

Untersuchung verschiedener Kodierungen von speziellen Cardinality Constraints für SAT

Exposé
Universität Bremen
Fachbereich 3

Jil Tietjen
<jiltietj@informatik.uni-bremen.de>

Erstgutachter: Prof. Dr. Rolf Drechsler
Zweitgutachter: Prof. Dr. Rüdiger Ehlers
Betreuung: Prof. Dr. Rolf Drechsler & Oliver Keszöcze

Bremen, 30. November 2015

INHALTSVERZEICHNIS

1	Problemstellung	3
2	Ziel und Fragestellung der Arbeit	3
3	Die Vorgehensweisen und Methoden	3
4	verwendete Quellen und Materialien	4
5	Zeitplan der Bachelorarbeit	5

1 PROBLEMSTELLUNG

Im Zuge dieser Arbeit soll untersucht werden, inwiefern verschiedene Kodierungen die Cardinality Constraints in eine Klauselmenge hinsichtlich ihrer Laufzeit und ihres Speicherverbrauchs effizient verkleinern beziehungsweise vereinfachen. Hierbei wird ebenfalls der Sonderfall ≤ 1 betrachtet. Dafür kann eine CNF angegeben werden, deren Korrektheit einfach zu beweisen ist.

Es werden alle unten genannten Kodierungen miteinander verglichen. Die vereinfachte Klauselmenge wird in den z3-Solver gegeben, mit dessen Hilfe herausgefunden werden kann, ob für die Klauselmenge eine erfüllende Belegung existiert oder nicht.

2 ZIEL UND FRAGESTELLUNG DER ARBEIT

Es soll festgestellt werden, ob es mit Hilfe der Kodierungen wirklich einfacher geworden ist, herauszufinden, ob die Klauselmenge erfüllbar ist oder nicht. Ist mit den Kodierungen keine Verbesserung festzustellen, könnte es eventuell helfen die bestehenden Kodierungen abzuwandeln und zu verbessern. Durch diese Untersuchung kann die effizienteste Kodierung gefunden werden, mit der Cardinality Constraints als Klauselmenge dargestellt werden können. Zusätzlich soll betrachtet werden, wie sich der Spezialfall ≤ 1 im Allgemeinen für n Variablen und für 4, 5, 8 und 9 Variablen im Speziellen verhält, da sich hier Anwendungsfälle für Biochips ergeben.

3 DIE VORGEHENSWEISEN UND METHODEN

Folgende Kodierungen sollen für Pseudo Boolesche Constraints untersucht werden:

- naiv für ≤ 1 mit Aufzählungen und Ausschluss aller Paarungen
- Sinz (Sinz, 2005)
 - sequentieller Counter
 - paralleler Counter
- Bailleux (Bailleux u. Boufkhad, 2003)
 - sequentieller Counter (anhand von Knuth (Knuth, 2015))
 - paralleler Counter
- Sorting Networks
 - naiv

– Niklas Eén und Niklas Sörensson (Een u. Soerenesson, 2006)

Diese Kodierungen werden in Java implementiert, so dass deren Effizienz getestet werden kann. Für die Verfahren von Sinz und Bailleux wird bei der Programmierung das Vorgehen von Knuth angewendet. Als Testbench wird das k-Damen-Problem (Knuth) umgesetzt, da dies für das Finden einer möglichen Belegung der Klauseln adaptiert werden kann und es ein anerkanntes Problem in der Informatik ist. Eventuell werden weitere Beispiele von Knuth zum Testen benutzt.

4 VERWENDETE QUELLEN UND MATERIALIEN

Folgende Quellen bilden die Grundlage für diese Arbeit:

- [Bailleux u. Boufkhad 2003] BAILLEUX, O. ; BOUFKHAD, Y.: *Efficient CNF Encoding of Boolean Cardinality Constraints*. Principles and Practice of Constraint Programming, 2003
- [Een u. Soerenesson 2006] EEN, N. ; SOERENSSON, N.: *Translating Pseudo-Boolean Constraints into SAT*. Journal on Satisfiability, Boolean Modeling and Computation 2, 2006
- [Knuth 2015] KNUTH, D. E.: *The Art of Computer Programming*. <http://www-cs-faculty.stanford.edu/~uno/fasc6a.ps.gz>. Version: 2015
- [Sinz 2005] SINZ, C.: *Towards an Optimal CNF Encoding of Boolean Cardinality Constraints*. Symbolic Computation Group, WSI for Computer Science, 2005

5 ZEITPLAN DER BACHELORARBEIT

Zeitraum	Tätigkeit
bis 09.2.2016	Exposé, Literaturarbeit, Lösungsansätze finden/definieren
09.2.2016	Anmeldung Bachelorarbeit
09.2.2016 – 12.4.2016	Programmierung
12.4.2016 – 31.5.2016	Bachelorarbeit schreiben
31.5.2016 – 8.6.2016	Korrektur und Drucken/Binden
09.6.2016	Abgabe Bachelorarbeit

Tabelle 1: Zeitplan