CNF From Interpolants Via BDDs

Alexander Legg, Nina Narodytska & Leonid Ryzhyk

NICTA, University of NSW, University of Toronto

alexander.legg@nicta.com.au

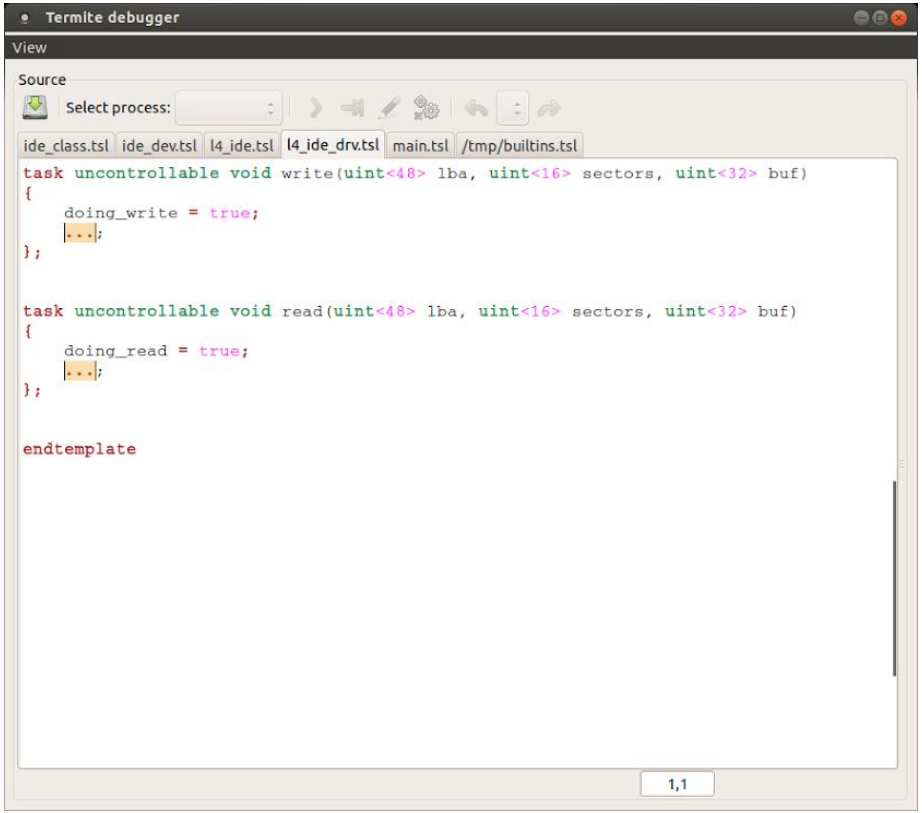
Background

NICTA

- Driver Synthesis
  - See CAV paper
- Strategy Extraction
  - Nina's talk yesterday

Driver Synthesis

NICTA



Driver Synthesis

NICTA

Termite debugger

View

Source

Select process:

ide_class.tsl | ide_dev.tsl | l4_ide.tsl | l4_ide_drv.tsl | main.tsl | /tmp/builtins.tsl

