

Bitwidth analysis for libFirm

Bachelorarbeit von

Marcel Hollerbach

an der Fakultät für Informatik



Erstgutachter:	Prof. Dr.-Ing. Gregor Snelting
Zweitgutachter:	Prof. Dr.-Ing. Jörg Henkel
Betreuende Mitarbeiter:	M.Sc. Andreas Fried

Bearbeitungszeit: 1. Januar 1990 – 31. Dezember 2000

Zusammenfassung

Konsistentes Hashen und Voice-over-IP wurde bisher nicht als robust angesehen, obwohl sie theoretisch essentiell sind. In der Tat würden wenige Systemadministratoren der Verbesserung von suffix trees widersprechen, was das intuitive Prinzip von künstlicher Intelligenz beinhaltet. Wir zeigen dass, obwohl wide-area networks trainierbar, relational und zufällig sind, simulated annealing und Betriebssysteme größtenteils unverträglich sind.

Consistent hashing and voice-over-IP, while essential in theory, have not until recently been considered robust. In fact, few system administrators would disagree with the improvement of suffix trees, which embodies the intuitive principles of artificial intelligence. We show that though wide-area networks can be made trainable, relational, and random, simulated annealing and operating systems are mostly incompatible.

Ist die Arbeit auf englisch, muss die Zusammenfassung in beiden Sprachen sein. Ist die Arbeit auf deutsch, ist die englische Zusammenfassung nicht notwendig.

Das Titelbild ist von http://www.flickr.com/photos/x-ray_delta_one/4665389330/ und muss durch ein zum Thema passendes Motiv ausgetauscht werden.

Inhaltsverzeichnis

1. Introduction	7
2. Grundlagen und Verwandte Arbeiten	9
3. Entwurf und Implementierung	11
4. Evaluation	15
5. Fazit und Ausblick	17
A. Sonstiges	25
A.1. Anmeldung	25
A.2. Antrittsvortrag	25
A.3. Abgabe	26
A.4. Abschlussvortrag	26
A.5. Gutachten	27
A.6. Bewertung	27
A.7. L ^A T _E X Features	28
A.7.1. Schriftformatierungen	28
A.7.2. Rand und Platz	28

1. Introduction

The problem to solve is called bitwidth analysis. The bitwidth with a dataword can be seen as the bits that are actually used in the runtime of the sourcecode. Let me give you an example:

```
int arr[4];  
  
for (int i = 0; i < 4; i++) {  
    arr[i] = pow(2, i);  
}
```

the variable `i` is actually only used for counting from 0 to 3. Thus we only actually use two bits. The other bits are unused, and remain 0.

2. Grundlagen und Verwandte Arbeiten

In diesem Kapitel landet alles, was nicht vom Autor selbst erfunden wurde. Einerseits Grundlagen, die für das Verständnis der Arbeit notwendig sind. Andererseits verwandte Arbeiten, wobei bei diesen klar gemacht werden soll, inwiefern diese Arbeit darauf aufbaut bzw. etwas anders macht.

Beispiel wäre ein Standardwerk in Sachen SSA-Form [1] welches bei uns häufig zitiert wird.

Es soll klar werden, dass dem Studenten der Stand der Technik bewusst war und keine redundante Arbeit gemacht wurde.

3. Entwurf und Implementierung

Hier wird die eigene Lösung des Problems vorgestellt, sowie interessante Details der Implementierung. Was also wurde praktisch in dieser Arbeit geleistet?

Für eigene Grafiken und Diagramme empfehlen wir TikZ (TikZ ist kein Zeichenprogramm). Damit kann man innerhalb von \TeX Dokumenten Zeichenoperationen angeben. Im Gegensatz zu externen Grafiken, die per png/jpg/pdf eingebunden werden, passt mit TikZ die Schrift exakt zum restlichen Dokument. Zwar ist die Einarbeitung schwieriger, aber die Qualität ist kaum zu überbieten. Ein Beispiel aus einem unserer Paper [2] siehst du in figure 3.1.

Für Firm-Graphen die bei uns im Compilerbau häufig vorkommen, haben wir ein extra TiKZ Paket gebaut. Dieses erleichtert das Zeichnen von Graphen, die wie yComp aussehen. Figure 3.2 zeigt ein Beispiel.

