IA symbolique pour les réseaux biologiques complexes

GB5 BIMB – année 2023–2024

Symbolic AI for
Complex Regulatory Networks (n.2)

Jean-Paul Comet

Université Côte d'Azur

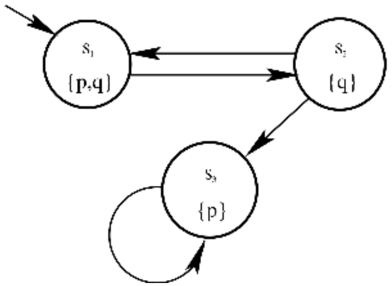
6 décembre 2023

Plan

A model ≡ a Kripke Structure

A Kripke Structure is a triple $K = (S, R, L)$ where :

- S is the set of possible states,
- $R \subseteq V \times V$ is a relation between states,
- $L : S \rightarrow 2^A$ is a Labelling function which associates with each state the subset of atomic formulas (A) satisfied at the considered state.



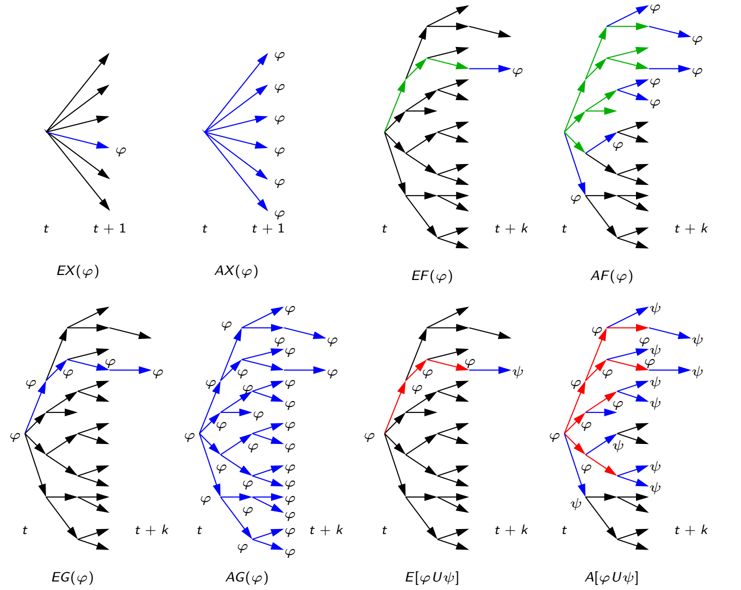
CTL ≡ Computational Tree Logic

- Atoms = comparisons : $(x = 2), (y > 0) \ldots$
  Connectors = $(\varphi_1 \wedge \varphi_2), (\varphi_1 \Rightarrow \varphi_2) \ldots$
  Modalities = made of 2 letters :

First letter	Second lettre
A = for All path choices	X = neXt state F = for some Future state
E = there Exists a choice	G = for all future state (Globally) U = Until

- Examples :
  AX (y=1) : the concentration of y equals 1 in all possible next state.
  EG(x=0) : there exists at least one path starting from the initial state along which x is constantly/globally zero.

Modalities Semantics on paths (1)



CÔTE D'AZUR
IA symbolique & réseaux Biol.
Jean-Paul Comet
Introduction to model checking
Reminder of CTL
Equivalences
Choice : AF, EU, EX
AF, EU, EX
Pseudo-code

Modalities Semantics on paths (2)

Let s_0 be a state. The CTL semantics is defined inductively :

- $s_0 \models \top$ and $s_0 \not\models \bot$ $\quad \forall p \in AP, s_0 \models p$ **iff** $p \in L(s_0)$,
- $s_0 \models \neg\varphi$ **iff** $s0 \not\models \varphi$,
- $s_0 \models \varphi_1 \wedge \varphi_2$ (resp. $\varphi_1 \vee \varphi_2$) **iff** $s_0 \models \varphi_1$ **and** (resp. **or**) $s_0 \models \varphi_2$,
- $s_0 \models \varphi_1 \Rightarrow \varphi_2$ **iff** $s_0 \not\models \varphi_1$ **or** $s_0 \models \varphi_2$,
- $s_0 \models \varphi_1 \Leftrightarrow \varphi_2$ **iff** $s_0 \models (\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1)$,
- ...

