Widening ROBDDs with Prime Implicants

Neil Kettle[†], Andy King[†] and Tadeusz Strzemecki[‡]

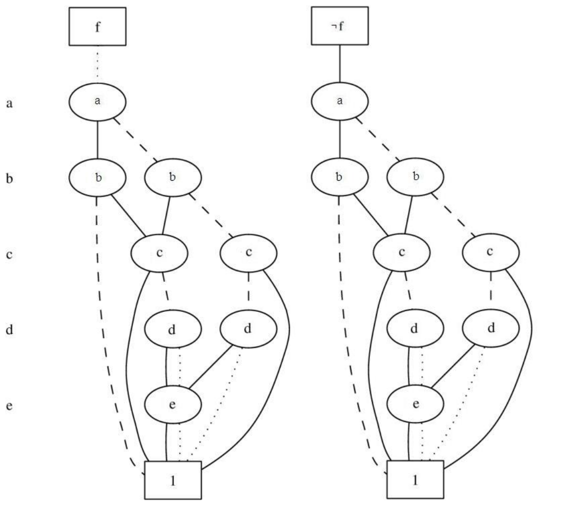
[†] University of Kent, Canterbury, UK
[‡] Fordham University, New York, USA

TACAS ETAPS-2006

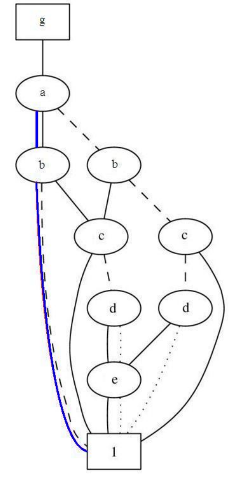
 

Reduced-Ordered Binary Decision Diagrams

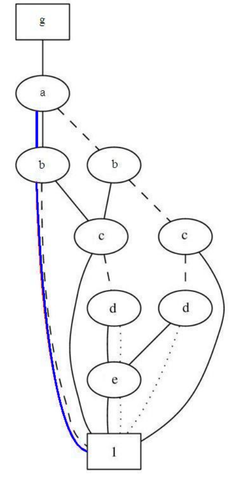
- ROBDDs have numerous applications in model checking, program analysis and abstract interpretation
- ROBDDs are remarkably tractable but problematic Boolean functions arise whose representation is excessive for any variable ordering
- Minimisation by variable reordering then is a limited solution, and this motivates the need for approximation



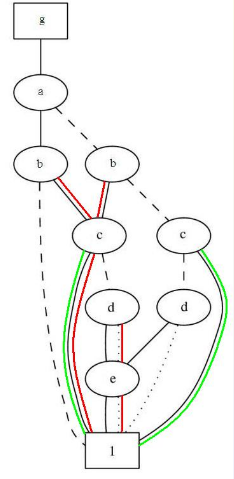
BDDs of $f = \neg c \wedge ((\neg a \wedge ((d \wedge \neg e) \vee (\neg b \wedge \neg d))) \vee (b \wedge (d \leftrightarrow \neg e)))$ and $g = \neg f$



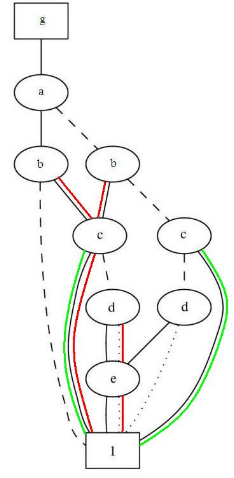
- A dense over-approximation of f can be obtained in terms of implicants of g
- A cube p is an implicant of g if $p \models g$, thus $(a \wedge \neg b)$ is an implicant of g



- A dense over-approximation of f can be obtained in terms of implicants of g
- A cube p is an implicant of g if $p \models g$, thus $(a \wedge \neg b)$ is an implicant of g
- Furthermore, $a \not\models g$ and $\neg b \not\models g$, thus $(a \wedge \neg b)$ is a prime implicant of g



- The cube $p = (b \wedge \neg d \wedge \neg e)$ is an implicant of $(g \wedge a \wedge c)$, $(g \wedge a \wedge \neg c)$, $(g \wedge \neg a \wedge c)$ and $(g \wedge \neg a \wedge \neg c)$
- The cube p is thus an implicant of g
- The implicant p is prime because
  - $(\neg d \wedge \neg e) \not\models g$ and
  - $(b \wedge \neg e) \not\models g$ and
  - $(b \wedge \neg d) \not\models g$

