Co-Synthesis of New Complex Selection Algorithms and their Human Comprehensible XML Documentation

Theorem, which was originally proven in 1976 by Appel and Haken who used computer programs to check a very large number of cases.

However, for a mathematician it is unsatisfying to know that there exists a solution or no solution for a problem, because thousands, hundreds of thousands or millions of states have been explored by a theorem prover whose rules have been verified, and at the same time, not be able to comprehend the tedious machine-generated proofs and not be able to draw conclusions from the automated proof.

context-specific rules as how to decompose the proof graph, when to generate

Fig. 1. Scenario for Knowledge-based Synthesis

or relationships between them have been defined. Figure 2 lists the major types of the SEAMLESS framework.

Basic types	*Bool, Integer, Constants, Vars, String*
Object logic	*Atom, Clause, Algebraic Expression, Constraint*
Synthesis types	*Precondition, Postcondition, Program, Proof*

Fig. 2. Types for Program Synthesis

- to facilitate the automatic synthesis of human comprehensible XML-based documents;
-

The advantage of using graphterms for encoding higher-order formulas is can be processed by the t

6 Experimental Results

A major advantage of combining higher-order program synthesis and document synthesis is that the generated proof terms describe programs at the sis or program documentation and merely exploit the correspondences between XML data and logical terms, e.g., [BDHG,LR03,SR01]. The presented framework distinguishes itself from related approaches (cf. [Pec04,GKP96,Oks05]) in which computer have been used for the complexity analysis of unsolved mathematical problems. The results synthesised by SEAMLESS are constructive, comprehensible and can be manually checked for correctness, provided that the sizes of the

Fig. 10. Poset P

lemma 2 *Let P be a poset as visualised in Figure 11. The 3-rd largest element of P is computed by at most 8 comparisons.*

Fig. 11. Poset P

Proof. algorithm 3

1. *Compare 3.79223(ad)-367(a17)1(.)]TJ12.2319-9.7316Td[((3))]TJF208.9664Tf17.21340Td[(a)]TJF178.9664Tf4.88 selecting the 2-ad l3rgest element takes at most f(.,10.0)=7 comparisons.*

(b) $a_5 < a_{17}$. a_{17}