ProfNet PlagiatService

-Prüfbericht-

für
Prof. Dr. Debora Weber-Wulff
Uni Kiel

Münster, den 27.07.2015

ProfNet
Institut für Internet-Marketing

ProfNet PlagiatService - Zusammenfassung

- Autor: Prof. Dr. Debora Weber-Wulff
- Titel: Contributions to Mechanical Pr ...
- Typ: Dissertation
- Abgabetermin: 31.12.1997
- Hochschule: Uni Kiel
- Fachbereich: Technische Fakultät
- Studiengang:
- Fachrichtung: Informatik
- 1. Gutachter: Prof. Dr. Hans Langmaack
- 2. Gutachter: Prof. Dr. Robert S. Boyer
- Prüfdatum: 27.07.2015

Dateigröße	491.649	Abbildungsverzeichnis	☐
Seiten	193	Abkürzungsverzeichnis	☐
Absätze	617	Anhang	☐
Sätze	3.094	Eidesstattliche Erklärung	☐
Wörter	70.300	Inhaltsverzeichnis	☒
Zeichen	357.963	Literaturverzeichnis	☒
Abbildungen	0	Quellenverzeichnis	☐
Tabellen	0	Stichwortverzeichnis	☒
Fußnoten	30	Sperrvermerk	☐
Literatur	0	Symbolverzeichnis	☐
Wörter (netto)	66.732	Tabellenverzeichnis	☐
		Vorwort	☐

Analysetyp	Indizien
Bauernopfer-Satz	23
Bauernopfer-Wort	1
Teilplagiat	1
Zitat-Veränderung	1
Zitierungsfehler	1

Anteil Fremdtexte (netto): 1 % (705 von 66.732 Wörtern)

Phrase-allgemein	97
Phrase-fachspezifisch	64
Zitat-Fremdtext-ohne Quelle	2
Zitat-im Text-ohne Quelle	16

Anteil Fremdtexte (brutto): 3 % (2.085 von 70.300 Wörtern)

28% Gesamtplagiatswahrscheinlichkeit

Alle Ergebnisse dieses Reports werden von der Software automatisch berechnet, so dass alle Angaben jeweils den Stand der Software-Entwicklung wiedergeben.

ProfNet PlagiatService - Ergebnis Textanalyse (alle Analysen)

Kriterium	Dimension	Prüfdokument	Erstprüfer	Fachbereich	Hochschule	Fachrichtung	Hausarbeiten	Seminararbeiten	Bachelor Thesen	Diplomarbeiten	Master Thesen	Dissertationen	Habilitationen	alle
Dokumente	Anzahl	1	1	2	6	421	218	243	299	1874	243	21000	175	183397
Abbildungen	Anzahl (Durchschnitt)	0	0	51	17	6	2	2	7	7	4	5	8	2
Absätze	Anzahl (Durchschnitt)	617	617	519	664	417	120	113	260	376	302	564	813	373
Fußnoten	Anzahl (Durchschnitt)	30	30	28	126	36	39	48	44	62	50	109	139	43
Literatur	Anzahl (Durchschnitt)	0	0	0	1	1	1	10	13	5	1	6	2	8
Sätze	Anzahl (Durchschnitt)	3094	3094	2379	3253	1752	507	489	1039	1543	1321	2444	3631	1313
Seiten	Anzahl (Durchschnitt)	193	193	182	167	116	35	30	75	108	94	164	207	78
Tabellen	Anzahl (Durchschnitt)	0	0	5	2	3	1	1	2	3	3	4	3	1
Wörter	Anzahl (Durchschnitt)	70300	70300	47873	56478	27749	8434	7655	16323	24039	21693	39357	58815	21661
Zeichen	Anzahl (Durchschnitt)	357963	357963	268399	367996	176629	55833	50661	107781	159629	138849	261386	404247	141446
Zitate	Anzahl (Durchschnitt)	164	164	139	447	172	82	61	109	165	143	224	376	136

Die statistischen Ergebnisse der Textanalyse des Prüfdokumentes werden mit den Ergebnissen aller analysieren Texte verglichen.

ProfNet PlagiatService - Ergebnis Textvergleich (alle Vergleiche)

Kriterium	Dimension	Prüfdokument	Erstprüfer	Fachbereich	Hochschule	Fachrichtung	Hausarbeiten	Seminararbeiten	Bachelor Thesen	Diplomarbeiten	Master Thesen	Dissertationen	Habilitationen	alle
Dokumente	Anzahl	1	1	2	6	355	69	33	280	1821	211	19840	164	37473
Mischpl.-eine	Anzahl (Durchschnitt)	0	0	0	0	1	1	6	1	1	1	3	3	3
Teilplagiat	Anzahl (Durchschnitt)	1	1	2	5	9	6	9	10	13	13	25	24	23
Mischpl.-mehrere	Anzahl (Durchschnitt)	0	0	0	1	2	2	2	2	3	3	6	4	6
Zitierungsfehler	Anzahl (Durchschnitt)	1	1	1	5	2	1	9	2	2	3	3	6	3
Bauernopfer	Anzahl (Durchschnitt)	0	0	0	1	2	1	0	1	2	2	2	3	2

28% Gesamtplagiatswahrscheinlichkeit

Die Textvergleichsergebnisse des Prüfdokumentes werden mit allen analysierten Texten verglichen. Die Plagiatswahrscheinlichkeit wird grob vom Programm automatisch berechnet.

