Das SymbolicData-Projekt

Olaf Bachmann (Kaiserslautern) und Hans-Gert Gräbe (Leipzig) [1]

Wir wollen mit diesem Bericht den Lesern des Computeralgebra-Rundbriefs das SymbolicData-Projekt, an dem wir im Rahmen der Benchmarking-Aktivitäten der Fachgruppe seit etwa einem Jahr arbeiten, kurz vorstellen. Wir konzentrieren uns dabei auf die Darstellung der Motivation, die zu diesem Projekt geführt hat, und einige grundsätzliche Überlegungen, aus denen der gewählte Ansatz deutlich wird. Für konkretere Fragen nach der Struktur der bisherigen Datensammlung und der erstellten Werkzeuge, der Art, wie man daraus Benchmark-Rechnungen erzeugen und starten sowie die Ergebnisse solcher Rechnungen auswerten kann, verweisen wir interessierte Leser auf die online-Repräsentation des Projekts auf unserer Web-Seite `www.SymbolicData.org`. Wie bei einem Open-Source-Projekt üblich sind die Daten und Werkzeuge offen verfügbar und können problemlos heruntergeladen, installiert und ausprobiert werden. Wir hoffen, auf diesem Wege neue Mitstreiter zu finden, die sich der Freiheiten, aber auch der Verantwortung bewusst sind, die die Mitarbeit an einem solchen Vorhaben mit sich bringt.

Motivation

Nach den Diskussionen auf der ISSAC'98 in Rostock über Benchmarking-Aktivitäten im Bereich der Computeralgebra haben wir zunächst begonnen, die bereits existierenden Materialien zu sichten und zu ordnen. Dabei stellte sich schnell heraus, dass es sinnvoll ist, diese Materialien in ein einheitliches elektronisches Format zu überführen und geeignete Werkzeuge zur Manipulation dieser Materialen bereitzustellen.

Entsprechend konzentrierten sich unsere bisherigen Aktivitäten auf die Fragestellungen, die im Untertitel des Projekts "An Electronic Repository of Tools and Data for Computer Algebra Benchmarks" deutlich werden.

Die bisherigen Hauptrichtungen des SymbolicData-Projekts lassen sich wie folgt umreißen:

1. Systematische Sammlung existierender Benchmark-Daten zu verschiedenen Fragestellungen des symbolischen Rechnens und Erstellung von Werkzeugen, mit denen diese Daten bequem gewartet und erweitert werden können.

2. Entwurf von Konzepten und Implementierung von Werkzeugen, unter denen sich glaubwürdige Benchmark-Rechnungen auf diesen Daten ausführen lassen.

3. Bereitstellung von Werkzeugen, mit denen man in den gesammelten Daten navigieren, diese nach verschiedenen Kriterien selektieren und in andere Formate transformieren kann.

In der ersten Phase des Projekts konzentrierten wir uns darauf, allgemeine Designprinzipien zu fixieren, die einerseits eine hohe Flexibilität und Erweiterbarkeit garantieren, andererseits aber auch genügend einfach und praktisch in ihrer Handhabung sind.

Wir haben uns dabei bemüht, stets die Umsetzung der Konzepte in Werkzeuge und die Anwendbarkeit der Werkzeuge auf größere Datenmengen zu erproben, so dass zum gegenwärtigen Zeitpunkt neben hinreichend bewährten Konzepten auch eine ganze Reihe von Werkzeugen und Datensammlungen zu zwei Benchmarkbereichen – polynomialen Systemen und mechanisiertem Theorembeweisen in der Geometrie – zur Verfügung stehen.

