1 Description

1.1 Python Programs

There are two programs, gen_microbenchmarks.py, which generates a single microbenchmark file, and runner.py, which generates several benchmark files, performs inference on them, and extracts the SMT solving time.

Use python3 gen_microbenchmarks.py --h to see the complete help menu and what all of the command line arguments mean,

```
-h, --help            show this help message and exit
--mult MULT           Number of multiplication constraints in each group
--mult-groups MULT_GROUPS
                      Number of groups of multiplications
--mult-end            Specify to annotate the end variable of the
                      multiplications in each group
--mult-perline        Specify to put one multiplication in each line and use
                      intermediate variables to store results
--mult-nocorrection Specify to disable using constants to ensure that the
                      number of variables equal the number of
                      multiplications
--mult-annot MULT_ANNOT
                      Percentage of starting variables for multiplication in
                      each group which should be annotated with some unit
--add ADD             Number of addition constraints in each group
--add-groups ADD_GROUPS
                      Number of groups of additions
--add-end             Specify to annotate the end variable of the additions
                      in each group
--add-perline         Specify to put one addition in each line and use
                      intermediate variables to store results
--add-nocorrection    Specify to disable using constants to ensure that the
                      number of variables equal the number of additions
--add-annot ADD_ANNOT
                      Percentage of starting variables for addition in each
                      group which should be annotated with some unit
--comp COMP           Number of comparison constraints in each group
--comp-groups COMP_GROUPS
                      Number of groups of comparison constraints
--comp-nocorrection Specify to disable using constants to ensure that the
                      number of variables equal the number of comparisons
--comp-annot COMP_ANNOT
                      Percentage of starting variables for comparison in
                      each group which should be annotated with some unit
```

In particular,

1. --mult-perline: There are two ways to multiply k variables together,

```
    // all in one line
    double result1 = starting1 * starting2 * ... * startingk;
    // one operation per line
```

```
double result1 = starting1 * starting2;
double result2 = result1 * starting3;
double result3 = result2 * starting4;
...
```

The default behavior is the first and this flag is used to get the second.

2. `--mult-nocorrection`: To get k multiplication operations, $k+1$ operands are needed. By default, k starting variables are created (starting variables are those which are initialized to 0 and later operated on) and the constant 1 is used as the last operand. This is to correct for the number of variables as the number of groups change, since the variables in each group are independent of the variables in other groups. For example, without correction, 10 groups of 10 multiplications each would result in $(10+1) \cdot 10 = 121$ variables whereas 1 group of 100 multiplications would result in 101 variables, even though both settings involve 100 multiplication operations. Use this flag to disable this correction and create $k+1$ starting variables for a group of k multiplications.

For example, to generate a file with 3 groups of 4 multiplications each, where there is one multiplication per line, no correction, and 75% of the starting variables are annotated, do: `python3 gen_microbenchmarks.py --mult 4 --mult-groups 4 --mult-perline --mult-nocorrection --mult-annot 75`.

2 Results

Averages across three replicates (takes around 1.5-2 hours to run in total) in milliseconds.

1. With correction, all operations in one line, not annotating the end result in each group, annotating 100% of starting variables.

		Number of Groups						
		1	5	10	15	20	25	30
Multiplications per group	1	41	37	55	51	60	69	66
	2	41	41	54	68	75	79	90
	3	42	60	63	77	91	96	112
	5	50	134	493	1402	2980	5446	14193
	7	47	207	828	2689	6308	17668	22678
	10	51	407	2224	6459	21559	38457	77021
	15	68	769	5096	23430	51603	183874	270886

		Number of Groups						
		1	5	10	15	20	25	30
Additions per group	1	37	49	77	113	188	307	525
	2	38	62	122	235	426	960	1304
	3	40	73	164	363	897	1296	2589
	5	45	192	388	1037	2069	6173	9123
	7	47	188	762	2045	5583	11689	19466
	10	54	387	1909	5426	13320	42211	63587
	15	56	854	5048	17302	74866	154856	247602

2. With correction, all operations in one line, annotating the end result in each group (arbitrarily for multiplication, which possibly causes inference failure, and correctly for addition), annotating 75% of starting variables.

		Number of Groups						
		1	5	10	15	20	25	30
Multiplications per group	1	31	50	67	80	125	163	241
	2	42	53	128	243	389	579	1134
	3	39	91	215	508	1027	1694	2651
	5	62	197	686	1759	3628	5605	9801
	7	56	296	1062	3283	6845	12501	20661
	10	71	447	2359	7095	14630	32354	58695
	15	76	942	5522	18594	49394	86678	174256

		Number of Groups						
		1	5	10	15	20	25	30
Additions per group	1	40	42	59	54	63	62	67
	2	45	44	56	70	78	82	95
	3	45	47	67	72	84	103	114
	5	37	56	89	102	140	138	165
	7	42	60	108	138	157	205	234
	10	52	84	127	225	236	247	297
	15	44	112	234	320	396	424	439

3. With correction, all operations in one line, annotating the end result in each group, annotating 100% of starting variables.

		Number of Groups						
		1	5	10	15	20	25	30
Multiplications per group	1	32	33	37	35	41	35	39
	2	39	31	35	28	33	36	44
	3	32	38	37	44	40	47	53
	5	36	36	47	50	51	61	63
	7	36	35	47	55	72	75	78
	10	34	52	63	80	96	96	112
	15	37	63	76	103	127	148	187

		Number of Groups						
		1	5	10	15	20	25	30
Additions per group	1	35	36	50	56	64	69	72
	2	36	44	63	62	64	83	96
	3	33	56	74	84	88	102	111
	5	32	55	84	100	125	139	164
	7	37	67	91	137	150	182	216
	10	39	86	135	167	205	263	319
	15	48	97	216	296	330	400	455