

Lehrstuhl für Informatik 2 (Programmiersprachen und Programmiermethodik)

Anschrift: Martensstr. 3, 91058 Erlangen

Tel.: +49.9131.85.27621

Fax.: +49.9131.85.28809

E-Mail: info@i2.informatik.uni-erlangen.de

Leitung:

Prof. Dr. Michael Philippsen

Emeritus:

Prof. em. Dr. rer. nat. Hans Jürgen Schneider

Sekretariat:

Agnes Brütting

Elfriede Schörmal

Wiss. Mitarbeiter:

Dr. -Ing. Ingrid Fischer

Dipl.-Inf. Michael Klemm

Dipl.-Inf. Thorsten Meinl

Dipl.-Inf. Dominic Schell

Ronald Veldema, Ph.D.

PD Dr.-Ing. habil. Peter Wilke

Gäste/Stipendiaten:

Franz Forman

Dipl.-Ing. (FH) Bernd Hardung

Dr. Klaus Kreisel

PD Dr.-Ing. Gabriella Kókai

Dr. Hermann Puhlmann

Dipl.-Inf. Szilvia Zvada

Externes Lehrpersonal:

Dipl.-Ing. Hans Holm Frühauf

Dr.-Ing. Jörg Nilson

1 Einführung

Im Mittelpunkt der Forschungsarbeiten des Lehrstuhls stehen parallele und verteilte Systeme und deren Programmierung sowie Programmiersysteme für eingebettete und mobile Systeme. Software (und deren Erstellung) für solche Systeme sollte nicht komplexer aber genauso portabel, wartbar und robust sein, wie heute für Einprozessorsysteme und Arbeitsplatzrechner. Langfristiges Ziel ist es, Anwendungen die verfügbare Rechen- und Kommunikationsleistung möglichst ungebremst zur Verfügung zu stellen bzw. aus sehr begrenzten Systemen ein Maximum an Möglichkeiten herauszuholen. Um diese Vision zu verwirklichen, bedarf es bei sich fortentwickelnden technologischen Grundlagen (von Parallelrechnern über verteilte Systeme bis hin zu allgegenwärtigem eingebettetem Rechnen der Zukunft) erstklassiger technischer Einzelbeiträge in mehreren Teilgebieten der Informatik (z.B. Entwurf von Programmiersprachen, Optimierungstechniken für Übersetzer, Software-Architekturen, Werkzeuge zum Software-Engineering, Laufzeitsystem, Betriebssystemintegration, Entwurfsmuster, Parallelität, etc.) sowie einer gebietsübergreifenden Arbeitsweise.

2 Forschungsschwerpunkte

Die laufenden Arbeiten des Lehrstuhls gliedern sich in vier ineinander greifende Puzzle-Steine/Arbeitsgruppen. Durch gegenseitiges Nehmen und Geben entstehen Synergie-Effekte, die die Leistungen der einzelnen Gruppen über sich hinaus wachsen lassen. Die **Übersetzerbau-Gruppe** arbeitet an neuen Analyse- und Optimierungstechniken, vor allem im Bereich der Laufzeitübersetzung. Fortschritte in diesen Bereichen werden von den Arbeitsgruppen **Cluster-Computing** und **Eingebettete Systeme** dringend zur Lösung ihrer jeweiligen Forschungsarbeiten benötigt. Andererseits stellen gerade diese Forschungsarbeiten ein umfangreiches Evaluationsfeld für die Übersetzerbaugruppe dar. Im Bereich der eingebetteten Systeme wird zur Zeit an der Minimierung des erforderlichen Speicherbedarfs, der besseren Ausnutzung der Register in statischen Betriebssystemen sowie der Energieeinsparung gearbeitet. Im Bereich des Cluster-Computings steht eine Anwendung aus der Medizin im Mittelpunkt: das Entdecken von gemeinsamen Molekülfragmenten. Die Lösung dieses Anwendungsproblems (mit Hilfe von weiterzuentwickelnden Data Mining-Techniken) erfordert so viel Rechenleistung, dass herkömmliche Programmier-Techniken für Rechnerbündel nicht ausreichen. Auch übliche Rechnerbündel reichen nicht - aus heterogenen Einheiten zusammengesetzte Grids sind erforderlich. Dies stellt harte Forderungen an die Übersetzerbaugruppe. Das vierte Puzzle-Teil, **Soft-Computing**, steht in Interaktion sowohl mit dem Arbeitsbereich Eingebettete Systeme als auch dem Bereich Cluster-Computing. Derzeit werden genetische Algorithmen in (eingebettete) Hardware umgesetzt. Geplant ist, genetische Algorithmen auf Rechnerbündeln zu implementieren, um die zur Verfügung stehende Rechenleistung zum schnelleren Durchforsten des Suchraums einzusetzen. Ein Arbeitsgebiet sind die Zeitplanungsverfahren - bekanntlich kann ein guter Stundenplan, der alle Randbedingungen erfüllt, nur mit heuristischer Suche gefunden werden. Genetische Algorithmen finden überraschend gute Lösungen in kurzer Zeit. Durch Verwendung von Rechnerbündeln ist ein weiterer Fortschritt zu erwarten, wenn der Übersetzerbau es ermöglicht, die nominelle Rechenleistung möglichst verlustfrei bis zur Anwendung zu bringen.

Graphische Darstellung (<http://www2.informatik.uni-erlangen.de/Forschung/puzzle.png>)

3 Forschungsprojekte

3.1 Automatische Analyse natürlicher Sprache

Projektleitung:

Dr.-Ing. Ingrid Fischer

Beteiligte:

Dr.-Ing. Ricarda Dormeyer

Laufzeit: 1.1.1999 - 31.12.2004

Kontakt:

Dr.-Ing. Ingrid Fischer

Tel.: +49.9131.85.27830, Fax: +49.9131.85.28809,

E-Mail: idfische@informatik.uni-erlangen.de

Die interessantesten und schwierigsten idiomatischen Beispiele enthalten lexikalische und syntaktische Modifikationen (z.B. "Er schloss die eiternde Wunde, in die er Salz gestreut hat"). Aber auch unveränderliche Idiome (z.B. "Er hat ins Gras gebissen") dürfen bei der automatischen Sprachanalyse nicht vernachlässigt werden. Der von uns verfolgte Ansatz zur Idiomanalyse beruht auf zwei Vermutungen. Erstens können Idiome analysierbar sein, das heißt, die einzelnen Komponenten können bei der Analyse des Idioms eine Rolle spielen, müssen es aber nicht. Zweitens spielen bei vielen Idiomen Metaphern eine aktive Rolle im Verstehensprozess, besonders wenn die Idiome analysierbar und modifizierbar sind. Diese Vermutungen werden im Rahmen des Projekts an verschiedenen Sprachverarbeitungssystemen getestet.

- Diskontinuierliche Konstituenten stehen in einem Satz nicht nebeneinander, obwohl sie eigentlich zusammengehören. In "Der Student hat ein Beispiel gefunden, das das Problem illustriert." gehört der Relativsatz "das das Problem illustriert" zu "ein Beispiel". Ebenso gehören der finite und der infinite Teil des Verbs zusammen. Dies führt zu sich überkreuzenden Kanten im Ableitungsbaum, was mit den üblichen kontextfreien Grammatiken nicht mehr modelliert werden kann. Zeichenketten generierende Hypergraphgrammatiken lösen dieses Problem auf elegante Weise. Um mit Hypergraphgrammatiken arbeiten zu können, entwickeln wir einen Parser, der dem allgemeinen Earley-Parser ähnlich ist.
- In der im Jahr 2004 fertig gestellten Dissertation von Ricarda Dormeyer wurden verschiedene Arten von Dependenzparsern untersucht und ein neuartiger Parser entwickelt. Für diesen Parser wurden Grammatiken für das Japanische und das Ungarische erstellt. Beides sind keine indo-europäischen Sprachen. Japanisch eignet sich sehr gut für die Verarbeitung mit Dependenzparsern, es zeichnet sich vor allem durch eine relativ freie Wortstellung aus, bei der das Verb am Ende des Satzes steht. Es wurden zwei Versionen der Grammatik entwickelt. In der ersten Variante arbeitet der Parser auf selbst entwickelten Worttrennungsregeln und einer eigenen Definition des kleinst möglichen Satzglieds. Die zweite Version wurde an den bestehen morphologischen Sequencer JUMAN angepasst.