Textstelle (Prüfdokument) S. 2

correct systems that were proven with the aid of a mechanical verification system which itself has been proven correct. No such mechanical verification system exists as yet. However, there are a number of verification systems such as the Boyer-Moore theorem prover NQTHM, the results of which are of high believability and integrity. As was seen in the preparation of this dissertation, it is not a trivial task to formulate the theorems that will state the correctness of a system

3% Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

provers that allow interaction that is user oriented and not system developer oriented. 1 Introduction Numerous papers have been written to describe challenge problems solved using the Boyer- Moore theorem prover, Nqthm' [3, 5].2 There are also papers on new tools and functionalities 'named after a directory on which the system was originally developed2 This system, and its Pc-Nqthm

- 1 Interaction with the Boyer-Moore Th..., 1994, S. 3

Textstelle (Prüfdokument) S. 6

partial parse trees. Looking up a state/symbol pair in an LR(1) parsing table returns an action, either shift, reduce, accept, shiftreduce, or error. The parser chooses the action determined by the state on the top of the state stack and the next symbol (i.e. token) in the input stream. The shiftreduce action does not seem to be motivated by proof concerns, but rather by an attempt to optimize the parser somewhat. He formulated a relation, slrrel, between the state stack

● **3%** Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

about the parse is saved in a state stack, and at each step the next parsing action is selected on the basis of the top element of the state stack and the next symbol in the input. A parsing action may either consume one symbol from the input and place a new state on the stack (a shift action), or it may

- **2** Bauer: Compiler Construction, 1976, S. 88

Textstelle (Prüfdokument) S. 9

done in the area of compiler construction, as they apply to the work at hand. The Boyer-Moore logic, as discussed in [BM79, BM88] permits the statement of recursive, side-effect free functions which are stated as s-expressions in the LISP-like language of the prover. Theorems, usually stating the equality of two terms or the implication of one term from another, can also be represented as s-expressions. They can be proved correct with the Boyer-Moore prover NQTHM 2 for functions that have already been defined in the current session, using the transformations described below. Definitions, lemmata,

● **10%** Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

of the Boyer-Moore theorem prover NQTHM and the computational logic is uses. The Boyer-Moore logic is a collection of recursive, side-effect free functions stated as s-expressions in the LISP-like language of the prover. Lemmata, usually stating the equality of two terms or the implication of one term from another, are also represented as s-expressions. A publically available copy of Bob Boyer and J Moore s theorem prover NQTHM or Matt Kaufmann s interactive proof checker version PC-NQTHM

- **3** Dagstuhl Seminar on High Integrity ..., 1995, S. 31

Textstelle (Prüfdokument) S. 9

NQTHM2 for functions that have already been defined in the current session, using the transformations described below. Definitions, lemmata, and other rules introduced during a session are referred to in the logic as 'events'. 2.6.1 Proof Method The theorem prover employs eight basic transformations when attempting to prove a lemma [BM88]: "decision procedures for propositional calculus, equality, and linear arithmetic term rewriting, based on axioms, definitions and previously proved lemmata application of verified user-supplied simplifiers called 'metafunctions' variable renaming to eliminate ' destructive' functions in favor of 'constructive' ones heuristic use of equality hypotheses generalization by the replacement of terms by type-restricted variables elimination of apparently irrelevant hypotheses mathematical induction" The theorem prover also contains many heuristics to orchestrate these basic techniques. No further detail on the mechanics of the proving techniques will be given here, but a short description of the syntax of the logic will enable the reader to understand the events that are presented in this thesis. A tiny example proof is included in order to present the 'flavor' of the proofs. 2.6.2 Syntax The syntax of NQTHM is very similar to that of Pure Lisp, but

● **3%** Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

theorem prover is a computer program that takes as input a term in the logic and repeatedly transforms it in an effort to reduce it to non-F. The theorem prover employs eight basic transformations: decision procedures for propositional calculus, equality, and linear arithmetic; term rewriting based on axioms, definitions and previously proved lemmas; application of verified user-supplied simplifiers

short example proof, the associativity of times, was printed in a handout and discussed in detail, examing the different transformations used in attempting to prove a lemma: decision procedures for propositional calculus, equality, and linear arithmetic; term rewriting based on axioms, definitions and previously proved lemmata; application of verified user-supplied simplifiers called " metafunctions"; variable renaming to eliminate "destructive" functions in favor of "constructive" ones; heuristic use of equality hypotheses; generalization by the replacement of terms by type-restricted variables; elimination of apparently irrelevant hypotheses; and mathematical induction. A list of theorems that have been proven was discussed, they include Mathematics Prime factorization uniqueness Unsolvability of the halting problem RSA public

- **4** A verified operating system kernel, 1987, S. 97
- **3** Dagstuhl Seminar on High Integrity ..., 1995, S. 31

Textstelle (Prüfdokument) S. 13

it states the induction scheme (in this case on the construction of natural numbers). Note that two of the goals pushed are important theorems, the right zero of times and the distributivity of plus through times. >(prove-lemma commutativity-of-times (rewrite) (equal (times x z) (times z x))) Give the conjecture the name *1. We will appeal to induction. Two inductions are suggested by terms in the conjecture, both of which are flawed. We limit our consideration to the two suggested by the largest number of non-primitive recursive functions in the conjecture. Since both of these are equally likely, we will choose arbitrarily. We will induct according to the following scheme: (AND (IMPLIES (ZEROP X) (p X Z)) (IMPLIES (AND (NOT (ZEROP X)) (p (SUB1 X) Z)) (p X Z))). Linear arithmetic, the lemma COUNT-NUMBERP, and the definition of ZEROP inform us that the measure (COUNT X) decreases according to the well-founded relation LESSP in each induction step of the scheme. The above induction scheme produces the following two new conjectures: Case 2. (IMPLIES (ZEROP X) (EQUAL (TIMES X Z) (TIMES Z X))). This simplifies, expanding the functions ZEROP, EQUAL, and TIMES, to the following two new conjectures: Case 2.2. (IMPLIES (EQUAL X 0) (EQUAL 0 (TIMES Z 0))). This again simplifies, obviously, to: (EQUAL 0 (TIMES Z 0)), which we will name +1.1. Case 2.1. (IMPLIES (NOT (NUMBERP X)) (EQUAL 0 (TIMES Z X))). Name the above subgoal +1.2. Case 1. (IMPLIES (AND (NOT (ZEROP X)) (EQUAL (TIMES (SUB1 X) Z) (TIMES Z (SUB1 X)))) (EQUAL (TIMES X Z) (TIMES Z X))). This simplifies, opening up ZEROP and TIMES, to the new conjecture: (IMPLIES (AND (NOT (EQUAL X 0)) (NUMBERP X) (EQUAL (TIMES (SUB1 X) Z) (TIMES Z (SUB1 X)))) (EQUAL (PLUS Z (TIMES Z (SUB1 X))) (TIMES Z X))). Applying the lemma SUB1-ELIM, replace X by (ADD1 V) to eliminate (SUB1 X). We employ the type restriction lemma noted when SUB1 was introduced to restrict the new variable. This produces the new conjecture: (IMPLIES (AND (NUMBERP V) (NOT (EQUAL (ADD1 V) 0)) (EQUAL (TIMES V Z) (TIMES Z V))) (EQUAL (PLUS Z (TIMES Z V)) (TIMES Z (ADD1 V)))), which further simplifies, obviously, to: (IMPLIES (AND (NUMBERP V) (EQUAL (TIMES V Z) (TIMES Z V))) (EQUAL (PLUS Z (TIMES V Z)) (TIMES Z (ADD1 V)))). We now use the above equality hypothesis by substituting (TIMES Z V) for (TIMES V Z) and throwing away the equality. This generates: (IMPLIES (NUMBERP V) (EQUAL (PLUS Z (TIMES Z V)) (TIMES Z (ADD1 V)))). Name the above subgoal +1.3. We will appeal to induction. There are three plausible inductions.

100% Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

plus y z)) (plus (times x y) (times x z)))) (prove-lemma times-addl (rewrite) (equal (times x (add1 y)) (if (numberp y) (plus x (times x y)) (fix x)))) (prove-lemma commutativity-of-times (rewrite) (equal (times x y) (times y x))) (prove-lemma commutativity2-of-times (rewrite) (equal (times x (times y z)) (times y (times x z)))) (prove-lemma associativity-of-times (rewrite) (equal (times (times x y) z) (times x (times y z)))) (prove-lemma equal-times-O (rewrite) (equal (equal (times x y) O) (or (zerop x) (zerop y)))) (prove-

Initialized with (BOOT-STRAP NQTHM) on November 9, 1992 08:32:18. >(load "checkpoints.lisp") Loading checkpoints.lisp Finished loading checkpoints.lisp T >(prove-lemma times-comm (rewrite) (equal (times x y) (times y x))) Give the conjecture the name *1. !!CHECKPOINT LEVEL 1; PRIORITY 4; ID 3 <<< Checkpoint 2, obtained from C-t n>>> We will appeal to induction. Two inductions are suggested by terms in the conjecture, both of which are flawed. We limit our consideration to the two suggested by the largest number of nonprimitive recursive functions in the conjecture. Since both of these are equally likely, we will choose arbitrarily. We will induct according to the following scheme: (AND (IMPLIES (ZEROP X) (p X Y)) (IMPLIES (AND (NOT (ZEROP X)) (p (SUB1 X) Y)) (p X Y))). Linear arithmetic, the lemma COUNT-NUMBERP, and the definition of ZEROP inform us that the measure (COUNT X) decreases according to the well-founded relation LESSP in each induction step of the scheme. The above induction scheme produces the following two new conjectures: Case 2. (IMPLIES (ZEROP X) 7(EQUAL (TIMES X Y) (TIMES Y X))).This simplifies, expanding the functions ZEROP, EQUAL, and TIMES, to the following two new conjectures: Case 2.2. (IMPLIES (EQUAL X 0) (EQUAL 0 (TIMES Y 0))). This again simplifies, obviously, to: !!CHECKPOINT LEVEL 1; PRIORITY 2; ID 7 <<< Checkpoint 3>>> Case 2.1. (IMPLIES (NOT (NUMBERP X)) (EQUAL 0 (TIMES Y X))). Name the above subgoal *1.2. Case 1. (IMPLIES (AND (NOT (ZEROP X)) (EQUAL (TIMES (SUB1 X) Y) (TIMES Y (SUB1 X)))) (EQUAL (TIMES X Y) (TIMES Y X))). This simplifies, opening up ZEROP and TIMES, to the new conjecture: !!CHECKPOINT LEVEL 3; PRIORITY 0; ID 12 <<< Checkpoint 6>>> (IMPLIES (AND (NUMBERP Z) (

- **5** Hunt FM8501, 1994, S. 148
- **6** An Assistant for Reading Nqthm Proo..., 1992, S. 6

ProfNet Institut für Internet-Marketing

Textstelle (Prüfdokument) S. 13

However, they merge into one likely candidate induction. We will induct according to the following scheme: (AND (IMPLIES (ZEROP Z) (p Z V)) (IMPLIES (AND (NOT (ZEROP Z)) (p (SUB1 Z) V)) (p Z V))). Linear arithmetic, the lemma COUNT-NUMBERP, and the definition of ZEROP establish that the measure (COUNT Z) decreases according to the well-founded relation LESSP in each induction step of the scheme. The above induction scheme leads to the following two new formulas: Case 2. (IMPLIES (AND (ZEROP Z) (NUMBERP V)) (EQUAL (PLUS Z (TIMES Z V)) (TIMES Z (ADD1 V)))). This simplifies, expanding the functions ZEROP, EQUAL, TIMES , PLUS, and NUMBERP, to: T. Case 1. (IMPLIES (AND (NOT (ZEROP Z)) (EQUAL (PLUS (SUB1 Z) (TIMES (SUB1 Z) V)) (TIMES (SUB1 Z) (ADD1 V))) (NUMBERP V)) (EQUAL (PLUS Z (TIMES Z V)) (TIMES Z (ADD1 V)))). This simplifies, applying SUB1-ADD1, and opening up ZEROP, TIMES, and PLUS, to the formula: (IMPLIES (AND (NOT (EQUAL Z 0)) (NUMBERP Z) (EQUAL (PLUS (SUB1 Z) (TIMES (SUB1 Z) V)) (TIMES (SUB1 Z) (ADD1 V))) (NUMBERP V)) (EQUAL (PLUS Z V (TIMES (SUB1 Z) V)) (ADD1 (PLUS V (TIMES (SUB1 Z) (

