A Full experimental data

This appendix will be made available online at

http://www.eprover.eu/E-eu/eqplanning.html

Unless otherwise specified, all results are obtained with E 3.0 using the command line `eprover -sR --proof-object -H'(1*weight11_ugg)' <problem>`. All times are in seconds. Since timing by the operating system is not exact, there is some noise, and differences in the few millisecond range are not significant.

A.1 Switches - all different

Full ground encoding In this setting, the initial and final states differ in all switches, and the actions are encoded as equations between full ground states. An example of size 3 is presented below. Results are in Table 3.

```
% Simple switchbank example. There are 3 switches
% that can be on or off. Actions flip a single switch.
%
% This is the ground version.
%
% Each possible action in each possible state is encoded
% in one rule, i.e. we have 2^n*n=24 action rules.

cnf(to_plan, negated_conjecture, f(on,off,on)!=f(off,on,off)).
cnf(g_0_on_on_on, axiom, f(on,on,on)=f(off,on,on)).
cnf(g_1_on_on_on, axiom, f(on,on,on)=f(on,off,on)).
cnf(g_2_on_on_on, axiom, f(on,on,on)=f(on,on,off)).
cnf(g_0_on_on_off, axiom, f(on,on,off)=f(off,on,off)).
cnf(g_1_on_on_off, axiom, f(on,on,off)=f(on,off,off)).
cnf(g_2_on_on_off, axiom, f(on,on,off)=f(on,on,on)).
cnf(g_0_on_off_on, axiom, f(on,off,on)=f(off,off,on)).
cnf(g_1_on_off_on, axiom, f(on,off,on)=f(on,on,on)).
cnf(g_2_on_off_on, axiom, f(on,off,on)=f(on,off,off)).
cnf(g_0_on_off_off, axiom, f(on,off,off)=f(off,off,off)).
cnf(g_1_on_off_off, axiom, f(on,off,off)=f(on,on,off)).
cnf(g_2_on_off_off, axiom, f(on,off,off)=f(on,off,on)).
cnf(g_0_off_on_on, axiom, f(off,on,on)=f(on,on,on)).
cnf(g_1_off_on_on, axiom, f(off,on,on)=f(off,off,on)).
cnf(g_2_off_on_on, axiom, f(off,on,on)=f(off,on,off)).
cnf(g_0_off_on_off, axiom, f(off,on,off)=f(on,on,off)).
cnf(g_1_off_on_off, axiom, f(off,on,off)=f(off,off,off)).
cnf(g_2_off_on_off, axiom, f(off,on,off)=f(off,on,on)).
cnf(g_0_off_off_on, axiom, f(off,off,on)=f(on,off,on)).
cnf(g_1_off_off_on, axiom, f(off,off,on)=f(off,on,on)).
cnf(g_2_off_off_on, axiom, f(off,off,on)=f(off,off,off)).
cnf(g_0_off_off_off, axiom, f(off,off,off)=f(on,off,off)).
cnf(g_1_off_off_off, axiom, f(off,off,off)=f(off,on,off)).
cnf(g_2_off_off_off, axiom, f(off,off,off)=f(off,off,on)).
```

	Instance size	Rewrite steps	Action equivalents	Prover time
alldiff_switch_gnd001.p.prf	1	1	1	0.003
alldiff_switch_gnd002.p.prf	2	2	2	0.002
alldiff_switch_gnd003.p.prf	3	3	3	0.002
alldiff_switch_gnd004.p.prf	4	4	4	0.002
alldiff_switch_gnd005.p.prf	5	5	5	0.003
alldiff_switch_gnd006.p.prf	6	6	6	0.004
alldiff_switch_gnd007.p.prf	7	7	7	0.009
alldiff_switch_gnd008.p.prf	8	8	8	0.020
alldiff_switch_gnd009.p.prf	9	9	9	0.046
alldiff_switch_gnd010.p.prf	10	10	10	0.114
alldiff_switch_gnd011.p.prf	11	11	11	0.290
alldiff_switch_gnd012.p.prf	12	12	12	0.794
alldiff_switch_gnd013.p.prf	13	13	13	2.272
alldiff_switch_gnd014.p.prf	14	14	14	6.850
alldiff_switch_gnd015.p.prf	15	15	15	22.405
alldiff_switch_gnd016.p.prf	16	16	16	93.106
alldiff_switch_gnd017.p.prf	17	17	17	407.151
alldiff_switch_gnd018.p.prf	18	18	18	1845.499
alldiff_switch_gnd019.p.prf	19	19	19	8076.025
alldiff_switch_gnd020.p.prf	20	20	20	42027.167

Table 3. Results on switch banks with full ground encoding

Action encoding with variables In this setting, the initial and final states differ in all switches, and the actions are encoded as equations using variables to represent the unchanging parts of the state. An example of size 3 is presented below. Results are in Table 4.

```
% Simple switchbank example. There are 3 switches
% that can be on or off. Actions flip a single switch.
%
% This is the version using first-order variables.
%
% The states of non-relevant switches are captured (and
% preserved) by variables, so we only need one rule to
% turn each switch on, and one to turn it off. Thus
% we have 2n=6 action rules.

cnf(to_plan, negated_conjecture, f(on,off,on)!=f(off,on,off)).
cnf(n_3_0_off, axiom, f(on,X1,X2) = f(off,X1,X2)).
cnf(n_3_0_on,  axiom, f(off,X1,X2) = f(on,X1,X2)).
cnf(n_3_1_off, axiom, f(X0,on,X2) = f(X0,off,X2)).
cnf(n_3_1_on,  axiom, f(X0,off,X2) = f(X0,on,X2)).
cnf(n_3_2_off, axiom, f(X0,X1,on) = f(X0,X1,off)).
cnf(n_3_2_on,  axiom, f(X0,X1,off) = f(X0,X1,on)).
```

	Instance size	Rewrite steps	Action equivalents	Prover time
alldiff_switch_var001.p.prf	1	1	1	0.006
alldiff_switch_var002.p.prf	2	2	2	0.005
alldiff_switch_var003.p.prf	3	3	3	0.004
alldiff_switch_var004.p.prf	4	4	4	0.003
alldiff_switch_var005.p.prf	5	5	5	0.003
alldiff_switch_var006.p.prf	6	6	6	0.003
alldiff_switch_var007.p.prf	7	7	7	0.003
alldiff_switch_var008.p.prf	8	8	8	0.003
alldiff_switch_var009.p.prf	9	9	9	0.003
alldiff_switch_var010.p.prf	10	10	10	0.003
alldiff_switch_var011.p.prf	11	11	11	0.003
alldiff_switch_var012.p.prf	12	12	12	0.003
alldiff_switch_var013.p.prf	13	13	13	0.003
alldiff_switch_var014.p.prf	14	14	14	0.003
alldiff_switch_var015.p.prf	15	15	15	0.003
alldiff_switch_var016.p.prf	16	16	16	0.003
alldiff_switch_var017.p.prf	17	17	17	0.003
alldiff_switch_var018.p.prf	18	18	18	0.003
alldiff_switch_var019.p.prf	19	19	19	0.003
alldiff_switch_var020.p.prf	20	20	20	0.003