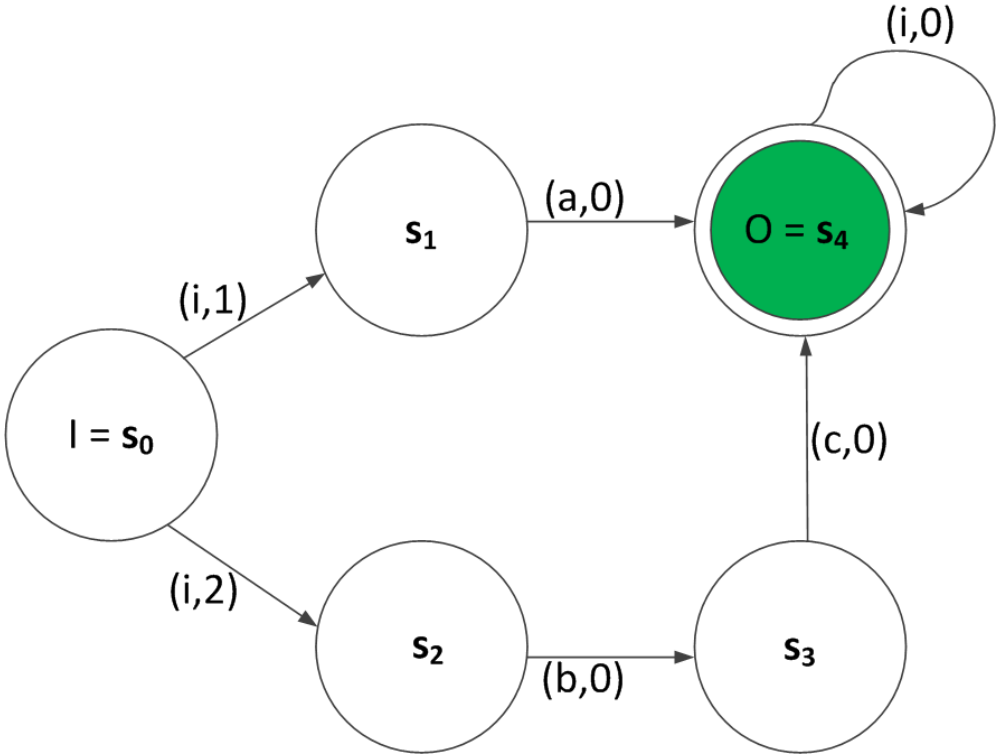
```
task uncontrollable void write(uint<48> lba, uint<16> sectors, uint<32> buf)
{
    doing_write = true;
    dev.rcmd_write_dev(/*any value*/6'h0 ++ 1'h1 ++ /*any value*/1'h0);
    dev.rcmd_write_lba_high0(os.r_lba[40:47]);
    dev.rcmd_write_lba_high0(os.r_lba[16:23]);
    dev.rcmd_write_lba_mid0(os.r_lba[32:39]);
    dev.rcmd_write_lba_mid0(os.r_lba[8:15]);
    dev.rcmd_write_lba_low0(os.r_lba[24:31]);
    dev.rcmd_write_lba_low0(os.r_lba[0:7]);
    dev.rcmd_write_sectors(os.r_sectors[8:15]);
    dev.rcmd_write_sectors(os.r_sectors[0:7]);
    dev.rcmd_write_errcmd(8'h35);
    dev.rdma_write_command(1'h0 ++ /*any value*/7'h0);
    dev.fill_prd(os.r_buf, dev.reg_sectors ++ dev.reg_sectors1);
    dev.rdma_write_command(1'h1 ++ /*any value*/2'h0 ++ 1'h0 ++ /*any value*/4'h0);
};


task uncontrollable void read(uint<48> lba, uint<16> sectors, uint<32> buf)
{
    doing_read = true;
    dev.rcmd_write_dev(/*any value*/6'h0 ++ 1'h1 ++ /*any value*/1'h0);
    dev.rcmd_write_lba_high0(os.r_lba[40:47]);
    dev.rcmd_write_lba_high0(os.r_lba[16:23]);
    dev.rcmd_write_lba_mid0(os.r_lba[32:39]);
    dev.rcmd_write_lba_mid0(os.r_lba[8:15]);
    dev.rcmd_write_lba_low0(os.r_lba[24:31]);
    dev.rcmd_write_lba_low0(os.r_lba[0:7]);
    dev.rcmd_write_sectors(os.r_sectors[8:15]);
    dev.rcmd_write_sectors(os.r_sectors[0:7]);
```

