

Solver setups

Name	Solver	Flags	Input
ACL2[-g]	ACL2	--time-limit 5 g: --generalize	SMT-LIB
CVC4[-Gen]	CVC4 (1.7)	--lang=smt2 --quant-ind --tlimit=5000 Gen: --conjecture-gen	SMT-LIB
CVC4-Gen-1.8	CVC4 (1.8)	--lang=smt2 --quant-ind --tlimit=5000 --conjecture-gen	
Imandra	Imandra	default mode with 5 second server timeout	functional program encoding
Vampire[-gcx]	Vampire	-ind struct -t 5 --input.syntax smtlib g: -indgen on c: -indoct on x: -to lpo -drc off	SMT-LIB
Zeno	Zeno	-t 5 --no-isa	functional program encoding
ZipperPosition	ZipperPosition	-t 5	.zf (native input format)
ZipRewrite		-t 5	.zf with definitions as rewrite rules

Benchmarks

benchmark set	example	count	ACL2	ACL2-g	CVC4	CVC4-Gen	CVC4-Gen-1.8	Imandra	Vampire	Vampire-c	Vampire-cx	Vampire-g	Vampire-gc	Vampire-gcx	Vampire-gx	Vampire-x	Zeno	ZipRewrite	ZipperPosition
combined/combined_nat_list	$\forall n, x. (cons(n + s(n), x) \text{ ++ } (x \text{ ++ } x) = (cons(s(n) + n, x) \text{ ++ } x) \text{ ++ } x)$	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
list/concat_assoc_1var_10occ	$\forall v_0. (((v_0 \text{ ++ } ((v_0 \text{ ++ } v_0) \text{ ++ } v_0)) \text{ ++ } (v_0 \text{ ++ } (((v_0 \text{ ++ } v_0) \text{ ++ } (v_0 \text{ ++ } v_0)) \text{ ++ } v_0)) = ((v_0 \text{ ++ } v_0) \text{ ++ } (((v_0 \text{ ++ } v_0) \text{ ++ } (v_0 \text{ ++ } (v_0 \text{ ++ } v_0))) \text{ ++ } (v_0 \text{ ++ } v_0))) \text{ ++ } v_0)$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	36	0
list/concat_assoc_1var_3occ	$\forall v_0. (v_0 \text{ ++ } (v_0 \text{ ++ } v_0) = (v_0 \text{ ++ } v_0) \text{ ++ } v_0)$	1	0	1	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0
list/concat_assoc_1var_4occ	$\forall v_0. (((v_0 \text{ ++ } v_0) \text{ ++ } v_0) \text{ ++ } v_0 = v_0 \text{ ++ } ((v_0 \text{ ++ } v_0) \text{ ++ } v_0))$	10	0	0	0	0	0	0	0	1	8	9	10	10	0	0	0	10	0
list/concat_assoc_1var_5occ	$\forall v_0. ((v_0 \text{ ++ } v_0) \text{ ++ } (v_0 \text{ ++ } (v_0 \text{ ++ } v_0)) = (v_0 \text{ ++ } ((v_0 \text{ ++ } v_0) \text{ ++ } v_0)) \text{ ++ } v_0)$	50	1	0	0	0	0	0	0	3	22	27	38	12	0	0	0	50	0
list/concat_assoc_1var_6occ	$\forall v_0. ((v_0 \text{ ++ } v_0) \text{ ++ } (v_0 \text{ ++ } ((v_0 \text{ ++ } v_0) \text{ ++ } v_0)) = v_0 \text{ ++ } (v_0 \text{ ++ } (((v_0 \text{ ++ } v_0) \text{ ++ } v_0) \text{ ++ } v_0)))$	50	0	0	0	0	0	0	0	1	3	13	17	6	0	1	0	50	0
