Verification of Dynamic Properties by Model Checking

Alexander Bockmayr
FU Berlin
SS 2014

Interaction graph: topology

\[ X_1 \in \{0, 1, 2, 3\} \]
\[ -, \theta_{13}, K_{13} \]
\[ +, \theta_{31}, K_{31} \]
\[ X_3 \in \{0, 1\} \]
\[ X_2 \in \{0, 1\} \]

State transition graph: dynamics

Stable states and cycles

Model checking

1. Temporal logic
Computation Tree Logics

- Atomic formulae: $p$, $q$, $r$, ...
- 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:
  - $A$: for every path
  - $E$: there exists a path

Linear time operators

$Xp$, $Fp$, $Gp$, $p \lor q$

CTL

- There exist different computation tree logics: CTL*, LTL, CTL, ...
- In CTL, each of the linear time operators $G$, $F$, $X$, and $U$ must be immediately preceded by a path quantifier.
- Example: $AG(EFp)$
- Mostly used CTL operators: $AG$, $AF$, $EG$, $EF$

For all paths ...

$AGp$, $AFp$

There exists a path ...

$EGp$, $EFp$

Examples

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

$EF \land \neg \Rightarrow$
Logic: Syntax and semantics

- **Syntax**
  - When a string 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

\[ M = (S, R, L) \times \]

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

CTL Syntax

- Atomic formulas in \( AP \) 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

Given a Kripke model \( M = (S, \rightarrow, L) \), a state \( s \in S \), and a CTL formula \( \phi \), the satisfaction relation

\[ M,s \models \phi \]

is inductively defined as follows:

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

CTL Semantics II

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

CTL Semantics III

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

Model checking
• M state transition graph or Kripke model
• \( \phi \) temporal logic formula
• Find all states \( s \) of \( M \) such that \( M, s \models \phi \)
• Efficient model checking algorithms and software tools exist for the logic CTL.

Equivalences

\[
\begin{align*}
AX f &= \neg EX (\neg f) \\
AG f &= \neg EF (\neg f) \\
EG f &= \neg AF (\neg f) \\
EF f &= E[true \ U f] \\
A[f U g] &= \neg E[\neg g U (\neg f \land \neg g)] \land AF g
\end{align*}
\]

Any CTL formula can be expressed using only the operators EX, EU, and AF.

Model checking algorithm

• **Input:** a CTL model \( M = (S, \rightarrow, L) \) and a CTL formula \( \phi \)
• **Output:** the set of states of \( M \) 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

- **false:** no state is labeled
- **p:** label \( s \) with \( p \) if \( p \in L(s) \)
- **f1 \land f2:** label \( s \) with \( f1 \land f2 \) if \( s \) is already labeled with both \( f1 \) and \( f2 \).
- **\neg f:** label \( s \) with \( \neg f \) if \( s \) is not already labeled with \( f \).

Model checking algorithm (III)

\( \neg EX f \):

Label \( s \) with \( EX f \) if one of its successors is labeled with \( f \).
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: Input

\[ M = \]

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

Example: EU

Label with E[\neg q \cup p] all states which satisfy p

Example: EU (contd)

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

Example: EU (contd)

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

Label with \( \text{EX}q \) any state \( s \) with one of its successors already labeled with \( q \)

Example: \( \lor \)

Label with \( \text{g} = \text{E}\lnot q\lor \text{EX}q \) any state \( s \) already labeled with \( \text{E}\lnot q\lor \text{EX}q \)

Example: AF

Label with \( \text{f} = \text{AF} (\text{E} \lnot q \lor \text{EX}q) \) any state already labeled with \( \text{g} = \text{E}\lnot q\lor \text{EX}q \)

Example: AF (contd)

Label any state \( s \) with \( \text{f} \) if all successors of \( s \) are already labeled with \( \text{f} \)

Example: Output

\[ M = \]

All states satisfy \( \text{AF} (\text{E} \lnot q \lor p) \lor \text{EX}q \)

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) reverse engineering
• Software package: http://smbionet.lami.univ-evry.fr

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.

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?

M. Delbrück (Nobel prize, 1969)

„Many systems in steady state can exhibit several different stable states under identical conditions. They can be shifted from one stable state to another by transient perturbations.“ (CNRS Symposium, Paris, 1948)

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

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

- Gene interaction graph
- State transition graph
  - asynchronous
  - non-deterministic
- Temporal logic
- Model checking
  - Querying (\neq simulation)
  - Reverse engineering

References