Table 4. Results on switch banks with frame variable encoding

Actions rewrite subterms In this setting, the initial and final states differ in all switches, and the actions are encoded as equations directly changing the value of a switch at the subterm level. An example of size 3 is presented below. Results are in Table 5.

```
% Simple switchbank example. There are 3 switches
% that can be on or off. Actions flip a single switch.
%
% This is the version using rewriting at subterm positions,
% using a single action rule that can flip any switch.
%
% There should be two axioms, on->off and off->on. However,
% since we require bidirectionality, both are covered by a
% single equation.

cnf(to_plan, negated_conjecture, f(on,off,on)!=f(off,on,off)).
cnf(onoff,axiom, on=off).
```

A.2 Switches - minimal difference, default ordering

The next three examples are encoded as above, but with a conjecture representing different initial and goal states. We only present one example (in the subterm action encoding) to demonstrate this difference. Results are in Tables 6, 7 and

	Instance size	Rewrite steps	Action equivalents	Prover time
alldiff_switch_sub001.p.prf	1	1	1	0.006
alldiff_switch_sub002.p.prf	2	2	2	0.005
alldiff_switch_sub003.p.prf	3	3	3	0.004
alldiff_switch_sub004.p.prf	4	4	4	0.003
alldiff_switch_sub005.p.prf	5	5	5	0.003
alldiff_switch_sub006.p.prf	6	6	6	0.003
alldiff_switch_sub007.p.prf	7	7	7	0.003
alldiff_switch_sub008.p.prf	8	8	8	0.003
alldiff_switch_sub009.p.prf	9	9	9	0.003
alldiff_switch_sub010.p.prf	10	10	10	0.002
alldiff_switch_sub011.p.prf	11	11	11	0.002
alldiff_switch_sub012.p.prf	12	12	12	0.002
alldiff_switch_sub013.p.prf	13	13	13	0.002
alldiff_switch_sub014.p.prf	14	14	14	0.002
alldiff_switch_sub015.p.prf	15	15	15	0.002
alldiff_switch_sub016.p.prf	16	16	16	0.002
alldiff_switch_sub017.p.prf	17	17	17	0.002
alldiff_switch_sub018.p.prf	18	18	18	0.002
alldiff_switch_sub019.p.prf	19	19	19	0.002
alldiff_switch_sub020.p.prf	20	20	20	0.002

Table 5. Results on switch banks with subterm action encoding

8. The prover nearly always generates the optimal ordering by default, but in the onediff_switch_gnd002 example (Table 6) it generates the less than optimal *on>off*.

```
% Simple switchbank example. There are 3 switches
% that can be on or off. Actions flip a single switch.
%
% This is the version using rewriting at subterm positions,
% using a single action rule that can flip any switch.
%
% There should be two axioms, on->off and off->on. However,
% since we require bidirectionality, both are covered by a
% single equation.

cnf(to_plan, negated_conjecture, f(on,on,on)!=f(off,on,on)).
cnf(onoff,axiom, on=off).
```

A.3 Switches - minimal difference, different orderings

These problems use the same encoding as above, but we explicitly specify the term ordering, using `--precedence='f>off>on'` for the first (Table 9) and `--precedence='f>on>off'` for the second (Table 10) set of experiments.

	Instance size	Rewrite steps	Action equivalents	Prover time
onediff_switch_gnd001.p.prf	1	1	1	0.003
onediff_switch_gnd002.p.prf	2	3	3	0.002
onediff_switch_gnd003.p.prf	3	1	1	0.002
onediff_switch_gnd004.p.prf	4	1	1	0.002
onediff_switch_gnd005.p.prf	5	1	1	0.003
onediff_switch_gnd006.p.prf	6	1	1	0.004
onediff_switch_gnd007.p.prf	7	1	1	0.008
onediff_switch_gnd008.p.prf	8	1	1	0.019
onediff_switch_gnd009.p.prf	9	1	1	0.046
onediff_switch_gnd010.p.prf	10	1	1	0.111
onediff_switch_gnd011.p.prf	11	1	1	0.283
onediff_switch_gnd012.p.prf	12	1	1	0.767
onediff_switch_gnd013.p.prf	13	1	1	2.136
onediff_switch_gnd014.p.prf	14	1	1	6.257
onediff_switch_gnd015.p.prf	15	1	1	20.225
onediff_switch_gnd016.p.prf	16	1	1	83.521
onediff_switch_gnd017.p.prf	17	1	1	346.459
onediff_switch_gnd018.p.prf	18	1	1	1607.884
onediff_switch_gnd019.p.prf	19	1	1	6946.928
onediff_switch_gnd020.p.prf	20	1	1	31387.790

Table 6. Results on switch banks with one switch difference, ground encoding

	Instance size	Rewrite steps	Action equivalents	Prover time
onediff_switch_var001.p.prf	1	1	1	0.005
onediff_switch_var002.p.prf	2	1	1	0.004
onediff_switch_var003.p.prf	3	1	1	0.003
onediff_switch_var004.p.prf	4	1	1	0.003
onediff_switch_var005.p.prf	5	1	1	0.002
onediff_switch_var006.p.prf	6	1	1	0.003
onediff_switch_var007.p.prf	7	1	1	0.003
onediff_switch_var008.p.prf	8	1	1	0.003
onediff_switch_var009.p.prf	9	1	1	0.003
onediff_switch_var010.p.prf	10	1	1	0.003
onediff_switch_var011.p.prf	11	1	1	0.003
onediff_switch_var012.p.prf	12	1	1	0.003
onediff_switch_var013.p.prf	13	1	1	0.003
onediff_switch_var014.p.prf	14	1	1	0.003
onediff_switch_var015.p.prf	15	1	1	0.003
onediff_switch_var016.p.prf	16	1	1	0.003
onediff_switch_var017.p.prf	17	1	1	0.003
onediff_switch_var018.p.prf	18	1	1	0.003
onediff_switch_var019.p.prf	19	1	1	0.003
onediff_switch_var020.p.prf	20	1	1	0.003

Table 7. Results on switch banks with one switch difference, frame variable encoding

	Instance size	Rewrite steps	Action equivalents	Prover time
onediff_switch_sub001.p.prf	1	1	1	0.007
onediff_switch_sub002.p.prf	2	1	1	0.004
onediff_switch_sub003.p.prf	3	1	1	0.004
onediff_switch_sub004.p.prf	4	1	1	0.003
onediff_switch_sub005.p.prf	5	1	1	0.003
onediff_switch_sub006.p.prf	6	1	1	0.003
onediff_switch_sub007.p.prf	7	1	1	0.003
onediff_switch_sub008.p.prf	8	1	1	0.003
onediff_switch_sub009.p.prf	9	1	1	0.003
onediff_switch_sub010.p.prf	10	1	1	0.003
onediff_switch_sub011.p.prf	11	1	1	0.003
onediff_switch_sub012.p.prf	12	1	1	0.002
onediff_switch_sub013.p.prf	13	1	1	0.002
onediff_switch_sub014.p.prf	14	1	1	0.002
onediff_switch_sub015.p.prf	15	1	1	0.002
onediff_switch_sub016.p.prf	16	1	1	0.002
onediff_switch_sub017.p.prf	17	1	1	0.002
onediff_switch_sub018.p.prf	18	1	1	0.002
onediff_switch_sub019.p.prf	19	1	1	0.002
onediff_switch_sub020.p.prf	20	1	1	0.002

