Symbolic model checking

Why?

Saves us from constructing a model's state space explicitly. Effective ”cure” for state space explosion problem.

How?

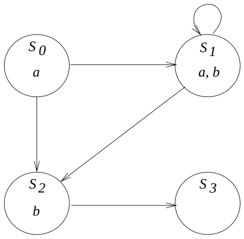
Sets of states and the transition relation are represented by formulas. Set operations are defined in terms of formula manipulations.

Data structures

ROBDDs - allow for efficient storage and manipulation of logic formulas.

Representing Models Symbolically

- A system state represents an interpretation (truth assignment) for a set of propositional variables V.
  - Formulas represent sets of states that satisfy it
    - False - $\emptyset$, True - S
    - a - set of states in which a is true - ($\{s_0, s_1\}$)
    - b - set of states in which b is true - ($\{s_1, s_2\}$)
    - $a \vee b$ = $\{s_0, s_1\} \cup \{s_1, s_2\}$ = $\{s_0, s_1, s_2\}$



- State transitions are described by relations over two sets of variables, V (source state) and V' (destination state)
  - Trans. from s_2 to s_3 is described by $(\neg a \wedge b \wedge \neg a' \wedge \neg b')$.
  - Trans. from s_0 to s_1 and s_2, and from s_1 to s_2 and to itself is described by $(a \wedge b')$.
  - Relation R is described by $(a \wedge b') \vee (\neg a \wedge b \wedge \neg a' \wedge \neg b')$

Model Checking using Sets of States

Computing $||\varphi||$

φ is $\top$	:	return S
φ is $\bot$	:	return $\emptyset$
φ is atomic	:	return $\{s \in S \mid \varphi \in L(s)\}$
φ is $\neg\varphi_1$	:	return $S \setminus ||\varphi_1||$
φ is $\varphi_1 \wedge \varphi_2$	:	return $||\varphi_1|| \cap ||\varphi_2||$
φ is $\varphi_1 \vee \varphi_2$	:	return $||\varphi_1|| \cup ||\varphi_2||$
φ is $AX\varphi_1$	:	return $||\neg EX\neg\varphi||$
φ is $EX\varphi_1$	:	return SAT$_{EX}(\varphi_1)$
φ is $EU\varphi_1$	:	return SAT$_{EU}(\varphi_1)$
φ is $EG\varphi_1$	:	return SAT$_{EG}(\varphi_1)$

Model Checking on Sets of States, Cont'd

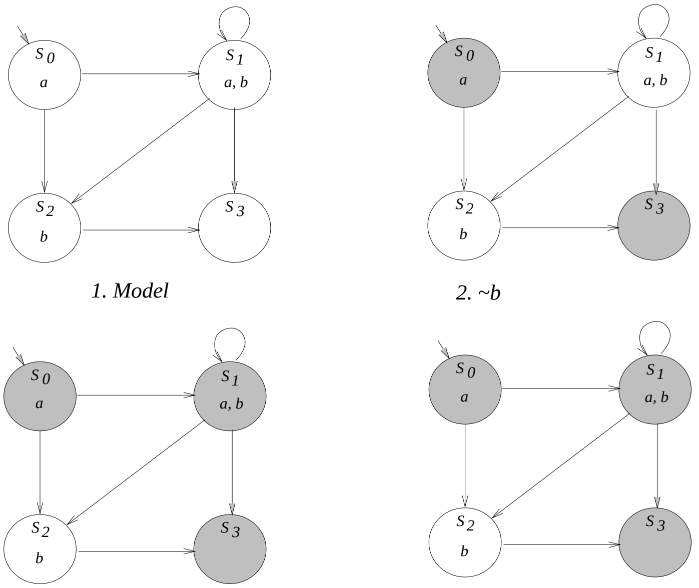
```
function SAT$_{EX}$(φ):
    return $\{s_0 \in S \mid s_0 \rightarrow s_1$ for some $s_1 \in ||\varphi||\}$

function SAT$_{EG}(\varphi)$
    $X$ := $\emptyset$; $Y$ := $S$;
    repeat
        $X := Y$
        $Y := ||\varphi|| \cap$ SAT$_{EX}(X)$
    until $X = Y$
    return $Y$
```

