1. CRN Syntax

Let $S = \{x_1, \dots, x_s\}$ be a finite set of molecular species names.

Def. A reaction is a quadruple (R, I, P, f), also noted $R \;/\; I \xrightarrow{f} P$
where R (resp. I, P) is a multiset of reactant species (resp. inhibitor, product species)
and $f: \mathbb{R}_+^S \to \mathbb{R}_+$ is a rate function (kinetic expression).

- Multisets are represented by linear expressions with integer stoichiometric coefficients
- A reaction catalyst is a molecular species that is both a reactant and a product (can also be an inhibitor).

Def. A CRN is a finite set of reactions.

E.g. reactions with

- Mass action law kinetics $\sum_j n_j \times x_j \xrightarrow{k \times \prod_j x_j^{n_j}} p$
- Michaelis-Menten kinetics $x \xrightarrow{V \times x/(K+x)} y$
- Hill kinetics $x \xrightarrow{V \times x^n/(K^n+x^n)} y$

Negative Hill kinetics $\emptyset/x \xrightarrow{V/K^n+x^n} y$

Well-formed Reactions

Def. A reaction (R, I, P, f) is well-formed if

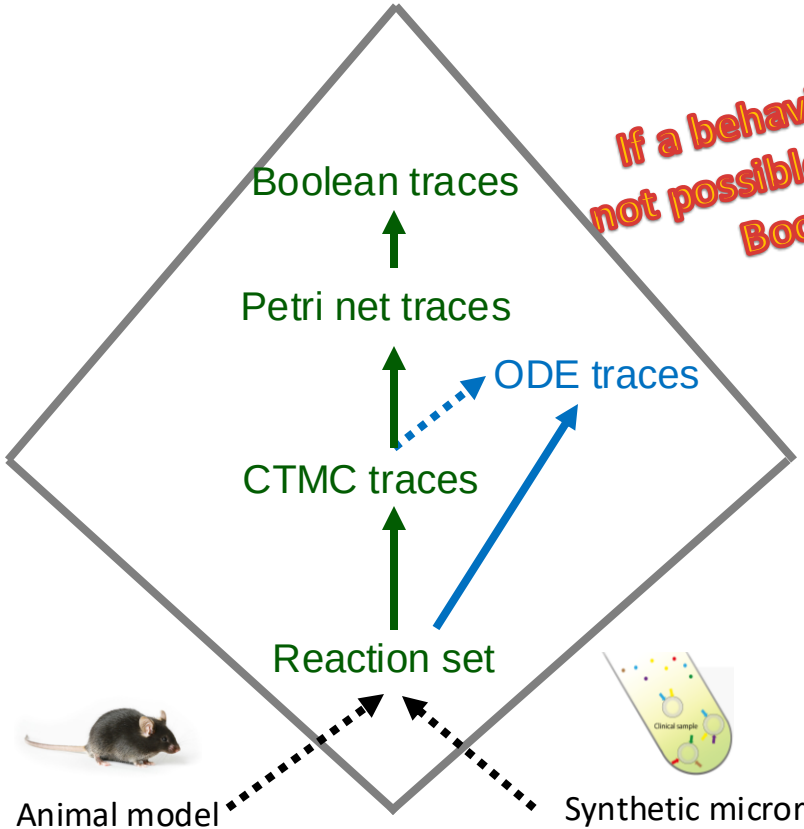
- $f: \mathbb{R}_+^S \to \mathbb{R}_+$ is a partially differentiable function
- $x_i \in R$ if and only if $\frac{\partial f}{\partial x_i}(x) > 0$ for some value $x \in \mathbb{R}_+^S$
- $x_i \in I$ if and only if $\frac{\partial f}{\partial x_i}(x) < 0$ for some value $x \in \mathbb{R}_+^S$.

Def. A reaction is strict if $R(x_i) > 0$ implies $f(x_1, \ldots, x_s) = 0$ whenever $x_i = 0$.

Prop. The ODE associated to a well-formed and strict reaction system (CRN) defines a positive system.

Fages, Gay, Soliman. Inferring Reaction Systems from Ordinary Differential Equations. *Theoretical Computer Science*, 599:64–78, 2015.