Table 8. Results on switch banks with one switch difference, subterm action encoding

	Instance size	Rewrite steps	Action equivalents	Prover time
off_ononediff_switch_sub001.p.prf	1	1	1	0.003
off_ononediff_switch_sub002.p.prf	2	1	1	0.003
off_ononediff_switch_sub003.p.prf	3	1	1	0.003
off_ononediff_switch_sub004.p.prf	4	1	1	0.002
off_ononediff_switch_sub005.p.prf	5	1	1	0.002
off_ononediff_switch_sub006.p.prf	6	1	1	0.002
off_ononediff_switch_sub007.p.prf	7	1	1	0.002
off_ononediff_switch_sub008.p.prf	8	1	1	0.002
off_ononediff_switch_sub009.p.prf	9	1	1	0.002
off_ononediff_switch_sub010.p.prf	10	1	1	0.002
off_ononediff_switch_sub011.p.prf	11	1	1	0.002
off_ononediff_switch_sub012.p.prf	12	1	1	0.002
off_ononediff_switch_sub013.p.prf	13	1	1	0.002
off_ononediff_switch_sub014.p.prf	14	1	1	0.002
off_ononediff_switch_sub015.p.prf	15	1	1	0.002
off_ononediff_switch_sub016.p.prf	16	1	1	0.002
off_ononediff_switch_sub017.p.prf	17	1	1	0.002
off_ononediff_switch_sub018.p.prf	18	1	1	0.002
off_ononediff_switch_sub019.p.prf	19	1	1	0.002
off_ononediff_switch_sub020.p.prf	20	1	1	0.002

Table 9. Results on switch banks with one switch difference, subterm action encoding, $\mathit{off} > \mathit{on}$

	Instance size	Rewrite steps	Action equivalents	Prover time
on_offonediff_switch_sub001.p.prf	1	1	1	0.005
on_offonediff_switch_sub002.p.prf	2	3	3	0.003
on_offonediff_switch_sub003.p.prf	3	5	5	0.003
on_offonediff_switch_sub004.p.prf	4	7	7	0.003
on_offonediff_switch_sub005.p.prf	5	9	9	0.002
on_offonediff_switch_sub006.p.prf	6	11	11	0.002
on_offonediff_switch_sub007.p.prf	7	13	13	0.002
on_offonediff_switch_sub008.p.prf	8	15	15	0.002
on_offonediff_switch_sub009.p.prf	9	17	17	0.002
on_offonediff_switch_sub010.p.prf	10	19	19	0.002
on_offonediff_switch_sub011.p.prf	11	21	21	0.002
on_offonediff_switch_sub012.p.prf	12	23	23	0.002
on_offonediff_switch_sub013.p.prf	13	25	25	0.002
on_offonediff_switch_sub014.p.prf	14	27	27	0.002
on_offonediff_switch_sub015.p.prf	15	29	29	0.002
on_offonediff_switch_sub016.p.prf	16	31	31	0.002
on_offonediff_switch_sub017.p.prf	17	33	33	0.002
on_offonediff_switch_sub018.p.prf	18	35	35	0.002
on_offonediff_switch_sub019.p.prf	19	37	37	0.002
on_offonediff_switch_sub020.p.prf	20	39	39	0.002

Table 10. Results on switch banks with one switch difference, subterm action encoding, *on*>*off*

A.4 Tower of Hanoi

Flat encoding In this encoding, a state is a flat term of the form $f(X_1, \ldots, X_n)$, where X_i encodes the ith disk and can take one of the values p_1, p_2, p_3, denoting peg one, two or three, respectively. The task is to move all disks from peg one to peg two. Results are in Table 11.

An example of size 3 is presented below.

```
% This is an encoding of the Tower of Hanoi puzzle.
% There are 3 differently sized disks sitting on one of the
% 3 pegs p1, p2, p3. A bigger disk may never sit on a smaller
% disk. One can move the top disk from one peg to another peg
% if this does nor violate the size constraint.
% The goal is to move all disks from p1 to p2.
%
% The encoding represents the disks as argument position
% in a term of the form f(arg1, arg2, ..., argn), where each
% arg can take the value p1, p2 or p3. The largest disk is on
% the left, the smallest on the right.
%
% There should be 6n=18 axioms (plus inital and goal state), one each
% to move disk k from any peg to any other peg (for all 6
% combinations of two distinct pegs). However, because the axioms
% are bidirectional, half of them are redundant, so we only need
```

```
% 3n = 9 axioms.

cnf(to_plan, negated_conjecture, f(p1,p1,p1)!=f(p2,p2,p2)).
cnf(h3_0p1p2p3, axiom, f(p1,p3,p3)=f(p2,p3,p3)).
cnf(h3_1p1p2p3, axiom, f(X0,p1,p3)=f(X0,p2,p3)).
cnf(h3_2p1p2p3, axiom, f(X0,X1,p1)=f(X0,X1,p2)).
cnf(h3_0p1p3p2, axiom, f(p1,p2,p2)=f(p3,p2,p2)).
cnf(h3_1p1p3p2, axiom, f(X0,p1,p2)=f(X0,p3,p2)).
cnf(h3_2p1p3p2, axiom, f(X0,X1,p1)=f(X0,X1,p3)).
cnf(h3_0p2p3p1, axiom, f(p2,p1,p1)=f(p3,p1,p1)).
cnf(h3_1p2p3p1, axiom, f(X0,p2,p1)=f(X0,p3,p1)).
cnf(h3_2p2p3p1, axiom, f(X0,X1,p2)=f(X0,X1,p3)).
```

	Instance size	Rewrite steps	Action equivalents	Prover time
hanoi_var001.p.prf	1	1	1	0.008
hanoi_var002.p.prf	2	6	6	0.005
hanoi_var003.p.prf	3	13	17	0.004
hanoi_var004.p.prf	4	27	47	0.004
hanoi_var005.p.prf	5	33	147	0.003
hanoi_var006.p.prf	6	46	436	0.004
hanoi_var007.p.prf	7	61	1301	0.004
hanoi_var008.p.prf	8	78	3894	0.003
hanoi_var009.p.prf	9	97	11671	0.003
hanoi_var010.p.prf	10	118	35000	0.003
hanoi_var011.p.prf	11	141	104985	0.003
hanoi_var012.p.prf	12	163	321495	0.003
hanoi_var013.p.prf	13	193	944795	0.004
hanoi_var014.p.prf	14	208	3188632	0.003
hanoi_var015.p.prf	15	238	9565923	0.004
hanoi_var016.p.prf	16	270	28697798	0.003
hanoi_var017.p.prf	17	304	86093425	0.004
hanoi_var018.p.prf	18	340	258280308	0.004
hanoi_var019.p.prf	19	378	774840959	0.004
hanoi_var020.p.prf	20	418	2324522914	0.004
hanoi_var021.p.prf	21	460	6973568781	0.004
hanoi_var022.p.prf	22	504	20920706384	0.005
hanoi_var023.p.prf	23	550	62762119195	0.005
hanoi_var024.p.prf	24	598	188286357630	0.005
hanoi_var025.p.prf	25	648	564859072937	0.006
hanoi_var026.p.prf	26	700	1694577218860	0.006
hanoi_var027.p.prf	27	754	5083731656631	0.007
hanoi_var028.p.prf	28	810	15251194969946	0.007
hanoi_var029.p.prf	29	868	45753584909893	0.007
hanoi_var030.p.prf	30	928	137260754729736	0.008

Table 11. Results on Tower of Hanoi, flat encoding, default ordering

Flat encoding, different orderings Tables 12 to 17 show the results for the Tower of Hanoi problem with flat encoding and different term orderings.