CÔTE D'AZUR
IA symbolique & réseaux Biol.
Jean-Paul Comet
Introduction to model checking
Reminder of CTL
Equivalences
Choice : AF, EU, EX
AF, EU, EX
Pseudo-code

Modalities Semantics on paths (2)

Let s_0 be a state. The CTL semantics is defined inductively :

- ...
- $s_0 \models AX\varphi$ **iff** for all successors s_1 of s_0, $s_1 \models \varphi$,
- $s_0 \models EX\varphi$
  iff there exists a successor s_1 of s_0 such that $s_1 \models \varphi$,
- $s_0 \models AG\varphi$
  iff $\forall s_i$ of each path $s_0 s_1 \dots s_i \dots$, one has $s_i \models \varphi$,
- $s_0 \models EG\varphi$
  iff $\exists$ a path $s_0 s_1 \dots s_i \dots$, tq $\forall s_i$, one has : $s_i \models \varphi$,
- $s_0 \models AF\varphi$ **iff** $\forall$ path $s_0 s_1 \dots s_i \dots$, $\exists j$ s.t. $s_j \models \varphi$,
- $s_0 \models EF\varphi$ **iff** $\exists$ a path $s_0 s_1 \dots s_i \dots$, $\exists j$ s.t. $s_j \models \varphi$,
- $s_0 \models A[\varphi_1 U \varphi_2]$ **iff** $\forall$ path $s_0 s_1 \dots s_i \dots$, $\exists j$ s.t.
  $s_j \models \varphi_2$ and $\forall i < j, s_i \models \varphi_1$,
- $s_0 \models E[\varphi_1 U \varphi_2]$ **iff** $\exists$ path $s_0 s_1 \dots s_i \dots$, $\exists j$ s.t.
  $s_j \models \varphi_2$ and $\forall i < j, s_i \models \varphi_1$

CÔTE D'AZUR
IA symbolique & réseaux Biol.
Jean-Paul Comet
Introduction to model checking
Reminder of CTL
Equivalences
Choice : AF, EU, EX
AF, EU, EX
Pseudo-code

Some equivalences [Ryan2000]

- $\neg AX\phi \equiv EX\neg\phi$
- $\neg AF\phi \equiv EG\neg\phi$
- $AF\phi \equiv A[\top U \phi]$
- $\neg EF\phi \equiv AG\neg\phi$
- $EF\phi \equiv E[\top\ U\ \phi\ ]$
- $A[pUq] \equiv \neg(E[\neg q U(\neg p \wedge \neg q)] \vee EG\neg q)$

We will use only connectors in $\{\bot, \top, \wedge, \neg, \begin{matrix} AF \\ EG \\ AU \end{matrix}, \begin{matrix} EX \\ AX \end{matrix}, EU\}$

Morgan laws :

$$\neg(A \wedge B) \equiv (\neg A) \vee (\neg B)$$
$$\neg(A \vee B) \equiv (\neg A) \wedge (\neg B)$$

Pseudo-code of function translate(φ)

if $\phi \equiv p$ (atom),		return	ϕ
if $\phi \equiv \top$		return	$\neg\bot$
if $\phi \equiv \bot$		return	ϕ
if $\phi \equiv \neg\psi$		return	$\neg$tr(ψ)
if $\phi \equiv \psi \wedge \psi'$		return	tr(ψ)$\wedge$ tr(ψ')
if $\phi \equiv \psi \vee \psi'$		return	$\neg$ ($\neg$tr(ψ)$\wedge$ $\neg$tr(ψ'))
if $\phi \equiv \psi \Rightarrow \psi'$		return	$\neg$ (tr(ψ)$\wedge$ $\neg$tr(ψ'))
if $\phi \equiv AX\psi$		return	$\neg EX\neg$tr(ψ)
if $\phi \equiv EX\psi$		return	EX tr(ψ)
if $\phi \equiv AF\psi$		return	AF tr(ψ)
if $\phi \equiv EF\psi$		return	E[$\neg\bot$ U tr(ψ)]
if $\phi \equiv AG\psi$		return	$\neg E$[$\neg\bot$ U $\neg$ tr(ψ)]
if $\phi \equiv EG\psi$		return	$\neg AF$[$\neg$ tr(ψ)]
if $\phi \equiv A[\psi U \psi']$	if $\psi = \top$	return	AF tr(ψ')
	else	return	$\neg E$[$\neg$ tr(ψ') U ($\neg$ tr(ψ) $\wedge$ $\neg$ tr(ψ'))] $\wedge AF$ tr(ψ')
if $\phi \equiv E[\psi U \psi']$		return	E[tr(ψ) U tr(ψ')]