Greedy Pebbling: Towards Proof Space Compression

Andreas Fellner * and Bruno Woltzenlogel Paleo **

fellner.a@gmail.com bruno@logic.at
Theory and Logic Group
Institute for Computer Languages
Vienna University of Technology

1 Experiments

All the pebbling algorithms and heuristics described in the previous sections have been implemented in the hybrid functional and object-oriented programming language Scala (www.scala-lang.org) as part of the Skeptik library for proof compression (github.com/Paradoxika/Skeptik) [2].

To evaluate the algorithms and heuristics, experiments were executed[1] on four disjoint sets of proof benchmarks (Table 1). TraceCheck_1 and TraceCheck_2 contain proofs produced by the SAT-solver PicoSAT [1] on unsatisfiable benchmarks from the SATLIB (www.satlib.org/benchm.html) library. The proofs[2] are in the TraceCheck proof format, which is one of the three formats accepted at the *Certified Unsat* track of the SAT-Competition. veriT_1 and veriT_2 contain proofs produced by the SMT-solver veriT (www.verit-solver.org) on unsatisfiable problems from the SMT-Lib (www.smtlib.org). These proofs[3] are in a proof format that resembles SMT-Lib's problem format and they were translated into pure resolution proofs by considering every non-resolution inference as an axiom.

Table 2 summarizes the results of the experiments. The two presented order finding algorithms are tested in combination with the four presented heuristics. The Children and LastChild heuristics were tested on all four benchmark sets. The Distance and Decay heuristics were tested on the sets TraceCheck_2 and veriT_2. The relative performance is calculated according to Formula 1, where f is an algorithm with a heuristic, P is the set of proofs the heuristic was tested on and G are all combinations of algorithms and heuristics that were tested on P. The time used to construct orders is measured in processed nodes per millisecond. Both columns show the best and worst result in boldface.

* Supported by the Google Summer of Code 2013 program.
** Supported by the Austrian Science Fund, project P24300.
[1] The Vienna Scientific Cluster VSC-2 (http://vsc.ac.at/) was used.
[2] SAT proofs: www.logic.at/people/bruno/Experiments/2014/Pebbling/tc-proofs.zip
[3] SMT proofs: www.logic.at/people/bruno/Experiments/2014/Pebbling/smt-proofs.zip

Name	**Number of proofs**	**Maximum length**	**Average length**
TraceCheck_1	2239	90756	5423
TraceCheck_2	215	1768249	268863
veriT_1	4187	2241042	103162
veriT_2	914	120075	5391

Table 1: Proof benchmark sets

Algorithm Heuristic	**Relative Performance** (%)	**Speed** (nodes/ms)
Bottom-Up		
Children	17.52	**88.6**
LastChild	**26.31**	84.5
Distance(1)	9.46	21.2
Distance(3)	-0.40	0.5
Top-Down		
Children	-27.47	0.3
LastChild	-31.98	1.9
Distance(1)	-70.14	0.6
Distance(3)	**-74.33**	**0.1**

Table 2: Experimental results

$$\text{relative\_performance}(f, P, G) = \frac{1}{|P|} * \sum_{\varphi \in P} \left(1 - \frac{s(\varphi, f(\varphi))}{avg_{g \in G} s(\varphi, g(\varphi))}\right) \quad (1)$$

Table 2 shows that the Bottom-Up algorithm constructs topological orders with much smaller space measures than the Top-Down algorithm. This fact is visualized in Figure 1, where each dot represents a proof φ and the x and y coordinates show the space of φ with the topological orders found by, respectively, the best Top-Down and Bottom-Up algorithms for φ. The LastChild heuristic produces the best results and the Children heuristic also performs well. The Distance heuristic produces the worst results, which could be due to the fact that the radius is too small for big proofs with thousands of proof nodes. The Decay heuristics unfortunately do not show an improvement of the underlying heuristic.