● **30%** Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

EQUAL (TIMES Z Y) (TIMES Y Z))) (EQUAL (PLUS Y (TIMES Y Z)) (TIMES Y (ADD1 Z)))). We now use the above equality hypothesis by substituting (TIMES Z Y) for (TIMES Y Z) and throwing away the equality. This generates: !!CHECKPOINT LEVEL 3; PRIORITY 2; ID 15 <<< Checkpoint 5>>> (IMPLIES (NUMBERP Z) (EQUAL (PLUS Y (TIMES Z Y)) (TIMES Y (ADD1 Z)))). 8Name the above subgoal *1.3.!!CHECKPOINT LEVEL 4; PRIORITY 4; ID 16 <<< Last checkpoint (8)>>> We will appeal to induction. There are three plausible inductions. They merge into two likely candidate inductions. However, only one is unflawed. We will induct according to the following scheme: (AND (IMPLIES (ZEROP Z) (p Y Z)) (IMPLIES (AND (NOT (ZEROP Z)) (p Y (SUB1 Z))) (p Y Z))). Linear arithmetic, the lemma COUNT-NUMBERP, and the definition of ZEROP establish that the measure (COUNT Z) decreases according to the well-founded relation LESSP in each induction step of the scheme. The above induction scheme leads to the following two new formulas: Case 2. (IMPLIES (AND (ZEROP Z) (NUMBERP Z)) (EQUAL (PLUS Y (TIMES Z Y)) (TIMES Y (ADD1 Z)))). This simplifies, expanding the functions ZEROP, NUMBERP, EQUAL, TIMES, and ADD1, to: (IMPLIES (EQUAL Z 0) (EQUAL (PLUS Y 0) (TIMES Y 1))), which again simplifies, trivially, to: !!CHECKPOINT LEVEL 4; PRIORITY 2; ID 20 <<< Checkpoint Lfounded relation LESSP in each induction step of the scheme. The above induction scheme leads to the following two new formulas: Case 2. (IMPLIES (AND (ZEROP Z) (NUMBERP Z)) (EQUAL (PLUS Y (TIMES Z Y)) (TIMES Y (ADD1 Z)))). This simplifies, expanding the functions ZEROP, NUMBERP, EQUAL, TIMES, and ADD1, to: (IMPLIES (EQUAL Z 0) (EQUAL (PLUS Y 0) (TIMES Y 1))), which again simplifies, trivially, to: !!CHECKPOINT LEVEL 4; PRIORITY 2; ID 20 <<< Checkpoint 7>>> (

- **6** An Assistant for Reading Nqthm Proo..., 1992, S. 7
- **6** An Assistant for Reading Nqthm Proo..., 1992, S. 8

Textstelle (Prüfdokument) S. 14

SUB1-ADD1, and opening up ZEROP, TIMES, and PLUS, to the formula: (IMPLIES (AND (NOT (EQUAL Z 0)) (NUMBERP Z) (EQUAL (PLUS (SUB1 Z) (TIMES (SUB1 Z) V)) (TIMES (SUB1 Z) (ADD1 V))) (NUMBERP V)) (EQUAL (PLUS Z V (TIMES (SUB1 Z) V)) (ADD1 (PLUS V (TIMES (SUB1 Z) (ADD1 V)))))). This again simplifies, using linear arithmetic, to: T. That finishes the proof of +1.3. So let us turn our attention to: (IMPLIES (NOT (NUMBERP X)) (EQUAL 0 (TIMES Z X))), named +1.2 above. We will try to prove it by induction. There is only one suggested induction. We will induct according to the following scheme: (AND (IMPLIES (ZEROP Z) (p Z X)) (IMPLIES (AND (NOT (ZEROP Z)) (p (SUB1 Z) X)) (p Z X))). Linear arithmetic, the lemma COUNT-NUMBERP, and the definition of ZEROP can be used to establish that the measure (COUNT Z) decreases according to the well-founded relation LESSP in each induction step of the scheme. The above induction scheme leads to the following two new formulas: Case 2. (IMPLIES (AND (ZEROP Z) (NOT (NUMBERP X))) (EQUAL 0 (TIMES Z X))). This simplifies, opening up the definitions of ZEROP, EQUAL, and TIMES, to: T. Case 1. (IMPLIES (AND (NOT (ZEROP Z)) (EQUAL 0 (TIMES (SUB1 Z) X)) (NOT (NUMBERP X))) (EQUAL 0 (TIMES Z X))). This simplifies, unfolding the definitions of ZEROP, TIMES, NUMBERP, PLUS, and EQUAL, to: T. That finishes the proof of +1.2. So we now return to: (EQUAL 0 (TIMES Z 0)), named +1.1 above. We will appeal to induction. There is only one plausible induction.

32% Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

and IMPLIES, to the following two new formulas: Case 2. (IMPLIES (AND (LESSP (LENGTH LST) (LENGTH LST)) (PROPERP LST)) (EQUAL (ROTATE (LENGTH LST) LST) LST)). However this again simplifies, using linear arithmetic, to: T. Case 1. (IMPLIES (AND (EQUAL (ROTATE (LENGTH LST) LST) (APPEND (NTHCDR (LENGTH LST) LST) (FIRSTN (LENGTH LST) LST))) (PROPERP LST)) (EQUAL (ROTATE (LENGTH LST) LST) LST)).

