Verification of Dynamic Properties by Model Checking

Alexander Bockmayr
FU Berlin
SS 2015

Interaction graph: topology

$$X_1 \in \{0, 1, 2, 3\}$$

$$X_2 \in \{0, 1\}$$

$$X_3 \in \{0, 1\}$$

Steady states and cycles

Model checking

State transition graph: dynamics

1. Temporal logic
Computation Tree Logics

• Atomic formulae/properties: p, q, r, …
  e.g. $p \equiv \text{"gene P is on"}$
• Boolean operators: $\neg$, $\land$, $\lor$, $\Rightarrow$
• Linear time operators
  – $Xp : p$ holds next time
  – $Fp : p$ holds sometimes in the future
  – $Gp : p$ holds globally in the future
  – $p \lor q : p$ holds until $q$ holds
• Path quantifiers
  $A : \text{for every path}, \quad E : \text{there exists a path}$

Linear time operators

For all paths …

There exists a path …

Examples

• $EF(p \land EF\ q) : \text{Can q hold after p holds ?}$
• $AF(AG\ p) : \text{Must the system reach a state where p holds forever ?}$
• $EG(p \Rightarrow EF(\neg p)) \land (\neg p \Rightarrow EF(p)) :$
  Can the system exhibit cyclic behavior w.r.t. property $p$?
Logic: Syntax and semantics

• Syntax
  When a sequence of symbols is a formula?

• Semantics
  What is the meaning of the formula?
  When is it true or false?

CTL Semantics

Given a Kripke model $M = (S, \rightarrow, L)$, a state $s \in S$, and a CTL formula $\phi$, the satisfaction relation $M, s \models \phi$ (shortly $s \models \phi$) is inductively defined as follows:

- $s \models p$ iff $p \in L(s)$
- $s \models \neg \phi$ iff not $s \models \phi$
- $s \models \phi \land \psi$ iff $s \models \phi$ and $s \models \psi$
- $s \models \phi \lor \psi$ iff $s \models \phi$ or $s \models \psi$

Kripke model

Formulas are interpreted over Kripke models $M = (S, \rightarrow, L)$

• $S$ is a finite set of states,
• $\rightarrow \subseteq S \times S$ : total transition relation (i.e., for each $s \in S$, there exists $s' \in S$ with $s \rightarrow s'$),
• Labeling $L : S \rightarrow 2^{AP}$ defines for each state $s$ the set $L(s)$ of atomic formulas true in $s$.

CTL Semantics II

- $s \models AF \phi$ iff for all paths $s = s_1 \rightarrow s_2 \rightarrow \ldots$ we have $s_i \models \phi$, for some $i \geq 1$.
- $s \models EF \phi$ iff for some path $s = s_1 \rightarrow s_2 \rightarrow \ldots$ we have $s_i \models \phi$, for some $i \geq 1$.
- $s \models AG \phi$ iff for all paths $s = s_1 \rightarrow s_2 \rightarrow \ldots$ we have $s_i \models \phi$, for all $i \geq 1$.
- $s \models EG \phi$ iff for some path $s = s_1 \rightarrow s_2 \rightarrow \ldots$ we have $s_i \models \phi$, for all $i \geq 1$.

CTL Syntax

• Atomic formulas are CTL formulas.
• If $\phi$ and $\psi$ are CTL formulas, then $\neg \phi, \phi \land \psi, \phi \lor \psi, AF \phi, EF \phi, AG \phi, EG \phi, AX \phi, EX \phi, A[\phi U \psi], E[\phi U \psi]$ are CTL formulas.

CTL Semantics III

- $s \models AX \phi$ iff for all $s' \in S$ with $s \rightarrow s'$ we have $s' \models \phi$.
- $s \models EX \phi$ iff for some $s' \in S$ with $s \rightarrow s'$ we have $s' \models \phi$.
- $s \models A[\phi U \psi]$ iff for all paths $s = s_1 \rightarrow s_2 \rightarrow \ldots$ there exists $i \geq 1$ such that $s_i \models \psi$ and $s_j \models \phi$, for all $1 \leq j < i$.
- $s \models E[\phi U \psi]$ iff for some path $s = s_1 \rightarrow s_2 \rightarrow \ldots$ there exists $i \geq 1$ such that $s_i \models \psi$ and $s_j \models \phi$, for all $1 \leq j < i$. 
2. Model checking algorithm

Model checking

- M Kripke model (or transition system)
- $\phi$ temporal logic formula
- Find all states $s$ of M such that $s \models \phi$
- Efficient model checking algorithms and software tools exist for the logic CTL.

Model checking algorithm

- Input: Kripke model $M = (S, \rightarrow, L)$ and CTL formula $\phi$
- Output: Set of states in $M$ that satisfy $\phi$
- Labeling algorithm: Label states of $M$ with the subformulas of $\phi$ that are satisfied there, starting with the smallest subformulas and working outwards towards $\phi$.