Grundsätzliche Überlegungen

Vom Ansatz her ist eine solche Datensammlung eine Datenbank-Applikation, wobei uns auf Grund der spezifischen Anwendung ein **objekt-relationales Konzept** besonders geeignet erschien. Mit einem solchen Ansatz kann man einerseits Querverbindungen zwischen verschiedenen Daten, etwa zwischen Problembeschreibungen, Ergebnissen, Hintergrundinformationen und Literaturquellen,

[1] published in "Computeralgebra-Rundbrief" 26, March 2000

gut erfassen, zum anderen aber auch günstig modulare, objektorientierte Konzepte bei der Erstellung der Werkzeuge für die Manipulation der verschiedenen Datenklassen verwirklichen.

Vor allem aus Gründen der Flexibilität haben wir uns, wenigstens im Moment, entschieden, für die Speicherung der primären Quellen unserer Daten keine klassische Datenbank zu nutzen, sondern diese direkt in einem **XML-artigen ASCII-Format** abzulegen. Wir gingen bei dieser Entscheidung davon aus, dass relevante Daten der Computeralgebra in Form von Strings vorliegen bzw. bequem als solche gespeichert werden können. Im vorliegenden Format sind sie auch reinen ASCII-Editoren zugänglich, aber zugleich in einer Weise gespeichert, die Anforderungen künftiger XML-Werkzeuge nahe kommt.

Die Verarbeitung und Manipulation dieser Daten erfolgt mit **Perl-Werkzeugen**. Perl ist eine Programmiersprache aus dem Open-Source-Bereich, die sich mit ihren leistungsstarken Stringmanipulations- und Skriptingfähigkeiten für diese Aufgaben als sehr geeignet erwiesen hat. Zur Strukturierung der Werkzeuge wird intensiv von den Modularisierungsmöglichkeiten Gebrauch gemacht, die Perl 5 bereit stellt.

Sowohl aus Gründen der Flexibilität als auch mit Blick auf potentielle Nutzer bleibt Perl aber im Hintergrund. Alle wichtigen Standard-Applikationen können über eine gut dokumentierte, flexible und intuitiv zu bedienende **einheitliche Schnittstelle** im üblichen Kommandozeilen-Format ausgeführt werden.

Um die notwendige Flexibilität bei der Erweiterung und Modifikation bereits bestehender Datenklassen (*Tabellen* in der Sprache der Datenbanken) sowie der Erzeugung neuer solcher Tabellen, etwa für weitere Benchmark-Bereiche, zu erreichen, ist die entsprechende Strukturinformation nicht in Perl festgeschrieben. Statt dessen lesen die Werkzeuge zur Laufzeit diese Informationen aus **Meta-Tabellen** ein, deren Attribute im selben XML-ASCII-Format vorliegen wie die Datenrecords selbst. Sogar das **Typkonzept**, das die Erstellung solcher Attributbeschreibungen unterstützt, liegt in diesem Format vor und kann damit leicht, unabhängig von den Werkzeugen und ohne Perlkenntnisse modifiziert und erweitert werden.

Der aktuelle Stand

Die hier vorgestellten konzeptionellen Überlegungen standen so natürlich noch nicht von Anfang an fest, sondern sind im Laufe der Zeit und intensiver Arbeit an den bisher gesammelten Daten und bei der Implementierung der nun verfügbaren Werkzeuge entstanden. Obwohl die Konzepte inzwischen eine gewisse innere Konsistenz aufweisen, sind zukünftige gut begründete Änderungen und Modifikationen, wie bei jedem Software-Projekt, nicht auszuschließen. Wir gehen allerdings davon aus, dass die über 1100 Datenrecords, die wir aus zwei großen Benchmark-Bereichen, dem Bereich der polynomialen Gleichungssysteme und dem Bereich des mechanisierten Geometrie-Theorembeweisens, gesammelt und mit unseren Werkzeugen erfolgreich gewartet und manipuliert haben, eine Gewähr dafür bieten, dass diese Änderungen eher evolutionären Charakter haben werden.