Anders ist die Situation für das Ungarische. In der Literatur existieren kaum Parser für das Ungarische und nur eine weitere Abhängigkeitsgrammatik für das Ungarische wurde gefunden. Es zeigte sich allerdings, dass die freie Wortstellung und die diskontinuierlichen Konstituenten problemlos mit dem Parser modelliert werden können. Problematisch bleibt aber die reichhaltige Wortbildung des Ungarischen. Die Anbindung einer Morphologie ist in dieser Sprache unumgänglich.

3.2 Cluster Computing

Projektleitung:

Prof. Dr. Michael Philippsen

Beteiligte:

Ronald Veldema, Ph.D.

Dipl.-Inf. Michael Klemm

Laufzeit: 1.4.2002 - 31.12.2006

Kontakt:

Prof. Dr. Michael Philippsen

Tel.: +49.9131.85-27625, Fax: +49.9131.85.28809,

E-Mail: philippsen@informatik.uni-erlangen.de

Das Programmieren von Rechnerbündeln befindet sich derzeit noch in einer sehr maschinennahen Phase. Als besonders dringend erscheint die Behandlung der non-uniformen Speicherzugriffshierarchie, die in verteilten Systemen neben Registern, Cache und Hauptspeicher um eine weitere Leistungsebene vertieft ist. Wie im folgenden skizziert, erfordern gute Lösungen Arbeiten auf allen Systemschichten.

Betriebssysteminteraktion: Welche Leistungsengpässe bestehen und wie können diese beseitigt werden? Wie kann der Kommunikationskanal - auch durch das Betriebssystem hindurch - weiter verschlankt werden (sowohl für Methodenaufrufe, die in der Regel gute Latenzzeiten verlangen, als auch für Datenstromübertragungen, bei denen es auf hohe Bandbreite ankommt)?

Übersetzerunterstützung, Laufzeitsystem: Können verteilte Systeme der Tradition von Einprozessorsystemen folgend auch weiterhin mit der Illusion eines transparenten Zugriffs programmiert werden? Kann man durch statische Analyse nebenläufigen objekt-orientierten Codes Informationen über bestehende Lokalitätsbeziehungen extrahieren? Welche Optimierungen sind dann möglich? Profitieren statische Analyse und Laufzeitmechanismen voneinander? Wie können durch Programmanalyse Pre-Fetch- und Post-Store-Kommandos im generierten Code erzeugt werden, durch die Kommunikation und Berechnung überdeckt und dadurch Wartezeiten vermieden werden?

Programmiersprache, Systementwurf: Programmierer haben in der Regel das anwendungsspezifische Hintergrundwissen über bestehende Lokalitätszusammenhänge. Um guten Code zu erzeugen, muss der Übersetzer auch über dieses Wissen verfügen. Im Idealfall sollte der Übersetzer durch statische und dynamische Analyse in der Lage sein, das Wissen aus dem Programm-Code zu extrahieren - dies ist außer in datenparallelen Sprachen bisher leider kaum möglich. Der alternative Ansatz, besteht im Prinzip darin, dass der Programmierer sein Wissen

explizit, z.B. mit Hilfe von leicht zu entdeckenden Pragmas, an den Übersetzer weitergibt. Offen ist, wie man diese Wissensweitergabe sinnvoll ausgestaltet. Sinnvoll bedeutet dabei erstens, dass es für den Programmierer bequem bleiben muss, und zweitens, dass das codierte/entdeckte Wissen aussagekräftig genug sein muss, um vom Übersetzer oder dem Laufzeitsystem effizienzsteigernd genutzt werden zu können.

Noch sehr schlecht verstanden ist generell, wie man verteilte Systeme von vorneherein so entwirft, dass Lokalität per Design vorhanden ist.

2004 wurde verstärkt an den Fähigkeiten des am Lehrstuhl entwickelten Übersetzers in Hinsicht auf Stabilität und den Anforderungen im Rahmen von Grid-Computing gearbeitet.

- Eine neue Programmiersprachen-Erweiterungen ermöglicht es dem Programmierer, Beziehungen zwischen Objekten der Programmiersprache Java zu beschreiben. Davon ausgehend kann der Übersetzer die in Beziehung stehenden Objekte kombinieren, d.h. zu einem Objekt zusammenzufassen. Werden zwei Objekte sehr häufig zusammen referenziert und ist eine Übertragung beider Objekte über das Netzwerk nötig, spart die Vereinigung der Objekte die für die Übertragung des zweiten Objektes ansonsten nötige Latenzzeit.
- Eine automatische Code-Transformation kann beliebige Code-Sequenzen aus einem Java-Programm in separate Funktionen austrennen. Ziel der Transformation ist die Vermeidung von Latenzzeiten, die entstehen, wenn die Originalfunktion während der Ausführung entfernte Objekte über das Netzwerk anfordert. Anstelle großer Datenmengen über das Netzwerk zu übertragen, wird nur der Methodenaufruf und eine Antwort-Nachricht übertragen. Dies spart die andernfalls entstehenden Objektanforderungen.
- Es wurde ferner overhead-freies Checkpointing für heterogene Rechnerbündel entwickelt. Durch dieses Verfahren ist es möglich, einen Aktivitätsträger (Thread) bzw. Prozess auf einem Rechner einzufrieren, auf einen anderen Rechner zu verschieben und dort mit der Ausführung fortzufahren. Bei diesem Vorgang können sich die beteiligten Architekturen zwischen Quell- und Zielsystem unterscheiden.

3.3 Entwicklung und Anwendung von Evolutionären Algorithmen

Projektleitung:

PD Dr.-Ing. Gabriella Kókai

Beteiligte:

Hans Holm Frühauf

Martin Böhner

Tonia Christ

Beginn: 1.1.1999

Förderer:

Frauenförderung aus dem Hochschul- und Wissenschaftsprogramm (HWP)

Mitwirkende Institutionen:

Fraunhofer Institute Integrierte Schaltungen

Kontakt:

PD Dr.-Ing. Gabriella Kókai

Tel.: +49.9131.85.28996, Fax: +49.9131.85.28809,

E-Mail: kokai@informatik.uni-erlangen.de

Das genetisch-logische Programmiersystem GeLog kombiniert Methoden genetischer Algorithmen und induktiv logischer Programmierung. Mit dem automatischen Lernsystem können aus einer induktiv logisch formulierten Aufgabenstellung logische Programme erzeugt werden, die die gegebene Aufgabe lösen. Zum Lernen verwendet GeLog einen genetischen Algorithmus. GeLog verfügt über automatische Parameteradaptation und Metaevolution und vermeidet dadurch die bekannten Nachteile genetischer Algorithmen, wie die große Anzahl von Einstellmöglichkeiten für Evolutionsparameter sowie das Hängenbleiben in lokalen Optima. Das System besitzt eine graphische Benutzeroberfläche, die die Arbeit mit GeLog einfach macht und eine graphische Darstellung der Ergebnisse erlaubt.

Im Jahr 2004 wurde GeLog mit einigen aus der Standard-Machine-Learning-Datenbank ausgewählten Beispielen getestet. Dieser Test basierte auf der 10-crossover-Methode. Auf dieselben Testbeispiele wurden andere auf einer Kombination von EA und ILP basierende Systemen angewendet. Der Vergleich zeigt, dass GeLog den anderen Systemen überlegen ist, besonders wenn die Daten verrauscht sind.