	Instance size	Rewrite steps	Action equivalents	Prover time
p1p2p3_hanoi_var001.p.prf	1	1	1	0.009
p1p2p3_hanoi_var002.p.prf	2	10	13	0.004
p1p2p3_hanoi_var003.p.prf	3	11	14	0.002
p1p2p3_hanoi_var004.p.prf	4	20	33	0.002
p1p2p3_hanoi_var005.p.prf	5	30	110	0.002
p1p2p3_hanoi_var006.p.prf	6	42	339	0.002
p1p2p3_hanoi_var007.p.prf	7	56	1024	0.002
p1p2p3_hanoi_var008.p.prf	8	72	3077	0.002
p1p2p3_hanoi_var009.p.prf	9	90	9234	0.002
p1p2p3_hanoi_var010.p.prf	10	110	27703	0.003
p1p2p3_hanoi_var011.p.prf	11	132	83108	0.003
p1p2p3_hanoi_var012.p.prf	12	155	354304	0.003
p1p2p3_hanoi_var013.p.prf	13	182	747958	0.003
p1p2p3_hanoi_var014.p.prf	14	209	3188658	0.003
p1p2p3_hanoi_var015.p.prf	15	239	9565951	0.003
p1p2p3_hanoi_var016.p.prf	16	271	28697828	0.003
p1p2p3_hanoi_var017.p.prf	17	305	86093457	0.004
p1p2p3_hanoi_var018.p.prf	18	341	258280342	0.004
p1p2p3_hanoi_var019.p.prf	19	379	774840995	0.004
p1p2p3_hanoi_var020.p.prf	20	419	2324522952	0.004
p1p2p3_hanoi_var021.p.prf	21	461	6973568821	0.005
p1p2p3_hanoi_var022.p.prf	22	505	20920706426	0.005
p1p2p3_hanoi_var023.p.prf	23	551	62762119239	0.005
p1p2p3_hanoi_var024.p.prf	24	599	188286357676	0.005
p1p2p3_hanoi_var025.p.prf	25	649	564859072985	0.006
p1p2p3_hanoi_var026.p.prf	26	701	1694577218910	0.006
p1p2p3_hanoi_var027.p.prf	27	755	5083731656683	0.007
p1p2p3_hanoi_var028.p.prf	28	811	15251194970000	0.007
p1p2p3_hanoi_var029.p.prf	29	869	45753584909949	0.007
p1p2p3_hanoi_var030.p.prf	30	929	137260754729794	0.008

Table 12. Results on Tower of Hanoi, flat encoding, $p_1 > p_2 > p_3$

List encoding with subterm actions In this encoding, a state is a recursively encoded list of positions, $f(X_1, f(X_2, f(\ldots, \bot)))$, where again X_i encodes the ith disk and can take one of the values p_1, p_2, p_3. The task is to move all disks from peg one to peg two. An example of size 3 is presented below. Results are in Table 18.

```
% This is an encoding of the Tower of Hanoi puzzle.
% There are 3 differently sized disks sitting on one of the
% 3 pegs p1, p2, p3. A bigger disk may never sit on a smaller
```

	Instance size	Rewrite steps	Action equivalents	Prover time
p1p3p2_hanoi_var001.p.prf	1	1	1	0.008
p1p3p2_hanoi_var002.p.prf	2	7	8	0.004
p1p3p2_hanoi_var003.p.prf	3	13	17	0.002
p1p3p2_hanoi_var004.p.prf	4	28	51	0.002
p1p3p2_hanoi_var005.p.prf	5	34	155	0.002
p1p3p2_hanoi_var006.p.prf	6	47	462	0.002
p1p3p2_hanoi_var007.p.prf	7	62	1381	0.002
p1p3p2_hanoi_var008.p.prf	8	79	4136	0.002
p1p3p2_hanoi_var009.p.prf	9	98	12399	0.002
p1p3p2_hanoi_var010.p.prf	10	119	37186	0.003
p1p3p2_hanoi_var011.p.prf	11	142	111545	0.003
p1p3p2_hanoi_var012.p.prf	12	168	354301	0.003
p1p3p2_hanoi_var013.p.prf	13	194	1003843	0.003
p1p3p2_hanoi_var014.p.prf	14	353	3188646	0.003
p1p3p2_hanoi_var015.p.prf	15	408	9565937	0.003
p1p3p2_hanoi_var016.p.prf	16	467	28697814	0.003
p1p3p2_hanoi_var017.p.prf	17	530	86093441	0.004
p1p3p2_hanoi_var018.p.prf	18	597	258280326	0.004
p1p3p2_hanoi_var019.p.prf	19	668	774840977	0.004
p1p3p2_hanoi_var020.p.prf	20	743	2324522934	0.004
p1p3p2_hanoi_var021.p.prf	21	822	6973568801	0.005
p1p3p2_hanoi_var022.p.prf	22	905	20920706406	0.005
p1p3p2_hanoi_var023.p.prf	23	992	62762119217	0.005
p1p3p2_hanoi_var024.p.prf	24	1083	188286357654	0.005
p1p3p2_hanoi_var025.p.prf	25	1178	564859072961	0.006
p1p3p2_hanoi_var026.p.prf	26	1277	1694577218886	0.006
p1p3p2_hanoi_var027.p.prf	27	1380	5083731656657	0.007
p1p3p2_hanoi_var028.p.prf	28	1487	15251194969974	0.007
p1p3p2_hanoi_var029.p.prf	29	1598	45753584909921	0.007
p1p3p2_hanoi_var030.p.prf	30	1713	137260754729766	0.008

Table 13. Results on Tower of Hanoi, flat encoding, $p_1 > p_3 > p_2$

```
% disk. One can move the top disk from one peg to another peg
% if this does nor violate the size constraint.
% The goal is to move all disks from p1 to p2.
%
% The encoding represents the disks as argument positions
% in a term of the form f(arg1, f(arg2, ..., f(argn, bot))),
% where each arg can take the value p1, p2 or p3. The largest
% disk is on % the left, the smallest on the right.
%
% There should be 6n=18 axioms (plus inital and goal state), one each
% to move disk k from any peg to any other peg (for all 6
% combinations of two distinct pegs). However, because the axioms
% are bidirectional, half of them are redundant, so we only need
% 3n = 9 axioms.
```