Model Checking on Sets of States, Cont'd

```
function SAT$_{EU}$(φ, ψ)
/* compute set of states satisfying $E[\varphi U \psi]$ */
    $X := \emptyset$; $Y := \emptyset$
    repeat
        $X := Y$
        $Y := ||\psi|| \cup (||\varphi|| \cap \text{SAT}_{EX}(X))$
    until $X = Y$
    return $Y$
```

Example: $M, s_2 \models \mathrm{E}[a\,U\,\neg b])$



1. Model

2. ~b

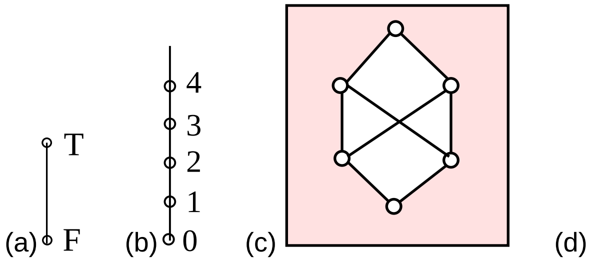
3. ~b \/ (a /\ EX E[a U ~b])

4. ~b \/ (a /\ EX E[a U ~b]) \/ (a /\ EX EX E[a U~b])

Lattice Theory

Def: A lattice is a partial order $(L, \leq)$ for which a unique greatest lower bound and a unique least upper bound exist for each pair of elements.
These are known as *join* $(a \sqcup b)$ and *meet* $(a \sqcap b)$.

Examples:



(a) (Bool, ⇒); (b) (Nat, $\leq$); (c) A non-lattice; (d) $\left(2^{\{a,b,c\}}, \subseteq\right)$

$\top$ (top) = $\sqcup L$
$\bot$ (bottom) = $\sqcap L$

Properties of Lattices

monotonicity $\quad a \leq a' \wedge b \leq b' \Rightarrow a \sqcap b \leq a' \sqcap b'$
$\quad a \leq a' \wedge b \leq b' \Rightarrow a \sqcup b \leq a' \sqcup b'$

idempotence $\quad a \sqcup a = a$
$\quad a \sqcap a = a$
commutativity $\quad a \sqcup b = b \sqcup a$
$\quad a \sqcap b = b \sqcap a$
associativity $\quad a \sqcup (b \sqcup c) = (a \sqcup b) \sqcup c$
$\quad a \sqcap (b \sqcap c) = (a \sqcap b) \sqcap c$
absorption $\quad a \sqcup (a \sqcap b) = a$
$\quad a \sqcap (a \sqcup b) = a$

In general, a function $F : L \rightarrow L$ is monotone if
$\forall x, y \in L \cdot x \leq y \Rightarrow F(x) \leq F(y)$.

Monotone Functions and Fixpoints

S — set of states, $F : P(S) \rightarrow P(S)$ — function on the powerset of S.

1. F is *monotone* if $\forall X, Y \subseteq S \cdot X \subseteq Y$ implies $F(X) \subseteq F(Y)$
2. $X \subseteq S$ is a *fixpoint* of F if $X = F(X)$

Examples:
1. $S = \{s_0, s_1\}$, $F(Y) = Y \cup \{s_0\}$
Is F monotone?

What are fixpoints of F?

2. $G(Y) =$ if $Y = s_0$ then $\{s_1\}$ else $\{s_0\}$

Is G monotone?

What are fixpoints of G?