list/concat_assoc_1var_7occ	$\forall v_0. ((v_0 \text{ ++ } v_0) \text{ ++ } ((v_0 \text{ ++ } ((v_0 \text{ ++ } v_0) \text{ ++ } v_0)) \text{ ++ } v_0) = (v_0 \text{ ++ } v_0) \text{ ++ } (((v_0 \text{ ++ } v_0) \text{ ++ } v_0) \text{ ++ } (v_0 \text{ ++ } v_0)))$	50	0	0	0	0	0	0	0	2	0	5	12	1	0	0	0	50	0
list/concat_assoc_1var_8occ	$\forall v_0. (v_0 \text{ ++ } (((v_0 \text{ ++ } v_0) \text{ ++ } ((v_0 \text{ ++ } v_0) \text{ ++ } v_0)) \text{ ++ } v_0) \text{ ++ } v_0 = (v_0 \text{ ++ } v_0) \text{ ++ } ((v_0 \text{ ++ } (v_0 \text{ ++ } (v_0 \text{ ++ } v_0))) \text{ ++ } (v_0 \text{ ++ } v_0)))$	50	0	0	0	0	0	0	0	0	1	1	2	1	0	0	0	47	0
list/concat_assoc_1var_9occ	$\forall v_0. (((v_0 \text{ ++ } v_0) \text{ ++ } (v_0 \text{ ++ } v_0)) \text{ ++ } (v_0 \text{ ++ } (v_0 \text{ ++ } (v_0 \text{ ++ } v_0)))) = (v_0 \text{ ++ } (v_0 \text{ ++ } v_0)) \text{ ++ } ((v_0 \text{ ++ } ((v_0 \text{ ++ } v_0) \text{ ++ } v_0)) \text{ ++ } (v_0 \text{ ++ } v_0)))$	50	0	0	0	0	0	0	0	1	1	1	2	1	0	0	0	47	0
list/concat_assoc_2var_4occ	$\forall x, y. (x \text{ ++ } (y \text{ ++ } (x \text{ ++ } x)) = (x \text{ ++ } y) \text{ ++ } (x \text{ ++ } x))$	1	0	1	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0
list/concat_assoc_3var_3occ	$\forall x, y, z. (x \text{ ++ } (y \text{ ++ } z) = (x \text{ ++ } y) \text{ ++ } z)$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
list/pref_1var_1.2occ	$\forall v_0. pref(v_0, v_0 \text{ ++ } v_0)$	1	0	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0

benchmark set	example	count	ACL2	ACL2-g	CVC4	CVC4-Gen	CVC4-Gen-1.8	Imandra	Vampire	Vampire.c	Vampire.cx	Vampire.g	Vampire.gc	Vampire.gcx	Vampire.gx	Vampire.x	Zeno	ZipRewrite	ZipperPosition
list/pref_1var_1.3occ	$\forall v_0. \text{pref}(v_0, (v_0 ++ v_0) ++ v_0)$	2	0	2	0	0	0	0	0	0	1	2	2	2	2	0	1	2	0
list/pref_1var_1.4occ	$\forall v_0. \text{pref}(v_0, v_0 ++ (v_0 ++ (v_0 ++ v_0)))$	5	0	5	0	0	0	0	0	0	2	5	5	5	5	0	2	5	0
list/pref_1var_1.5occ	$\forall v_0. \text{pref}(v_0, v_0 ++ ((v_0 ++ v_0) ++ (v_0 ++ v_0)))$	14	0	14	0	0	0	0	0	0	6	14	14	14	14	0	6	14	0
list/pref_1var_2.3occ	$\forall v_0. \text{pref}(v_0 ++ v_0, v_0 ++ (v_0 ++ v_0))$	2	0	2	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0
list/pref_1var_2.4occ	$\forall v_0. \text{pref}(v_0 ++ v_0, v_0 ++ ((v_0 ++ v_0) ++ v_0))$	5	0	5	0	0	0	0	0	0	1	0	2	2	0	0	1	5	0
list/pref_1var_2.5occ	$\forall v_0. \text{pref}(v_0 ++ v_0, (v_0 ++ (v_0 ++ (v_0 ++ v_0))) ++ v_0)$	14	0	14	0	0	0	0	0	0	3	0	5	5	0	0	3	14	0