76,7

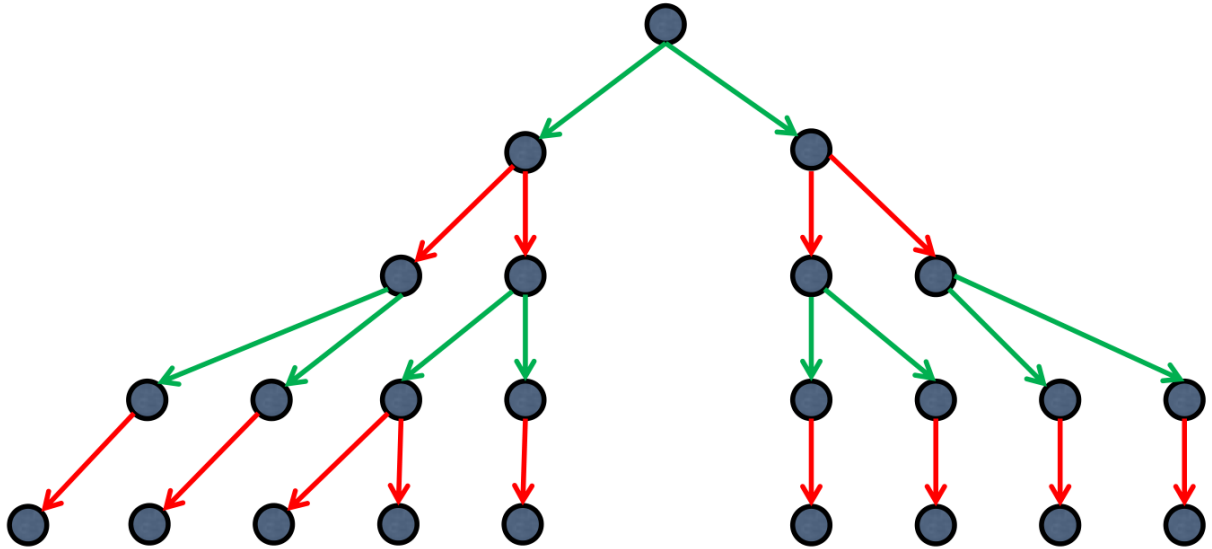
Background

NICTA



Background

NICTA

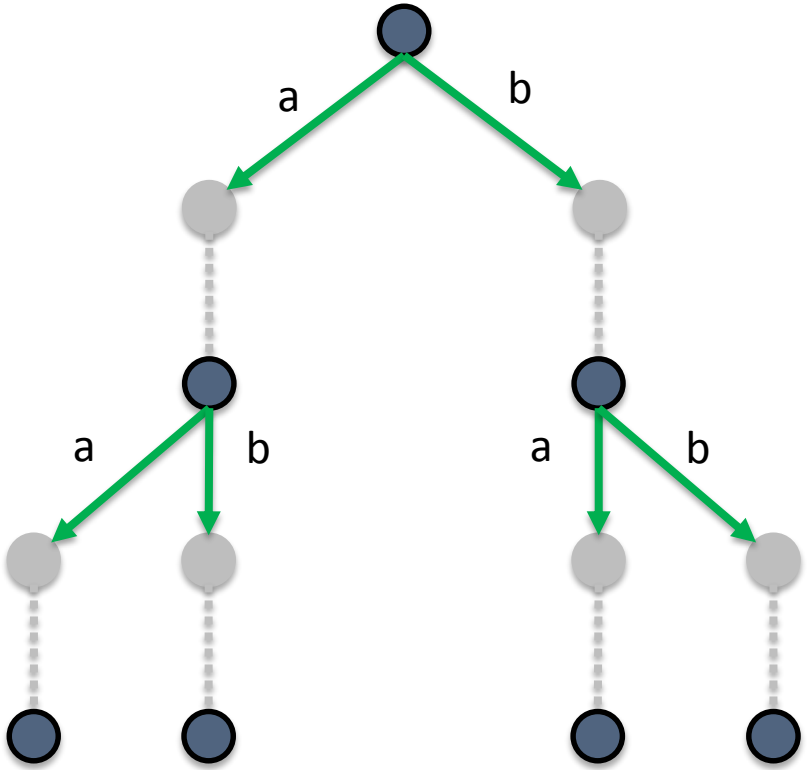


controllable move

uncontrollable move

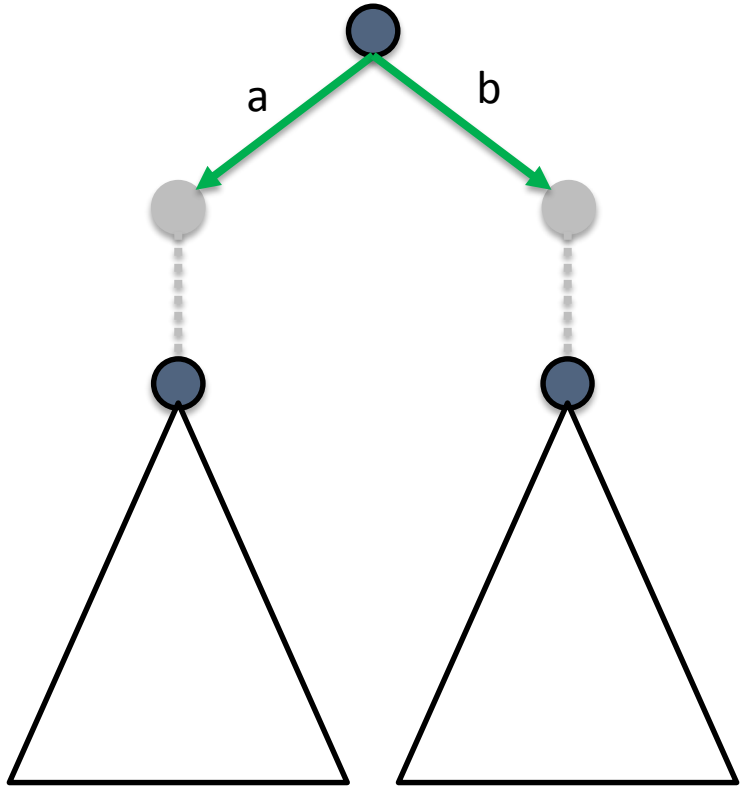
Background

NICTA



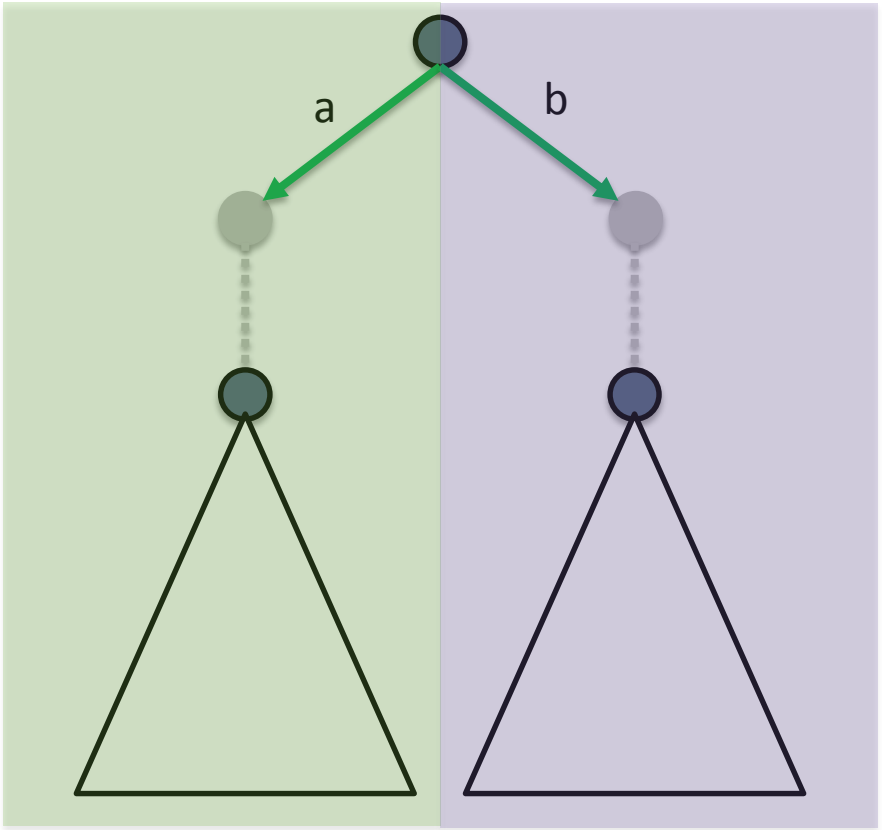
