Kickoff Meeting

Certifying Algorithms

Matthias Gehnen, Henri Lotze, Daniel Mock, Peter Rossmanith

28. März, 2021

Einführung

Das Thema

- Algorithmen erhalten Eingabe, berechnen Ausgabe

Das Thema

- Algorithmen erhalten Eingabe, berechnen Ausgabe
- Können wir der Ausgabe vertrauen?

Das Thema

- Algorithmen erhalten Eingabe, berechnen Ausgabe
- Können wir der Ausgabe vertrauen?
- Der Algorithmus kann fehlerhaft sein!

Lösungen?

- Pair Programming

Lösungen?

- Pair Programming
  - Programmierer machen auch zu zweit Fehler

Lösungen?

- Pair Programming
  - Programmierer machen auch zu zweit Fehler
- Testing

Lösungen?

- Pair Programming
  - Programmierer machen auch zu zweit Fehler
- Testing
  - Tests decken häufig nicht alles ab

Lösungen?

- Pair Programming
  - Programmierer machen auch zu zweit Fehler
- Testing
  - Tests decken häufig nicht alles ab
  - (und sind manchmal selber fehlerhaft)

Lösungen?

- Pair Programming
  - Programmierer machen auch zu zweit Fehler
- Testing
  - Tests decken häufig nicht alles ab
  - (und sind manchmal selber fehlerhaft)
- Formale Verifizierung

Lösungen?

- Pair Programming
  - Programmierer machen auch zu zweit Fehler
- Testing
  - Tests decken häufig nicht alles ab
  - (und sind manchmal selber fehlerhaft)
- Formale Verifizierung
  - Formale Verifizierung ist kompliziert und Aufwändig

Zertifizierende Algorithmen

- Algorithmen sollen *Nachweis* liefern, dass sie korrekt sind

Zertifizierende Algorithmen

- Algorithmen sollen *Nachweis* liefern, dass sie korrekt sind
- Weiterer Algorithmus prüft Eingabe, Ausgabe, Zertifikat

Zertifizierende Algorithmen

- Algorithmen sollen *Nachweis* liefern, dass sie korrekt sind
- Weiterer Algorithmus prüft Eingabe, Ausgabe, Zertifikat
  - Häufig probabilistisch, WHP korrekt

Zertifizierende Algorithmen

- Algorithmen sollen *Nachweis* liefern, dass sie korrekt sind
- Weiterer Algorithmus prüft Eingabe, Ausgabe, Zertifikat
  - Häufig probabilistisch, WHP korrekt
- Möglichst schnelle Prüfung!

Kleine Beispiele

- Ist ein gegebener, gerichteter Graph *azyklisch*?

Kleine Beispiele

- Ist ein gegebener, gerichteter Graph *azyklisch*?
  - Topologische Ordnung oder Zykel.

Kleine Beispiele

- Ist ein gegebener, gerichteter Graph *azyklisch*?
  - Topologische Ordnung oder Zykel.
- Gilt für zwei Matrizen $AB = C$?

Kleine Beispiele

- Ist ein gegebener, gerichteter Graph *azyklisch*?
  - Topologische Ordnung oder Zykel.
- Gilt für zwei Matrizen $AB = C$?
  - Vektor c erzeugen, $Cc = A \cdot Bc$? Wiederholen!

Das Thema

- Forschung von Konzipierung bis heute

Das Thema

- Forschung von Konzipierung bis heute
- Große Breite, Zeritifizierer für viele Probleme

Das Thema

- Forschung von Konzipierung bis heute
- Große Breite, Zeritifizierer für viele Probleme
- Konkrete zertifizierende Algorithmen

Das Thema

- Forschung von Konzipierung bis heute
- Große Breite, Zeritifizierer für viele Probleme
- Konkrete zertifizierende Algorithmen
- Ihr erhaltet (hoffentlich) einen guten Gesamtüberblick

Gliederung

- **Kickoff / Fragen** (Heute)
- **Themenvergabe / Fragen** (Donnerstag?)
  - Jeder bekommt ein Thema (1-2 Personen pro Thema)
  - Via Umfrage und kurzem Treffen
- **Literaturrechercheschulung der Infobib** (eigenständig)
- **Vorbereitungsphase** (mind. 4 Wochen)
- **Wöchentliche Präsentationen** (Anfang: 2. Maiwoche)
  - 25-30 Minuten Präsentation + 10-20 Minuten Diskussion
- **Ausarbeitung** (bis. 1. September)
  - Wichtigste Ideen zusammenfassen, höchstens 8 Seiten, LaTeX

Treffen

- Wöchentliche Treffen in Person
- Raum buchen wir nach Terminfestlegung
- Anwesenheitspflicht

Aufbau des Treffens

Wenn man nicht selber vorträgt:

- Dem Vortrag genau zuhören
- Eigene Fragen notieren
- An der Diskussion teilnehmen
- Feedback geben

Aufbau des Treffens

Wenn man präsentiert:

- Eine Auswahl des eigenen Themas präsentieren
- Innerhalb der Zeitbeschränkung bleiben
- Die Präsentation sollte für jeden verständlich sein
    - Insbesondere für eure Mitstudierenden!
- Anschließend Fragen beantworten