list/pref_1var_2.6occ	$\forall v_0. \text{pref}(v_0 ++ v_0, (v_0 ++ ((v_0 ++ v_0) ++ v_0)) ++ (v_0 ++ v_0))$	42	0	42	0	0	0	0	0	0	9	0	15	14	0	0	9	42	0
list/pref_1var_3.3occ	$\forall v_0. \text{pref}((v_0 ++ v_0) ++ v_0, v_0 ++ (v_0 ++ v_0))$	2	0	0	0	0	0	0	0	0	0	2	1	1	2	0	0	2	0
list/pref_1var_3.4occ	$\forall v_0. \text{pref}((v_0 ++ v_0) ++ v_0, v_0 ++ ((v_0 ++ v_0) ++ v_0))$	10	0	0	0	0	0	0	0	0	2	0	2	2	0	0	2	0	0
list/pref_1var_3.5occ	$\forall v_0. \text{pref}(v_0 ++ (v_0 ++ v_0), (v_0 ++ ((v_0 ++ v_0) ++ v_0)) ++ v_0)$	28	0	0	0	0	0	0	0	0	4	0	4	4	0	0	4	28	0
list/pref_1var_3.6occ	$\forall v_0. \text{pref}(v_0 ++ (v_0 ++ v_0), (v_0 ++ v_0) ++ (((v_0 ++ v_0) ++ v_0) ++ v_0))$	50	0	0	0	0	0	0	0	0	4	0	7	6	0	0	6	50	0
list/pref_1var_3.7occ	$\forall v_0. \text{pref}(v_0 ++ (v_0 ++ v_0), ((v_0 ++ v_0) ++ v_0) ++ (v_0 ++ ((v_0 ++ v_0) ++ v_0)))$	50	0	0	0	0	0	0	0	0	6	0	8	8	0	0	8	50	0
list/pref_1var_4.4occ	$\forall v_0. \text{pref}(v_0 ++ (v_0 ++ (v_0 ++ v_0)), ((v_0 ++ v_0) ++ v_0) ++ v_0)$	20	0	0	0	0	0	0	0	0	0	0	5	6	0	0	0	20	0
list/pref_1var_4.5occ	$\forall v_0. \text{pref}(v_0 ++ (v_0 ++ (v_0 ++ v_0)), v_0 ++ (((v_0 ++ v_0) ++ v_0) ++ v_0))$	50	0	0	0	0	0	0	0	0	5	0	7	6	0	0	5	0	0
list/pref_1var_4.6occ	$\forall v_0. \text{pref}((v_0 ++ v_0) ++ (v_0 ++ v_0), (((v_0 ++ v_0) ++ v_0) ++ v_0) ++ (v_0 ++ v_0))$	50	0	0	0	0	0	0	0	0	3	0	3	3	0	0	4	50	0
list/pref_1var_4.7occ	$\forall v_0. \text{pref}(((v_0 ++ v_0) ++ v_0) ++ v_0, ((v_0 ++ (v_0 ++ (v_0 ++ v_0))) ++ (v_0 ++ v_0)) ++ v_0)$	50	0	0	0	0	0	0	0	0	2	0	2	2	0	0	2	50	0
list/pref_1var_4.8occ	$\forall v_0. \text{pref}(v_0 ++ ((v_0 ++ v_0) ++ v_0), (v_0 ++ v_0) ++ ((v_0 ++ (v_0 ++ (v_0 ++ v_0))) ++ (v_0 ++ v_0)))$	50	0	0	0	0	0	0	0	0	2	0	2	2	0	0	2	50	0
list/pref_1var_5.5occ	$\forall v_0. \text{pref}((v_0 ++ v_0) ++ (v_0 ++ (v_0 ++ v_0)), (v_0 ++ (v_0 ++ v_0)) ++ (v_0 ++ v_0))$	50	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	50	0
list/pref_1var_5.6occ	$\forall v_0. \text{pref}(((v_0 ++ v_0) ++ v_0) ++ (v_0 ++ v_0), ((v_0 ++ v_0) ++ v_0) ++ ((v_0 ++ v_0) ++ v_0))$	50	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
list/pref_1var_5.7occ	$\forall v_0. \text{pref}(v_0 ++ (((v_0 ++ v_0) ++ v_0) ++ v_0), ((v_0 ++ v_0) ++ (v_0 ++ (v_0 ++ (v_0 ++ v_0)))) ++ v_0)$	50	0	0	0	0	0	0	0	0	1	0	1	1	0	0	2	50	0
list/pref_1var_5.8occ	$\forall v_0. \text{pref}(((v_0 ++ v_0) ++ v_0) ++ (v_0 ++ v_0), ((v_0 ++ v_0) ++ (v_0 ++ (v_0 ++ v_0))) ++ ((v_0 ++ v_0) ++ v_0))$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	50	0