ID 16 <<< Last checkpoint (8)>>> We will appeal to induction. There are three plausible inductions. They merge into two likely candidate inductions. However, only one is unflawed. We will induct according to the following scheme: (AND (IMPLIES (ZEROP Z) (p Y Z)) (IMPLIES (AND (NOT (ZEROP Z)) (p Y (SUB1 Z))) (p Y Z))). Linear arithmetic, the lemma COUNT-NUMBERP, and the definition of ZEROP establish that the measure (COUNT Z) decreases according to the well-founded relation LESSP in each induction step of the scheme. The above induction scheme leads to the following two new formulas: Case 2. (IMPLIES (AND (ZEROP Z) (NUMBERP Z)) (EQUAL (PLUS Y (TIMES Z Y)) (TIMES Y (ADD1 Z)))). This simplifies, expanding the functions ZEROP, NUMBERP, EQUAL, TIMES, and ADD1, to: (IMPLIES (EQUAL Z 0) (EQUAL (PLUS Y 0) (TIMES Y 1))), which again

LEVEL 1; PRIORITY 2; ID 7 <<< Checkpoint 3>>> Case 2.1. (IMPLIES (NOT (NUMBERP X)) (EQUAL 0 (TIMES Y X))). Name the above subgoal *1.2. Case 1. (IMPLIES (AND (NOT (ZEROP X)) (EQUAL (TIMES (SUB1 X) Y) (TIMES Y (SUB1 X)))) (EQUAL (TIMES X Y) (TIMES Y X))). This simplifies , opening up ZEROP and TIMES, to the new conjecture: !!CHECKPOINT LEVEL 3; PRIORITY 0; ID 12 <<< Checkpoint Lour main theorem right away. Maybe we ll be lucky. >(prove-lemma rotate-length () (implies (listp lst) (equal (rotate (length lst) lst) lst))) Name the conjecture *1. We will appeal to induction. There is only one plausible induction. We will induct according to the following scheme: (AND (IMPLIES (AND (LISTP LST) (p (CDR LST))) (p LST)) (IMPLIES (NOT (LISTP LST)) (p LST))). Linear arithmetic and the

ID 16 <<< Last checkpoint (8)>>> We will appeal to induction. There are three plausible inductions. They merge into two likely candidate inductions.

- **7** An Instructive Example for Beginnin..., 1990, S.
- **6** An Assistant for Reading Nqthm Proo..., 1992, S. 8
- **6** An Assistant for Reading Nqthm Proo..., 1992, S. 7
- **7** An Instructive Example for Beginnin..., 1990, S.

Textstelle (Prüfdokument) S. 15

We will induct according to the following scheme: (AND (IMPLIES (ZEROP Z) (p Z)) (IMPLIES (AND (NOT (ZEROP Z)) (p (SUB1 Z))) (p Z))). Linear arithmetic, the lemma COUNT-NUMBERP, and the definition of ZEROP establish that the measure (COUNT Z) decreases according to the well-founded relation LESSP in each induction step of the scheme. The above induction scheme generates the following two new formulas : Case 2. (IMPLIES (ZEROP Z) (EQUAL 0 (TIMES Z 0))). This simplifies, opening up the definitions of ZEROP, TIMES, and EQUAL, to: T. Case 1. (IMPLIES (AND (NOT (ZEROP Z)) (EQUAL 0 (TIMES (SUB1 Z) 0))) (EQUAL 0 (TIMES Z 0))). This simplifies, expanding the definitions of ZEROP, TIMES, PLUS, and EQUAL, to : T. That finishes the proof of +1.1, which also finishes the proof of *1. Q.E.D. [0.0 1.2 0.5] COMMUTATIVITY-OF-TIMES > 2.6.5 Compiler Proofs with the Boyer-Moore Prover

43% Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

However, only one is unflawed. We will induct according to the following scheme: (AND (IMPLIES (ZEROP Z) (p Y Z)) (IMPLIES (AND (NOT (ZEROP Z)) (p Y (SUB1 Z))) (p Y Z))). Linear arithmetic, the lemma COUNT-NUMBERP, and the definition of ZEROP establish that the measure (COUNT Z) decreases according to the well-founded relation LESSP in each induction step of the scheme. The above induction scheme leads to the following two new formulas: Case 2. (IMPLIES (AND (ZEROP Z) (NUMBERP Z)) (EQUAL (PLUS Y (TIMES Z Y)) (TIMES Y (ADD1 Z)))). This simplifies, expanding the functions ZEROP, NUMBERP, EQUAL, TIMES, and ADD1, to: (IMPLIES (EQUAL Z 0) (EQUAL (PLUS Y 0) (TIMES Y 1))),

LEVEL 1; PRIORITY 2; ID 7 <<< Checkpoint 3>>> Case 2.1. (IMPLIES (NOT (NUMBERP X)) (EQUAL 0 (TIMES Y X))). Name the above subgoal *1.2. Case 1. (IMPLIES (AND (NOT (ZEROP X)) (EQUAL (TIMES (SUB1 X) Y) (TIMES Y (SUB1 X)))) (EQUAL (TIMES X Y) (TIMES Y X))). This simplifies , opening up ZEROP and TIMES, to the new conjecture: !!CHECKPOINT LEVEL 3; PRIORITY 0; ID 12 <<< Checkpoint 4>>> (IMPLIES (AND (NOT (EQUAL X 0)) (

- 6 An Assistant for Reading Nqthm Proo..., 1992, S. 8
- 6 An Assistant for Reading Nqthm Proo..., 1992, S. 7

Textstelle (Prüfdokument) S. 15

usage of the Boyer-Moore prover in proving theorems about compilers or portions of compilers. The previous sections discussed work with different verifiers in the area of scanning and parsing. This section will briefly mention the use of the Boyer-Moore prover in the area of code generation. Short Stack A number of researchers at CLInc have used NQTHM to prove a remarkably complex compiler to be correct. Starting from an s-expression representation for the abstract syntax of a subset

● **3%** Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

of the use of an automated reasoning assistant to check the reliability of detailed proofs and software. What is new about the current effort is the use of the Boyer-Moore prover (slightly extended) to check a non-trivial logic proof related to the correctness of an actual implementation, though it must be conceded that the actual Pc-