Fixpoints (Cont'd)

Greatest fixpoint:

$$Y = F(Y) \wedge \forall X \cdot X = F(X) \Rightarrow X \subseteq Y$$

Computing greatest fixpoint:

$$\top \supseteq F(\top) \supseteq F(F(\top)) \supseteq ... \supseteq F^i(\top) = F^{i+1}(\top)$$

Least fixpoint:

$$Y = F(Y) \wedge \forall X = F(X) \Rightarrow Y \subseteq X$$

Computing least fixpoint:

$$\bot \subseteq F(\bot) \subseteq F(F(\bot) \subseteq ... \subseteq F^i(\bot) = F^{i+1}(\bot)$$

$F^i(X)$ means ”F applied i times”.

Fixpoints (Cont'd)

Can a monotone function have several fixpoints?
If F is a monotone function, is lfp(F) = gfp(F)? (or $\mu X.F(X) = \nu X.F(X)$)

Exercises:
Let $H_1, H_2, H_3 : \mathcal{P}(\{1,\dots,10\}) \rightarrow \mathcal{P}(\{1,\dots,10\})$. Let $\forall Y \subseteq \{1,\dots,10\}$.
Which of the functions are monotone:

- $H_1(Y) = Y - \{1,4,7\}$
- $H_2(Y) = \{2,5,9\} - Y$
- $H_3(Y) = \{1,2,3,4,5\} \cap (\{2,4,8\} \cup Y)$

What are greatest and least fixpoints of H_3?

Theorem (Knaster-Tarski)

Let $(L, \leq)$ be a lattice, $F : L \rightarrow L$ be a monotone function. Then,
$\mu X.F(X) = F^{n+1}(\bot)$ and $\nu X.F(X) = F^{n+1}(\top)$, where $n = \mathit{height}(L)$.

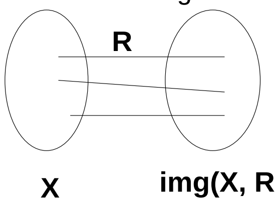
Proof:

1. $\bot \subseteq F^1(\bot) \subseteq F^2(\bot) \subseteq \dots \subseteq F^i(\bot) \forall i \geq 1$.
$F^1(\bot)$ needs to contain at least one element. F^2 – at least two. Thus, F^{n+1} should contain $n+2$ elements. This cannot happen since L has only $n+1$ unique elements.

2. Suppose X is another fixpoint. We should show that $F^{n+1}(\bot) \subseteq X$.
The proof goes by induction:
(a). $\bot \subseteq X$
(b). $F(\bot) \subseteq F(X) = X$ since F is monotone.
Thus, $F^i(\bot) \subseteq X \ \forall i \geq 0$, including the i which is $n+1$.

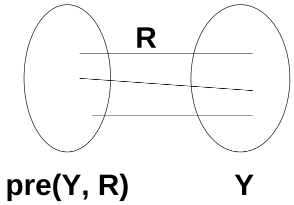
Forward and Backward Image

- Forward image:



$img(X, R)$ = $\{s' \mid \exists s \in X \ \wedge \ (s,s') \in R\}$
$||img(X,R)||(s') = \exists s \cdot X(s) \wedge R(s,s')$
$||img(X,R)||(s') = \exists s \in (X \cap R^{-1}(s'))$
$||img(X,R)|| = \exists V \cdot X \ \wedge \ R^{-1}$

- Backward (pre)image:



$pre(Y, R)$ = $\{s \mid \exists s' \in Y \ \wedge \ (s,s') \in R\}$
$||pre(Y,R)||(s) = \exists s' \cdot Y(s') \wedge R(s,s')$
$||pre(Y,R)|| = \exists V' \cdot Y' \wedge R$

Theorem: pre and img ($post$) are monotone.