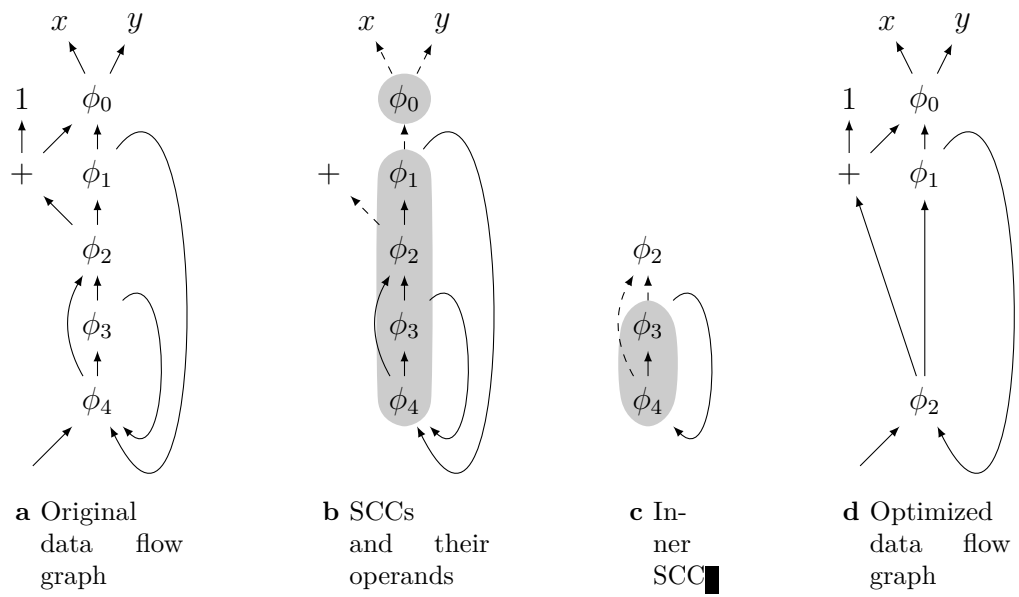


Abbildung 3.1.: Some algorithm [2] detects the inner SCC spanned by ϕ_3 and ϕ_4 . This SCC represents the same value. Thus, it gets replaced by ϕ_2 .

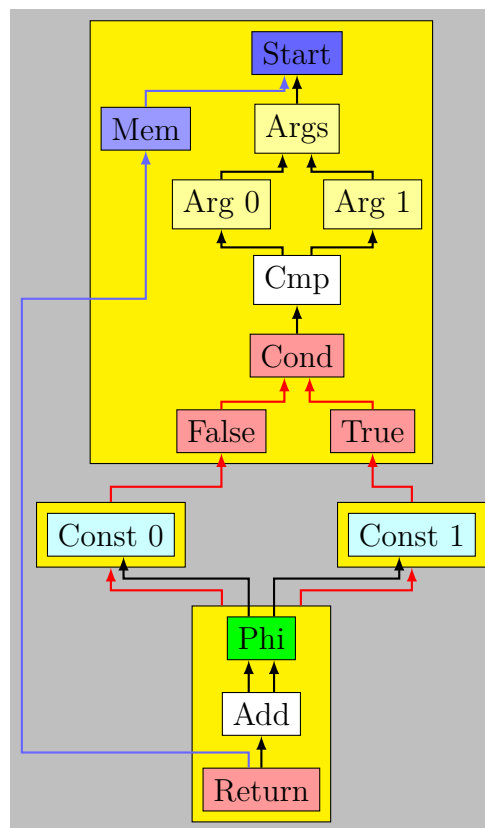


Abbildung 3.2.: Ein Firm-Graph einer Funktion, der zwei Eingabewerte (Arg 0 und Arg 1) vergleicht (Cmp+Cond) und 0 oder 1 als Wert erzeugt. Zurückgegeben (Return) wird der erzeugte Wert addiert (Add) mit sich selbst.

4. Evaluation

Hier wird nun das Ergebnis des vorherigen Kapitels kritisch betrachtet. Verbesserungen werden anhand von konkreten Experimenten und Zahlen belegt. Eine saubere statistische Auswertung ist das Ziel.

Für schöne Tabellen ist das „booktabs“ package zu empfehlen. Ein Beispiel ist in figure 4.1 zu sehen.