- **8** Response to FM91 Survey of Formal M..., 1992, S. 23

Textstelle (Prüfdokument) S. 18

on-line. Mathematics Prime factorization uniqueness [BM79] Unsolvability of the halting problem [BM84b] RSA public key encryption algorithm is invertible [BM84c] GauB Law of Quadratic Reciprocity [Rus92] Church-Rosser Theorem [Sha85] Godel's incompleteness theorem [Sha86] Irrationality of the square root of 2 Exponent two version of Ramsey's Theorem Schroeder-Bernstein Theorem Koening's Tree Lemma Group Theory lemmata [Yu90] Wilson's Theorem [Rus85] Turing Completeness of Pure Lisp [BM84a] Hardware Hypothetical processor FM8501 [Hun87] Motorola MC 68020 [BY91] Processor FM 9001 Railroad gate controller Fuzzy logic controller Parameterized hardware modules [VCDM90, VVCDM92] Synchronous circuits [Bro89] Theorem proving Ground resolution prover Theorem about generalization [Kau91] Various Short Stack (Compiler for Gypsy to FM8501 machine code) [BHMY89] Towers of Hanoi MACH Kernel specification Scheduling theorem for real-time operating system Implementation of an applicative language with Dynamic Storage Allocation Simple real-time control problem (cross-wind navigational system) 2.6.6 Suitableness for this Proof People have often advised me, during this proof attempt, to switch to a different theorem prover. None, it would seem, are exactly well-

26% Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

factorization uniqueness Unsolvability of the halting problem RSA public key encryption algorithm is invertable Gauß s Law of Quadratic Reciprocity Chruch-Rosser Theorem Gödel s incompleteness theorem Irrationality of the square root of 2 Exponent two version of Ramsey s Theorem Schroeder-Bernstein Theorem Koening s Tree Lemma Group Theory lemmataWilson s Theorem Hardware Hypothetical processor FM8501 Motorola MC 68020 Processor FM 9001 Railroad gate controller Fuzzy logic controller Parameterized hardware modules Synchronous circuits VHDL formalization Theorem proving Ground resolution prover Theorem about generalization Various Short Stack (Compiler for Gypsy to FM8501 machine code) Towers of Hanoi MACH Kernel specification Scheduling theorem for real-time operating system Implementation of an applicative language with Dynamic Storage Allocation Problems associated with the prover tend to be found in the large effort in learning how to use the system, in a proper statement of the

- **3** Dagstuhl Seminar on High Integrity ..., 1995, S. 31
- **3** Dagstuhl Seminar on High Integrity ..., 1995, S. 32

Textstelle (Prüfdokument) S. 24

GOTO, as it is when used to construct the viable prefix recognizer for an LR-Parser. 3.2 A Constructive Proof Rabin and Scott made use of existential quantification in their proof of the automaton equivalence. The basic version of the Boyer-Moore theorem prover does not provide existential quantification[4] . My intent is to do the proof completely from first principles without resorting to such "higher" constructs. Section 3.3 will describe a proof using explicit quantification as conducted by a researcher at CLInc.

4 Although there is an extension included in the new version of the prover, NQTHM-1992, which provides a mechanism for introducing it.

1% Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

Report , Department of Computer Sciences, The University of Texas at Austin, November, 1988. [51] M. Kaufmann. A Formal Semantics and Proof of Soundness for the Logic of the NQTHM Version of the Boyer- Moore Theorem Prover. Technical Report , Institute for Computing Science, University of Texas at Austin, Austin, TX 78712, 1986. ICSCA Internal Note 229. [52] M. Kaufmann. A User s Manual for an Interactive Enhancement to the

- **9** Mechanically verifying concurrent p..., 1992, S. 334

Textstelle (Prüfdokument) S. 90

creating a list of the elements in reverse stack order. This way of reading the stack is necessary for stating and proving invariants of parsing. The function signatures are: is-stack Any ? B emptystack > Stack push Element X Stack > Stack pop Stack > Stack top Stack > Element is-empty Stack > B pop-n No X Stack > Stack top-n N0 X Stack > Element* stack-length Stack > N from-bottom Stack > Element* A shell can be used for implementing this basic type. Event: Add the shell push, with bottom object

● **1%** Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

Figure N.2: Algebraic definition of an Integer Stack ADT Domains: Nat (the natural numbers Stack (of natural numbers) Bool (boolean values) Functions: newStack: () -> Stack push : (Nat, Stack) -> Stack pop: Stack -> Stack top: Stack -> Nat empty : Stack -> Bool Axioms: pop(push(N,S)) = S top(push(N,S)) = N empty(push(N,S)) = false empty(newStack()) = true Errors: pop(newStack()) top(newStack()) where N in Nat and S in

- **10** Introduction to programming languages, 1996, S.

Textstelle (Prüfdokument) S. 151

table generation, one would have to first generate the table to know how many states are possible k steps away from the current state. However, as J Moore remarked once when I was complaining to him about this, "there's no such thing as a free lunch". Partial correctness is really only half of what one needs. As many theorems as desired can be proved partially correct, but if there is no strong argument that the functions involved indeed terminate, then the

● **0%** Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

tale ????????, ???????. ??????????? ??, ??????, ?????????? ???????????: to dine with Duke Humphrey ???????? ??? ????? (?????? ?? ????); to cut off with a shilling ?????? ????-???? ??????????. ? ?????????? ???????????, take smb. by the button, ? bend smb. s ear ?????????? "??????????? ??????????? ??????????? , ????? ? ????? ??????????? ???-???? ????-????". ????????? ?? ??????? ?????? ? ??????????- ??? ???????????: "????? ?? ????????" ? "????????? ???". ??????? ?? ?????? (???????) ??? ????? "??- ????????, ????????? ?????????? ? ???-?., ????? ?? ? ???-?.", ???????? (???) ???? "????????, ??????? ?? ??????? ???????????, ??? ????? ???-???? ??????????", ?????? (??????? ? ?. ?.) ? ???? "????? ?????????? ?? ???????? ?????????? ? ??????, ???????? ????? ????-????" ???????? ???? ?? ???? ??????? ??????? ???- ???????? ?????? ??????????? ?? ?? ??????? "????". ?????? ?? ????? ???? ???????? ? ????????, ??????? ????? ??????? ? ??????????? ?????? ??. ??- ??????? A hedge between keeps friendship green ???????? ??? ????????? ??, ???????? ?????????: ???? ? ???????? ?????????? ???????? ?? ????????? ???? ??????? (? ??????????) ?????????? (??? ?? ????? ?????? hedge). ???????????, ?? ????????, ???????? ???????? ?????????? ? ??? ???????, ? ??? ????????. There s no such a thing as a free lunch ?????????, ???????????: ?????? ???-?? ????????, ??? ???? ?? ????? ?????? ??????, ?? ??? ???? ? ????? ?????? ?????????. ?????? ? ?????????? ? ???????? XX ????, ??????????? ? ???, ? ???????? ???????, ??????? ???? ? ????????? ? ????????. ???????: Money does n?t grow on trees. With time and patience the leaf of the mulberry becomes satin ?? ???????? ? ??? ???????? ? ??????? ???? ?????? ???????. ??.: ???????? ? ???? ??? ????????. He who would eat the nut must first crack