Some additional heuristics, not described in this work, designed specifically for Top-Down Pebbling were tested on small benchmark sets. These heuristics

Decay γ	Depth d	Combination *com*	Performance **Improvement** (%)	Speed (nodes/ms)
0.5	1	avg	0.50	47.7
0.5	1	max	0.40	47.0
0.5	7	avg	0.85	14.0
0.5	7	max	0.76	15.3
3	1	avg	0.48	**64.0**
3	1	max	0.43	64.4
3	7	avg	0.21	15.3
3	7	max	**0.94**	15.3

Table 3: Improvement of LastChild using Decay Heuristic

aimed at doing local pebbling without having to calculate full spheres. For example pebbling nodes that allow other nodes to be unpebbled in the next move can be preferred. Unfortunately, none of the additional heuristics showed promising results.

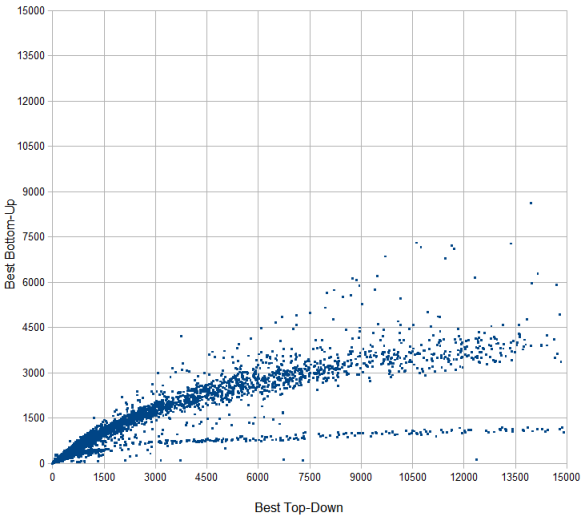

Fig. 1: Spaces obtained with best Bottom-Up and Top-Down heuristics

The Bottom-Up algorithm does not only produce better results, it is also much faster, as can be seen in the last column of Table 2. The reason probably is the number of comparisons that the algorithms make. For Bottom-Up the set N of possible choices consists of the premises of a single node only, and usually $|N| \in O(1)$ (e.g. for a binary resolution proof, $N \leq 2$ always). For Top-Down the set N is the set of currently pebbleable nodes, which can be large (e.g. for a perfect binary tree with $2n-1$ nodes, initially $|N| = n$). Possibly for some heuristics, Top-Down algorithms could be made more efficient by using, instead of a set, an ordered sequence of pebbleable nodes together with their memorized heuristic evaluations.

Unsurprisingly the radius used for the Distance Heuristic has a severe impact on the speed, which decreases rapidly as the maximum radius increases. With radius 5, only a few small proofs were processed in a reasonable amount of time.

Figure 2 relates for each proof the smallest space measure obtained by all algorithms tested on the respective proof and its length in number of nodes. Note that the y-axis, showing the space measures, scale is lower by a factor of 100 compared to the scale of the x-axis, which displays the proof lengths. On average the smallest space measure of a proof is 44,1 times smaller than its length. This shows the impact that the usage of deletion information together with well constructed topological orders can have. When these techniques are used, on average 44,1 times less memory is required for storing nodes in memory while proof processing.

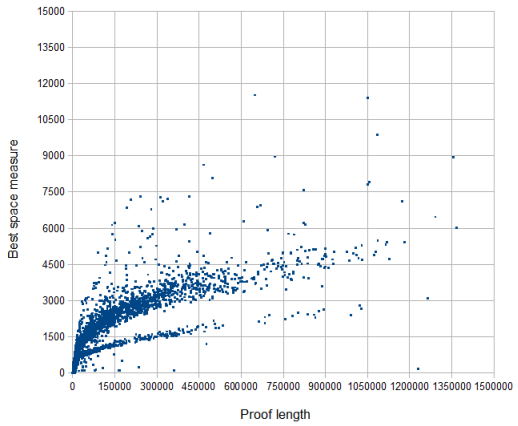

Fig. 2: Best space measure compared to proof length

References

1. Biere, A.: Picosat essentials. JSAT 4(2-4), 75–97 (2008)
2. Paleo, B.W., Boudou, J., Fellner, A.: Skeptik system description