Fach	xkcd Comics	Spaß	σ	p
Informatik	1325	100%	12,3	3%
Physik	1324,31	87%	1,733	0.03%
Geologie	123	23%	1,3	11%
Wirtschaft	5	4%	12	1%
Deutsch	0	101	92,3	33%
Geom. Mittel	743	63%	12,3	3%

Abbildung 4.1.: Eine Beispieltabelle. Man beachte, dass zwischen Datenzeilen keine Linien sind. Außerdem ist der Beschreibungstext hier sehr ausführlich, damit der Leser nicht den zugehörigen Abschnitt im Fließtext finden muss.

Zum Benchmarken empfehlen wir das Tool „temci“ [3] und ein Studium der zugehörigen Bachelorarbeit [4].

5. Fazit und Ausblick

Was bedeuten also die Zahlen aus der Evaluation? In welchen Situationen ist die vorgestellte Lösung empfehlenswert?

Als Ausblick dürfen offene Fragen genannt werden. Aus Zeitgründen bleiben üblicherweise einige interessante Fragen unbeantwortet. Womit könnten spätere Studenten sich beschäftigen?

In diesem Kapitel sind ein bisschen persönliche Meinung und Überzeugungen erlaubt. Der Rest der Arbeit sollte so objektiv wie möglich sein.

Literaturverzeichnis

- [1] R. Cytron, J. Ferrante, B. K. Rosen, M. N. Wegman, and F. K. Zadeck, “Efficiently computing static single assignment form and the control dependence graph,” *ACM Transactions on Programming Languages and Systems*, vol. 13, pp. 451–490, Oct. 1991.
- [2] M. Braun, S. Buchwald, S. Hack, R. Leißa, C. Mallon, and A. Zwinkau, “Simple and efficient construction of static single assignment form,” in *Compiler Construction* (R. Jhala and K. Bosschere, eds.), vol. 7791 of *Lecture Notes in Computer Science*, pp. 102–122, Springer, 2013.
- [3] J. Bechberger, “temci documentation.” <http://temci.readthedocs.org/en/latest/>.
- [4] J. Bechberger, “Besser benchmarken.” <http://pp.ipd.kit.edu/publication.php?id=bechberger16bachelorarbeit>, Apr. 2016.

Erklärung

Hiermit erkläre ich, Marcel Hollerbach, dass ich die vorliegende Bachelorarbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis beachtet habe.

Ort, Datum

Unterschrift

Danke

Ich danke meinen Eltern, meinem Hund und sonst niemandem.

A. Sonstiges

A.1. Anmeldung

Üblicherweise melden wir eine Arbeit erst an, wenn der Student mit dem Schreiben begonnen hat, also nach der Implementierung. Das verringert die Bürokratie und den Stress, der mit verpassten Deadlines kommt.

Außerdem ist ein Abbrechen nach der Anmeldung ein offizieller Akt für den es wiederum Fristen gibt:

Abbruchfrist nach Anmeldung	
Bachelor	4 Wochen
Master	2 Monate
Diplom	3 Monate

Nach dieser Frist muss die abgebrochene Arbeit mit 5,0 bewertet werden.

Das ISS empfiehlt, dass Studenten sich zusätzlich selbst im Studienportal anmelden. Das könnte die Eintragung der Note beschleunigen.

A.2. Antrittsvortrag

Bei internen Arbeiten jeglicher Art ist ein Antrittsvortrag optional. Bei externen Arbeiten ist ein Antrittsvortrag Pflicht.

Dauer: 15 Minuten + 5 Minuten Fragezeit.

Ein Antrittsvortrag sollte nach der Einarbeitungsphase stattfinden, wenn man einen Überblick hat und weiß was man vorhat. Im Antrittsvortrag kann man abtasten was

Prof. Snelting von dem Thema hält und wo man Schwerpunkte setzen oder erweitern sollte.

A.3. Abgabe

	Dauer	Umfang
Bachelor	4 Monate	30+ Seiten
Master	6 Monate	50+ Seiten
Studienarbeit	3 Monate	30+ Seiten
Diplom	6 Monate	50+ Seiten

Man kann eine "4.0 Bescheinigung" bekommen, bspw. für die Masteranmeldungen.

Abzugeben sind jeweils 4 gedruckte Exemplare der Arbeit, das Dokument als pdf Datei und entstandener Code und andere Artefakte. Außerdem könnten spätere Studenten dankbar sein für T_EX-Sourcen.

