PuMoC: A CTL Model-Checker For Sequential Programs

Fu Song and Tayssir Touili

http://sist.shanghaitech.edu.cn/faculty/songfu/Projects/PuMoC

Motivation

- CTL model checking for sequential programs
- Existing software model-checking tools only support LTL and/or reachability properties
- It is important to have a software model-checker for CTL and CTL with predicates over the stack.
- Example, standard CTL property needed for Windows drivers:

$$\mathbf{AG}(\mathit{DeviceAdd} \Longrightarrow \mathbf{EF}\mathit{DeviceCreate})$$

- Example, CTL property with predicates over the stack needed for Windows drivers:

$$\mathbf{AG}(\mathit{DeviceAdd} \Longrightarrow \mathbf{EF}(\mathit{DeviceCreate} \wedge \text{before } \mathit{DeviceAdd} \text{ ret.}))$$

Our tool: PuMoC

Given a program and a CTL property, it checks whether the program satisfies the property. PuMoC can:

- perform Data Flow Analysis of Java Programs
- verify Pushdown Systems (PDS)
- verify Boolean Programs
- verify C and Java Programs

Approach

Sequential Program $\models$ CTL Property

[Esparza and Schwoon, CAV'01]
[Reps et al, SAS'03]

Pushdown Systems $\models$ CTL Property

Model Checking Engine

PDS $\models$ CTL

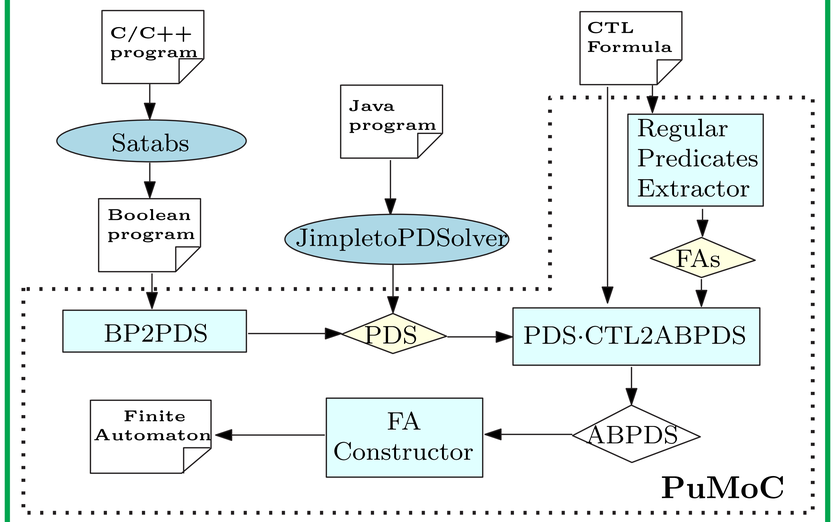
Emptiness problem of
Alternating Büchi Pushdown Systems (ABPDS)

Finite Automaton
recognizing potential infinite set of
configurations from which the ABPDS has an accepting run

PDS $\models$ CTL $\Longleftrightarrow$ The finite automaton is nonempty

[Song and Touili, CONCUR'10]

Implementation



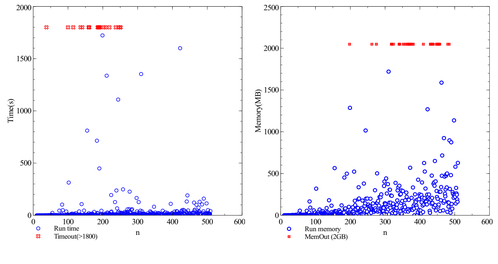
Satabs: [Kroening et al, TACAS'05]

JimpletoPDSolver: [Hague and Ong, SPIN'10]

Experiments

Verify Random PDSs

- 500 random PDSs
- $10 < |P| < 510$
- $|P|^2 < \Delta < 2|P|^2$
- $2 < |\psi| < 15$



Data Flow Analysis

- $def(x)$ holds at a control point n, iff the statement at n assigns some value to x, e.g., $x := 1$
- $use(x)$ holds at a control point n, iff the statement at n uses x, e.g., $y := x + 2$
- $\psi_1 = \mathbf{E}[\neg def(x)\mathbf{U}use(x)]$: x is used without being defined
- $\psi_2 = \mathbf{AG}(def(x) \Longrightarrow \mathbf{EF}use(x))$: whenever x is defined, it is eventually used

Program	#LOC	ψ_1			ψ_2		
		PDSolver	Our tool: PuMoC		PDSolver	Our tool: PuMoC	
		Time(s)	Time(s)	Mem(MB)	Time(s)	Time(s)	Mem(MB)
RegAction	3k	5.14	0.71	5.23	19.89	15.95	11.88
ELFDump	6k	7.63	1.43	9.19	50.62	37.27	20.64
FOP2PDF	17k	108.4	10.33	26.73	311.66	262.51	81.76
DOM2PDF	18k	54.53	11.92	29.21	167.15	254.68	88.52
DisAction	54k	458.77	134.18	87.09	>2000	1616.09	386.82
CFGAction	90k	1129.99	544.45	143.39	>2000	1734.81	514.56