Im Jahr 2004 gelang es uns erneut, genetische Algorithmen im Rahmen eines Kooperationsprojekts mit der Industrie einzusetzen. Wir entwickelten verschiedene Hardware-basierte Optimierer zur Echtzeitsteuerung von adaptiven Antennen. Moderne, intelligente Antennen erzeugen räumlich eng begrenzte Strahlenbündel, die sich elektronisch durch Einstellungen von einer Reihe von Hardware-Parametern während des Betriebs schwenken lassen. Diese Einstellungen haben erheblichen Einfluss auf die Empfangsqualität der Signale mobiler Sender innerhalb der Reichweite der Gruppenantenne.

Ziel der Optimierung war eine bestmögliche (messbare) Empfangsqualität gewünschter mobiler Teilnehmer und eine maximale Unterdrückung störender bzw. unerwünschter Teilnehmer. Aufgrund der hohen Mobilität der Sender ist es notwendig, die Optimierungsstrategie als hochparallele Hardware zu entwerfen, um die strengen Echtzeitbedingungen zur Nachführung der Richtcharakteristik der Antenne zu erfüllen. Innerhalb der Untersuchungen wurden die wichtigsten Verfahren des Soft-Computing auf ihre Eignung evaluiert und die erfolgreichsten implementiert. Zusätzlich zu den Algorithmen wurden alternative Hardwarestrukturen für ASIC und FPGAs entworfen und validiert.

3.4 Graphen und Graphtransformationen

Projektleitung:

Prof. em. Dr. Hans Jürgen Schneider

Beteiligte:

Dr.-Ing. Ingrid Fischer

Laufzeit: 1.1.1995 - 1.10.2006

Kontakt:

Prof. em. Dr. Hans Jürgen Schneider

Tel.: +49.9131.85.27620, Fax: +49.9131.85.28809,

E-Mail: schneider@informatik.uni-erlangen.de

Graphen werden an vielen Stellen als intuitives Hilfsmittel zur Verdeutlichung komplizierter Sachverhalte verwendet. Außerhalb der Informatik trifft dies z.B. auf die Biologie oder Chemie zu, wo Moleküle graphisch modelliert werden. Innerhalb der Informatik werden Daten- bzw. Kontrollflussdiagramme, als Entity-Relationship-Diagramme oder Petri-Netze, zur Visualisierung sowohl von Software- als auch von Hardware-Architekturen häufig verwendet. Graphgrammatiken und Graphtransformationen kombinieren Ideen aus den Bereichen Graphentheorie, Algebra, Logik und Kategorientheorie, um Veränderungen an Graphen formal zu beschreiben.

Am Lehrstuhl für Programmiersysteme werden Graphen und Graphtransformationen in verschiedenen Bereichen eingesetzt.

- Die zugrundeliegende Theorie ist ein attraktives Hilfsmittel, äußerst unterschiedliche Strukturen in einer einheitlichen Weise zu beschreiben, z.B. die unterschiedlichen Modelle für asynchrone Prozesse: Petri-Netze basieren auf "gewöhnlichen" Graphen, Statecharts verwenden hierarchische Graphen, die parallele logische Programmierung kann mit Hilfe sogenannter Dschungel graphentheoretisch interpretiert werden, und die Aktorsysteme lassen sich als Graphen darstellen, deren Markierungsalphabet eine Menge von Termgraphen ist.
- Im Doppel-Pushout-Ansatz wird die linke Seite einer Produktion üblicherweise als injektiv vorausgesetzt, da der nichtinjektive Fall zu mehrdeutigen Ergebnissen führt. Die Behandlung von Markierungsänderungen mit Hilfe eines strukturierten Alphabetes führt aber auch im injektiven Fall zu Mehrdeutigkeiten. Die meisten Autoren lösen dieses Problem, indem sie nur die Transformation des zugrundeliegenden Graphen kategoriell, die Markierungsänderung aber separat behandeln. Die Arbeiten im Jahr 2004 dienten einer kategoriellen Charakterisierung aller Lösungen, wobei auf ein Ergebnis von Kreowski und Ehrig aus dem Jahre 1976 zurückgegriffen wurde. Es konnte gezeigt werden, dass damit wichtige Anwendungsgebiete (Petri-Netze, Termgraphen, Datenbanken) abgedeckt werden.

- Mit Hilfe von Graphtransformationen werden nicht nur allgemeine Graphen generiert und analysiert, auch auf Zeichenketten basierende Sprachen und ihre Probleme können bearbeitet werden. Ein typisches Beispiel ist die kontextsensitive Sprache $a^n b^n c^n$. Sie kann mit kontextfreien Ersetzungsgrammatiken für Hyperkanten modelliert werden. Um mit diesen Hypergraphgrammatiken arbeiten zu können, wurde ein Parser entwickelt, der dem allgemeinen Earley-Parser ähnlich ist. Im Vergleich zum ursprünglichen Earley-Algorithmus wurde das Konzept der inaktiven Elemente erweitert. Ein Element wird nicht erst inaktiv, wenn es alle seine Komponenten gefunden hat, sondern kann auch an anderen Stellen inaktiv werden und so die Weiterverarbeitung ermöglichen. Die neue Variante des *Predictor* sorgt dafür, dass zwischenzeitlich inaktive Kanten, wieder zu aktiven umgewandelt werden.

3.5 International Collegiate Programming Contest an der FAU

Projektleitung:

Prof. Dr. Michael Philippsen

Beteiligte:

Dr.-Ing. Ingrid Fischer

Dipl.-Inf. Thorsten Meinel

Tilmann Spiegelhauer

Christian Riess

Beginn: 1.11.2002

Die Association for Computing Machinery (ACM) richtet seit vielen Jahren den **International Collegiate Programming Contest (ICPC)** aus. Dabei sollen Teams aus je drei Studenten in fünf Stunden neun Programmieraufgaben lösen. Als Erschwernis kommt hinzu, dass nur ein Computer pro Gruppe zur Verfügung steht. Die Aufgaben erfordern solide Kenntnisse von Algorithmen aus allen Gebieten der Informatik, wie z.B. Graphen, Kombinatorik, Zeichenketten und Geometrie.

Der ICPC wird jedes Jahr in drei Stufen ausgetragen. Zuerst werden innerhalb der Universitäten in lokalen Ausscheidungen die maximal drei Teams bestimmt, die dann zu den regionalen Wettbewerben entsandt werden. Erlangen liegt im Einzugsbereich des **Southwestern European Regional Contest (SWERC)**, an dem u.a. auch Teams aus der Schweiz, Frankreich, Spanien und Portugal teilnehmen. Die Sieger aller regionalen Wettbewerbe der Welt (und einige Zweitplatzierte) erreichen die World Finals, die im Frühjahr des darauffolgenden Jahres stattfinden.

Seit 2002 nimmt die FAU wieder regelmäßig an den SWERCs teil und konnte bereits im ersten Jahr ein Team zu den World Finals 2003 nach Los Angeles entsenden. Auch in 2004 erreichte ein Team der FAU einen beachtlichen vierten Platz unter 55 weiteren Mannschaften.

Die Auswahl der Teams der FAU und die Vorbereitung auf den SWERC werden vom Lehrstuhl 2 in Zusammenarbeit mit Studenten durchgeführt. Dazu werden pro Jahr zwei lokale Wettbewerbe und mehrere Vorbereitungswettbewerbe veranstaltet. Außerdem gibt es im Sommersemester das Hauptseminar "Hallo Welt für Fortgeschrittene", das einen tieferen Einblick in alle möglichen Arten von Algorithmen bietet.