Das SymbolicData-Team hat neben diesen 1100 Datenrecords mehr als 40 Perl-Module mit über 15 000 Zeilen Code geschrieben und über 20 Aktionen in der einheitlichen Schnittstelle zusammengeführt. Der folgende kurze alphabetische Überblick über die existierenden Tabellen mag den Lesern ein Gefühl für die Gesamtstruktur der bisher gesammelten Daten geben:

Tabelle BIB sammelt bibliographische Informationen im BibTeX-Format, kurze Abstracts und Querverweise zu den Tabellen `GEO`, `INTPS` und `PROBLEMS`.

Tabelle CAS enthält allgemeine Beschreibungen einzelner Computer-Algebra-Software.

Tabelle CASCONFIG enthält Konfigurationen einzelner CAS, die für verschiedene Benchmark-Rechnungen erforderlich sind und Querverweise zu den Tabellen `CAS` und `COMP`.

Tabelle COMP enthält Informationen über die auszuführenden Benchmarkrechnungen selbst.

Tabelle COMPREPORT enthält Reports von Benchmark-Rechnungen.

Tabelle COMPRESULTS enthält Resultate von Benchmark-Rechnungen.

Tabelle GEO: Eine Sammlung von Problemen aus dem Bereich des mechanisierten Geometrie-Theorembeweisens mit Querverweisen zu den Tabellen `INTPS` und `PROBLEMS`.

Tabelle INTPS: Eine Sammlung von Problemen aus dem Bereich der polynomialen Gleichungssysteme mit Querverweisen zu den Tabellen `BIB` und `PROBLEMS`.

Tabelle MACHINE: Zusammenstellung der Computer, auf denen Benchmark-Rechnungen ausgeführt wurden. Querverweise zur Tabelle `CASCONFIG`.

Tabelle PERSON: Informationen über die am SymbolicData-Projekt beteiligten Personen.

Tabelle PROBLEMS: Detailliertere Hintergrund-Informationen und Kommentare zu einzelnen Problemen, etwa eine Beschreibung, ein Verweis auf Originalarbeiten, relevanter CAS-Code und/oder einige Schlüsselworte.

Wir haben mit ersten Benchmark-Rechnungen zu Gröbnerbasen begonnen, die verschiedene Koeffizientenbereiche und Termordnungen verwenden. Diese Benchmark-Rechnungen laufen mit 10 Versionen verschiedener CAS auf den über 500 `INTPS`-Records. Weitere Benchmark-Rechungen sind in Vorbereitung, wobei wir auf die exzellenten Bedingungen am UMS MEDICIS (`www.medicis.polytechnique.fr/medicis`) zurückgreifen können. Auch unsere von der Fachgruppe CA gesponsorte Web-Site ist dort physisch beheimatet.

Wie weiter?

Aus unserer Sicht sind die Konzepte und Werkzeuge inzwischen so weit gereift, dass sie einem größeren Personenkreis zur Begutachtung und Weiterentwicklung zur Verfügung gestellt werden können. Eine Weiterentwicklung wird vor allem dann notwendig werden, wenn sich aus Anwendungen in anderen als den bisher untersuchten Benchmark-Bereichen Stellen ergeben sollten, an denen die Konzepte noch nicht allgemein genug sind. Wir suchen deshalb den Kontakt zu Gruppen, die solche Anwendungen studieren, über relevante Datenmengen verfügen und bereit sind, diese unter den Bedingungen der Gnu Public License mit anderen zu teilen.

Obwohl wir dies auf ein Minimum zurückgeschraubt haben, wird man beim Erschließen neuer Benchmark-Bereiche, besonders für semantische Aspekte, nicht ohne ein gewisses Maß an Perl-Programmierung auskommen. Das SymbolicData-Team steht mit Rat und im Rahmen seiner Möglichkeiten auch mit tatkräftiger Unterstützung bereit, diesen Programmieraufwand zu bewältigen. Allerdings wird das Projekt auf längere Sicht nur erfolgreich sein können, wenn es auch personelle Verstärkung erfährt, die wir gern unbürokratisch und zu gleichberechtigten Bedingungen in unser Team integrieren.