list/pref_1var_5.9occ	$\forall v_0. \text{pref}((v_0 ++ v_0) ++ ((v_0 ++ v_0) ++ v_0), (((v_0 ++ v_0) ++ v_0) ++ (v_0 ++ v_0)) ++ (v_0 ++ v_0)) ++ (v_0 ++ v_0))$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	50	0
list/pref_2var_1.2occ	$\forall x, y. \text{pref}(x, x ++ y)$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
list/reverse_expressions	$\forall x. (x ++ (\text{rev}(x) ++ x) = (x ++ \text{rev}(x)) ++ x)$	4	0	0	0	0	0	1	0	0	1	2	2	2	2	0	0	2	0
nat/add_assoc_1var_10occ	$\forall v_0. (v_0 + ((v_0 + v_0) + ((v_0 + v_0) + (((v_0 + v_0) + v_0) + (v_0 + v_0)))) = ((v_0 + (v_0 + v_0)) + v_0) + (((v_0 + v_0) + (v_0 + v_0)) + (v_0 + v_0)))$	50	0	0	0	50	0	0	0	0	3	0	0	1	0	0	2	43	0
nat/add_assoc_1var_3occ	$\forall v_0. (v_0 + (v_0 + v_0) = (v_0 + v_0) + v_0)$	1	0	1	0	1	0	0	0	0	1	1	1	1	1	0	1	1	0
nat/add_assoc_1var_4occ	$\forall v_0. ((v_0 + (v_0 + v_0)) + v_0 = v_0 + ((v_0 + v_0) + v_0))$	10	0	0	0	10	0	0	0	0	4	8	10	10	10	0	2	10	0
nat/add_assoc_1var_5occ	$\forall v_0. (((v_0 + v_0) + v_0) + v_0 + v_0 = ((v_0 + v_0) + (v_0 + v_0)) + v_0)$	50	1	0	0	50	0	0	0	0	12	13	22	41	17	0	6	50	0
nat/add_assoc_1var_6occ	$\forall v_0. (((v_0 + (v_0 + v_0)) + v_0) + (v_0 + v_0) = v_0 + ((v_0 + (v_0 + v_0)) + (v_0 + v_0)))$	50	3	0	0	50	0	0	0	0	12	2	14	31	9	0	11	50	0

benchmark set	example	count	ACL2	ACL2.g	CVC4	CVC4-Gen	CVC4-Gen-1.8	Imandra	Vampire	Vampire.c	Vampire.cx	Vampire.g	Vampire.gc	Vampire.gcx	Vampire.gx	Vampire.x	Zeno	ZipRewrite	ZipperPosition
nat/add_assoc_1var_7occ	$\forall v_0.(((v_0 + v_0) + v_0) + (v_0 + (v_0 + v_0))) + v_0 = ((v_0 + v_0) + (v_0 + (v_0 + v_0))) + (v_0 + v_0)$	50	0	0	0	50	0	0	0	0	6	1	5	12	1	0	2	50	0
nat/add_assoc_1var_8occ	$\forall v_0.((v_0 + v_0) + (((v_0 + (v_0 + v_0)) + (v_0 + v_0)) + v_0) = ((v_0 + ((v_0 + v_0) + v_0)) + v_0) + (v_0 + (v_0 + v_0)))$	50	0	0	0	50	0	0	0	0	2	0	0	3	0	0	1	50	0
nat/add_assoc_1var_9occ	$\forall v_0.((v_0 + (v_0 + v_0)) + ((v_0 + ((v_0 + (v_0 + v_0)) + v_0)) + v_0) = v_0 + (((v_0 + (v_0 + v_0)) + ((v_0 + v_0) + (v_0 + v_0))) + v_0))$	50	0	0	0	50	0	0	0	0	1	0	0	0	0	0	0	47	0
nat/add_assoc_3var	$\forall x, y, z. (x + (y + z) = (x + y) + z)$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nat/add_comm	$\forall x, y. (x + y = y + x)$	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nat/add_comm_with_id	$\forall x, y. (id(x) + y = y + x)$	1	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
