The Industrial Success of Verification Tools Based on Stålmarck's Method

Arne Borälv
arne@lk.se

Logikkonsult NP AB,
Swedenborgsgatan 2,
S-118 48 Stockholm, Sweden,
http://www.lk.se

Abstract. Stålmarck's Method is a patented natural deduction proof method with a novel proof-theoretic notion of *proof depth*, defined as the largest number of nested assumptions in the proof. An implementation of the method, called Prover, has been used as proof engine in various commercial tools since 1990, and is now integrated in a formal verification framework called NP-Tools. Prover searches for shallow subformula proofs, which has proven to be an efficient strategy for solving many industrial problems, the largest of which today consists of several 100,000's of sub-formulas. Stålmarck's method is in industrial use, for instance in the areas of telecom service specification analysis, analysis of railway interlocking software, analysis of programmable controllers and analysis of aircraft systems. The method seems suitable also for hardware verification.

1 Railway Interlocking Software: First Tool

In 1989, ADtranz Signal had problems with system availability due to run-time errors ("double values") in their computerized interlockings, which caused their interlocking to enter a safe state at times. This problem made extensive testing necessary, but since computerized interlocking is a complex task, even a very long test phase could not give enough coverage. The runtime errors remained.

Logikkonsult found that a sufficient condition for ensuring that this type of error could not occur was that certain boolean formulas representing the generic interlocking software program were unsatisfiable. Proving this seemed to work in practise, so the tool CVT [SS90] was developed. Proving this property for a complete program is done automatically by a simple pressing of a button; the proof takes roughly about one minute CPU time on a Sparc 10. The tool was released in 1990 and has been part of ADtranz' development environment since. CVT is also regularly used by the Swedish National Railroad Administration for comparing different revisions of interlocking software (a compare fuctionality of interlocking software programs is built-in in CVT). The reported benefits from ADtranz are a 90% reduction of the test phase time, and an overall development cost reduction of more than 15%. Furthermore, since the introduction of the tool, no run-time errrors due to double values have occured.

2 Railway Interlocking Software: Second Tool

Inspired by the success of the CVT tool for proving properties of generic software, an attempt to analyse complete railyard interlockings was made [BÅ95]. Replacing parts of the rigourous system-level test phase with formal proofs can potentially save a lot of time, and also improve quality as given by rigorous formal proofs. A new tool SVT was developed, which is a translator for complete (or partial) railyard interlockings into the general-purpose formal verification framework NP-Tools (see Section 3). Using SVT and NP-Tools, a logic model of any of ADtranz' interlockings can be produced automatically, and safety requirements on the system level can be proved. The approach has already proved its usefulness and errors have been found in interlockings, even after the standard test phase [Bor97][1]. A medium-sized railyard model consists of some 100,000 sub-formulas; the largest system analyzed (in which both proofs and countermodels for properties have been found) consists of 350,000 sub-formulas. In the latter case, the proof times were about 20 seconds after an initial 1-saturation [Wid96] of the system (done only once per system) which took 100 minutes CPU time on an HP 9000/715 work-station. A commercial release of SVT is planned in 1997.

2.1 Proof Logging and Proof Checking

In order to replace (parts of) ADtranz' extensive system-level test phase with formal proofs, ADtranz requires that Prover generated proofs must be possible to check separately. Therefore, the Prover implementation (and thereby also NP-Tools) is augmented with the possibility of logging proofs, which thereafter can be checked by a (relatively simple) proof checker program. This strenghtens the confidence in the proofs generated, especially since the proof checker can be written using a rigorous formal approach, for instance the well-known B-Method.

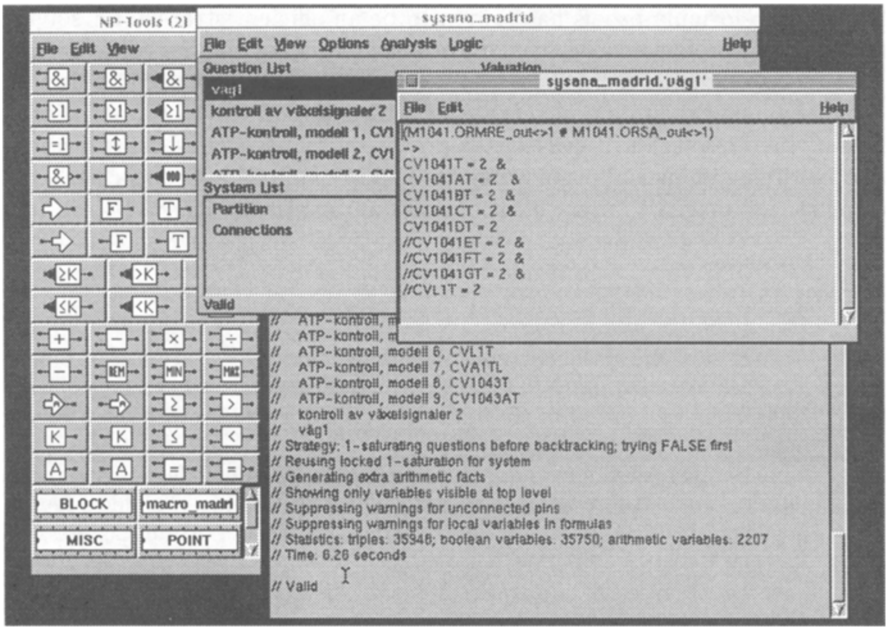
3 NP-Tools

The foremost usage of Prover today is in the general-purpose formal verification framework NP-Tools [E$^+$96], developed by Logikkonsult. NP-Tools offers a set of automatic translators, a graphical user interface for construction of designs and integration of automatically imported systems and a system analysis facility. Figure 1 illustrates a system analysis where a set of requirements have been proved for an interlocking software system.