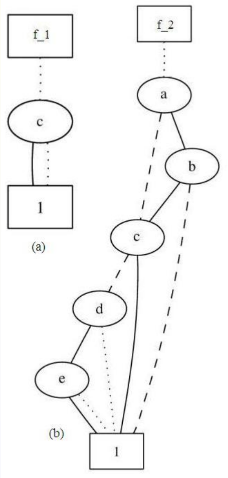


- The cube $p = (b \wedge \neg d \wedge \neg e)$ is an implicant of $(g \wedge a \wedge c)$, $(g \wedge a \wedge \neg c)$, $(g \wedge \neg a \wedge c)$ and $(g \wedge \neg a \wedge \neg c)$
- The cube p is thus an implicant of g
- The implicant p is prime because
  - $(\neg d \wedge \neg e) \not\models g$ and
  - $(b \wedge \neg e) \not\models g$ and
  - $(b \wedge \neg d) \not\models g$
- The cube c is an implicant of $(g \wedge a \wedge b)$, $(g \wedge a \wedge \neg b)$, $(g \wedge \neg a \wedge b)$ and $(g \wedge \neg a \wedge \neg b)$
- The cube c is thus an implicant of g
- The implicant c is prime since $true \not\models g$

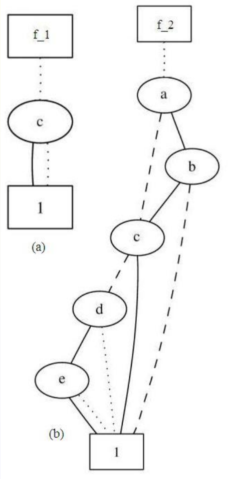
- The set of all prime implicants is $primes(g) = \{c, (d \wedge e), (a \wedge \neg b), (b \wedge \neg d \wedge \neg e)\}$
- Since
  - $c \models g$, it follows that $f = \neg g \models \neg c$
  - $(d \wedge e) \models g$, it follows that $f = \neg g \models \neg(d \wedge e)$
  - $(a \wedge \neg b) \models g$, it follows that $f = \neg g \models \neg(a \wedge \neg b)$
  - $(b \wedge \neg d \wedge \neg e) \models g$, it follows that $f = \neg g \models \neg(b \wedge \neg d \wedge \neg e)$
- This leads to a family of approximations (widenings)

$$\nabla_k(f) = \bigwedge \{\neg p \mid p \in primes(\neg f) \wedge \|p\| \leq k\}$$

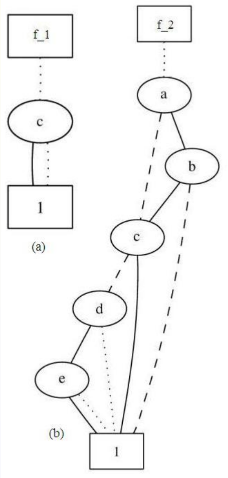
where $\|p\|$ denotes the number of propositional variables in p



- $\nabla_1(f) = f_1$ has 16 truth assignments and 1 node
- $\nabla_2(f) = f_2$ has 9 truth assignments and 5 nodes
- $\nabla_3(f) = f$ has 7 truth assignments and 8 nodes



- $\nabla_1(f) = f_1$ has 16 truth assignments and 1 node
- $\nabla_2(f) = f_2$ has 9 truth assignments and 5 nodes
- $\nabla_3(f) = f$ has 7 truth assignments and 8 nodes
- Observe $f \models \nabla_3(f) \models \nabla_2(f) \models \nabla_1(f)$



- $\nabla_1(f) = f_1$ has 16 truth assignments and 1 node
- $\nabla_2(f) = f_2$ has 9 truth assignments and 5 nodes
- $\nabla_3(f) = f$ has 7 truth assignments and 8 nodes
- Observe $f \models \nabla_3(f) \models \nabla_2(f) \models \nabla_1(f)$
- Generally
    - Tuneable: ∇_k is never less precise than ∇_{k-1}
    - Predictable: if $f_1 \models f_2$ then $\nabla_k(f_1) \models \nabla_k(f_2)$
    - Swings both ways: $f \models \nabla_k(f)$ and $\neg\nabla_k(\neg f) \models f$
    - Ordering independent: ∇_k approximates f rather than its representation