	Instance size	Rewrite steps	Action equivalents	Prover time
p2p1p3_hanoi_var001.p.prf	1	1	1	0.005
p2p1p3_hanoi_var002.p.prf	2	9	9	0.004
p2p1p3_hanoi_var003.p.prf	3	10	10	0.002
p2p1p3_hanoi_var004.p.prf	4	19	25	0.002
p2p1p3_hanoi_var005.p.prf	5	29	82	0.002
p2p1p3_hanoi_var006.p.prf	6	41	251	0.002
p2p1p3_hanoi_var007.p.prf	7	55	756	0.002
p2p1p3_hanoi_var008.p.prf	8	71	2269	0.002
p2p1p3_hanoi_var009.p.prf	9	89	6806	0.002
p2p1p3_hanoi_var010.p.prf	10	109	20415	0.003
p2p1p3_hanoi_var011.p.prf	11	131	61240	0.003
p2p1p3_hanoi_var012.p.prf	12	154	236206	0.003
p2p1p3_hanoi_var013.p.prf	13	181	551130	0.003
p2p1p3_hanoi_var014.p.prf	14	208	2125776	0.003
p2p1p3_hanoi_var015.p.prf	15	238	6377305	0.003
p2p1p3_hanoi_var016.p.prf	16	270	19131890	0.003
p2p1p3_hanoi_var017.p.prf	17	304	57395643	0.004
p2p1p3_hanoi_var018.p.prf	18	340	172186900	0.004
p2p1p3_hanoi_var019.p.prf	19	378	516560669	0.004
p2p1p3_hanoi_var020.p.prf	20	418	1549681974	0.004
p2p1p3_hanoi_var021.p.prf	21	460	4649045887	0.005
p2p1p3_hanoi_var022.p.prf	22	504	13947137624	0.005
p2p1p3_hanoi_var023.p.prf	23	550	41841412833	0.005
p2p1p3_hanoi_var024.p.prf	24	598	125524238458	0.006
p2p1p3_hanoi_var025.p.prf	25	648	376572715331	0.006
p2p1p3_hanoi_var026.p.prf	26	700	1129718145948	0.006
p2p1p3_hanoi_var027.p.prf	27	754	3389154437797	0.007
p2p1p3_hanoi_var028.p.prf	28	810	10167463313342	0.007
p2p1p3_hanoi_var029.p.prf	29	868	30502389939975	0.007
p2p1p3_hanoi_var030.p.prf	30	928	91507169819872	0.008

Table 14. Results on Tower of Hanoi, flat encoding, $p_2 > p_1 > p_3$

```
cnf(to_plan, negated_conjecture, f(p1,f(p1,f(p1,bot)))!=f(p2,f(p2,f(p2,bot)))).
cnf(h3_0p1p2p3, axiom, f(p1,f(p3,f(p3,bot)))=f(p2,f(p3,f(p3,bot)))).
cnf(h3_1p1p2p3, axiom, f(p1,f(p3,bot))=f(p2,f(p3,bot))).
cnf(h3_2p1p2p3, axiom, f(p1,bot)=f(p2,bot)).
cnf(h3_0p1p3p2, axiom, f(p1,f(p2,f(p2,bot)))=f(p3,f(p2,f(p2,bot)))).
cnf(h3_1p1p3p2, axiom, f(p1,f(p2,bot))=f(p3,f(p2,bot))).
cnf(h3_2p1p3p2, axiom, f(p1,bot)=f(p3,bot)).
cnf(h3_0p2p3p1, axiom, f(p2,f(p1,f(p1,bot)))=f(p3,f(p1,f(p1,bot)))).
cnf(h3_1p2p3p1, axiom, f(p2,f(p1,bot))=f(p3,f(p1,bot))).
cnf(h3_2p2p3p1, axiom, f(p2,bot)=f(p3,bot)).
```

Recursive encoding, different orderings Tables 19 to 24 show the results for the Tower of Hanoi problem with recursive encoding and different term orderings.

	Instance size	Rewrite steps	Action equivalents	Prover time
p2p3p1_hanoi_var001.p.prf	1	1	1	0.005
p2p3p1_hanoi_var002.p.prf	2	6	6	0.003
p2p3p1_hanoi_var003.p.prf	3	13	17	0.002
p2p3p1_hanoi_var004.p.prf	4	27	47	0.002
p2p3p1_hanoi_var005.p.prf	5	33	147	0.002
p2p3p1_hanoi_var006.p.prf	6	46	436	0.002
p2p3p1_hanoi_var007.p.prf	7	61	1301	0.002
p2p3p1_hanoi_var008.p.prf	8	78	3894	0.002
p2p3p1_hanoi_var009.p.prf	9	97	11671	0.002
p2p3p1_hanoi_var010.p.prf	10	118	35000	0.003
p2p3p1_hanoi_var011.p.prf	11	141	104985	0.003
p2p3p1_hanoi_var012.p.prf	12	163	321495	0.003
p2p3p1_hanoi_var013.p.prf	13	193	944795	0.003
p2p3p1_hanoi_var014.p.prf	14	208	3188632	0.003
p2p3p1_hanoi_var015.p.prf	15	238	9565923	0.003
p2p3p1_hanoi_var016.p.prf	16	270	28697798	0.003
p2p3p1_hanoi_var017.p.prf	17	304	86093425	0.004
p2p3p1_hanoi_var018.p.prf	18	340	258280308	0.004
p2p3p1_hanoi_var019.p.prf	19	378	774840959	0.004
p2p3p1_hanoi_var020.p.prf	20	418	2324522914	0.004
p2p3p1_hanoi_var021.p.prf	21	460	6973568781	0.004
p2p3p1_hanoi_var022.p.prf	22	504	20920706384	0.005
p2p3p1_hanoi_var023.p.prf	23	550	62762119195	0.005
p2p3p1_hanoi_var024.p.prf	24	598	188286357630	0.005
p2p3p1_hanoi_var025.p.prf	25	648	564859072937	0.006
p2p3p1_hanoi_var026.p.prf	26	700	1694577218860	0.006
p2p3p1_hanoi_var027.p.prf	27	754	5083731656631	0.006
p2p3p1_hanoi_var028.p.prf	28	810	15251194969946	0.007
p2p3p1_hanoi_var029.p.prf	29	868	45753584909893	0.007
p2p3p1_hanoi_var030.p.prf	30	928	137260754729736	0.008

Table 15. Results on Tower of Hanoi, flat encoding, $p_2 > p_3 > p_1$

	Instance size	Rewrite steps	Action equivalents	Prover time
p3p1p2_hanoi_var001.p.prf	1	1	1	0.004
p3p1p2_hanoi_var002.p.prf	2	5	6	0.003
p3p1p2_hanoi_var003.p.prf	3	11	17	0.002
p3p1p2_hanoi_var004.p.prf	4	19	53	0.002
p3p1p2_hanoi_var005.p.prf	5	29	161	0.002
p3p1p2_hanoi_var006.p.prf	6	41	485	0.002
p3p1p2_hanoi_var007.p.prf	7	55	1457	0.002
p3p1p2_hanoi_var008.p.prf	8	71	4373	0.002
p3p1p2_hanoi_var009.p.prf	9	89	13121	0.002
p3p1p2_hanoi_var010.p.prf	10	109	39365	0.003
p3p1p2_hanoi_var011.p.prf	11	131	118097	0.003
p3p1p2_hanoi_var012.p.prf	12	156	354293	0.003
p3p1p2_hanoi_var013.p.prf	13	181	1062881	0.003
p3p1p2_hanoi_var014.p.prf	14	353	3188646	0.003
p3p1p2_hanoi_var015.p.prf	15	408	9565937	0.003
p3p1p2_hanoi_var016.p.prf	16	467	28697814	0.003
p3p1p2_hanoi_var017.p.prf	17	530	86093441	0.004
p3p1p2_hanoi_var018.p.prf	18	597	258280326	0.004
p3p1p2_hanoi_var019.p.prf	19	668	774840977	0.004
p3p1p2_hanoi_var020.p.prf	20	743	2324522934	0.004
p3p1p2_hanoi_var021.p.prf	21	822	6973568801	0.005
p3p1p2_hanoi_var022.p.prf	22	905	20920706406	0.005
p3p1p2_hanoi_var023.p.prf	23	992	62762119217	0.005
p3p1p2_hanoi_var024.p.prf	24	1083	188286357654	0.005
p3p1p2_hanoi_var025.p.prf	25	1178	564859072961	0.006
p3p1p2_hanoi_var026.p.prf	26	1277	1694577218886	0.006
p3p1p2_hanoi_var027.p.prf	27	1380	5083731656657	0.007
p3p1p2_hanoi_var028.p.prf	28	1487	15251194969974	0.007
p3p1p2_hanoi_var029.p.prf	29	1598	45753584909921	0.007
p3p1p2_hanoi_var030.p.prf	30	1713	137260754729766	0.008