nat/add_mix_2var_3occ	$\forall v_0, v_1. ((v_1 + v_1) + v_0 = v_0 + (v_1 + v_1))$	30	2	6	2	5	2	18	2	2	17	20	23	30	30	2	12	30	3
nat/add_mix_2var_4occ	$\forall v_0, v_1. ((v_1 + v_0) + (v_1 + v_0) = (v_1 + (v_1 + v_0)) + v_0)$	50	2	5	0	4	0	11	0	0	19	6	14	44	29	0	17	36	3
nat/add_mix_2var_5occ	$\forall v_0, v_1. ((v_0 + v_1) + (v_0 + (v_1 + v_1)) = v_0 + ((v_1 + v_1) + (v_1 + v_0)))$	50	1	2	1	7	1	12	1	1	4	4	5	15	9	1	6	21	1
nat/add_mix_2var_6occ	$\forall v_0, v_1. (v_1 + (((v_0 + v_0) + v_1) + v_1) + v_1) = (((v_1 + v_0) + v_1) + (v_0 + v_1)) + v_1$	50	0	1	0	1	0	0	0	0	4	2	3	3	2	0	3	7	0
nat/add_mix_2var_7occ	$\forall v_0, v_1. ((v_0 + (v_1 + (v_1 + v_0))) + (v_1 + (v_1 + v_0))) = v_1 + (((v_1 + (v_0 + v_1)) + v_0) + (v_1 + v_0)))$	50	0	0	0	4	0	2	0	0	1	1	1	2	0	0	2	2	1
nat/add_mix_3var_3occ	$\forall v_0, v_1, v_2. ((v_1 + v_0) + v_2 = (v_2 + v_1) + v_0)$	50	6	4	35	38	36	47	29	27	48	24	27	45	50	50	36	50	44
nat/add_mix_3var_4occ	$\forall v_0, v_1, v_2. ((v_1 + v_0) + (v_1 + v_2) = (v_1 + (v_0 + v_1)) + v_2)$	50	2	0	0	2	0	12	1	1	13	5	8	35	28	1	9	31	12
nat/add_mix_3var_5occ	$\forall v_0, v_1, v_2. ((v_1 + v_2) + ((v_1 + v_1) + v_0) = v_0 + (v_1 + ((v_2 + v_1) + v_1)))$	50	2	0	0	1	0	2	0	0	7	2	2	16	3	0	7	13	0
nat/add_mix_3var_6occ	$\forall v_0, v_1, v_2. ((v_2 + v_0) + ((v_2 + v_1) + (v_1 + v_0))) = (v_1 + (v_2 + ((v_2 + v_1) + v_0))) + v_0$	50	0	0	0	1	0	2	0	0	1	0	1	3	0	0	0	1	1
nat/add_mix_3var_7occ	$\forall v_0, v_1, v_2. ((v_0 + (v_2 + v_0)) + (v_2 + ((v_1 + v_0) + v_2))) = ((v_1 + ((v_2 + v_2) + v_0)) + v_0) + (v_0 + v_2)$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
nat/add_mix_4var_4occ	$\forall v_0, v_1, v_2, v_3. ((v_2 + (v_3 + v_1)) + v_0 = (v_1 + (v_3 + v_2)) + v_0)$	50	6	1	11	21	13	20	8	7	31	6	9	26	29	34	24	14	11
nat/add_mix_4var_5occ	$\forall v_0, v_1, v_2, v_3. ((v_1 + v_0) + (v_2 + (v_2 + v_3))) = (v_2 + ((v_1 + v_0) + v_3)) + v_2$	50	1	0	1	1	1	3	0	0	3	0	3	6	4	1	4	4	1
nat/add_mix_4var_6occ	$\forall v_0, v_1, v_2, v_3. (v_2 + (((v_0 + v_1) + v_3) + v_2) + v_2) = (v_2 + (v_3 + ((v_0 + v_1) + v_2))) + v_2$	50	0	0	0	0	0	0	0	0	2	0	0	1	0	0	1	3	0
nat/add_mix_4var_7occ	$\forall v_0, v_1, v_2, v_3. (((v_1 + v_0) + v_0) + (v_2 + (v_3 + v_3))) + v_1 = v_0 + ((v_1 + v_1) + (v_3 + (v_3 + (v_0 + v_2))))$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
nat/add_mix_5var_5occ	$\forall v_0, v_1, v_2, v_3, v_4. ((v_0 + v_2) + ((v_1 + v_3) + v_4) = v_0 + (v_4 + ((v_1 + v_3) + v_2)))$	50	1	0	1	1	1	5	0	0	4	0	1	3	1	8	11	6	2
