**Hochschule Karlsruhe**
University of Applied Sciences

Fakultät für **Informatik**
**und Wirtschaftsinformatik**

**Prof. Dr. Carsten Sinz**
**Lehrgebiet Software-Entwicklung**

# Übungsblatt 1 zur Vorlesung Practical SAT Solving

Ausgabe: 09.10.2023 – Abgabe: 23.10.2023 (3+4), 06.11.2023 (1+2)

### Aufgabe 1 (Graphfärbung durch inkrementelles SAT-Solving (7(+7) Punkte)):

Implementieren Sie ein Programm zur Lösung des Graphfärbungsproblems per inkrementellem SAT-Solving in C oder C++ unter Verwendung des IPASIR-Interfaces. Das Programm soll auf der Kommandozeile ein Argument entgegennehmen, den Dateinamen einer DIMACS-Graph-Datei. Für diesen soll dann die kleinste Zahl gefunden werden, für die eine Färbung der Knoten des Graphen existiert, so dass benachbarte Knoten unterschiedliche Farben annehmen. Ausgabe des Programms soll die Anzahl der benötigen Farben, sowie für jeden Knoten dessen Farbe.

Hinweise und Beispiele zur Implementierung eines inkrementellen Solvers können Sie auf der github-Seite `https://github.com/biotomas/ipasir` finden.

Testdaten sind verfügbar unter `http://mat.gsia.cmu.edu/COLOR/instances.html` (verwenden Sie die *.col-Instanzen, bei *.col.b-Dateien handelt es sich um ein Binärformat).

Der schnellste Solver erhält 7 Bonuspunkte.

### Aufgabe 2 (Sudoku Challenge (7(+7) Punkte)):

Schreiben Sie ein Programm (in einer Programmiersprache Ihrer Wahl) zur Lösung von verallgemeinerten Sudoku-Puzzles mit Hilfe eines SAT-Solvers.

Ein verallgemeinertes Sudoku-Puzzle der Ordnung $n$ ist ein $n^2 \times n^2$ großes Feld, das aus $n^2$ Unterblöcken der Größe $n \times n$ besteht. Die einzelnen Felder sollen so mit den Zahlen $1, \ldots, n^2$ gefüllt werden, dass

- in jeder Zeile jede Zahl genau einmal vorkommt,
- in jeder Spalte jede Zahl genau einmal vorkommt und
- in jedem Unterblock jede Zahl genau einmal vorkommt.

Bei dem 9x9-Standard-Soduko (`https://de.wikipedia.org/wiki/Sudoku`) handelt es sich um das verallgemeinerte Sudoku-Puzzel der Ordnung 3.

Benchmark-Dateien zum Testen Ihres Programms finden Sie unter `https://carstensinz.de/hka/files/sat/sudokus.zip`.

Das beste Programm (Anzahl und Laufzeit zur Lösung der Benchmark-Probleme) bekommt einen Bonus von 7 Punkten.

### Aufgabe 3 (Van-der-Waerden-Zahlen (3 Punkte)):

Berechnen Sie, wie viele Variablen und Klausen die SAT-Codierung aus der Vorlesung benötigt, um die Eigenschaft $W(2, k) > n$ zu prüfen.

**Aufgabe 4 (Pythagoräische Tripel (6 Punkte)):**

Finden Sie mit Hilfe eines SAT Solvers eine Lösung (d.h. eine Färbung) für die ersten 1000 Zahlen (1, 2, ..., 1000) des Pythagoräischen-Triple-Problems.

Schätzen Sie die Anzahl der Variablen und Klauseln ab für die SAT-Codierung aus der Vorlesung für das Pythagoräische-Triple-Problem bis zur Zahl $n$.