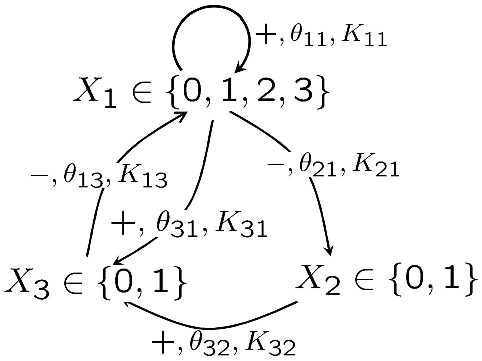
Verification of Dynamic Properties by Model Checking

Alexander Bockmayr

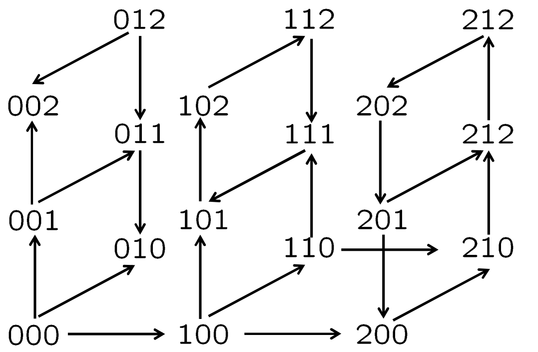
FU Berlin

SS 2017

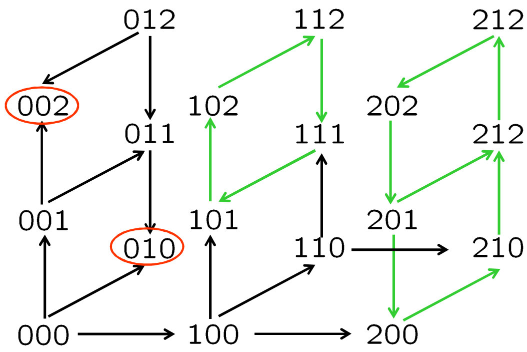
Interaction graph: topology



State transition graph: dynamics

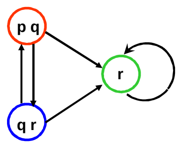


Steady states and cycles

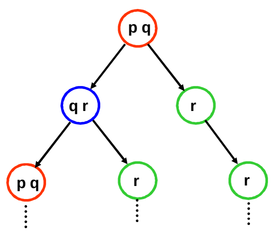


Model checking

Transition system or Kripke model



Infinite computation tree



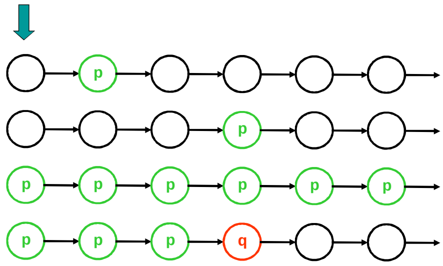
check dynamics properties using *some* temporal logic

1. Temporal logic

Computation Tree Logics

- Atomic formulae/properties: p, q, r, …
  e.g. p ≡ "gene P is on"
- Boolean operators: ¬ , ∧ , ∨ , ⇒
- 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 U q : p holds until q holds
- Path quantifiers
  A : for every path, E : there exists a path

Linear time operators



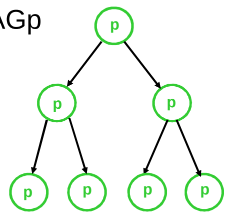
Xp
Fp
Gp
pUq

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 (EF p)
- Mostly used CTL operators :
  AG, AF, EG, EF

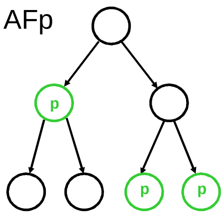
For all paths …

AGp



always

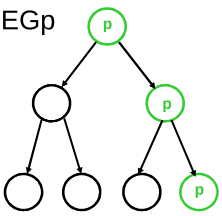
AFp



sometimes

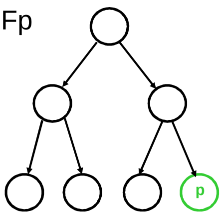
There exists a path …

EGp



always

EFp



sometimes

Examples

- EF(p ∧ EF q): Can q hold after p holds ?
- AF(AG p): Must the system reach a state where p holds forever ?
- EG((p ⇒ EF(¬ p)) ∧ (¬ p ⇒ 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 ?

22 May 2017

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.

22 May 2017

CTL Syntax

- Atomic formulas are CTL formulas.
- If ϕ and ψ are CTL formulas, then

  $\neg \phi$, $\phi \wedge \psi$, $\phi \vee \psi$,

  AF ϕ, EF ϕ,

  AG ϕ, EG ϕ,

  AX ϕ, EX ϕ,

  A[ϕ U ψ], E[ϕ U ψ]

are CTL formulas.