nat/add_mix_5var_6occ	$\forall v_0, v_1, v_2, v_3, v_4. ((v_3 + (v_2 + (v_4 + v_2))) + (v_1 + v_0) = v_1 + (v_3 + ((v_2 + (v_4 + v_0)) + v_2)))$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0
nat/add_mix_5var_7occ	$\forall v_0, v_1, v_2, v_3, v_4. ((v_1 + v_0) + ((v_1 + (v_0 + v_3)) + (v_2 + v_4))) = (v_3 + v_0) + ((v_1 + ((v_4 + v_0) + v_1)) + v_2)$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nat/add_s_1var_2occ	$\forall x. (s(x) + x = s(x + x))$	2	1	2	1	2	1	1	1	1	2	2	2	2	2	1	1	2	1
nat/add_s_zero_mix_12	$\forall v_0, v_1. (s(s(v_0) + (s(s(s(v_1)))) + ((v_1 + s(v_1)) + s(v_0)))) = s(v_0) + ((v_0 + s(s(s(v_1 + s(v_1)))) + s(s(v_1))))$	50	0	0	2	6	2	11	3	2	15	4	3	20	20	20	4	12	2

benchmark set	example	count	ACL2	ACL2-g	CVC4	CVC4-Gen	CVC4-Gen-1.8	Inandra	Vampire	Vampire.c	Vampire.cx	Vampire.g	Vampire.gc	Vampire.gcx	Vampire.gx	Vampire.x	Zeno	ZipRewrite	ZipperPosition
nat/add_s_zero_mix_15	$\forall v_0, v_1, v_2. (s(s(s(s(s(v_1)) + (s(s(v_2) + s(v_0)) + s(v_2)))))) + s(v_0) = s(((s(s(s(v_2)))) + s(v_1)) + s(v_0)) + s(v_0 + s(s(s(v_2))))))$	50	0	0	1	1	0	4	1	1	2	1	1	3	5	7	0	3	1
nat/add_s_zero_mix_18	$\forall v_0, v_1, v_2, v_3, v_4. (s(s(((v_3 + s(s(v_1) + v_0)) + (s(s(v_4)) + s(s(s(v_2 + s(v_4)))))) + s(v_2))) = s(v_0) + (s(v_4) + s(s(s(s(s(s((v_4 + v_3) + s(v_1)) + (s(v_2) + v_2))))))))))$	50	0	0	0	0	0	1	0	0	1	0	0	1	2	5	1	0	0
nat/add_s_zero_mix_21	$\forall v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8. (s(s(s(((s(v_8) + v_0) + s(s((s(v_1) + s(s(v_4))) + v_3))) + s(s(v_2) + (v_7 + (v_6 + s(v_5)))))) = v_8 + (((s(v_1) + (v_7 + s(s(s(v_6) + s(v_0)))))) + (v_4 + s(v_3))) + s(v_5 + s(s(s(v_2))))))$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nat/add_s_zero_mix_24	$\forall v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7. (((s(s(zero)) + zero) + s(s(s(v_3)))) + (s(s(((v_2 + s(v_7)) + s(v_5)) + (s(s(v_6)) + s(v_1))) + s(v_4))) + s(v_0)) = ((v_3 + s(v_4)) + s(s(((s(s(v_2)) + s(v_1)) + (s(s(s(v_0 + s(zero)))) + zero)) + (s(v_7) + v_6)))) + s(s(s(v_5))))$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nat/add_s_zero_mix_27	$\forall v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9. (s(s(s(s(s(s(s(s(s(s(s(s(s(s(v_0) + ((v_8 + s(v_5)) + v_3)))))) + (s(v_2) + (s(v_7) + v_6)))))) + (s((v_9 + s(v_1)) + s(v_0)) + v_4) = s((s(s(s(s(v_2)) + v_8)) + v_4) + s(v_0)) + (s(s(v_1)) + (s(v_9) + s(s(s(v_5 + (s(v_2) + v_7))) + s(v_6 + s(v_3))))))$	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nat/add_s_zero_mix_3	$\forall v_0, v_1. (v_0 + s(v_1) = v_1 + s(v_0))$	34	14	18	32	34	31	29	32	32	34	34	34	34	34	32	30	34	32