3.6 Optimierung der Funktionsverteilung im Fahrzeugnetzwerk

Projektleitung:

PD Dr.-Ing. Gabriella Kókai

Beteiligte:

Prof. Dr. Michael Philippsen
Dipl.-Ing. (FH) Bernd Hardung

Laufzeit: 1.1.2003 - 31.12.2006

Förderer:

Audi AG, Ingolstadt

Kontakt:

PD Dr.-Ing. Gabriella Kókai
Tel.: +49 .9131.85-28996, Fax: +49.9131.85.28809,
E-Mail: kokai@informatik.uni-erlangen.de

Moderne Automobile haben komplexe, verteilte Systeme von elektronischen Steuergeräten, die mittels verschiedener Bussysteme vernetzt sind. Bis zu 80 Netzknoten befinden sich in einem einzelnen Fahrzeug. Die Steuergeräte realisieren eine große Anzahl von vernetzten Funktionen, die aus Software- und Hardware-Komponenten zusammensetzen. In diesem Projekt werden Möglichkeiten untersucht, Funktionen bei der Erstellung der Systemarchitektur des Fahrzeugnetzwerks unterschiedlich zu plazieren, ohne das Ziel ihrer Wiederverwendung zu gefährden. Anschließend wird untersucht, nach welchen Kriterien Funktionen optimal in diesem Netzwerk verteilt werden können. Dazu werden verschiedene Verteilungskriterien wie z. B. Kosten oder Komplexität beleuchtet. Weiterhin wird eine Methode entwickelt, die es darauf aufbauend erlaubt, Funktionsverteilungen in gegebenen Netzwerktopologien zu bewerten und zu optimieren. Die Ergebnisse werden in einem Optimierungssystem implementiert und auf ihre Praxistauglichkeit hin überprüft.

2004 wurden zunächst Untersuchungen zur Wiederverwendung von Software-Komponenten in Fahrzeugnetzwerken durchgeführt. Zudem wurde begonnen, ein Optimierungsmodell für die Funktionsverteilung in Fahrzeugnetzwerken zu entwickeln. Verschiedene Kriterien und Randbedingungen wurden bereits untersucht und konnten berücksichtigt werden. Außerdem arbeiten wir an einem Optimierungs-Framework. Dieses soll die Validierung der theoretischen Überlegungen an praktischen Beispielen erlauben. Gleichzeitig liegt ein Fokus der Arbeit auf der Austauschbarkeit der Optimierungsalgorithmen.

3.7 Parallele Suche in Moleküldatenbanken

Projektleitung:

Prof. Dr. Michael Philippsen

Beteiligte:

Dr.-Ing. Ingrid Fischer

Dipl.-Inf. Thorsten Meinl

Stichwörter:

Fragmente; Moleküle; Suche; Datenbank; parallel; verteilt

Beginn: 1.4.2004

Förderer:

Bayerisch-Kalifornisches Hochschulzentrum

Das Forschungsprojekt **ParMol** befasst sich mit der verteilten Suche häufiger Fragmente in Moleküldatenbanken. In seinem Rahmen sollen neue Verfahren entwickelt werden, um in großen Datenbanken interessante Zusammenhänge zwischen chemischen Strukturen zu finden. Diese Zusammenhänge sollen als Molekülfragmente, d.h. in Form von Zusammenhangsgraphen dargestellt werden. Ein Anwendungsgebiet ist z.B. die in vitro Vorhersage von toxischen Seiteneffekten von Medikamentenkandidaten schon vor in vivo oder klinischen Tests. Im Gegensatz zu bisher bekannten Verfahren sollen auch ähnliche Strukturen gefunden werden, die biologisch oder chemisch die gleiche Wirkung zeigen. Um den dadurch entstehenden deutlich höheren Rechenbedarf zu decken, werden die entwickelten Verfahren parallelisiert und so entworfen, dass sie sich auch für die Bearbeitung auf verteilten Ressourcen eignen.

Das Projekt wurde im April 2004 zusammen mit dem ALTANA Lehrstuhl für Bioinformatik und Information Mining an der Universität Konstanz ins Leben gerufen. Folgende Schwerpunkte wurden gesetzt:

- Die unscharfe Suche in Moleküldatenbanken wurde anhand von Kohlenstoffketten untersucht. Die Länge dieser Ketten hat keine Auswirkungen, verschieden lange Ketten wurden als gleich betrachtet. Die Grenzen können dabei vom Benutzer festgelegt werden.
- Zwei der wesentlichen Algorithmen zum Data Mining in Moleküldatenbanken wurden kombiniert. Die Algorithmen MoFa und FSG haben ihre Vor- und Nachteile in unterschiedlichen Bereichen. MoFa braucht sehr viel Speicher, FSG hat eine hohe Laufzeit. Wird die Suche mit FSG gespeichert und ab einer gewissen Molekülgröße an MoFa übertragen, entsteht ein hybrider Algorithmus, der besser als beide Einzelteile arbeitet.
- Die Größe der zu untersuchenden Datenbanken stellt viele Algorithmen vor Probleme. Deshalb wurden Pruningstrategien untersucht, mit deren Hilfe der Suchraum verkleinert wird.

3.8 SIGNAL - Sofortprogramm Informatik am Gymnasium - Nachqualifikation von Lehrkräften (ehemals NELLI)

Projektleitung:

Prof. Dr. Michael Philippsen

Beteiligte:

Franz Forman

Dr. Klaus Kreisel

Dr. Hermann Puhlmann

Laufzeit: 1.9.2002 - 31.8.2006

Förderer:

Bayerisches Staatsministerium für Unterricht und Kultus

Bayerisches Staatsministerium für Wissenschaft, Forschung und Kunst

Europäische Kommission

Mitwirkende Institutionen:

Ludwig-Maximilian-Universität München (Prof. Dr. Hofmann und Prof. Dr. Kröger)

Technische Universität München (Prof. Dr. Hubwieser)

Universität Passau (Prof. Dr. Freitag)

Kontakt:

Prof. Dr. Michael Philippsen

Tel.: +49.9131.85.27625, Fax: +49.9131.85.28809,

E-Mail: philippsen@informatik.uni-erlangen.de

Deutschland will im Bereich Informationsgesellschaft einen Spitzenplatz in Europa einnehmen. Um dieses Ziel zu erreichen, müssen junge Menschen bereits in der Schule eine fundierte Ausbildung in Informatik erhalten. Konsequenterweise führte Bayern im Schuljahr 2004/05 das Pflichtfach Informatik - zunächst als Teil des Faches "Natur und Technik" - an allen Gymnasien ein. Jedes der rund 400 bayerischen Gymnasien sollte ab diesem Zeitpunkt über mindestens eine Lehrkraft mit der Fakultas Informatik verfügen.

Der immense Bedarf an Lehrkräften kann nicht über die traditionelle Lehramtsausbildung an den Universitäten gedeckt werden. Aus diesem Grund wurde im Jahr 2000 das Projekt Nelli (Netzgestützter Lehrverbund zur Lehrerausbildung in Informatik) gestartet und ein elektronisches Angebot von speziell zugeschnittenen Studien- und Unterrichtsmaterialien entwickelt. Der Lehrstuhl hat hierzu das Algorithmik-Modul beigetragen. Im September 2001 nahmen bayernweit im Rahmen von Pilotkursen 75 Lehrkräfte das Studium der Informatik auf, davon 15 in Erlangen. Im Herbst 2003 haben diese Lehrerinnen und Lehrer die Erweiterungsprüfung in Informatik abgelegt.

Der erfolgreiche Verlauf der Pilotkurse ermutigte zum Folgeprojekt SIGNAL (Sofortprogramm Informatik am Gymnasium - Nachqualifikation von Lehrkräften). In den Jahren 2002 bis 2004

startete an der Universität Erlangen-Nürnberg jeweils ein zweijähriger Kurs mit rund 25 Teilnehmern.

Im ersten Kursjahr lernen die Teilnehmer überwiegend zu Hause mit elektronisch bereitgestellten Materialien. Sie geben über einen Gruppenarbeits-Server Übungen ab, die vom Tutor des Kurses korrigiert werden. Zum Erfahrungsaustausch und persönlichen Kontakt finden zudem monatliche Treffen statt. Im zweiten Kursjahr finden wöchentliche Präsenzveranstaltungen an der Universität statt. Dabei besuchen die Teilnehmer auch reguläre Veranstaltungen aus dem Angebot für das Informatikstudium.