22 May 2017

CTL Semantics

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

$M,s \vDash \phi$ (shortly $s \vDash \phi$)

is inductively defined as follows:

- $s \vDash p$ iff $p \in L(s)$
- $s \vDash \neg \phi$ iff $s \vDash \phi$ does not hold
- $s \vDash \phi \wedge \psi$ iff both $s \vDash \phi$ and $s \vDash \psi$ hold
- $s \vDash \phi \vee \psi$ iff $s \vDash \phi$ or $s \vDash \psi$ or both hold

22 May 2017

CTL Semantics II

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

22 May 2017

CTL Semantics III

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

22 May 2017

2. Model checking algorithm

Model checking

- M Kripke model (or transition system)
- ϕ temporal logic formula
- Find all states s of M such that

  $s \vDash \phi$
- Efficient model checking algorithms and software tools exist for the logic CTL.

Equivalences

- AX f = ¬ EX (¬ f)
- AG f = ¬ EF (¬ f)
- EG f = ¬ AF (¬ f)
- EF f = E[**true** U f]
- A[f U g] = ¬ E[¬ g U (¬ f ∧ ¬ g)] ∧ AF g

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

Model checking algorithm

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

Model checking algorithm (II)

- Let g be a subformula of ϕ and suppose all immediate subformulas of g have already been labeled.
- Determine states to be labeled by g a follows:

If g is

✓ false : no state is labeled

✓ p : label s with p if p ∈ L(s)

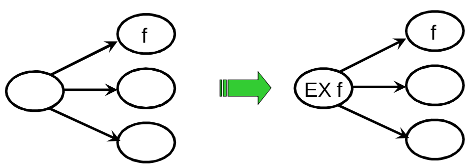
✓ $f_1 \wedge f_2$: label s with $f_1 \wedge f_2$ if s is already labeled with both f_1 and f_2.

✓ ¬ f : label s with ¬ f if s is not already labeled with f.

Model checking algorithm (III)

✓EX f :

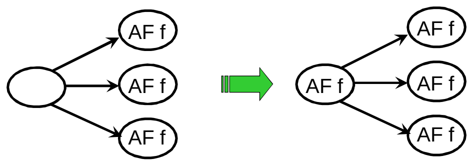
label s with EX f if one of its successors is labeled with f.



Model checking algorithm (IV)

✓ AF f :

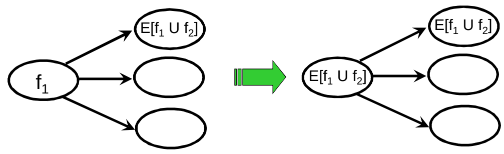
- If any state is labeled with f, label it with AF f.
- Repeat : label any state with AF f if all successor states are labeled with AF f, until there is no change.



Model checking algorithm (V)

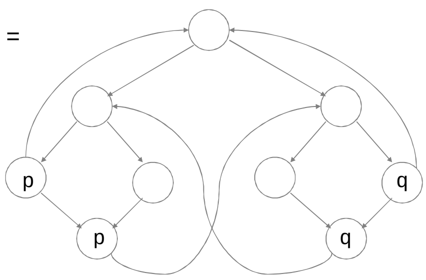
✓ $E[f_1 \text{ U } f_2]$:

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



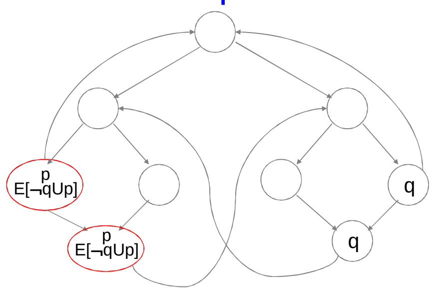
Example: Input

M =



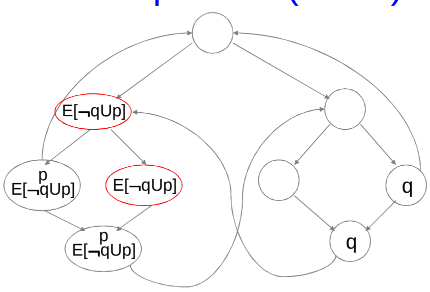
ϕ = AF(E[¬q U p] v EXq)

Example: EU



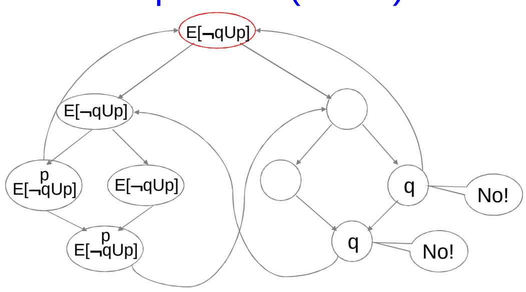
Label with E[¬qUp] all states which satisfy p