Verify C and Java Programs

- 4 real-world Java programs taken from JimpleToPDSolver
- 4 real-world Java programs of SciMark2
- 7 real-world Java programs of JBDD
- C source code of two bounded model checkers (verbs and verds)

	Program	#LOC	Time(s)	Mem(MB)
Java Prog.	Namer	60	0.03	0.62
	cmdline	3k	5.72	32.14
	readCmdLine	78k	525.32	149.62
	usage	95k	3009.50	581.28
	FFT	1k	11.33	8.74
	Bench	3k	13.69	21.76
	HTTPPost	26k	17.95	100.28
	Applet	159k	4148.21	535.35
	Equivalence	335	0.04	1.82
	Queens	665	2.87	5.03
	Queens2	885	2.33	9.01
	Knight	1k	0.38	9.40
	DimacsSlover	1k	0.33	7.87
	interface	1k	23.86	26.17
	IQueens	109k	2440.32	411.67
	jlink	37k	2092.60	281.77
	JUnitTestRunner	26k	32.24	141.46
	DefaultDepDes	28k	1390.64	198.45
	IBiblioHelper	89k	4648.64	348.58

	Program	#LOC	Time(s)	Mem(MB)
Java Prog.	EntryPoint	11k	20.14	66.39
	ProcessDestroyer	11k	128.22	87.06
	TestProcess	1k	0.19	5.89
	IvyAuthenticator	3k	40.81	27.91
	JunitTestRunnerTest	28k	1427.90	179.40
	Diagnostics	59k	1353.17	238.09
	DirectoryScanner	17k	626.41	113.68
	IntrospectionHelper	5k	3.92	36.03
	Launcher	18k	1341.98	185.55
	KeySubst	3k	7.40	15.60
	IPlanetEjbc	22k	703.68	175.50
C Prog.	progreconstruct	4k	0.01	0.07
	cs2bool	4k	0.01	0.07
	specs	4k	0.01	0.08
	cnfsat	4k	0.01	0.07
	qmdwritecnf	8k	0.01	0.11
	treedopost	8k	128.22	87.06
	CNFspec2model	8k	0.19	5.89
	qmd2model	8k	40.81	27.91

Verify Windows Drivers

We checked 1461 versions of 30 Windows drivers against two properties: an API usage rule r_1 and a lock/unlock rule r_2.

Program	No.	Avg. #LOC	r_1		r_2	
			Avg. Time(s)	Avg. Mem(MB)	Avg. Time(s)	Avg. Mem(MB)
1394	10	7.9k	48.80	30.45	27.82	9.80
bluetooth	37	10.32k	67.83	34.38	28.11	10.43
SD	14	6.9k	27.42	19.57	7.10	5.93
PLX9x5x	16	13.2k	119.14	45.72	40.53	14.01
amcc5933	14	10.0k	54.50	32.26	16.84	9.40
cancel	69	3.4k	20.95	12.43	4.01	4.34
Echo	68	5.4k	28.34	19.10	7.46	5.66
event	20	4.8k	32.04	18.24	7.22	5.35
pcidrv	72	29.1k	422.58	115.56	181.52	36.58
perfcounters	16	2.2k	11.84	8.58	2.57	2.82
portio	26	4.9k	23.17	15.17	6.00	4.96
registry	35	11.4k	150.85	46.36	56.83	14.91
toaster_wdm_bus	43	9.6k	91.19	37.93	31.91	12.15
toaster_wdm_func	183	10.1k	90.05	40.44	32.97	12.99
toaster_wdm_toastmon	41	4.5k	32.15	17.58	8.24	5.70
toaster_wdm_filter	231	4.0k	26.36	15.58	6.65	5.11
toaster_kmdf	165	5.0k	19.45	15.13	5.19	4.89
tracing	62	2.8k	15.90	10.54	3.64	3.54
firefly	17	4.2k	12.96	10.21	3.14	3.45
hidmapper	29	2.5k	13.65	9.33	2.71	3.14
hidusbfx2	17	4.4k	24.83	16.62	4.48	3.95
HBtnKey	34	5.5k	48.06	21.76	13.29	6.98
hiddigi	65	9.2k	84.00	35.46	28.90	11.25
kbfiltr	15	6.5k	25.28	19.30	6.86	6.20
moufiltr	14	5.0k	13.04	11.79	3.15	3.88
vserial	9	4.2k	17.33	14.77	4.65	4.81
smscir	10	14.8k	293.78	57.25	117.33	18.50
network	59	43.8k	1283.84	171.42	594.93	52.95
serial	46	16.1k	174.61	63.55	69.19	20.56
storage	84	57.3k	923.24	224.42	401.03	69.38