State Partitioning

NICTA



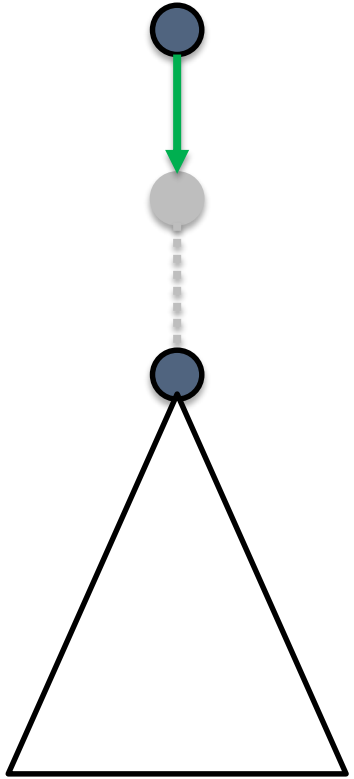
State Partitioning

NICTA



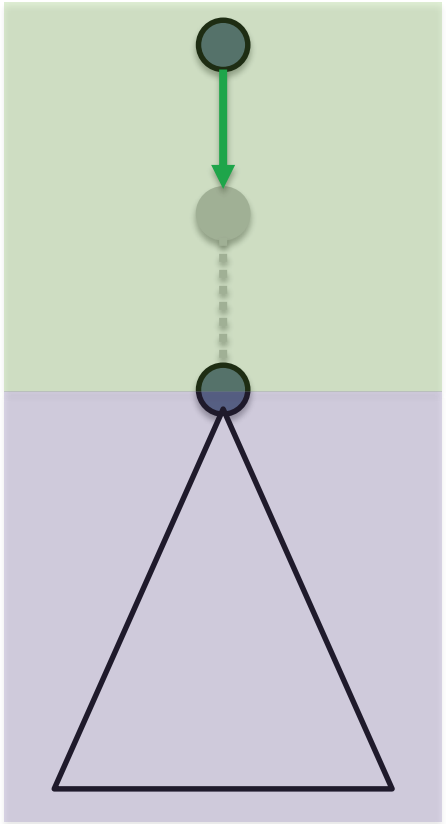
Next State Operation

NICTA



Next State Operation

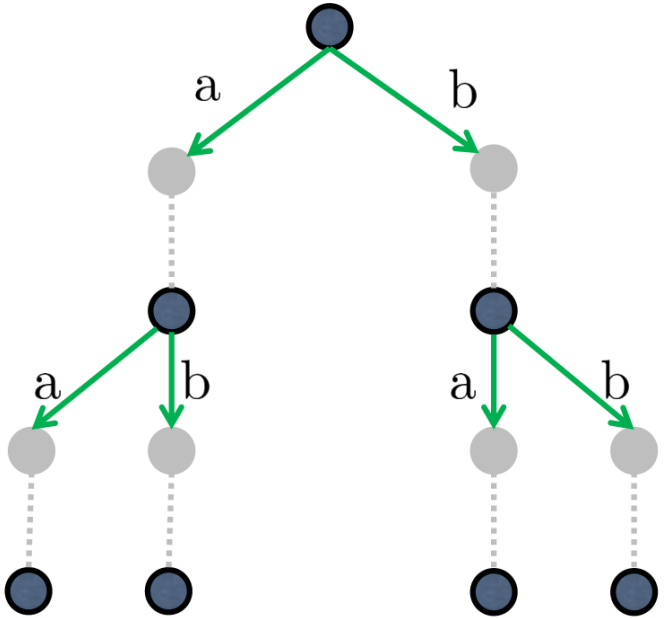
NICTA



Strategy Extraction

NICTA

- 1) Our interpolants get reused
  - We need small interpolants
  - We need small CNF
- 2) Our interpolants are over small sets of variables
  - Interpolants are state sets (over state variables)
  - An efficient representation exists