Example: EU (contd)



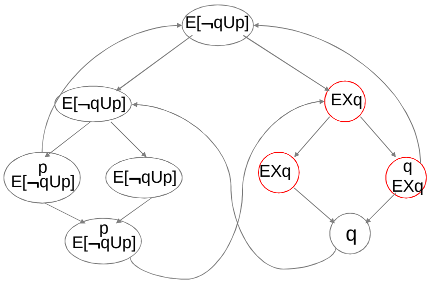
Label any state s with E[¬qUp] if it is labeled with ¬q and at least one of its successor is already labeled with E[¬qUp]

Example: EU (contd)



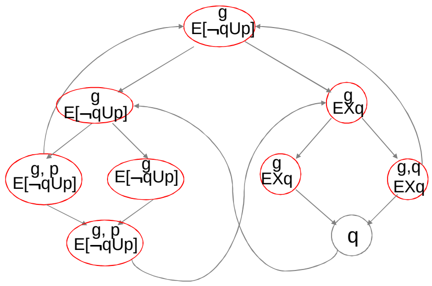
Label any state s with E[¬qUp] if it is labeled with ¬q and at least one of its successor is already labeled with E[¬qUp]

Example: EX



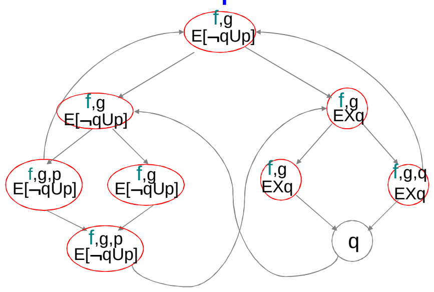
Label with EXq any state s with one of its successors already labeled with q

Example: ∨



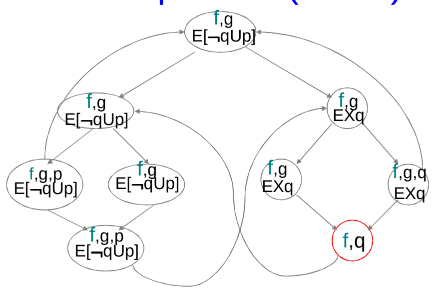
Label with g= E[¬qUp] v EXq any state s already labeled with E[¬qUp] or EXq

Example: AF



Label with f= AF(E[¬qUp]vEXq) any state already labeled with g= E[¬qUp]vEXq

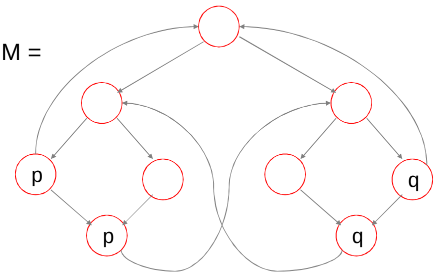
Example: AF (contd)



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

Example: Output

M =



All states satisfy AF(E[¬q U p] v EXq)

3. Biological application

Model checking for Thomas networks

Bernot/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) ➡ reverse engineering
- Software package: SMBioNet (Richard 04)
- Recent software developed at FU: PyBoolNet (Klarner 16), TREMPPI (Streck 15)

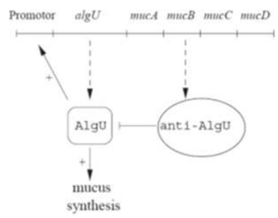
Application : *Pseudomonas aeruginosa*

- 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

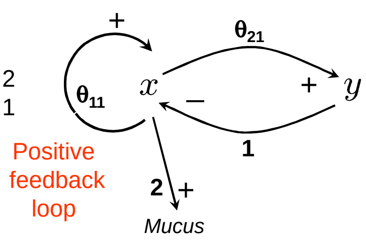


Is this biological knowledge compatible with a model exhibiting multiple steady states, where one state regularly produces mucus and the other does not ?

Formalization

Two cases :

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



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

- E. Clarke, O. Grumberg, D.A. Peled: Model checking, MIT Press, 2000
- M. R. A. Huth, M. D. Ryan: Logic in Computer Science, Cambridge Univ. Press, 2000
- G. Bernot, J.-P. Comet, A. Richard and J. Guespin: Application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic. J. Theor. Biol. 229, 339-347, 2004
- H. Klarner: Contributions to the Analysis of Qualitative Models of Regulatory Networks, Dissertation, FU Berlin, 2014
- A. Streck: Toolkit for Reverse Engineering of Molecular Pathways via Parameter Identification, Dissertation, FU Berlin, 2015