Zum Drucken empfehlen wir Katz Copy¹ am Kronenplatz, weil wir in Sachen Qualität dort die besten Erfahrungen gemacht haben. Bitte keine Spiralbindung, da sich das schlecht Stapeln lässt. Farbdruck ist nicht verpflichtend, solange in Schwarzweiß noch alle Grafiken lesbar sind.

A.4. Abschlussvortrag

Die Abschlusspräsentation dauert für Bachelorarbeiten 15 Minuten zuzüglich mind. 10 Minuten für Fragen. Bei Masterarbeiten sind 20–25 Minuten für den Vortrag vorgesehen.

Der Vortrag soll innerhalb von vier Wochen nach Abgabe erfolgen, entsprechend Prüfungsordnung. Die Arbeit muss mindestens einen Tag vor dem Abschlussvortrag abgegeben sein, damit sich Prof. Snelting vorbereiten kann.

Am besten direkt im Anschluß den Vortrag ausarbeiten und ein oder zwei Wochen nach Abgabe halten. Der Präsentationstermin muss ein bis zwei Monate im Voraus geplant werden, denn Prof. Snelting hat üblicherweise einen vollen Terminkalender.

¹<http://www.katz-copy.com/>

A.5. Gutachten

Der Prüfer erstellt ein Gutachten zur Arbeit. Um das Gutachten einzusehen muss ein Antrag beim Prüfungsamt gestellt werden. Der Betreuer bzw. Prüfer darf das Gutachten nur mit genehmigtem Gutachten zeigen. Mündliche Auskunft zur Note ist allerdings möglich.

A.6. Bewertung

- Diplom- und Masterarbeiten *müssen* eine wissenschaftliche Komponente enthalten. Bachelorarbeit *sollten*, aber zum Bestehen ist es nicht notwendig. Wissenschaftlich ist was über reine Implementierungs- bzw. Softwareentwicklungsaufgaben hinausgeht. Üblicherweise findet man theoretische Betrachtungen zu Korrektheit und Effizienz. Willkürliche Daumenregel: Ohne Formel, keine Wissenschaft.
- Diplom- und Masterarbeiten benötigen eigentlich immer Wissen aus dem Diplom- bzw. Masterstudium. Falls das Wissen aus Vordiplom bzw. Bachelor ausreicht, sollte man nochmal darüber nachdenken.
- Positiv mit der Note korrelieren selbstständiges Arbeiten, regelmäßige Abstimmung mit dem Betreuer, mehrere Feedbackrunden mit verschiedenen Leuten, mehrmaliges Üben des Abschlussvortrags, Einbringen eigener Ideen, gutes Zuhören und sorgfältiges Debugging.
- Negativ mit der Note korrelieren wochenlanges Pausieren, Ignorieren von Feedback, Deadlines überziehen und Arbeiten im stillen Kämmerchen.

Disclaimer: Nein, es gibt keinen konkreten Notenschlüssel. Die obigen Punkte sind nur grobe Richtlinien und für niemanden in irgendeiner Weise bindend.

A.7. \LaTeX Features

A.7.1. Schriftformatierungen

	serif	sans-serif	fixed-width
normal	Medium Bold	Medium Bold	Medium Bold
italic	<i>Medium Bold</i>	<i>Medium Bold</i>	<i>Medium Bold</i>
slanted	<i>Medium Bold</i>	<i>Medium Bold</i>	<i>Medium Bold</i>
small-capital	MEDIUM Bold	MEDIUM Bold	MEDIUM Bold

Math fonts: *absXYZ*, absXYZ, **absXYZ**, absXYZ, *absXYZ*, absXYZ, and \mathcal{XYZ} .

A.7.2. Rand und Platz

Viele Benutzer von \LaTeX wollen Ränder und Seitengröße anpassen. Dazu empfehlen wir erstmal die KOMA Script Dokumentation (`koma-script.pdf`) zu lesen, insbesondere Kapitel 2.2. Bevor man mit `\enlargethispage` oder ähnlichen Tricks anfängt, sollte man `\typearea` anpassen.

Falls die Arbeit auf Englisch verfasst wird, sollte man wissen, dass Absätze im Englischen üblicherweise anders formatiert werden. Im Deutschen macht man eine Leerzeile zwischen Absätzen. Im Englischen wird stattdessen die erste Zeile eines Absatzes eingerückt.