- **11** 1052 1045 1044 1030 1040 - 1044 104..., 1051, S. 271

Textstelle (Prüfdokument) S. 9

2 A publicly available copy of Robert S. Boyer and J Strother Moore's theorem prover NQTHM, or Matt Kaufmann's interactive proof checker version PC-NQTHM, can be obtained from Internet host ftp.cli.com (192.31.85.129) by anonymous ftp.

● **9%** Einzelplagiatswahrscheinlichkeit

Textstelle (Originalquellen)

the equality of two terms or the implication of one term from another, are also represented as s-expressions. A publically available copy of Bob Boyer and J Moore s theorem prover NQTHM or Matt Kaufmann s interactive proof checker version PC-NQTHM can be obtained from Internet host ftp.cli.com (192.31.85.129) by anonymous ftp. A World Wide Web home page is offered by Computational Logic at httpwwwclicom A short example proof, the associativity of times, was printed in a handout and discussed

- **3** Dagstuhl Seminar on High Integrity ..., 1995, S. 31

Quellenverzeichnis

1 Interaction with the Boyer-Moore Theorem Prover: A Tutorial Study, 1994
http://www.dtic.mil/cgi-bin/GetTRDoc%3FAD%3DADA358632
2 Bauer: Compiler Construction , 1976
3 Dagstuhl Seminar on High Integrity Programmable Electronic Systems, 1995
http://www.dagstuhl.de/Reports/95/9509.pdf
4 A verified operating system kernel, 1987
http://ftp://128.83.120.139/pub/boyer/cli-reports/011.pdf
5 Hunt FM8501 , 1994
http://link.springer.com/book/10.1007%2F3-540-57960-5
6 An Assistant for Reading Nqthm Proof Output, 1992
http://ftp://ftp.cs.utexas.edu/pub/boyer/cli-reports/085.pdf
7 An Instructive Example for Beginning Users of the Boyer-Moore, 1990
http://ftp://ftp.cs.utexas.edu/pub/boyer/cli-notes/note-185.pdf
8 Response to FM91 Survey of Formal Methods: Nqthm and Pc-Nqthm, 1992
http://ftp://ftp.cs.utexas.edu/.snapshot/nightly.5/pub/boyer/cli-reports/075.pdf
9 Mechanically verifying concurrent programs, 1992
http://ftp://www.cs.utexas.edu/pub/boyer/cli-reports/071.pdf
10 Introduction to programming languages, 1996
http://www.worldcolleges.info/sites/default/files/aaby.pdf
11 1052 1045 1044 1030 1040 - 1044 1048 1057 1050 1059 1056 1057 1071 1050 1057 1045 1056 1045 1044 1054 1042 1048 1065 1045 1050 1054 1053 1062 1045 1055 1058 1059 1040 1051 1030 1047, 1051
http://lingvj.oa.edu.ua/assets/files/full/2013/n38/NZ_Vyp_38.pdf

ProfNet
Institut für Internet-Marketing

Glossar

- **Ampel** — Entsprechend der Gesamtwahrscheinlichkeit wird ein Rating der Schwere durch die Ampelfarbe berechnet: grün (bis 19 %) = wenige Indizien unterhalb der Bagatellschwelle; gelb (20 bis 49 %) - deutliche Indizien enthalten, die eine Plagiatsbegutachtung durch den Prüfer notwendig machen; rot (ab 50 %) = Plagiate liegen mit sehr hoher Wahrscheinlichkeit vor, die eine Täuschungsabsicht dokumentieren. Bei publizierten Dissertationen sollte ein offizielles Verfahren zur Prüfung und/oder zum Entzug des Doktortitels eröffnet werden.
- **Anteil Fremdtexte (brutto)** — Anteil aller durch die Software automatisch gefundenen Bestandteile aus anderen Texten am Prüftext (von mindestens 7 Wörtern) in Prozent und Anzahl der Wörter gemessen. Dabei wird noch keine Interpretation auf Plagiatsindizien oder korrekte Übernahmen (z.B. Zitat, Literaturquelle) vorgenommen.
- **Anzahl Fremdtext (netto)** — Anteil aller durch die Software automatisch gefundenen und als Plagiatsindizien interpretierten Bestandteile aus anderen Texten am Prüftext (von mindestens 7 Wörtern) in Prozent und Anzahl der Wörter gemessen.
- **Bauernopfer** — Fehlende Quellenangabe bei einer inhaltlichen oder wörtlichen Textübernahme, wobei die Originalquelle an anderer Stelle des Textes (außerhalb des Absatzes, des Satzes, des Hablsatzes oder des Wortes) angegeben wird.
- **Compilation** — Zusammensetzen des Textes als "Patchwork" aus verschiedenen nicht oder unzureichend zitierten Quellen.
- **Eigenplagiat** — Übernahme eines eigenen Textes des Autors ohne oder mit unzureichender Kennzeichnung des Autors. Auch wenn hier nur eigene Texte und Gedanken übernommen werden, handelt es sich um eine Täuschung. Der Prüfer geht davon aus, dass es sich hier um neue Texte und Gedanken des Autors handelt.
- **Einzelplagiatswahrscheinlichkeit** — Grobe Berechnung der Wahrscheinlichkeit des Vorliegens eines Plagiates des einzelnen Treffers (oder der Treffer) auf einer Seite im Prüfbericht.
- **Gesamtplagiatswahrscheinlichkeit** — Berechnung der Wahrscheinlichkeit des Vorliegens von Plagiaten durch Verknüpfung der Indizienanzahl, des Netto-Fremdtextanteils und der Schwere der

einzelnen Plagiatsindizien.