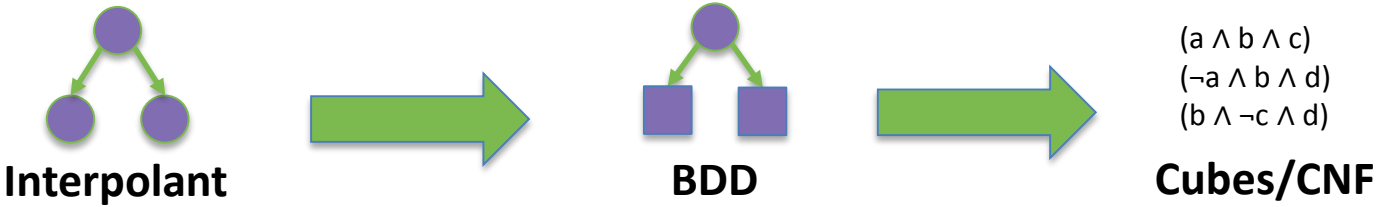
CNF Via BDDs

NICTA

- BDDs provide efficient representation
  - Interpolants are redundant and potentially large
  - BDDs are canonical
- CNF from BDD is simple and efficient
  - Get the shortest path to False
  - Block that path and repeat
- BDDs can explode
  - Small number of variables

Experimental Set Up

NICTA



Results

	Average	Maximum
Interpolant Size	67.66 nodes	1410 nodes
Interpolant Time	0.24 sec	1.50 sec
BDD Size	14.8 nodes	58 nodes
BDD Time	< 0.01 sec	< 0.01 sec
Cube Size	2.05 clauses	12 clauses
Cube Time	< 0.01 sec	< 0.01 sec

Runs of EvaSolver: 36
Total Interpolants: 872

Scalability

- Time spent solving 1297.37s
- Time spent on interpolants 232.33s (17%)

- BDDs do not contribute significantly to time
- Interpolant size increases with state space

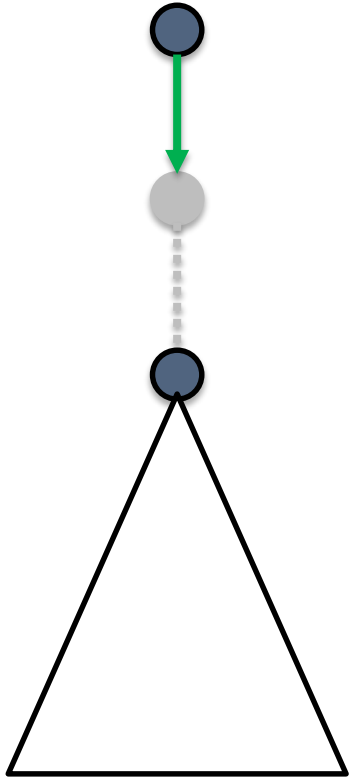
Related Work

NICTA

- ALLSAT
  - Alternative to interpolation
- Sweeping
  - Reduces circuit
  - Doesn’t give CNF
- Interpolants in CNF (CAV ‘13)
  - Needs domain specific solution

Next State Operation

NICTA



ALLSAT

NICTA

- Existential quantification
  - Find solution (via SAT)
  - Block solution
  - Repeat

ALLSAT

NICTA

- Existential quantification
  - Find solution (via SAT)
  - Block solution
  - Repeat
- Problem:

	$(x \wedge x')$	$(\neg x \wedge \neg x')$
After Projection:	(x')	$(\neg x')$

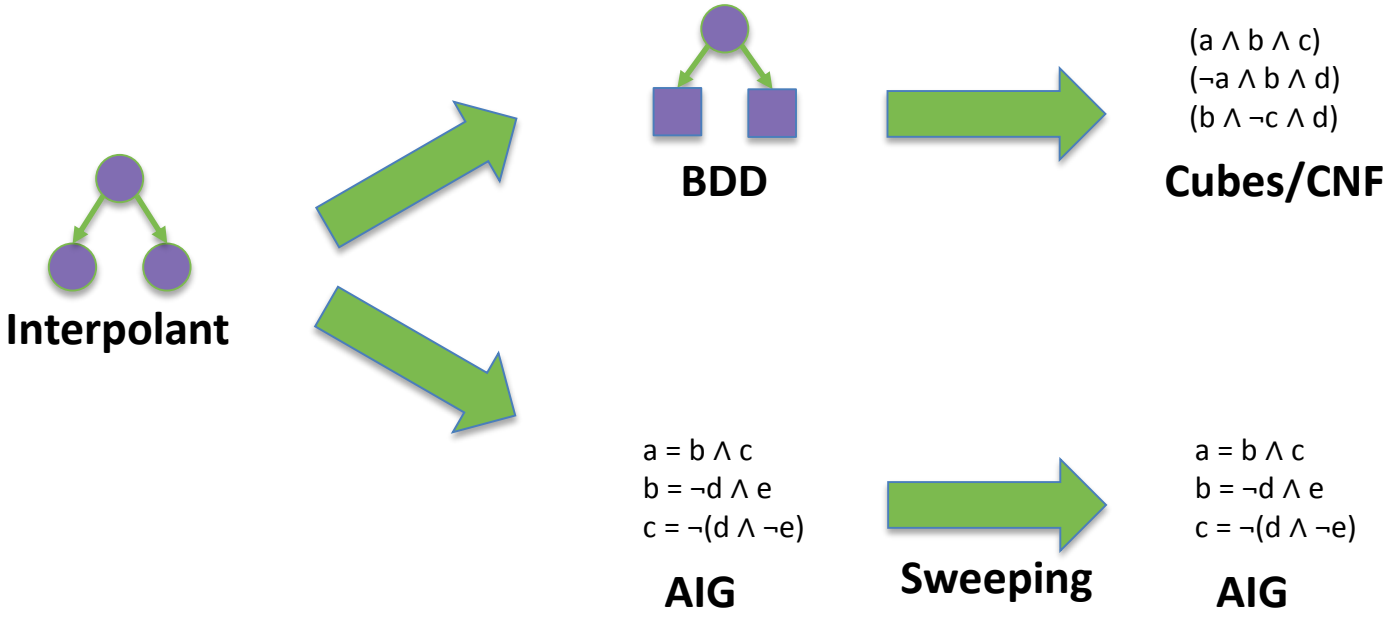
Related Work

NICTA

- ALLSAT
  - Alternative to interpolation
- Sweeping
  - Reduces circuit
  - Doesn’t give CNF
- Interpolants in CNF (CAV ‘13)
  - Needs domain specific solution

Experimental Set Up

NICTA



Results

	Average	Maximum
Interpolant Size	67.66 nodes	1410 nodes
Interpolant Time	0.24 sec	1.50 sec
Cube Size	2.05 clauses	12 clauses
Cube Time	< 0.01 sec	< 0.01 sec
Post-sweeping Size	14.89 nodes	80 nodes
Sweeping Time	< 0.01 sec	< 0.01 sec

Runs of EvaSolver: 36
Total Interpolants: 872

Third Party Libraries

NICTA

- PeRIPLO (University of Lagano)
  - Interpolant library
  - Backed by MiniSAT
  - Performs some redundancy detection
- CUDD (CU Boulder)
  - BDD library
- IImc (CU Boulder)
  - Model checking library
  - Sweeping algorithms (BDD, SAT, Cut)

Other Talks

NICTA

12:35 pm, July 21st, CAV
N. Narodytska, A. Legg, F. Bacchus, L. Ryzhyk and A. Walker
Solving Games without Controllable Predecessor

14:50 pm, July 21st, CAV
P. Cerny, T. Henzinger, A. Radhakrishna, L. Ryzhyk and T. Tarrach
Regression-free Synthesis for Concurrency

09:00 am, July 24th, SYNT
Leonid Ryzhyk
Automatic Device Driver Synthesis Project (Invited Talk, OSDI'14)

Questions