Darüberhinaus wird seit Herbst 2003 das reguläre Lehramtsstudium Informatik für alle Schulformen angeboten; dazu wird eine Professur **Didaktik der Informatik** am Lehrstuhl eingerichtet werden. Siehe Informationsseite (<http://www2.informatik.uni-erlangen.de/~philippsen/lehramt/>)

Im Vergleich mit den anderen Bundesländern wird Bayern nach Abschluss dieses Programms einen Spitzenplatz in Bezug auf den Stellenwert der Informatik an den Schulen einnehmen.

3.9 Übersetzung für eingebettete Systeme

Projektleitung:

Prof. Dr. Michael Philippsen

Beteiligte:

Dr.-Ing. Volker Barthelmann

Dipl.-Inf. Dominic Schell

Ronald Veldema, Ph.D.

Laufzeit: 1.1.2001 - 31.12.2006

Förderer:

Deutsche Forschungsgemeinschaft

3Soft GmbH

Kontakt:

Prof. Dr. Michael Philippsen

Tel.: +49.9131.85.27625, Fax: +49.9131.85.28809,

E-Mail: philippsen@informatik.uni-erlangen.de

Ob im Auto oder zur Steuerung von Maschinen, in vielen elektronischen Geräten sind eingebettete Systeme ein unverzichtbarer Bestandteil. Mit zunehmender Komplexität und Größe der Anwendungen ergibt sich die Anforderung, Software für eingebettete Systeme in modernen Programmiersprachen zu entwickeln. Diese Sprachen wurden jedoch insbesondere für Desktop-Systeme entwickelt, welche mit verschwenderisch vielen Ressourcen ausgestattet sind (z.B. Prozessoren mit einem Takt im Gigahertz-Bereich, viel Hauptspeicher, Strom usw.). Eingebettete Systeme verfügen meist nur über sehr eingeschränkte Ressourcen. Möchte man hier den Einsatz moderner Programmiersprachen ermöglichen, so ist es das vornehmlichste Ziel, den mit der Verwendung dieser Sprachen einhergehenden hohen Ressourcenbedarf zu reduzieren.

Am Lehrstuhl werden insbesondere die folgenden drei Teilgebiete untersucht:

1.) Stromsparoptimierung. Für batteriebetriebene Systeme wie z.B. Mobiltelefone oder PDAs ist es wichtig, maßvoll mit der ihnen zur Verfügung stehenden Energie umzugehen. Um Strom zu sparen, besitzen viele Prozessoren die Möglichkeit, dynamisch zur Laufzeit ihre Taktfrequenz und Versorgungsspannung herab- und wieder heraufzusetzen. Diese Technologien werden Dynamic-Frequency-Scaling (DFS) und Dynamic-Voltage-Scaling (DVS) genannt. Am Lehrstuhl wurde erforscht, inwiefern Übersetzer und Betriebssystem zusammenarbeiten können, um diese Technologien auszunutzen. Führt der Prozessor Programmteile aus, die viele Daten aus dem Speicher laden bzw. hineinschreiben, so muss er oft auf den meist wesentlich langsameren (Arbeits-)Speicher warten. Takt und Spannung können in diesem Fall herunterskaliert werden, ohne dass dies aus Sicht der Rechenzeit ins Gewicht fällt. Frühere Ansätze haben versucht, DVS entweder über das Betriebssystem oder direkt vom auszuführenden Programm aus zu realisieren. Bei unserem Ansatz wird das DVS vom Betriebssystem gesteuert, der Übersetzer bettet jedoch Anweisungen in das Programm ein, welche dem Betriebssystem Informationen über den Rechenzeitbedarf des aktuellen Programmabschnittes mitteilt. Das Betriebssystem erlangt somit Kenntnis über den aktuellen Rechenzeitbedarf einer jeden Anwendung und kann die jeweils optimale Frequenz und Versorgungsspannung bestimmen und einstellen. Die Arbeit wurde 2004 abgeschlossen, wobei der reine Kern Algorithmus ohne zusätzliche Übersetzerunterstützung Energieeinsparungen von bis zu 25% bei einer um 19% erhöhten Ausführungszeit erzielen konnte. Die zusätzliche Unterstützung durch den Übersetzer konnte jedoch nur geringe Energieeinsparungen erzielen. In einer Anwendung mit stark unterschiedlicher Dichte von Hauptspeicherzugriffen konnte die Energieeinsparungen von 16,5% auf 18% erhöht werden.

2.) Speicherbereinigung für Code. Da der Arbeitsspeicher (neben Energie) eine weitere beschränkte Resource im Bereich eingebetteter Systeme ist, sind Techniken erforderlich, die den Arbeitsspeicherverbrauch eines Programmes verringern. Die gegenwärtig verfolgte Idee beruht darauf, dass nicht der gesamte Programmcode einer Anwendung komplett im Arbeitsspeicher liegen muss, um das Programm auszuführen. Das Laden des benötigten Codes wird normalerweise vom Betriebssystem übernommen, welches mittels virtuellem Speicher die Möglichkeit besitzt, Programmcode seitenbasiert in den Speicher zu laden bzw. wieder zu verdrängen. Die Größe der Seiten liegt dabei meist im Bereich mehrerer Kilobytes. Die Seiten enthalten mehr als nur den minimal benötigten Code. Bedingt durch ihre Architektur besitzen eingebettete Systeme oft keine Einheit zur Verwaltung von virtuellem Speicher oder sie müssen ganz ohne ein Betriebssystem auskommen. Die am Lehrstuhl entwickelte automatische Speicherverwaltung für Programmcode lädt den Code nicht seitenweise in den Speicher, sondern es werden nur kleine Code-Fragmente (Basisblöcke) geladen, welche garantiert vollständig zur Ausführung kommen. Wenn es zu einem Speicherengpass kommt, wird nicht mehr benötigter Code automatisch erkannt und aus dem Speicher entfernt. Die Laufzeitinformationen, welche die Speicherverwaltung über das auszuführende Programm hat, können verwendet werden, um das Programm dynamisch zu optimieren. Das Potential dieser Optimierungen soll in der Zukunft untersucht werden.

Im Jahr 2004 wurden verschiedene Entwurfalternativen für den automatischen Speicherbereiniger untersucht, wobei für die folgenden beiden Optionen jeweils ein lauffähiger Prototyp erstellt wurden:

- Der am Lehrstuhl entwickelte Übersetzer Jackal wurde so erweitert, dass er den erzeugten Code in Fragmente zerteilt und diese in eine Datei schreibt. Relevante Informationen können bei diesem Ansatz leicht gefunden werden, da sie dem Übersetzer bekannt sind und

in die Datei mit eingebracht werden können. Als Problem erwies sich jedoch, dass der vollständige Quellcode inklusive aller Bibliotheken mit dem angepassten Übersetzer übersetzt werden muss.

- Der zweite Ansatz arbeitet direkt mit ausführbaren Programmen, welche durch zusätzliche Informationen ergänzt werden. Diese werden in einem für jedes Programm nur einmal anfallenden Vorverarbeitungsschritt durch ein Analysewerkzeug ermittelt. Das auszuführende Programm wird zusammen mit den Informationen an den autom. Speicherbereiniger übergeben, welcher das Programm ausführt. Dieser Ansatz ist genereller, da kein spezieller Übersetzer notwendig ist. Jedoch kann das Analysewerkzeug nicht mehr auf die Informationen zurückgreifen, die zur Übersetzungszeit bekannt sind, sondern muss relevante Informationen rekonstruieren. Dies wird im Jahr 2005 angegangen.