• Ghostwritersuche	Über den statistischen Vergleich der Texte (Stilometrie) wird eine Wahrscheinlichkeit berechnet, ob die Texte von demselben Autor stammen.
• Indizien	Dieser Prüfbericht gibt nur die von der Software automatisch ermittelten Indizien auf eine bestimmte Plagiatsart wieder. Die Feststellung eines Plagiats kann nur durch den Gutachter erfolgen.
• Literaturanalyse	Die im Prüftext enthaltenen Literatureinträge im Literaturverzeichnis werden analysiert: Wird die Quelle im Text zitiert? Handelt es sich um eine wissenschaftliche Quelle? Wie alt sind die Quellen?
• Mischplagiat - eine Quelle	Der Text wird hierbei aus verschiedenen Versatzstücken einer einzigen Quelle zusammengesetzt, also gemischt.
• Mischplagiat - mehrere Quellen	Der Text wird hierbei aus verschiedenen Versatzstücken aus verschiedenen Quellen zusammengesetzt, also gemischt.
• Phrase	Die übernommenen Textstellen stellen allgemeintypische oder fachspezifische Wortkombinationen der deutschen Sprache dar, die viele Autoren üblicherweise verwenden. Solche Übernahmen gelten nicht als Plagiate.
• Plagiat	Übernahme von Leistungen wie Ideen, Daten oder Texten von anderen - ohne vollständige oder ausreichende Angabe der Originalquelle.
• Plagiatsanalyse	Gefundene gleiche Textstellen (= Treffer) werden durch die Software automatisch auf spezifische Plagiatsindizien analysiert.
• Plagiatssuche	Mit Hilfe von Suchmaschinen wird im Internet, in der Nationalbibliothek und im eigenen Dokumentenbestand nach Originalquellen mit gleichen oder ähnlichen Textstellen gesucht. Diese Quellen werden alle vollständig Wort für Wort mit dem Prüftext verglichen. Plagiatsindizien werden für Textstellen ab 7 Wörtern berechnet.

Glossar

- Plagiatswahrscheinlichkeit — Grobe Berechnung der Wahrscheinlichkeit des Vorliegens eines Plagiates auf der Basis der Plagiatsindizien. Die Ampel zeigt drei Ergebnisse an: grün - keine Wahrscheinlichkeit des Vorliegens eines Plagiates und somit keine weitere Überprüfung notwendig, gelb - mögliches Vorliegen eines Plagiates und somit eine weitere Überprüfung empfohlen, rot - hohe Wahrscheinlichkeit des Vorliegens eines Plagiates und somit weitere Überprüfung unbedingt notwendig.
- Stilometrie — Texte werden dabei einzeln nach statistischen Kennzahlen (z.B. durchschnitliche Länge der Wörter, Häufigkeit bestimmter Wörter) analysiert. Sind diese Kennzahlen für zwei Texte ähnlich, liegt hier statistisch der gleiche "Stil" und somit mit hoher Sicherheit der selbe Autor vor.
- Teilplagiat — Ein Textbestandteil einer Quelle wurde vollständig ohne ausreichende Zitierung kopiert.
- Textanalyse — Der einzelne Text wird durch die Software automatisch für sich allein analysiert, z.B nach statistischen Kennzahlen, benutzter Literatur, Rechtschreibfehlern oder Bestandteilen. Je nach Stand der Softwareentwicklung sind die absoluten Ergebnisse (z.B. Erkennung von Abbildungen, Fußnoten, Tabellen, Zitaten) im einzelnen eingeschränkt aussagefähig. Aufgrund der immer für alle Texte durchgeführten Analysen sind die relativen Unterschiede zwischen den Spalten (z.B. Diplomarbeit vs. Dissertation) uneingeschränkt aussagefähig.
- Textvergleich — Jeder Text wird mit anderen älteren Texten vollständig verglichen. Gefundene gleiche Texstellen werden in einem weiteren Schritt z.B. auf Plagiatsindizien hin untersucht.
- Übersetzungsplagiat — Nutzung eines fremdsprachigen Textes durch Übersetzung.
- Verschleierung — Ein Text wird ohne eindeutige Kennzeichnung (i.d.R. durch Anführungszeichen) Wort für Wort übernommen, aber mit Angabe der Quelle in der Fußnote. Dadurch wird der Prüfer getäuscht, der von einer nur inhaltlichen Übernahme ausgehen muss.
- Vollplagiat — Der gesamte Text wird vollständig ohne Zitierung kopiert.

Glossar

• Zitat - wörtlich	Übernommener Text wird z.B. mit Anführungszeichen korrekt dargestellt. Dieses wörtliche Zitat darf keine Veränderungen, Ergänzungen oder Auslassungen enthalten. Fehlt für das Zitat nach der Plagiatssuche ein Nachweis in einer Originalquelle, so wird der Treffer als "Zitat-wörtlich-im Text" bezeichnet.
• Zitat - wörtlich - Veränderung	Einzelne Wörter einer korrekt gekennzeichneten wörtlichen Übernahme werden verändert oder weggelassen, ohne dass der Sinn verändert wird. Z.B.: " Unternehmung" wird durch "Unternehmen" ersetzt.
• Zitat - wörtlich - Verdrehung	In dem korrekt gekennzeichneten übernommenen wörtlichen Text wird der Sinn durch Austauschung einzelner Wörter deutlich verändert. Beispiel: " überentwickelten" statt "unterentwickelten".
• Zitierungsfehler	Arbeitsbezeichnung für eine wörtliche Textübernahme, die nur als inhaltliche Textübernahme (Paraphrase) gekennzeichnet wird.