Jeder, der eine interessante (kontextuelle) Hyperkantenersetzungs-Grammatik mit GRAPPA bearbeitet, bekommt zwei Grappa!

Antworten

Drei Anwendungsgebiete, die allerdings alle schon bekannt sind, wurden angesprochen.

Natürliche Sprachverarbeitung (NLP)

In der Gruppe um Kevin Knight an der *University of Southern California* in Los Angeles werden Hykerkanten-Ersetzungsgrammatiken benutzt, um "*abstract meaning representations*" (AMR) für Sätze in natürlicher Sprache zu definieren und zu parsieren.[2] In der Folge wurde von der Gruppe ein HR-Parser namens *Bolinas* entwickelt [4]. Auch in der Gruppe von Alexander Koller wurde "der effizienteste Parser der Welt" für eine ähnliche Beschreibungssprache entwickelt [7]. Die Arbeit [6] hat gezeigt, dass auch CHR-Grammatiken für die Definition von AMR nützlich sein können.

Auf der AMR-Seite werden Corpora von mit AMR "*getaggten*" Sätzen zur Verfügung gestellt und auch maschinell erlernte Grammtiken, die Sätze mit AMR "*taggen*" können. Diese Grammatikren sind *groß* bis **riesig** und *hochgradig mehrdeutig*, weshalb es keine PTD- und PSR-Parser für sie geben kann, wohl aber GPSR-Parser.

Eine unserer nächsten Arbeiten wird sein, GPSR-Parser für AMR zu bauen und sie mit den oben genannten Parsierern zu vergleichen.

[1] Sprachen, die schon in [8,5] oder in [2] als Beispiele auftauchen, gelten nicht als *interessant*!
[2] Siehe https://amr.isi.edu/.

Modelltransformation

Bei der Modelltransformation geht es darum, die Bezüge zwischen zwei Software-Modelle zu spezifizieren. Dies wird u.a. mit Tripel-Graph-Grammatiken (TGG) getan. Diese Grammatiken definieren zwei Modelle und deren Relation als Graphen.

Eine wichtige Anwendungen von TGGs sind:

- das Ableiten von Vorwärts- und Rückwärts-Transformationen von einem Modell ins andere.
- Das automatische inkrementelle Synchronisieren der Modelle, wenn eines davon geändert wird.

Als Grammatiken werden zumeist monotone reduktive Spezifikationen ohne Nichtterminale benutzt. Die Graphen repräsentieren Objekte, weshalb die Knoten Typen, Attribute und Untertypen besitzen und die Kanten Multiplizitäten haben.

Shape Analysis

Bei der "Gestalt-Analyse" geht es darum nachzuweisen, dass ein (typischerweise imperatives) Programm P die Integrität seiner Datenstrukturen (als doppelt verkettete Liste, Blatt-verbundener Baum o.ä.) bewahrt.

[2] haben reduktive Spezifikationen dafür benutzt, die Gestalt von Datenstrukturen mit Zeigern zu spezifizieren und nachzuweisen, dass die Operationen von C-Programmem diese Gestalt bewahren.

In der Gruppe von Thomas Noll an der RWTH Aachen werden (erweiterte) HR-Grammatiken benutzt, um die Gestalt von Datenstrukturen zu definieren [9,1].

In [3] haben Heike Werheim u.a. HR-Grammatiken benutzt, um zu zeigen, das Modelltransformationen die Gestalt der Modelle bewahren.

Literatur

1. H. Arndt, C. Jansen, C. Matheja, and T. Noll. Graph-based shape analysis beyond context-freeness. In E. B. Johnsen and I. Schaefer, editors, *Software Engineering and Formal Methods - 16th International Conference, SEFM 2018, Held as Part of STAF 2018, Toulouse, France, June 27-29, 2018, Proceedings*, volume 10886 of *Lecture Notes in Computer Science*, pages 271–286. Springer, 2018.
2. A. Bakewell, D. Plump, and C. Runciman. Specifying pointer structures by graph reduction. Technical report, Department of Computer Science, University of York, Nov. 2002.
3. G. Besova, D. Steenken, and H. Wehrheim. Grammar-based model transformations. In M. Ganzha, L. A. Maciaszek, and M. Paprzycki, editors, *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, September 7-10, 2014.*, pages 1601–1610, 2014.

4. D. Chiang, J. Andreas, D. Bauer, K. M. Hermann, B. Jones, and K. Knight. Parsing graphs with hyperedge replacement grammars. In *Proc. 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 924–932, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.
5. F. Drewes, A. Habel, and H.-J. Kreowski. Hyperedge replacement graph grammars. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation. Vol. I: Foundations*, chapter 2, pages 95–162. World Scientific, Singapore, 1997.
6. F. Drewes and A. Jonsson. Contextual hyperedge replacement grammars for abstract meaning representations. In *13th Intl. Workshop on Tree-Adjoining Grammar and Related Formalisms (TAG+13)*, pages 102–111, 2017.
7. J. Groschwitz, A. Koller, and C. Teichmann. Graph parsing with s-graph grammars. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1481–1490. The Association for Computer Linguistics, 2015.
8. A. Habel. *Hyperedge Replacement: Grammars and Languages.* Number 643 in Lecture Notes in Computer Science. Springer, 1992.
9. J. Heinen, C. Jansen, J. Katoen, and T. Noll. Verifying pointer programs using graph grammars. *Sci. Comput. Program.*, 97:157–162, 2015.