3.) RAM-Optimierung. Eine 2004 abgeschlossene Dissertation untersuchte, inwieweit im Bereich gewisser eingebetteter Systeme eine engere Zusammenarbeit von Übersetzern und statischen Betriebssystemen Vorteile bei Effizienz und Sicherheit bringt. Das Interesse lag hier speziell auf kleinen eingebetteten Systemen, die in Hochsprachen wie C programmiert werden und statische Echtzeitbetriebssysteme benutzen. Solche Systeme werden z.B. in der Automobilindustrie verstärkt eingesetzt. Einsparungen von einigen Dutzend Bytes können aufgrund der hohen Stückzahlen insgesamt Kosten in Millionenhöhe einsparen. Eine Studie, die den Overhead eines C-Übersetzers gegenüber direkt in Assembler entwickelten Programmen auf 100 - 1000% schätzt, zeigt das noch große Einsparungspotential durch Optimierungen zur Übersetzungszeit zu erwarten sind. Konkret wurden drei Möglichkeiten untersucht, den RAM-Verbrauch solcher Systeme zu reduzieren:

- Der Einfluss klassischer Übersetzeroptimierungen auf den RAM-Verbrauch wurde gemessen. Eine geschickte Auswahl aus der Menge der klassischen Optimierungen konnte den RAM-Verbrauch in einer kleinen Testreihe um fast 20% senken.
- Modulübergreifende Stackanalyse im Übersetzer wurde mit einem kommerziell erhältlichen Post-Linktime Analysewerkzeug verglichen. Testergebnisse zeigen eine verbesserte Handhabung und höhere Zuverlässigkeit als die existierenden Ansätze. Durch statische Analyse des worst-case Stackverbrauchs konnten der Verschnitt an Stackspeicher minimiert und das Risiko für Stacküberläufe fast eliminiert werden.
- Als neues Verfahren wurden die von statischen Betriebssystemen gespeicherten Taskkontexte (d.h. Registersätze, die bei Unterbrechung eines Tasks gesichert werden müssen) minimiert. Für jeden Task werden möglichst wenige Register gesichert (die sowohl Werte enthalten als auch überschrieben werden können) und auch nur so viel RAM reserviert. Eine um Inter-Task Registervergabe erweiterte Implementierung für ein kommerziell in der Automobilindustrie eingesetztes Betriebssystem zeigt die Machbarkeit dieser Verfahren. Es gelang, sowohl die Handhabung zu verbessern als auch den Speicherbedarf einiger Benchmarks um meist 30%-60% zu reduzieren.

3.10 Zeitplanungs-Algorithmen

Projektleitung:

PD Dr.-Ing. habil. Peter Wilke

Stichwörter:

Time-Tabling-Probleme

Laufzeit: 1.1.2004 - 31.12.2005

Kontakt:

PD Dr.-Ing. habil. Peter Wilke

Tel.: +49.9131.85-27624, Fax: +49.9131.85-.8809,

E-Mail: wilke@informatik.uni-erlangen.de

Zeitpläne müssen in vielen unterschiedlichen Bereichen erstellt werden, z.B. in der Schulstundenplanung oder der Personaleinsatzplanung. Da es sehr mühsam ist, komplexe Zeitpläne wie Schulstundenpläne per Hand zu erstellen, werden die meisten Zeitpläne computerunterstützt generiert. Dazu wurde am Lehrstuhl in den vergangenen Jahren eine Software entwickelt, die es ermöglicht, die Planung unter zu Hilfenahme verschiedener Optimierung-Algorithmen durchzuführen.

Im Berichtsjahr wurde diese Software um zwei weitere Algorithmen, Tabu Search und Simulated Annealing, erweitert und um eine grafische Oberfläche ergänzt.

Tabu Search, Simulated Annealing und der Genetische Algorithmus wurden auf drei reale Beispielprobleme angewendet. Durch Vergleich der Ergebnisse wurde die Eignung von Tabu Search und Simulated Annealing zum Lösen von Zeitplanungsproblemen untersucht. Es hat sich herausgestellt, dass die von Tabu Search erzeugten Zeitpläne deutlich besser als die Zeitpläne des Genetischen Algorithmus sind, und dass Tabu Search in relativ kurzer Zeit gute Pläne generieren kann. Die von Simulated Annealing erstellten Zeitpläne übertreffen in ihrer Qualität sowohl den Genetischen Algorithmus als auch Tabu Search.

Die grafische Bedienoberfläche ist für die Verwendung durch Benutzer mit stark unterschiedlichen Kenntnissen konzipiert. Durch die Verwaltung der Benutzer ist es möglich, die Sichten auf die Stundenpläne ebenso wie den Grad der Einflussnahme des Benutzers auf die Algorithmen zu steuern.

Projektunabhängige Publikationen

1. Barta, Csongor; Dormeyer, Ricarda; Fischer, Ingrid: Word Order and Discontinuities in a Dependency Grammar for Hungarian . In: Alexin, Zoltán; Csendes, Dóra (Hrsg.) : Proceedings of the 2nd Conference on Hungarian Computational Linguistics (MSZNY 2004) (2nd Conference on Hungarian Computational Linguistics (MSZNY 2004) Szeged, Hungary 9-10 December 2004). Szeged, Hungary : Juhász Nyomda, 2004, S. 19-27.
2. Barthelmann, Volker: Advanced Compiling Techniques to reduce RAM Usage of Static Operating Systems . Erlangen, Universität, Diss., 2004. - 145 Seiten.
3. Borgelt, Christian; Meinl, Thorsten; Berthold, Michael R.: Advanced Pruning Strategies

- to Speed Up Mining Closed Molecular Fragments . In: Thissen, Wil; Wieringa, Peter; Pantic, Maja; Ludema, Marcel (Hrsg.) : IEEE SMC 2004 Conference Proceedings (2004 IEEE Conference on Systems, Man & Cybernetics Den Haag, The Netherlands 2004-10-10 - 2004-10-13). 2004, S. 4565-4570. - ISBN 0-7803-8567-5
4. Böhm, Niko; Kókai, Gabriella; Mandl, Stefan: Evolving a Heuristic Function for the Game of Tetris . In: Tobias Scheffer (Veranst.) : Proc. Lernen, Wissensentdeckung und Adaptivität LWA - 2004 (Lernen, Wissensentdeckung und Adaptivität LWA - 2004 Berlin, Germany 04.-06. Oktober 2004). 2004, S. 118-122.
 5. Dormeyer, Ricarda: Syntaxanalyse auf Basis der Dependenzgrammatik . Erlangen-Nürnberg, Friedrich-Alexander Universität, Diss., 2004. - 174 Seiten.
 6. Fischer, Ingrid; Meinel, Thorsten: Graph Based Molecular Data Mining - An Overview . In: Thissen, Wil; Wieringa, Peter; Pantic, Maja; Ludema, Marcel (Hrsg.) : IEEE SMC 2004 Conference Proceedings (2004 IEEE Conference on Systems, Man & Cybernetics Den Haag, The Netherlands 2004-10-10 - 2004-10-13). 2004, S. 4578-4582. - ISBN 0-7803-8567-5
 7. Fischer, Ingrid: Kann man dem Computer einen Bären aufbinden? Metaphern und Idiome in der automatischen Sprachverarbeitung .Vortrag: Vom Abakus zum Quantencomputer III - Entwicklungsgeschichte und die philosophisch-mathematischen Grundlagen der Rechentechnik, Cauchy-Forum-Nürnberg (CFN) e.V. - Interdisziplinäres Forum für Mathematik und ihre Grenzgebiete, Nürnberg, 09.12.2004
 8. Fischer, Ingrid: Modeling Discontinuous Constituents with Hypergraph Grammars . In: Pfaltz, John L.; Nagl, Manfred; Böhlen, Boris (Hrsg.) : Applications of Graph Transformations with Industrial Relevance (AGTIVE 2003) (Second International Workshop of Applications of Graph Transformations with Industrial Relevance Charlottesville, VA, USA 27. September - 1. Oktober 2003). Berlin : Springer-Verlag, 2004, S. 163 - 169. (Lecture Notes On Computer Science, Nr. 3062)
 9. Fühner, Tim; Erdmann, Andreas; Farkas, Richard; Tollkühn, Bernd; Kókai, Gabriella: Genetic Algorithms to Improve Mask and Illumination Geometries in Lithographic Imaging Systems . In: Raidl, G. (Hrsg.) : Proc. Applications of Evolutionary Computing (1st European Workshop on Hardware Optimisation (EVOHOT2004) <http://evonet.dcs.napier.ac.uk/eurogp2004/hotindex.html> April 3-5, 2004). Bd. 3005. Berlin : Springer Verlag, 2004, S. 208-219. - ISBN 3-540-21378-3
 10. Hardung, Bernd; Kölzow, Thorsten; Krüger, Andreas: Reuse of Software in Distributed Embedded Automotive Systems . In: Buttazzo, Giorgio (Hrsg.) : Proceedings of the 4th ACM International Conference on Embedded Software (EMSOFT 2004 University of Pavia, Italy Sept. 27-29, 2004). Bd. 306. New York, NY, USA : ACM Press, 2004, S. 203-210. - ISBN 1-58113-860-1
 11. Klemm, Michael; Veldema, Ronald; Philippsen, Michael: Latency Reduction in Software-DSMs by Means of Dynamic Function Splicing . In: Gonzales, Teofilo (Hrsg.) : IASTED (Veranst.) : Proc. of the 16th IASTED Intl. Conf. on PDCS (Parallel and Distributed Computing and Systems Cambridge 2004-11-09 - 2004-11-11). 2004, S. 362-367. - ISBN 0-88986-421-7
 12. Kókai, Gabriella; Frühauf, Hans Holm; Feng, Xu : Development of a Hardware Based Genetic Optimizer to Adjust Smart Antenna Receiver . In: International Journal of Embedded Systems Special Issue on Hardware-Software Codesign for Systems-on-Chip