Model checking algorithm (II)

- Let $g$ be a subformula of $\phi$ and suppose all immediate subformulas of $g$ have already been labeled.
- Determine states to be labeled by $g$ as follows:
  - $\neg f$: label $s$ with $\neg f$ if $s$ is not already labeled with $f$.
  - $p$: label $s$ with $p$ if $p \in L(s)$.
  - $f_1 \land f_2$: label $s$ with $f_1 \land f_2$ if $s$ is already labeled with both $f_1$ and $f_2$.
  - $\neg f$: label $s$ with $\neg f$ if $s$ is not already labeled with $f$.

Model checking algorithm (III)

- $\neg \neg f /\land f$:
  - Label $s$ with $\neg \neg f$ if one of its successors is labeled with $f$.

Equivalences

- $AX f = \neg EX (\neg f)$
- $AG f = \neg EF (\neg f)$
- $EG f = \neg AF (\neg f)$
- $EF f = E[true U f]$
- $A[f U g] = \neg E[\neg g U (\neg f \land \neg g)] \land AF g$

Any CTL formula can be expressed using only the operators EX, EU, and AF.
Model checking algorithm (IV)

- **AF f:**
  - If any state is labeled with f, label it with AF f.
  - Repeat: label any state with AF f if all successor states are labeled with AF f, until there is no change.

Model checking algorithm (V)

- **E[f₁ U f₂]:**
  - If any state is labeled with f₂, label it with E[f₁ U f₂].
  - Repeat: label any state with E[f₁ U f₂] if it is labeled with f₁ and at least one of its successors is labeled with E[f₁ U f₂], until there is no change.

Example: EU

Label with E[¬q U p] all states which satisfy p.

Example: EU (contd)

Label any state s with E[¬q U p] if it is labeled with ¬q and at least one of its successor is already labeled with E[¬q U p].

Example: Input

M =

\[ \phi = \text{AF}(E[\neg q U p] \lor \text{EX}q) \]

Example: EU (contd)

Label any state s with E[¬q U p] if it is labeled with ¬q and at least one of its successor is already labeled with E[¬q U p].
Example: EX

Label with EXq any state s with one of its successors already labeled with q

Example: AF (contd)

Label any state s with f if all successors of s are already labeled with f

Example: \( \lor \)

Label with \( g = E[\neg q U p] \lor EXq \) any state s already labeled with \( E[\neg q U p] \) or EXq

Example: Output

M =

All states satisfy AF(\( E[\neg q U p] \lor EXq \))

Example: AF

Label with f= AF(\( E[\neg q U p] \lor EXq \)) any state already labeled with g= \( E[\neg q U p] \lor EXq \)

3. Biological application
SMBioNet

Bernot/Comet/Richard/Guespin 2004

- Model checking applied to gene regulatory networks
- Input
  - Regulatory network
  - Functional circuits, steady states
  - Biological properties formulated in CTL
- Output: List of compatible models (each defined by its logical parameters)

Application: *Pseudomonas aeruginosa*

- Bacteria commonly present in the environment.
- They secrete mucus only in lungs affected by cystic fibrosis (major cause of mortality).
- Bacteria isolated from cystic fibrosis’ lungs continue to grow in laboratory for many generations (mucoid phenotype).
- A majority of these bacteria present a mutation (elimination of the anti-AlgU).

Biological question

- Is the mutation the cause of the passage to the mucoid state, or could the mucoidy be induced by an epigenetic phenomenon?
- In this case, the mutation could be favored by another mechanism.

Formalization

Two cases:
1. $\theta_{11} = 1 < \theta_{21} = 2$
2. $\theta_{11} = 2 > \theta_{21} = 1$

What are the possible dynamic behaviors of this network?

Model selection

- Many possible models
- Various combinations for logical parameters
- Use model checking to find out whether there exist models satisfying certain biological properties.
- SMBioNet software

Reverse engineering

Regulatory network

Is this biological knowledge compatible with a model exhibiting multiple steady states, where one state regularly produces mucus and the other does not?
Biological properties

- **Multiple steady states**: positive circuit should be functional, i.e., its characteristic state has to be steady.
- **Temporal logic properties**
  - Mucus is produced regularly
    \[(x = 2) \implies AX AF(x = 2)\]
  - Mucus is never produced when starting in basal state
    \[(x = 0) \implies AG(\neg(x = 2))\]

Example

- 648 parameter settings at the beginning
- Snoussi constraints: 56 parameter sets resp. 38 different Kripke models
- Functionality of positive circuit: 19 models
- CTL formulas: 4 models
  - Epigenetic hypothesis is compatible with the model

References


Conclusion

- Discrete modeling
- Transition systems
- Temporal logic
- Model checking
  - Analyse all possible trajectories (≠ simulation)
  - Query the model / test properties
  - Reverse engineering
- Formal vs. numerical methods