Hierarchy of Semantics



If a behavior is not possible in the Boolean semantics, it is not possible in the stochastic semantics for any reaction rates. Boolean model-checking can exhibit rare events

Thm. (abstract interpretation ↑) Galois connections between the domains of syntactical, stochastic, Petri net and Boolean trace semantics

Fages, Soliman. Abstract Interpretation and Types for Systems Biology. *Theoretical Computer Science*, 403(1):52–70, 2008.

Thm. (approximation ⋮) [Gillespie 1971 Kurtz 1978, 1992] When the volume tends to infinity the ODE trace approximates the *mean* stochastic trace

Thm. (equality) [Buscemi Fages CMSB 2024] Under graphical conditions on the ancestors of polymolecular reactions, the ODE trajectory equals the *mean* stochastic trace.

2. Computable Real Numbers and Functions

Classical definitions of *computable analysis* based on Turing machines

Definition. A real number *r* is computable if there exists a Turing machine with

Input: precision $p \in \mathbb{N}$

Output: rational number $q \in \mathbb{Q}$ with $|r-q| < 2^{-p}$

Examples. Rational numbers, limits of computable Cauchy sequences ($\lim_{p,q\to\infty} |x_p - x_q| = 0$), π, e, …

Definition. A real function $f:\mathbb{R}\to\mathbb{R}$ is computable if there exists a Turing machine that computes *f(x)* with an oracle (Turing machine) for *x*.

Examples. Polynomials, trigonometric functions, analytic functions ($f(x) = \sum_{n=0}^{+\infty} a_n (x - x_0)^n$) …

Counter-examples. x=0, $\lceil x \rceil$ are not computable (undecidable on x=0.000…) discontinuous functions are not computable

Decision problem $w \in \mathcal{L}$: analog encoding by a real function $f:\mathbb{R}\to\mathbb{R}$?

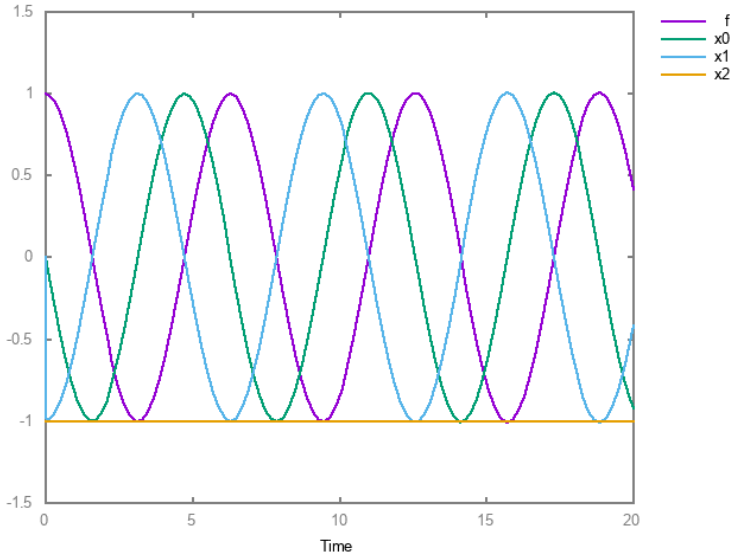
Input encoding *e*: $\mathcal{L} \to \mathbb{R}$ problem encoding by f: accept w if f(e(w)) >1 reject if <-1

General Purpose Analog Computer [Shannon 1941]

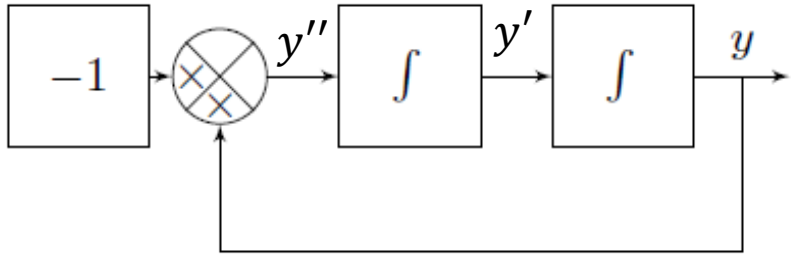
Shannon's formalization of the Differential Analyser by GPAC circuits

A time function if GPAC-generated if it is the output of some unit of a GPAC circuit built from:

1. Constant unit
2. Sum unit
3. Product unit
4. Integral $\int y\,dx$ unit (dt by default)



What does this GPAC circuit compute ?