(2004), S. -

13. Meinl, Thorsten; Borgelt, Christian; Berthold, Michael R.: Discriminative Closed Fragment Mining and Perfect Extensions in MoFa . In: Onaindia, Eva; Staab, Steffen (Hrsg.) : STAIRS 2004 - Proceedings of the Second Starting AI Researchers' Symposium (STAIRS 2004 Valencia, Spain 2004-08-22 - 2004-08-23). Bd. 109, 1. Aufl. Amsterdam : IOS Press, 2004, S. 3-14. (Frontiers in Artificial Intelligence and Applications Bd. 109) - ISBN 1 58603 451 0
14. Meinl, Thorsten: Erweiterte Fragmentsuche in Moleküldatenbanken . Erlangen, Friedrich-Alexander-Universität Erlangen-Nürnberg, Dipl-Arb., 2004. - 78 Seiten.
15. Meinl, Thorsten; Berthold, Michael R.: Hybrid Fragment Mining with MoFa and FSG . In: Thissen, Wil; Wieringa, Peter; Pantic, Maja; Ludema, Marcel (Hrsg.) : IEEE SMC 2004 Conference Proceedings (2004 IEEE International Conference on Systems, Man & Cybernetics Den Haag, The Netherlands 2004-10-10 - 2004-10-13). 2004, S. 4559-4564. - ISBN 0-7803-8567-5
16. Meinl, Thorsten; Borgelt, Christian; Berthold, Michael R.: Mining fragments with fuzzy chains in molecular databases . In: Kok, Joost N.; Washio, Takashi (Hrsg.) : Proceedings of the Workshop W7 on Mining Graphs, Trees and Sequences (MGTS '04) (ECML/PKDD 2004 Pisa, Italy 2004-09-20 - 2004-09-24). 2004, S. 49-60.
17. Puhmann, Hermann; Humbert, Ludger: Informatische Bildung und PISA Standards - zur Umsetzung für die informatische Bildung . In: CD Austria, Sonderheft des bm:bwk (2004), Nr. 5, S. 21-24
18. Schneider, Hans Jürgen: Introducing the Parallel Random Access Machine Together with FORTRAN 90/95 . In: Bulletin of the European Association for Theoretical Computer Science (EATCS) 82 (2004), S. 299-314
19. Seifert, Sebastian; Fischer, Ingrid: Parsing String Generating Hypergraph Grammars . In: Ehrig, Hartmut; Engels, Gregor; Parisi-Presicce, Francesco; Rozenberg, Grzegorz (Hrsg.) : Graph Transformations (Second International Conference on Graph Transformations (ICGT 04) Rome, Italy September, 28 - October, 1). Berlin : Springer-Verlag, 2004, S. 352 - 267. (Lecture Notes On Computer Science, Nr. 3256)
20. Vanyi, Robert; Zvada, Szilvia: Syntactically Correct Genetic Programming . In: Cagnoni, S.; Keijzer, M.; O'Neill, M.; Ryan, C. (Hrsg.) : Proc. of the Grammatical Evolution (GEWS 2004) (Satellite Workshop to Genetic and Evolutionary Computation Conference (GECCO) 2004 Seattle, Washington/ USA June 26-3, 2004). 2004, S. CD-ROM.
21. Vanyi, Robert; Zvada, Szilvia; Frühauf, Hans Holm; Kókai, Gabriella: Generating Resource-Optimal FIR Structures with Derivation Tree Based Genetic Programming . In: submitted to Journal Annals of Operations Research Special Issue on STOCHASTIC SEARCH ALGORITHMS (2004), S. -
22. Veldema, Ronald; Philippsen, Michael: Using Object Combining for Object Prefetching in DSM Systems . In: - (Hrsg.) : 11th Workshop on Compilers for Parallel Computers (CPC 2004 Seon July 7-9, 2004). 2004, S. -.
23. Zvada, Szilvia; Vanyi, Robert: Improving Grammar Based Evolutionary Algorithms via Attributed Derivation Trees . In: Keijzer, M.; O'Reilly, U. (Hrsg.) : Proc. of the 7th European Conference on Genetic Programming (EuroGP2004 Coimbra, Portugal April 5-7, 2004). Bd. LNCS 3003. Berlin : Springer Verlag, 2004, S. 208-219.