Table 16. Results on Tower of Hanoi, flat encoding, $p_3 > p_1 > p_2$

	Instance size	Rewrite steps	Action equivalents	Prover time
p3p2p1_hanoi_var001.p.prf	1	1	1	0.005
p3p2p1_hanoi_var002.p.prf	2	4	4	0.003
p3p2p1_hanoi_var003.p.prf	3	11	17	0.002
p3p2p1_hanoi_var004.p.prf	4	19	53	0.002
p3p2p1_hanoi_var005.p.prf	5	29	161	0.002
p3p2p1_hanoi_var006.p.prf	6	41	485	0.002
p3p2p1_hanoi_var007.p.prf	7	55	1457	0.002
p3p2p1_hanoi_var008.p.prf	8	71	4373	0.002
p3p2p1_hanoi_var009.p.prf	9	89	13121	0.002
p3p2p1_hanoi_var010.p.prf	10	109	39365	0.003
p3p2p1_hanoi_var011.p.prf	11	131	118097	0.003
p3p2p1_hanoi_var012.p.prf	12	154	314927	0.003
p3p2p1_hanoi_var013.p.prf	13	181	1062881	0.003
p3p2p1_hanoi_var014.p.prf	14	208	3188632	0.003
p3p2p1_hanoi_var015.p.prf	15	238	9565923	0.003
p3p2p1_hanoi_var016.p.prf	16	270	28697798	0.003
p3p2p1_hanoi_var017.p.prf	17	304	86093425	0.003
p3p2p1_hanoi_var018.p.prf	18	340	258280308	0.004
p3p2p1_hanoi_var019.p.prf	19	378	774840959	0.004
p3p2p1_hanoi_var020.p.prf	20	418	2324522914	0.004
p3p2p1_hanoi_var021.p.prf	21	460	6973568781	0.004
p3p2p1_hanoi_var022.p.prf	22	504	20920706384	0.005
p3p2p1_hanoi_var023.p.prf	23	550	62762119195	0.005
p3p2p1_hanoi_var024.p.prf	24	598	188286357630	0.005
p3p2p1_hanoi_var025.p.prf	25	648	564859072937	0.006
p3p2p1_hanoi_var026.p.prf	26	700	1694577218860	0.006
p3p2p1_hanoi_var027.p.prf	27	754	5083731656631	0.006
p3p2p1_hanoi_var028.p.prf	28	810	15251194969946	0.007
p3p2p1_hanoi_var029.p.prf	29	868	45753584909893	0.007
p3p2p1_hanoi_var030.p.prf	30	928	137260754729736	0.008

Table 17. Results on Tower of Hanoi, flat encoding, $p_3 > p_2 > p_1$

	Instance size	Rewrite steps	Action equivalents	Prover time
hanoi_sub001.p.prf	1	1	1	0.007
hanoi_sub002.p.prf	2	4	4	0.004
hanoi_sub003.p.prf	3	9	15	0.004
hanoi_sub004.p.prf	4	16	50	0.004
hanoi_sub005.p.prf	5	25	157	0.003
hanoi_sub006.p.prf	6	36	480	0.003
hanoi_sub007.p.prf	7	49	1451	0.003
hanoi_sub008.p.prf	8	64	4366	0.003
hanoi_sub009.p.prf	9	81	13113	0.003
hanoi_sub010.p.prf	10	100	39356	0.003
hanoi_sub011.p.prf	11	121	118087	0.003
hanoi_sub012.p.prf	12	144	354282	0.003
hanoi_sub013.p.prf	13	169	1062869	0.003
hanoi_sub014.p.prf	14	196	3188632	0.003
hanoi_sub015.p.prf	15	225	9565923	0.003
hanoi_sub016.p.prf	16	256	28697798	0.003
hanoi_sub017.p.prf	17	289	86093425	0.003
hanoi_sub018.p.prf	18	324	258280308	0.004
hanoi_sub019.p.prf	19	361	774840959	0.004
hanoi_sub020.p.prf	20	400	2324522914	0.004
hanoi_sub021.p.prf	21	441	6973568781	0.004
hanoi_sub022.p.prf	22	484	20920706384	0.004
hanoi_sub023.p.prf	23	529	62762119195	0.004
hanoi_sub024.p.prf	24	576	188286357630	0.005
hanoi_sub025.p.prf	25	625	564859072937	0.005
hanoi_sub026.p.prf	26	676	1694577218860	0.005
hanoi_sub027.p.prf	27	729	5083731656631	0.005
hanoi_sub028.p.prf	28	784	15251194969946	0.006
hanoi_sub029.p.prf	29	841	45753584909893	0.006
hanoi_sub030.p.prf	30	900	137260754729736	0.007

Table 18. Results on Tower of Hanoi, recursive encoding, default ordering

	Instance size	Rewrite steps	Action equivalents	Prover time
p1p2p3_hanoi_sub001.p.prf	1	1	1	0.009
p1p2p3_hanoi_sub002.p.prf	2	5	6	0.003
p1p2p3_hanoi_sub003.p.prf	3	10	21	0.003
p1p2p3_hanoi_sub004.p.prf	4	18	68	0.002
p1p2p3_hanoi_sub005.p.prf	5	28	211	0.002
p1p2p3_hanoi_sub006.p.prf	6	40	642	0.002
p1p2p3_hanoi_sub007.p.prf	7	54	1937	0.002
p1p2p3_hanoi_sub008.p.prf	8	70	5824	0.002
p1p2p3_hanoi_sub009.p.prf	9	88	17487	0.002
p1p2p3_hanoi_sub010.p.prf	10	108	52478	0.003
p1p2p3_hanoi_sub011.p.prf	11	130	157453	0.003
p1p2p3_hanoi_sub012.p.prf	12	154	472380	0.003
p1p2p3_hanoi_sub013.p.prf	13	180	1417163	0.003
p1p2p3_hanoi_sub014.p.prf	14	208	4251514	0.003
p1p2p3_hanoi_sub015.p.prf	15	238	12754569	0.004
p1p2p3_hanoi_sub016.p.prf	16	270	38263736	0.004
p1p2p3_hanoi_sub017.p.prf	17	304	114791239	0.004
p1p2p3_hanoi_sub018.p.prf	18	340	344373750	0.004
p1p2p3_hanoi_sub019.p.prf	19	378	1033121285	0.005
p1p2p3_hanoi_sub020.p.prf	20	418	3099363892	0.005
p1p2p3_hanoi_sub021.p.prf	21	460	9298091715	0.005
p1p2p3_hanoi_sub022.p.prf	22	504	27894275186	0.006
p1p2p3_hanoi_sub023.p.prf	23	550	83682825601	0.006
p1p2p3_hanoi_sub024.p.prf	24	598	251048476848	0.007
p1p2p3_hanoi_sub025.p.prf	25	648	753145430591	0.007
p1p2p3_hanoi_sub026.p.prf	26	700	2259436291822	0.007
p1p2p3_hanoi_sub027.p.prf	27	754	6778308875517	0.008
p1p2p3_hanoi_sub028.p.prf	28	810	20334926626604	0.008
p1p2p3_hanoi_sub029.p.prf	29	868	61004779879867	0.009
p1p2p3_hanoi_sub030.p.prf	30	928	183014339639658	0.009