nat/add_s_zero_mix_30	$\forall v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}. (s(((s(s(v_5) + s(v_9)) + v_7) + v_2) + ((s(zero) + v_{10}) + s(v_4))) + (s(s(v_6)) + s(v_0))) + s(s(s(s(s(v_{11}) + v_1))) + s(s(v_{11}) + (v_8 + s(v_3)))) = s(s(s(s(v_3 + (v_{10} + (((s(s(v_1)) + s(s(v_6 + s(s(s(s(s(v_2))) + (v_8 + v_4)))) + v_{11})) + v_0) + s(s(zero) + s(s(v_7 + v_9)))))) + v_5) + v_{11})$	35+50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nat/add_s_zero_mix_6	$\forall v_0. (s(s(v_0) + s(s(zero))) = s(s(s(s(zero) + v_0))))$	50	7	8	29	33	26	37	32	33	44	33	32	50	49	37	25	47	34
nat/add_s_zero_mix_9	$\forall v_0, v_1, v_2, v_3. (s(v_2 + s(s(s(v_3 + s(v_0)))) + v_1)) = s(s(s(v_0) + v_1)) + (v_2 + s(s(v_3)))$	50	7	4	17	18	10	30	12	14	31	10	8	37	40	31	13	30	12
nat/equal	$\forall x, y, z. (equal(x, y, z) = and(x = y, y = z))$	4	1	0	2	4	2	2	1	1	1	2	2	2	2	1	0	4	2
nat/leq_lvar_1.2occ	$\forall v_0. (v_0 \leq v_0 + v_0)$	1	0	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
nat/leq_lvar_1.3occ	$\forall v_0. (v_0 \leq (v_0 + v_0) + v_0)$	2	0	2	0	0	0	0	0	0	1	2	2	2	2	0	2	2	0
nat/leq_lvar_1.4occ	$\forall v_0. (v_0 \leq v_0 + (v_0 + (v_0 + v_0)))$	5	0	5	0	0	0	0	0	0	3	5	5	5	5	0	4	5	0
nat/leq_lvar_1.5occ	$\forall v_0. (v_0 \leq v_0 + (v_0 + (v_0 + (v_0 + v_0))))$	14	0	14	0	0	0	0	0	0	7	14	14	14	14	0	12	14	0
nat/leq_lvar_2.3occ	$\forall v_0. (v_0 + v_0 \leq v_0 + (v_0 + v_0))$	2	0	2	0	0	0	0	0	0	1	2	2	1	1	0	2	0	0
nat/leq_lvar_2.4occ	$\forall v_0. (v_0 + v_0 \leq (v_0 + v_0) + (v_0 + v_0))$	5	0	5	0	0	0	0	0	0	2	1	2	2	0	0	2	5	0
nat/leq_lvar_2.5occ	$\forall v_0. (v_0 + v_0 \leq (v_0 + (v_0 + (v_0 + v_0))) + v_0)$	14	0	14	0	0	0	0	0	0	5	0	5	5	0	0	6	14	0
nat/leq_lvar_2.6occ	$\forall v_0. (v_0 + v_0 \leq v_0 + ((v_0 + v_0) + (v_0 + (v_0 + v_0))))$	42	0	42	0	0	0	0	0	0	13	0	16	14	0	0	11	42	0
nat/leq_lvar_3.3occ	$\forall v_0. ((v_0 + v_0) + v_0 \leq v_0 + (v_0 + v_0))$	2	0	2	0	2	0	0	0	0	0	2	2	2	2	0	2	2	0
nat/leq_lvar_3.4occ	$\forall v_0. (v_0 + (v_0 + v_0) \leq v_0 + (v_0 + (v_0 + v_0)))$	10	0	0	0	0	0	0	0	0	2	0	2	2	0	0	4	0	0
nat/leq_lvar_3.5occ	$\forall v_0. ((v_0 + v_0) + v_0 \leq v_0 + ((v_0 + (v_0 + v_0)) + v_0))$	28	0	0	0	0	0	0	0	0	4	0	5	4	0	0	8	28	0

[illegible]