Symbolic Calculation of EXb for System on Slide 97

Symbolic representation of the transition relation is:

$$R = (a \wedge b') \vee (\neg a \wedge b \wedge \neg a' \wedge \neg b')$$

Symbolic computation using pre-image on Slide 107.

$$\begin{array}{ll}
 & ||EXb|| \\
= & pre(b,R) \\
= & \exists a',b' \cdot R \wedge b' \\
= & \exists a',b' \cdot ((a \wedge b') \vee (\neg a \wedge b \wedge \neg a' \wedge \neg b')) \wedge b' \\
= & \exists a',b' \cdot ((a \wedge b') \wedge b') \vee ((\neg a \wedge b \wedge \neg a' \wedge \neg b') \wedge b') \\
= & \exists a',b' \cdot (a \wedge b') \vee \mathsf{f} \\
= & \exists a',b' \cdot (a \wedge b') \\
= & \exists b' \cdot (a \wedge b') \\
= & (a \wedge \mathsf{t}) \vee (a \wedge \mathsf{f}) \\
= & a
\end{array}$$

That is, $||EXb||$ is true in a state s iff $s \models a$.

Correctness Arguments: SAT$_{EU}$

Intuition: least fixpoint - finite number of iterations

$E[\varphi U \psi] = \psi \vee (\varphi \wedge EXE[\varphi U \psi])$ or
$||E[\varphi U \psi]|| = ||\psi|| \cup (||\varphi|| \cap ||EXE[\varphi U \psi]||)$
So, $||E[\varphi U \psi]||$ is a fixpoint of $G(X) = ||\psi|| \cup (||\varphi|| \cap ||EX\, X||)$

Theorem: For G as defined above and $n = |S|$,
1. G is monotone
2. $||E[\varphi U \psi]|| = \mu X.G(X)$

Proof

1. Monotonicity. Take $X, Y \subseteq S$, $X \subseteq Y$.
We need to show $G(X) \subseteq G(Y)$
$G(X) = ||\psi|| \cup (||\varphi|| \cap ||EXX||$
$\subseteq ||\psi|| \cup (||\varphi|| \cap ||EXY||$
$= G(Y)$
2. Show that
$\forall X \subseteq S \cdot G(X) = X \Rightarrow X \supseteq ||E[\varphi U \psi]||$
Proof is by induction on the length of prefix of the path along which $\varphi U \psi$ is satisfied: there is a path $s_0, s_1, \ldots$ and $j \geq 0$ s.t. $s_j \models \psi \wedge \forall l < i, s_l \models \varphi$.
If this length is 0, then it can be computed by $G^1(\emptyset) = ||\psi||$
Inductive hypothesis: G^{i+1} computes $E[\varphi U \psi]$ for length up to i.
Inductive case: Consider the path $s_0, s_1, \ldots$. For state s_1, inductive hypothesis holds. Since $(s_0, s_1) \in R, s_0 \models \varphi$ and $s_0 \models EX(G^{i+1}(\emptyset))$, thus, $s_0 \in G^{i+2}$.

Correctness Arguments: SAT$_{EG}$

Intuition: greatest fixpoint: infinite number of iterations

$EG\,\varphi = \varphi \wedge EXEG\,\varphi$ or $||EG\,\varphi|| = ||\psi|| \cap \{s \mid \exists s'\ s \rightarrow s' \wedge s' \in ||EG\,\varphi||\}$
So, $||EG\,\varphi||$ is a fixpoint of $F(X) = ||\varphi|| \cap ||EX\ X||$

Theorem: Let F be defined above and $n = |S|$.
1. F is monotone
2. $||EG\,\varphi|| = \nu X.F(X)$

Proof:

1. Monotonicity. Obvious because of monotonicity of EX.

2. Show that

$$\forall X \subseteq S \cdot F(X) = X \Rightarrow X \subseteq ||EG\,\varphi||$$

So, take an element $s \in X$ and show that it is in $||EG\,\varphi||$.
Take $s_0 \in X$. $F(X) = ||\varphi|| \cap EX\ X$, so clearly, $||\varphi||(s_0)$ holds.
By mathematical induction, construct a path s_0, s_1, ... such that $||\varphi||(s_i)$ holds.
So, $s_0 \in ||EG\,\varphi||$.