Implementation

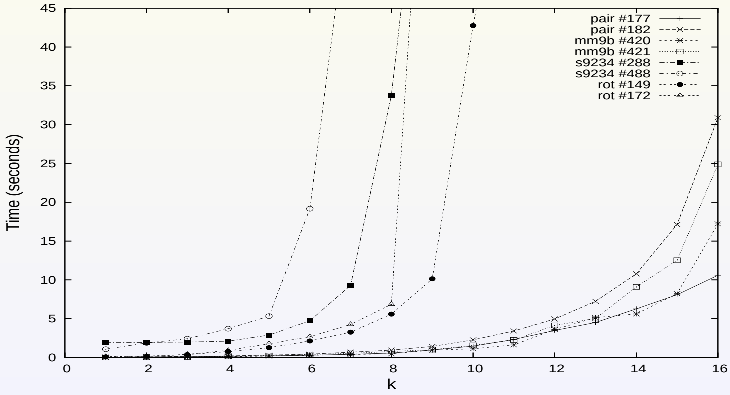
- The complexity of finding the shortest prime implicant is in $GC(\log^2 n, coNP)$-complete
- Coudert and Madre [1] proposed an algorithm for enumerating all primes whose complexity is related to the size of an ROBDD encoding of the primes and not the number of primes
- We overlay the algorithm with a constraint which ensures that the number of propositional variables in any prime does not exceed a given k
- This also reduces the size of all intermediate ROBDDs

[1] O. Coudert and J. C. Madre, "A New Graph Based Prime Computation Technique", in Logic Synthesis and Optimization, Kluwer, pages 33–57, 1993.

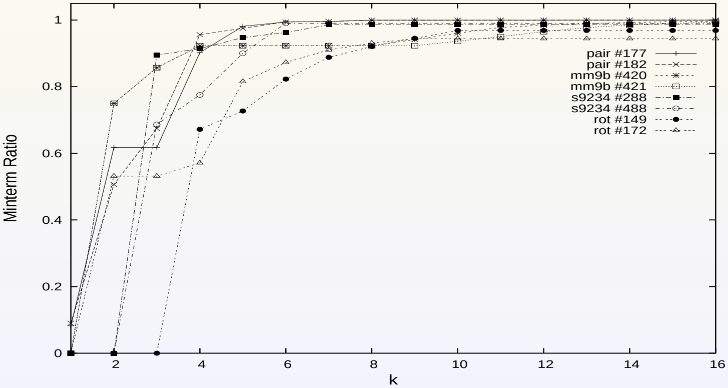
Experimental Results

ID		Approximation size	Approximation minterms	Ratios size	Ratios minterms	Time	Notes
Ravi	pair #177	8382	3.40×10^{14}	0.32	1.83	4.61	**q:** 0.94
	pair #182	9711	1.47×10^{15}	0.29	1.81	6.32	**q:** 0.84
	mm9b #420	933	1.88×10^{9}	0.01	1.16	10.85	**q:** 0.75
	mm92 #421	722	1.88×10^{9}	0.01	1.16	11.96	**q:** 0.84
	s9234 #288	15	5.68×10^{22}	0.01	1.58	1086.12	**q:** 0.88
	s9234 #488	11	2.89×10^{22}	0.01	1.49	2321.68	**q:** 0.92
Shiple	pair #177	8385	1.72×10^{15}	0.32	10.85	4.86	**q:** 0.92
	pair #182	9714	8.06×10^{15}	0.29	9.93	6.35	**q:** 0.81
	mm9b #420	933	1.88×10^{9}	0.01	1.16	12.39	**q:** 0.75
	mm92 #421	722	1.88×10^{9}	0.01	1.16	13.10	**q:** 0.84
	s9234 #288	15	5.68×10^{22}	0.01	1.58	1057.62	**q:** 0.87
	s9234 #488	11	2.89×10^{22}	0.01	1.49	2562.30	**q:** 0.92
Our	pair #177	11027	2.06×10^{14}	0.42	1.11	0.58	**k:** 5
	pair #182	7301	8.32×10^{14}	0.22	1.03	0.85	**k:** 6
	mm9b #420	44334	1.68×10^{9}	0.47	1.02	6.38	**k:** 12
	mm92 #421	39718	1.69×10^{9}	0.41	1.05	8.19	**k:** 11
	s9234 #288	75	3.64×10^{22}	0.01	1.01	20.36	**k:** 7
	s9234 #488	103	1.96×10^{22}	0.01	1.01	47.53	**k:** 6

Experimental Results



Experimental Results



Conclusions

- The approximation appears to be competitive with current existing methods
- By increasing k until a timeout is exceeded, we obtain a so-called anytime approach to ROBDD approximation
- Details are provided in the paper as to how prime implicants can be used to widen for time as well as space