Symbolic Model Checking

10^{20} States and Beyond

Burch Clarke McMillan Dill Hwang

Seminal Papers in Verification

March 23, 2012

Outline

The Mu-Calculus

The Mu-Calculus is similar to standard first-order logic.

- Does not include relational symbols or constant symbols.
- Relational symbols are replaced by relational variables.
- $\mu P[R]$ denotes the least fixed point of an n-ary relational term R and P is an n-ary relational variable.

Symbolic Model Checking

- Use BDDs as internal representation
- Recursively translate formula to BDD
- CTL expressions can be translated into efficient BDD operations.
- FalseBDD and TrueBDD correspond to trees with only one terminal node, 0 or 1 respectively.

Translating formulas

- Over the structure of formulas & terms

BDD_f: Formulas

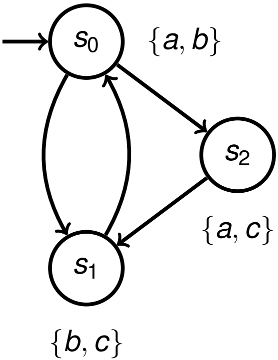
f is individual var	$\mathrm{BDDAtom}(f)$
$f = f_1 \wedge f_2$	$\mathrm{BDDAnd}(\mathrm{BDD}_f(f_1), \mathrm{BDD}_f(f_2))$
$f = \neg f_1$	$\mathrm{BDDNegate}(\mathrm{BDD}_f(f_1))$
$f = \exists x.f$	$\mathrm{BDDExists}(x, \mathrm{BDD}_f(f_1))$
$f = R(x_1, \ldots, x_n)$	$\mathrm{BDD}_R(R)\langle d_1 \leftarrow x_1, \ldots, d_n \leftarrow x_n\rangle$

BDD_R: Terms

R is relational var	$I_R(R)$
$R = \lambda x_1, \ldots, x_n.f$	$\mathrm{BDD}_f(f)\langle x_1 \leftarrow d_1, \ldots, x_n \leftarrow d_n\rangle$
$R = \mu P[R']$	$\mathrm{FixedPoint}(P, R', \mathrm{FalseBDD})$

- $\mathrm{AF}\ f_1 = \mu Z\ .\ f_1 \vee \mathrm{AX}\ Z$
- $\mathrm{EF}\ f_1 = \mu Z\ .\ f_1 \wedge \mathrm{EX}\ Z$
- $\mathrm{A}[f_1\ \mathrm{U}\ f_2] = \mu Z\ .\ f_2 \vee (f_1 \wedge \mathrm{AX}\ Z)$
- $\mathrm{E}[f_1\ \mathrm{U}\ f_2] = \mu Z\ .\ f_2 \vee (f_1 \wedge \mathrm{EX}\ Z)$

- The set of atomic prepositions $AP = \{a, b, c\}$
- The set of states $S = \{s_0, s_1, s_2\}$
- The set of transitions $T = \{(s_0, s_1), (s_1, s_0), (s_0, s_2), (s_2, s_1)\}$
- The labelling function $L = \{(s_0, \{a, b\}), (s_1, \{b, c\}), (s_2, \{a, c\})\}$

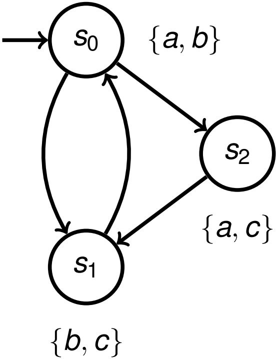


- CTL formulae:

$$f = EX\ c$$

- Mu-Calculus:

$$R = \lambda s[\exists t[c(t) \wedge T(s,t)]]$$



States are described by means of a vector of boolean variables

$$s_i = (x_1, x_2)$$

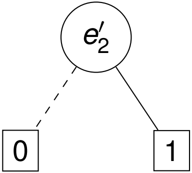
Boolean vectors can be represented as formulas

$$s_0 = \neg e_1 \wedge e_2, s_1 = \neg e_1 \wedge e_2, s_2 = e_1 \wedge e_2$$

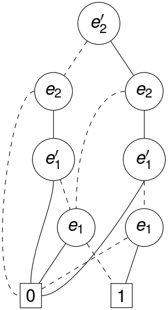
Trasitions, described by the pairs (s_i, s_i'), can be represented as

$$s_i \wedge s_i'$$

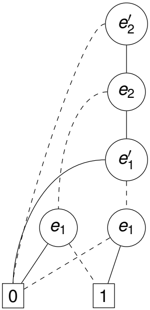
$c(t)$



$T(s,t)$



$$c(t) \wedge T(s, t)$$



$$\exists t[c(t) \wedge T(s,t)]$$

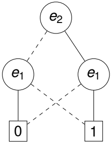
$$e'_2 = 0 \wedge e'_1 = 0$$

$$\boxed{0}$$

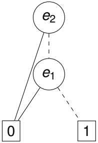
$$e'_2 = 0 \wedge e'_1 = 1$$

$$\boxed{0}$$

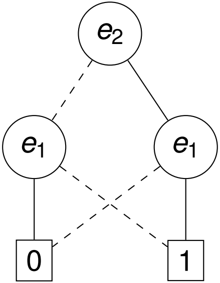
$$e'_2 = 1 \wedge e'_1 = 0$$



$$e'_2 = 1 \wedge e'_1 = 1$$



$$\exists t[c(t) \wedge T(s,t)] = [c(t) \wedge T(s,t)]_{e_2'=0, e_1'=0} \vee [c(t) \wedge T(s,t)]_{e_2'=0, e_1'=1} \vee \ldots$$



$\{s_0, s_2\}$

- Symbolic model checking allows larger models (many magnitudes).
- Interesting result: BDDs grow linearly
- State space very large
- Execution time still rises quickly

Symbolic Model Checkers

- **Most hardware design companies have their own Symbolic Model Checker(s)**
  - Intel, IBM, Motorola, Siemens, ST, Cadence, ...
  - very advanced tools
  - proprietary technolgy!
- **On the academic side**
  - CMU SMV [McMillan]
  - VIS [Berkeley, Colorado]
  - Bwolen Yang's SMV [CMU]
  - NuSMV [CMU, IRST, UNITN, UNIGE]
  - ...