Verification of Dynamic Properties by Model Checking

Alexander Bockmayr
FU Berlin
SS 2017

Interaction graph: topology

State transition graph: dynamics

Steady states and cycles

Model checking

1. Temporal logic
Computation Tree Logics

- Atomic formulae/properties: $p, q, r, \ldots$
- e.g. $p \equiv "gene P is on"
- Boolean operators: $\neg, \land, \lor, \Rightarrow$
- Linear time operators
  - $X_p: p$ holds next time
  - $F_p: p$ holds sometimes in the future
  - $G_p: p$ holds globally in the future
  - $p \lor q: p$ holds until $q$ holds
- Path quantifiers
  - $A$: for every path, $E$: there exists a path

Linear time operators

For all paths ...

Examples

- $EF(p \land EF\ q)$: Can $q$ hold after $p$ holds?
- $AF(AG\ p)$: 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?

Kripke model

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

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

CTL Syntax

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

CTL Semantics

Given a Kripke model $M = (S, \to, L)$, 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 $s \models \phi$ does not hold
• $s \models \phi \land \psi$ iff both $s \models \phi$ and $s \models \psi$ hold
• $s \models \phi \lor \psi$ iff $s \models \phi$ or $s \models \psi$ or both hold

CTL Semantics II

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

CTL Semantics III

• $s \models AX \phi$ iff for all $s' \in S$ with $s \to s'$ we have $s' \models \phi$.
• $s \models EX \phi$ iff for some $s' \in S$ with $s \to s'$ we have $s' \models \phi$.
• $s \models A[\phi U \psi]$ iff for all paths $s = s_1 \to s_2 \to ...$ 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 $s = s_1 \to s_2 \to ...$ there exists $i \geq 1$ such that $s_i \models \psi$ and $s_j \models \phi$, for all $1 \leq j < i$. 

22 May 2017
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:
  - If $g$ is
    - $\text{false}$: no state is labeled
    - $p$: 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)

- $\Box f$:
  - label $s$ with $\Box f$ if one of its successors is labeled with $f$.

Equivalences

- $\Box f = \neg \Diamond (\neg f)$
- $\forall f = \neg \exists f$:
- $\forall f = \neg \Diamond (\neg f)$
- $\forall f = \neg \Box (\neg f)$
- $\Box f = E[\text{true} U f]$
- $A[f U g] = \neg E[\neg g U (\neg f \land \neg g)] \land AF g$

Any CTL formula can be expressed using only the operators $\Box$, $E[\_U \_]$, and $AF$. 
Model checking algorithm (IV)

\[ \text{AF} f : \]
- 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)

\[ \text{E}[f_1 \cup f_2] : \]
- If any state is labeled with \( f_2 \), label it with E[\( f_1 \cup f_2 \)].
- Repeat: label any state with E[\( f_1 \cup f_2 \)] if it is labeled with \( f_1 \) and at least one of its successors is labeled with E[\( f_1 \cup f_2 \)], until there is no change.

Example: Input

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

Example: EU

Label with E[\( \neg q U p \)] all states which satisfy \( p \)

Example: EU (contd)

Label any state \( s \) with E[\( \neg q U p \)] if it is labeled with \( \neg q \) and at least one of its successor is already labeled with E[\( \neg q U p \)]

Example: EU (contd)

Label any state \( s \) with E[\( \neg q U p \)] if it is labeled with \( \neg q \) and at least one of its successor is already labeled with E[\( \neg q U p \)]
Example: EX
Label with EXq any state s with one of its successors already labeled with q

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

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

Example: AF (contd)
Label any state s with \( f \) if all successors of s are already labeled with \( f \)

Example: Output
\[ M = \]
All states satisfy AF(E[\neg \neg q \cup p] \lor EXq)

3. Biological application
Model checking for Thomas networks

Beriot/Comet/Richard/Guespin 2004

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

Recent software developed at FU:

PyBoolNet (Klarner 16), TREMPPI (Streck 15)

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 it be induced by an epigenetic phenomenon (stable change of phenotype without mutation)?
- In this case, the mutation could be favored later on by another mechanism (production of a toxic inhibitor complex).

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?

Formalization

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

Positive feedback loop

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
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
    \[(x = 2) \Rightarrow AX AF(x = 2)\]
  - Mucus is never produced when starting in basal state
    \[(x = 0) \Rightarrow AG(\neg (x = 2))\]

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

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

References