$$y' = \frac{dy}{dt}$$

$$y'' = \frac{d^2y}{dt^2} = -y$$

$$y(t) = cos(t) \quad y'(t) = -sin(t)$$

if $y(0) = 1, y'(0) = 0$

CRN Implementation of GPAC Units

Mass action law kinetics reaction network with output concentration stabilizing on the result of the operation applied to the input concentrations

Positive constant units: molecular concentrations

Product unit $z = x.y$

$$x + y \xrightarrow{k.x.y} x + y + z$$
$$z \xrightarrow{k.z} \_$$
$$\frac{dz}{dt} = k(xy - z)$$
$$= 0 \text{ when } z = x.y$$

Sum unit $z = x + y$

$$x \xrightarrow{k.x} x + z$$
$$y \xrightarrow{k.y} y + z$$
$$z \xrightarrow{k.z} \_$$
$$\frac{dz}{dt} = k(x + y - z)$$
$$= 0 \text{ when } z = x + y$$

Time integral $z = \int x\, dt$ unit

$$x \xrightarrow{x} x + z$$
$$\frac{dz}{dt} = x$$
$$z = \int_0^T x\, dt$$

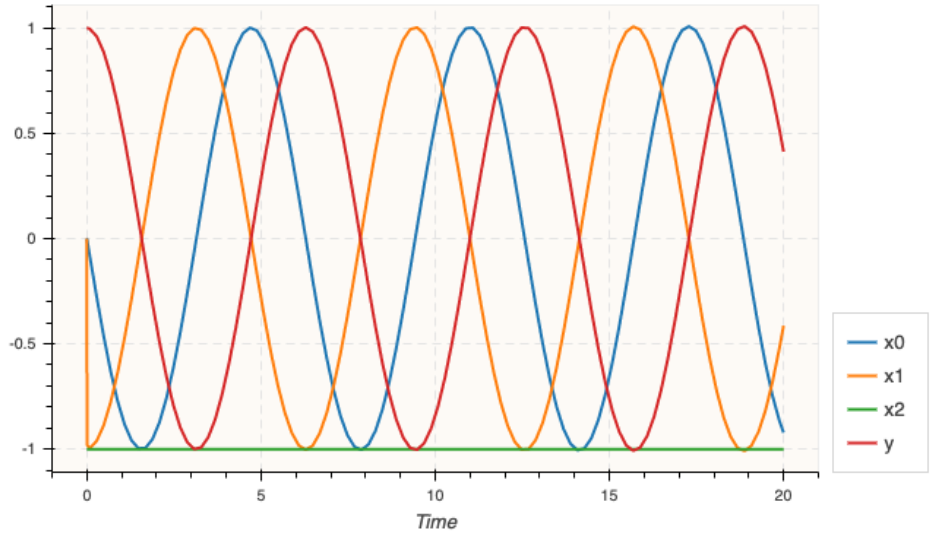
Polynomial ODE Initial Value Problems (PIVP)

Graça and Costa 2003's formalization of GPAC generated functions

Definition. A real time function $f:R_+ \to R$ is PIVP-generable iff there exist a vector of polynomials $p \in R^n[R^n]$ and of initial values $y(0) \in R^n$

and a solution function $y:R_+ \to R^n$ such that $y'(t)=p(y(t))$ and $\mathbf{f(t)=y_1(t)}$

Example. y=cos(t)



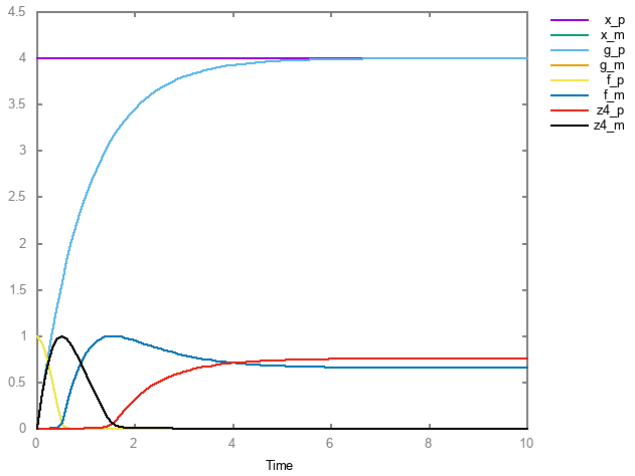
Closure properties:

f+g, f-g, f.g, 1/f, ,f ◦g, y s.t. y′ =f(y) are GPAC-generable if f, g are.