Mögliche Struktur der Präsentation

- Kurze Einführung
- Motivation: Weshalb ist das Thema interessant?
- Notwendiges Hintergrundwissen, typische Techniken
- Inhalte vorstellen
- Abschließend Zusammenfassung mit offenen Fragen
- Diskussion fördern

Tipps: Vor der Präsentation

- **Versteht das Thema**, Quellen recherchieren
- Gliederung erstellen, Themen/Fokus auswählen
- Einfache, verständliche Beispiele um Ideen zu präsentieren
- Mögliche Fragen und offene Themen für Diskussion finden

Tipps: Während der Präsentation

- **Langsam** präsentieren. Nicht jeder wird euch sofort verstehen
- Kontext geben, ggf. auf vorherige Vorträge referenzieren
- *Idee* eines Beweises präsentieren
- Trotzdem in der Lage sein Detailfragen zu beantworten

Tipps für eure Folien

- Deutsch *oder* Englisch
- Nutzt `Beamer` mit LaTeX
- Aufgeräumte Folien: Lieber ein Bild als eine Textwand
- Diese Folien sind viel zu voll und ein schlechtes Vorbild

Materialien für einen erfolgreichen Vortrag

- Z.B. `http://ianparberry.com/pubs/speaker.pdf`
- Learning by doing
- Achtet darauf, was bei anderen gelobt und kritisiert wurde

Ausarbeitung

- Nutzt gerne die selbe Struktur wie im Vortrag
- LaTeX ist verpflichtend (Anleitung: `https://www.latex-tutorial.com/tutorials/`)
- 8 Seiten
- Vorlage ist auf unserer Website verfügbar
- Richtig zitieren (wissenschaftliche Ausarbeitungen etc.)

Aber:

- Nicht einfach das Paper nacherzählen

Diskussiongruppe?

Telegram, Signal, WhatsApp, ...

Probleme?

- Wir benoten nur euren Vortrag und die Ausarbeitung
- Redet mit uns, wenn ihr Fragen habt
- Abmeldung ohne Fehlversuch in den nächsten drei Wochen möglich

Die Themen

Themen

1. Program result checking: A new approach to making programs more reliable

   Blum's wegweisendes, erster Papier in diesem Bereich.

   Konzept und einige zertifizierende Algorithmen.

   1-2 Personen

2. Self-Testing/Correcting with Applications to Numerical Problems

   Erweiterung des Originalpapiers: Fehler nicht nur *erkennen*, sondern auch *korrigieren*

   2 Personen

Themen

3. Checking the correctness of Memories

   Checker für Programme mit unzuverlässigem Speicher

   Stacks, Queues, RAM

   1-2 Person

4. Checking Linked Data Structures

   Selbiges wie oben für Linked Lists, Trees und Graphen

   1 Person

5. Reflections on the Pentium Division Bug

   Die Hardwareseite: Checker für Basisoperationen

   1 Person

Themen

6. Linear-time certifying recognition algorithms and forbidden induced subgraphs

   Zertifikate zur Erkennung von bestimmten Graphklassen.

   1-2 Personen

7. Efficient Planarity Testing / On the use of program checking in LEDA

   Wie können wir zertifizieren, ob ein Graph (nicht) planar ist?

   2 Personen

Themen

8. Program Checkers for Probability Generation
   Wie checken wir Programme, die Objekte angeblich nach einer Wahrscheinlichkeitsverteilung generieren?
   1 Person

9. Designing Checkers for Programs that Run in Parallel
   Erweiterung des Checkingframeworks auf parallel arbeitende Algorithmen.
   1-2 Person

Themen

10. Spot-Checkers
    Falls checking zu teuer ist, können wir auch mit einem abgeschwächten Modell annehmbare Ergebnisse erlangen?
    1-2 Personen

11. A Framework for the Verification of Certifying Computations Wir können auch Checkern nicht blind vertrauen. Diese zu verifizieren ist viel einfacher, als die Programme, die sie checken zu verifizieren.
    1 Person

Themen

12? Designing Programs that Check Their Work - Optional!

Welche Probleme sind überhaupt checkbar?

2 Personen

Achtung: Grundwissen über Komplexitätsklassen nötig!

Schwieriges Thema!

Anmerkungen zu Partnerarbeit

- Einige Themen sind auf 2 Personen ausgelegt
- Teilt den Vortrag sinnvoll auf
- Es ist schlecht, wenn eine Person alles Einfache macht
- Gemeinsame Ausarbeitungen bis 16 Seiten (oder 2×8)

Wöchentlichen Termin finden

Regelmäßiger Termin

- Montag 14:30
- Dienstag 10:30, 12:30, 14:30
- Donnerstag 10:30, 12:30

Wie geht es weiter?

- Wir schicken euch: Doodle, Papiere
- Ihr tragt euch bis Donnerstag ein und wählt 5 Prioritäten
- Ihr kümmert euch eigenständig um eine Literaturrechercheschulung
- Wir treffen uns am Donnerstag zur Themenvergabe (Termin folgt)