4 Veröffentlichungen und Vorträge

4.1 Veröffentlichungen

1. Barta, Csongor; Dormeyer, Ricarda; Fischer, Ingrid: Word Order and Discontinuities in a Dependency Grammar for Hungarian . In: Alexin, Zoltán; Csendes, Dóra (Hrsg.) : Proceedings of the 2nd Conference on Hungarian Computational Linguistics (MSZNY 2004) (2nd Conference on Hungarian Computational Linguistics (MSZNY 2004) Szeged, Hungary 9-10 December 2004). Szeged, Hungary : Juhász Nyomda, 2004, S. 19-27.
2. Barthelmann, Volker: Advanced Compiling Techniques to reduce RAM Usage of Static Operating Systems . Erlangen, Universität, Diss., 2004. - 145 Seiten.
3. Borgelt, Christian; Meinl, Thorsten; Berthold, Michael R.: Advanced Pruning Strategies to Speed Up Mining Closed Molecular Fragments . In: Thissen, Wil; Wieringa, Peter; Pantic, Maja; Ludema, Marcel (Hrsg.) : IEEE SMC 2004 Conference Proceedings (2004 IEEE Conference on Systems, Man & Cybernetics Den Haag, The Netherlands 2004-10-10 - 2004-10-13). 2004, S. 4565-4570. - ISBN 0-7803-8567-5
4. Böhm, Niko; Kókai, Gabriella; Mandl, Stefan: Evolving a Heuristic Function for the Game of Tetris . In: Tobias Scheffer (Veranst.) : Proc. Lernen, Wissensentdeckung und Adaptivität LWA - 2004 (Lernen, Wissensentdeckung und Adaptivität LWA - 2004 Berlin, Germany 04.-06. Oktober 2004). 2004, S. 118-122.
5. Dormeyer, Ricarda: Syntaxanalyse auf Basis der Dependenzgrammatik . Erlangen-Nürnberg, Friedrich-Alexander Universität, Diss., 2004. - 174 Seiten.
6. Fischer, Ingrid; Meinl, Thorsten: Graph Based Molecular Data Mining - An Overview . In: Thissen, Wil; Wieringa, Peter; Pantic, Maja; Ludema, Marcel (Hrsg.) : IEEE SMC 2004 Conference Proceedings (2004 IEEE Conference on Systems, Man & Cybernetics Den Haag, The Netherlands 2004-10-10 - 2004-10-13). 2004, S. 4578-4582. - ISBN 0-7803-8567-5
7. Fischer, Ingrid: Kann man dem Computer einen Bären aufbinden? Metaphern und Idiome in der automatischen Sprachverarbeitung .Vortrag: Vom Abakus zum Quantencomputer III - Entwicklungsgeschichte und die philosophisch-mathematischen Grundlagen der Rechentechnik, Cauchy-Forum-Nürnberg (CFN) e.V. - Interdisziplinäres Forum für Mathematik und ihre Grenzgebiete, Nürnberg, 09.12.2004
8. Fischer, Ingrid: Modeling Discontinuous Constituents with Hypergraph Grammars . In: Pfaltz, John L.; Nagl, Manfred; Böhlen, Boris (Hrsg.) : Applications of Graph Transformations with Industrial Relevance (AGTIVE 2003) (Second International Workshop of Applications of Graph Transformations with Industrial Relevance Charlottesville, VA, USA 27. September - 1. Oktober 2003). Berlin : Springer-Verlag, 2004, S. 163 - 169. (Lecture Notes On Computer Science, Nr. 3062)
9. Fühner, Tim; Erdmann, Andreas; Farkas, Richard; Tollkühn, Bernd; Kókai, Gabriella: Genetic Algorithms to Improve Mask and Illumination Geometries in Lithographic Imaging Systems . In: Raidl, G. (Hrsg.) : Proc. Applications of Evolutionary Computing (1st European Workshop on Hardware Optimisation (EVOHOT2004) <http://evonet.dcs.napier.ac.uk/eurogp2004/hotindex.html> April 3-5, 2004). Bd. 3005. Berlin : Springer Verlag, 2004, S. 208-219. - ISBN 3-540-21378-3

10. Hardung, Bernd; Kölzow, Thorsten; Krüger, Andreas: Reuse of Software in Distributed Embedded Automotive Systems . In: Buttazzo, Giorgio (Hrsg.) : Proceedings of the 4th ACM International Conference on Embedded Software (EMSOFT 2004 University of Pavia, Italy Sept. 27-29, 2004). Bd. 306. New York, NY, USA : ACM Press, 2004, S. 203-210. - ISBN 1-58113-860-1
11. Klemm, Michael; Veldema, Ronald; Philippsen, Michael: Latency Reduction in Software-DSMs by Means of Dynamic Function Splicing . In: Gonzales, Teofilo (Hrsg.) : IASTED (Veranst.) : Proc. of the 16th IASTED Intl. Conf. on PDCS (Parallel and Distributed Computing and Systems Cambridge 2004-11-09 - 2004-11-11). 2004, S. 362-367. - ISBN 0-88986-421-7
12. Kókai, Gabriella; Frühauf, Hans Holm; Feng, Xu : Development of a Hardware Based Genetic Optimizer to Adjust Smart Antenna Receiver . In: International Journal of Embedded Systems Special Issue on Hardware-Software Codesign for Systems-on-Chip (2004), S. -
13. Meinel, Thorsten; Borgelt, Christian; Berthold, Michael R.: Discriminative Closed Fragment Mining and Perfect Extensions in MoFa . In: Onaindia, Eva; Staab, Steffen (Hrsg.) : STAIRS 2004 - Proceedings of the Second Starting AI Researchers' Symposium (STAIRS 2004 Valencia, Spain 2004-08-22 - 2004-08-23). Bd. 109, 1. Aufl. Amsterdam : IOS Press, 2004, S. 3-14. (Frontiers in Artificial Intelligence and Applications Bd. 109) - ISBN 1 58603 451 0
14. Meinel, Thorsten: Erweiterte Fragmentsuche in Moleküldatenbanken . Erlangen, Friedrich-Alexander-Universität Erlangen-Nürnberg, Dipl.-Arb., 2004. - 78 Seiten.
15. Meinel, Thorsten; Berthold, Michael R.: Hybrid Fragment Mining with MoFa and FSG . In: Thissen, Wil; Wieringa, Peter; Pantic, Maja; Ludema, Marcel (Hrsg.) : IEEE SMC 2004 Conference Proceedings (2004 IEEE International Conference on Systems, Man & Cybernetics Den Haag, The Netherlands 2004-10-10 - 2004-10-13). 2004, S. 4559-4564. - ISBN 0-7803-8567-5
16. Meinel, Thorsten; Borgelt, Christian; Berthold, Michael R.: Mining fragments with fuzzy chains in molecular databases . In: Kok, Joost N.; Washio, Takashi (Hrsg.) : Proceedings of the Workshop W7 on Mining Graphs, Trees and Sequences (MGTS '04) (ECML/PKDD 2004 Pisa, Italy 2004-09-20 - 2004-09-24). 2004, S. 49-60.
17. Puhlmann, Hermann; Humbert, Ludger: Informatische Bildung und PISA Standards - zur Umsetzung für die informatische Bildung . In: CD Austria, Sonderheft des bm:bwk (2004), Nr. 5, S. 21-24
18. Schneider, Hans Jürgen: Introducing the Parallel Random Access Machine Together with FORTRAN 90/95 . In: Bulletin of the European Association for Theoretical Computer Science (EATCS) 82 (2004), S. 299-314
19. Seifert, Sebastian; Fischer, Ingrid: Parsing String Generating Hypergraph Grammars . In: Ehrig, Hartmut; Engels, Gregor; Parisi-Presicce, Francesco; Rozenberg, Grzegorz (Hrsg.) : Graph Transformations (Second International Conference on Graph Transformations (ICGT 04) Rome, Italy September, 28 - October, 1). Berlin : Springer-Verlag, 2004, S. 352 - 267. (Lecture Notes On Computer Science, Nr. 3256)
20. Vanyi, Robert; Zvada, Szilvia: Syntactically Correct Genetic Programming . In: Cagnoni, S.; Keijzer, M.; O'Neill, M.; Ryan, C. (Hrsg.) : Proc. of the Grammatical Evolution (GEWS 2004) (Satellite Workshop to Genetic and Evolutionary Computation

Conference (GECCO) 2004 Seattle, Washington/ USA June 26-3, 2004). 2004, S. CD-ROM.

21. Vanyi, Robert; Zvada, Szilvia; Frühauf, Hans Holm; Kókai, Gabriella: Generating Resource-Optimal FIR Structures with Derivation Tree Based Genetic Programming . In: submitted to Journal Annals of Operations Research Special Issue on STOCHASTIC SEARCH ALGORITHMS (2004), S. -
22. Veldema, Ronald; Philippsen, Michael: Using Object Combining for Object Prefetching in DSM Systems . In: - (Hrsg.) : 11th Workshop on Compilers for Parallel Computers (CPC 2004 Seon July 7-9, 2004). 2004, S. -.
23. Zvada, Szilvia; Vanyi, Robert: Improving Grammar Based Evolutionary Algorithms via Attributed Derivation Trees . In: Keijzer, M.; O'Reilly, U. (Hrsg.) : Proc. of the 7th European Conference on Genetic Programming (EuroGP2004 Coimbra, Portugal April 5-7, 2004). Bd. LNCS 3003. Berlin : Springer Verlag, 2004, S. 208-219.

4.2 Vorträge

5 Promotionen

6 Studien- und Diplomarbeiten

6.1 Studienarbeiten

6.2 Diplomarbeiten

7 Bachelor- und Master Theses

7.1 Bachelor Theses

7.2 Master Theses