Table 19. Results on Tower of Hanoi, recursive encoding, $p_1 > p_2 > p_3$

	Instance size	Rewrite steps	Action equivalents	Prover time
p1p3p2_hanoi_sub001.p.prf	1	1	1	0.006
p1p3p2_hanoi_sub002.p.prf	2	5	6	0.003
p1p3p2_hanoi_sub003.p.prf	3	11	17	0.002
p1p3p2_hanoi_sub004.p.prf	4	21	54	0.002
p1p3p2_hanoi_sub005.p.prf	5	35	161	0.002
p1p3p2_hanoi_sub006.p.prf	6	53	486	0.002
p1p3p2_hanoi_sub007.p.prf	7	75	1457	0.002
p1p3p2_hanoi_sub008.p.prf	8	101	4374	0.002
p1p3p2_hanoi_sub009.p.prf	9	131	13121	0.002
p1p3p2_hanoi_sub010.p.prf	10	165	39366	0.002
p1p3p2_hanoi_sub011.p.prf	11	203	118097	0.002
p1p3p2_hanoi_sub012.p.prf	12	245	354294	0.003
p1p3p2_hanoi_sub013.p.prf	13	291	1062881	0.003
p1p3p2_hanoi_sub014.p.prf	14	341	3188646	0.003
p1p3p2_hanoi_sub015.p.prf	15	395	9565937	0.003
p1p3p2_hanoi_sub016.p.prf	16	453	28697814	0.003
p1p3p2_hanoi_sub017.p.prf	17	515	86093441	0.003
p1p3p2_hanoi_sub018.p.prf	18	581	258280326	0.003
p1p3p2_hanoi_sub019.p.prf	19	651	774840977	0.004
p1p3p2_hanoi_sub020.p.prf	20	725	2324522934	0.004
p1p3p2_hanoi_sub021.p.prf	21	803	6973568801	0.004
p1p3p2_hanoi_sub022.p.prf	22	885	20920706406	0.004
p1p3p2_hanoi_sub023.p.prf	23	971	62762119217	0.005
p1p3p2_hanoi_sub024.p.prf	24	1061	188286357654	0.005
p1p3p2_hanoi_sub025.p.prf	25	1155	564859072961	0.005
p1p3p2_hanoi_sub026.p.prf	26	1253	1694577218886	0.005
p1p3p2_hanoi_sub027.p.prf	27	1355	5083731656657	0.006
p1p3p2_hanoi_sub028.p.prf	28	1461	15251194969974	0.006
p1p3p2_hanoi_sub029.p.prf	29	1571	45753584909921	0.006
p1p3p2_hanoi_sub030.p.prf	30	1685	137260754729766	0.007

Table 20. Results on Tower of Hanoi, recursive encoding, $p_1 > p_3 > p_2$

	Instance size	Rewrite steps	Action equivalents	Prover time
p2p1p3_hanoi_sub001.p.prf	1	1	1	0.005
p2p1p3_hanoi_sub002.p.prf	2	4	4	0.003
p2p1p3_hanoi_sub003.p.prf	3	8	13	0.002
p2p1p3_hanoi_sub004.p.prf	4	14	38	0.002
p2p1p3_hanoi_sub005.p.prf	5	22	111	0.002
p2p1p3_hanoi_sub006.p.prf	6	32	328	0.002
p2p1p3_hanoi_sub007.p.prf	7	44	977	0.002
p2p1p3_hanoi_sub008.p.prf	8	58	2922	0.002
p2p1p3_hanoi_sub009.p.prf	9	74	8755	0.002
p2p1p3_hanoi_sub010.p.prf	10	92	26252	0.002
p2p1p3_hanoi_sub011.p.prf	11	112	78741	0.003
p2p1p3_hanoi_sub012.p.prf	12	134	236206	0.003
p2p1p3_hanoi_sub013.p.prf	13	158	708599	0.003
p2p1p3_hanoi_sub014.p.prf	14	184	2125776	0.003
p2p1p3_hanoi_sub015.p.prf	15	212	6377305	0.003
p2p1p3_hanoi_sub016.p.prf	16	242	19131890	0.004
p2p1p3_hanoi_sub017.p.prf	17	274	57395643	0.004
p2p1p3_hanoi_sub018.p.prf	18	308	172186900	0.004
p2p1p3_hanoi_sub019.p.prf	19	344	516560669	0.004
p2p1p3_hanoi_sub020.p.prf	20	382	1549681974	0.005
p2p1p3_hanoi_sub021.p.prf	21	422	4649045887	0.005
p2p1p3_hanoi_sub022.p.prf	22	464	13947137624	0.005
p2p1p3_hanoi_sub023.p.prf	23	508	41841412833	0.006
p2p1p3_hanoi_sub024.p.prf	24	554	125524238458	0.006
p2p1p3_hanoi_sub025.p.prf	25	602	376572715331	0.006
p2p1p3_hanoi_sub026.p.prf	26	652	1129718145948	0.007
p2p1p3_hanoi_sub027.p.prf	27	704	3389154437797	0.007
p2p1p3_hanoi_sub028.p.prf	28	758	10167463313342	0.008
p2p1p3_hanoi_sub029.p.prf	29	814	30502389939975	0.008
p2p1p3_hanoi_sub030.p.prf	30	872	91507169819872	0.008