3.1 NP-Tools Applications

NP-Tools is currently evaluated and used by Saab AB (formerly Saab Military Aircraft) and the Swedish Defence Material Adminstration for verification

[1] The case study consisted of about 35,000 sub-formulas, and the proof time for some 50 requirements was about 15 CPU seconds on an HP 9000/715 work-station.



```
(M1041.ORMRE_out<>1 # M1041.ORSA_out<>1)
->
CV1041T = 2  &
CV1041AT = 2  &
CV1041BT = 2  &
CV1041CT = 2  &
CV1041DT = 2
//CV1041ET = 2  &
//CV1041FT = 2  &
//CV1041GT = 2  &
//CVL1T = 2
```

```
//    ATP-kontroll, m
//    ATP-kontroll, m
//    ATP-kontroll, modell 6, CVL1T
//    ATP-kontroll, modell 7, CVA1TL
//    ATP-kontroll, modell 8, CV1043T
//    ATP-kontroll, modell 9, CV1043AT
//    kontroll av växelsignaler 2
//    väg1
// Strategy: 1-saturating questions before backtracking; trying FALSE first
// Reusing locked 1-saturation for system
// Generating extra arithmetic facts
// Showing only variables visible at top level
// Suppressing warnings for unconnected pins
// Suppressing warnings for local variables in formulas
// Statistics: triples: 35948; boolean variables. 35750; arithmetic variables. 2207
// Time. 6.26 seconds

// Valid
```

Fig. 1. An NP-Tools System Analysis with proven requirements

of safety properties, e.g., in CASE tool designs. In this context, an automatic NP-Tools translator from Verilog's ASA tool has been developed [Bol96] and a Statecharts translator is under development [Mei97a]. Saab are also currently exploring NP-Tools for FMEA and FTA. The Swedish avionics consultant firm LUTAB are regularly using NP-Tools for assessing safety properties in electronic designs and airborne software.

ABB Network Partner are currently using NP-Tools for verification of PLC programs, based on the IEC standard 1131-3. An NP-Tools translator is under development.

Volvo Bus uses NP-Tools for verifying and analysing formal specifications, for instance proving that the specifications are unambigous, handles all input combinations and that safety requirements hold [Mei97b]. Also Volvo Car Corporation have used NP-Tools for developing formal specifications.

Recently, NP-Tools has been used to verify VHDL designs, with promising results compared to commercial VHDL verification tools.

4 Stålmarck's Method

The patented [Stå92] Stålmarck Method [Stå94] is a natural deduction proof system with a novel proof-theoretic notion of *proof depth* [Wid96, Har96]. The depth of a proof is the largest number of nested assumptions in the proof. Searching for shallow sub-formula proofs has proven to be an efficient strategy for solving many industrial problems, as reported for a few applications here. The decision procedure was originally defined for boolean formulas only, but has in a natural way been extended to finite domain integer arithmetic. In 1994 a former version of Prover, the (resolution based) Otter prover and a BDD based prover were used for verifying industrial problems in the railway field. Prover was clearly the only one that managed to prove all properties automatically [GKvV94].

References

[BÅ95] Arne Borälv and Herman Ågren. Feasibility Study SVT. Technical Report U-95002, Logikkonsult NP AB, 1995. Internally published at Logikkonsult.

[Bol96] Hans Bolinder. LSA 1.0. Technical Report NPT-01-07-0-3, Issue 2 Rev 1, Logikkonsult NP AB, 1996.

[Bor97] Arne Borälv. A Fully Automated Approach for Proving Safety Properties in Interlocking Software Using Automatic Theorem-Proving. Technical report, Logikkonsult NP AB, March 1997. Submitted to the World Congress on Railway Research 1997 (WCRR'97).

[E$^+$96] Love Ekenberg et al. *Reference Manual, NP-Tools 2.2.* Logikkonsult NP AB, October 1996. NPT-01-07-02 2.0.

[GKvV94] J.F. Groote, J.W.C. Koorn, and S.F.M. van Vlijmen. The safety guaranteeing system at station Hoorn-Kersenboogerd. Technical report, Department of Philosophy – Utrecht University, 1994.

[Har96] John Harrison. The Stålmarck Method as a HOL Derived Rule. In *Theorem Proving in Higher Order Logics*, pages 221–234. TPHOLs'96, Springer Verlag, 1996.

[Mei97a] Karl Meinke. Axiomatic Semantics and Automatic Verification of Statecharts. Technical report, Logikkonsult NP AB, April 1997.

[Mei97b] Karl Meinke. Industrial Formal Methods: A Case Study in Public Transport Vehicles, February 1997. In Formal Methods Europe Tour 2, 1997. http://www.ifad.dk/projects/tour2.html.

[SS90] Mårten Säflund and Gunnar Stålmarck. Modelling and Verifying Systems and Software in Propositional Logic. In *Proceedings of IFAC/EWICS/SARS Symposium SAFECOMP '90*, pages 31–36. Pergamon Press, 1990.

[Stå92] Gunnar Stålmarck. A System for Determining Propositional Logic Theorems by Applying Values and Rules to Triplets that are Generated from a Formula, 1992. Swedish Patent No. 467 076 (approved 1992), U.S. Patent No. 5 276 897 (1994), European Patent No. 0403 454 (1995).

[Stå94] Gunnar Stålmarck. A Proof Theoretic Concept of Tautological Hardness. Unpublished manuscript, 1994.

[Wid96] Filip Widebäck. Stålmarck's Notion of *n*-saturation. Technical Report NP-K-FW-200, Logikkonsult NP AB, January 1996.