Symbolic Model-Checking Algorithm on BDDs

Procedure MC(p)
Case

```
  $p \in A$            :  **return** *Build*(“p”)
  $p = \neg\varphi$         :  **return** *Apply*(’$\neg$’, MC($\varphi$))
  $p = \varphi \wedge \psi$       :  **return** *Apply*(’$\wedge$’, MC($\varphi$), MC($\psi$))
  $p = \varphi \vee \psi$       :  **return** *Apply*(’$\vee$’, MC($\varphi$), MC($\psi$))
  $p = EX\varphi$        :  **return** *existQuantify*($V'$,
                              *Apply*(‘$\wedge$’, $R$, *Prime*(MC($\varphi$)))
  $p = AX\varphi$        :  **return** *Apply*(’$\neg$’, MC($EX\ \neg\varphi$))
  $p = E[\varphi U \psi]$     :  $Q_0$ = *Build*(’$\bot$’)
                          $Q_{i+1}$ = *Apply*(’$\vee$’, $Q_i$, *Apply*(’$\vee$’, MC($\psi$),
                             *Apply*(’$\wedge$’, MC($\varphi$), MC($EX\ Q_i$)))
                          **return** $Q_n$ when $Q_n = Q_{n+1}$
  $p = EG\varphi$        :  $Q_0$ = *Build*(’$\top$’)
                          $Q_{i+1}$ = *Apply*(’$\wedge$’, MC($\varphi$), MC($EX\ Q_i$))))
                          **return** $Q_n$ when $Q_n = Q_{n+1}$
```

How do all CTL operators look like?

AG f = $\mu Z. f \wedge$ AX Z

EG f = $\nu Z. f \wedge$ EX Z

AF f = $\nu Z. f \vee$ AX Z

EF f = $\mu Z. f \vee$ EX Z

E[f U g] = $\mu Z. g \vee (f \wedge$ EX Z)

Symbolic Fairness

- Let $C = \{\psi_1, \psi_2, \ldots, \psi_k\}$ be fairness constraints.
- Recall, we only need to know how to compute $||E_C G\varphi||$
- A set $Z = ||E_C G\varphi||$ if it is the largest set such that
  1. $Z \subseteq ||\varphi||$
  2. for all fairness constraints ψ_i, and all states $s \in Z$, there exists a path of length *one* or more to a state in $||\psi_i||$, going only through states in $||\varphi||$.
- Symbolically
  - $\nu Z \cdot \varphi \wedge \bigwedge_{i=1}^{k} EXE[\varphi\, U\, (Z \wedge \psi_i)]$
  - BTW: formula not expressible in CTL
  - Note: EU recomputed at each iteration of EG!
  - Complexity: square in $|S|$

Witnesses and Counterexamples

- Witness for $||EX\varphi||(s)$
  - s_1 is a witness iff it is in $img(\{s\},R) \cap ||\varphi'||$
- Witness for $||E[\varphi U \psi]||(s_0)$
  - From the algorithm: $s \in Q_i$ iff there exists a path of *at most* i steps from s to a state in $||\psi||$, going only through states in $||\varphi||$.
  - Find the smallest i such that $s_0 \in Q_i$
  - Let s_1 be a witness to $||EX\ Q_{i-1}||(s_0)$, s_2 a witness to $||EX\ Q_{i-2}||(s_1)$, etc.
  - $s_0, s_1, s_2, \ldots$ is the witness for EU

Witnesses and Counterexamples (Cont'd)

- Witness for $||EG\varphi||(s)$
  - Need to find a looping path from s, going through states in $||\varphi||$.
  - $||\varphi \wedge EXE[\varphi\, U\ (\varphi \wedge \{s\})]||(s)$ means – there exists a path from s to itself going only through states in $||\varphi||$.
  - If $||\varphi \wedge EXE[\varphi\, U\ (\varphi \wedge \{s\})]||(s)$ holds, then apply algorithm for EU
  - Otherwise,
    - $*$ find a witness s_1 for $||\varphi \wedge EXEG\varphi||(s)$
    - $*$ repeat from s_1
  - Why does this terminate?