Table 21. Results on Tower of Hanoi, recursive encoding, $p_2 > p_1 > p_3$

	Instance size	Rewrite steps	Action equivalents	Prover time
p2p3p1_hanoi_sub001.p.prf	1	1	1	0.004
p2p3p1_hanoi_sub002.p.prf	2	4	4	0.003
p2p3p1_hanoi_sub003.p.prf	3	9	15	0.002
p2p3p1_hanoi_sub004.p.prf	4	16	50	0.002
p2p3p1_hanoi_sub005.p.prf	5	25	157	0.002
p2p3p1_hanoi_sub006.p.prf	6	36	480	0.002
p2p3p1_hanoi_sub007.p.prf	7	49	1451	0.002
p2p3p1_hanoi_sub008.p.prf	8	64	4366	0.002
p2p3p1_hanoi_sub009.p.prf	9	81	13113	0.002
p2p3p1_hanoi_sub010.p.prf	10	100	39356	0.002
p2p3p1_hanoi_sub011.p.prf	11	121	118087	0.002
p2p3p1_hanoi_sub012.p.prf	12	144	354282	0.003
p2p3p1_hanoi_sub013.p.prf	13	169	1062869	0.003
p2p3p1_hanoi_sub014.p.prf	14	196	3188632	0.003
p2p3p1_hanoi_sub015.p.prf	15	225	9565923	0.003
p2p3p1_hanoi_sub016.p.prf	16	256	28697798	0.003
p2p3p1_hanoi_sub017.p.prf	17	289	86093425	0.003
p2p3p1_hanoi_sub018.p.prf	18	324	258280308	0.003
p2p3p1_hanoi_sub019.p.prf	19	361	774840959	0.004
p2p3p1_hanoi_sub020.p.prf	20	400	2324522914	0.004
p2p3p1_hanoi_sub021.p.prf	21	441	6973568781	0.004
p2p3p1_hanoi_sub022.p.prf	22	484	20920706384	0.004
p2p3p1_hanoi_sub023.p.prf	23	529	62762119195	0.005
p2p3p1_hanoi_sub024.p.prf	24	576	188286357630	0.005
p2p3p1_hanoi_sub025.p.prf	25	625	564859072937	0.005
p2p3p1_hanoi_sub026.p.prf	26	676	1694577218860	0.005
p2p3p1_hanoi_sub027.p.prf	27	729	5083731656631	0.005
p2p3p1_hanoi_sub028.p.prf	28	784	15251194969946	0.006
p2p3p1_hanoi_sub029.p.prf	29	841	45753584909893	0.006
p2p3p1_hanoi_sub030.p.prf	30	900	137260754729736	0.006

Table 22. Results on Tower of Hanoi, recursive encoding, $p_2 > p_3 > p_1$

	Instance size	Rewrite steps	Action equivalents	Prover time
p3p1p2_hanoi_sub001.p.prf	1	1	1	0.004
p3p1p2_hanoi_sub002.p.prf	2	5	6	0.003
p3p1p2_hanoi_sub003.p.prf	3	11	17	0.002
p3p1p2_hanoi_sub004.p.prf	4	21	54	0.002
p3p1p2_hanoi_sub005.p.prf	5	35	161	0.002
p3p1p2_hanoi_sub006.p.prf	6	53	486	0.002
p3p1p2_hanoi_sub007.p.prf	7	75	1457	0.002
p3p1p2_hanoi_sub008.p.prf	8	101	4374	0.002
p3p1p2_hanoi_sub009.p.prf	9	131	13121	0.002
p3p1p2_hanoi_sub010.p.prf	10	165	39366	0.002
p3p1p2_hanoi_sub011.p.prf	11	203	118097	0.002
p3p1p2_hanoi_sub012.p.prf	12	245	354294	0.003
p3p1p2_hanoi_sub013.p.prf	13	291	1062881	0.003
p3p1p2_hanoi_sub014.p.prf	14	341	3188646	0.003
p3p1p2_hanoi_sub015.p.prf	15	395	9565937	0.003
p3p1p2_hanoi_sub016.p.prf	16	453	28697814	0.003
p3p1p2_hanoi_sub017.p.prf	17	515	86093441	0.003
p3p1p2_hanoi_sub018.p.prf	18	581	258280326	0.003
p3p1p2_hanoi_sub019.p.prf	19	651	774840977	0.004
p3p1p2_hanoi_sub020.p.prf	20	725	2324522934	0.004
p3p1p2_hanoi_sub021.p.prf	21	803	6973568801	0.004
p3p1p2_hanoi_sub022.p.prf	22	885	20920706406	0.004
p3p1p2_hanoi_sub023.p.prf	23	971	62762119217	0.005
p3p1p2_hanoi_sub024.p.prf	24	1061	188286357654	0.005
p3p1p2_hanoi_sub025.p.prf	25	1155	564859072961	0.005
p3p1p2_hanoi_sub026.p.prf	26	1253	1694577218886	0.005
p3p1p2_hanoi_sub027.p.prf	27	1355	5083731656657	0.005
p3p1p2_hanoi_sub028.p.prf	28	1461	15251194969974	0.006
p3p1p2_hanoi_sub029.p.prf	29	1571	45753584909921	0.006
p3p1p2_hanoi_sub030.p.prf	30	1685	137260754729766	0.006

Table 23. Results on Tower of Hanoi, recursive encoding, $p_3 > p_1 > p_2$

	Instance size	Rewrite steps	Action equivalents	Prover time
p3p2p1_hanoi_sub001.p.prf	1	1	1	0.003
p3p2p1_hanoi_sub002.p.prf	2	4	4	0.004
p3p2p1_hanoi_sub003.p.prf	3	9	15	0.002
p3p2p1_hanoi_sub004.p.prf	4	16	50	0.002
p3p2p1_hanoi_sub005.p.prf	5	25	157	0.002
p3p2p1_hanoi_sub006.p.prf	6	36	480	0.002
p3p2p1_hanoi_sub007.p.prf	7	49	1451	0.002
p3p2p1_hanoi_sub008.p.prf	8	64	4366	0.002
p3p2p1_hanoi_sub009.p.prf	9	81	13113	0.002
p3p2p1_hanoi_sub010.p.prf	10	100	39356	0.002
p3p2p1_hanoi_sub011.p.prf	11	121	118087	0.002
p3p2p1_hanoi_sub012.p.prf	12	144	354282	0.003
p3p2p1_hanoi_sub013.p.prf	13	169	1062869	0.003
p3p2p1_hanoi_sub014.p.prf	14	196	3188632	0.003
p3p2p1_hanoi_sub015.p.prf	15	225	9565923	0.003
p3p2p1_hanoi_sub016.p.prf	16	256	28697798	0.003
p3p2p1_hanoi_sub017.p.prf	17	289	86093425	0.003
p3p2p1_hanoi_sub018.p.prf	18	324	258280308	0.003
p3p2p1_hanoi_sub019.p.prf	19	361	774840959	0.004
p3p2p1_hanoi_sub020.p.prf	20	400	2324522914	0.004
p3p2p1_hanoi_sub021.p.prf	21	441	6973568781	0.004
p3p2p1_hanoi_sub022.p.prf	22	484	20920706384	0.004
p3p2p1_hanoi_sub023.p.prf	23	529	62762119195	0.005
p3p2p1_hanoi_sub024.p.prf	24	576	188286357630	0.005
p3p2p1_hanoi_sub025.p.prf	25	625	564859072937	0.005
p3p2p1_hanoi_sub026.p.prf	26	676	1694577218860	0.005
p3p2p1_hanoi_sub027.p.prf	27	729	5083731656631	0.005
p3p2p1_hanoi_sub028.p.prf	28	784	15251194969946	0.006
p3p2p1_hanoi_sub029.p.prf	29	841	45753584909893	0.006
p3p2p1_hanoi_sub030.p.prf	30	900	137260754729736	0.006

Table 24. Results on Tower of Hanoi, recursive encoding, $p_3 > p_2 > p_1$