PIVP-Computable Function f(x)

Definition. [Graça Costa 03 J. Complexity] A real function f:R→R is PIVP-computable if there exists vectors of polynomials $p \in R^n[R^n]$ and $q \in R^n[R]$ and a function y: $R^n \to R^n$ such that $y'(t)=p(y(t))$, $\mathbf{y(0)=q(x)}$ and $\mathbf{|y_1(t)-f(x)|<y_2(t)}$ with $y_2(t) \geq 0$ decreasing for t>1 and $\lim_{t\to\infty} y_2(t) = 0$

Example. y=cos(4)



Reconciles Digital and Analog Computation !

Theorem (analog characterization of Turing computability).

[Bournez Campagnolo Graça Hainry 07 J. Complex]

A real function is computable (by Turing machine) iff it is PIVP-computable.

Normal Form Theorem

Theorem (abstract CRN normal form)

A real function is computable if and only if it is computable by a system of elementary reactions of the form

_ => z or x => x+z or x+y => x+y+z

plus annihilation reactions x+y => _ all with mass action law kinetics

Realistic CRN:

- formal annihilations by complexations (e.g. in a stable inactive complex)
- formal syntheses by modifications (e.g. phosphorylation with kinases)

Concrete CRN: search mapping with real enzymes (e.g. Brenda database)

- Easier for CRN with rate independence property
- Robustness w.r.t. parameter perturbations (extrinsic variability)
- Robustness w.r.t. stochastic simulations (intrinsic variability)

5. Logical Gates

Assuming concentrations in [0, 1]

And: C = A /\ B

[C] = min([A],[B]) `A+B => C` (destructive on A, B, rate-independent)

or

[C]=[A]*[B] $\frac{dC}{dt} = A * B - C$ (non-destructive on A, B)

```
MA(k) for A+B => A+B+C
MA(k) for C => _
```

(any rate constant k but the same for both reactions)

Or: C = A \/ B

[C]=[A]+[B]-[A]*[B] $\frac{dC}{dt} = A + B - A * B - C$ (non-destructive on A, B)

```
MA(k) for A => A+C
MA(k) for B => B+C
k*A*B for A+B+C => A+B
MA(k) for C => _
```

(not well-formed, should use C+ C-)

Not: C = ¬ A

[C]=1-[A] $\frac{dC}{dt} = 1 - A - C$

```
k for - => C
k*A for A+C => A
MA(k) for C => _
```

(not well-formed, should use C+ C-)

1. Chemical Reaction Kinetics

Molecular species: $A_1, \ldots, A_m$

|A|=Number of molecules A

[A]=Concentration of A in the solution: [A] = |A| / Volume

dimension volume^{-1} , e.g. unit ML^{-1}, *noted also A by abuse of notation*

Molecular solution: multiset of molecules S, S', …

linear expression with stoichiometric coefficients $S = k1 * A_1 + \ldots + c_n * A_n$

Reaction: multiset rewriting rule given with a rate function f for S => S'

Rate function f gives the number of reactions per time and volume units: dimension volume^{-1}time^{-1}

determines the velocity of our « chemical computer »

Well-formed reaction: $A \in S \iff \frac{\partial f}{\partial A} \neq 0$ (catalyst if >0, inhibitor if <0) and $A \in S \wedge A = 0 \Rightarrow f(A) = 0$

Reaction Rate Functions

Mass action law kinetics (proportionality)

k*A for A => B

k*A*B for A+B => C

k*A^m*B^n for m*A + n*B => R

Guldberg and Waage, 1864

Henri-Michaelis-Menten kinetics (saturation)

Vm*A/(Km+A) for A => B

Victor Henry (X) 1903

Michaelis and Menten 1913

Hill kinetics (cooperativity, sigmoid velocity)

Vm*A^n/(Km+A^n) for A => B

Archibald Hill 1910

Origin and justification of these other rate functions?

By model reduction of a detailed mass action CRN, e.g. by elimination of the enzyme variables

ODE Semantics of a CRN

To a set of species $\{A_1, \ldots, A_m\}$ with real valued concentrations
and a set of reactions $\{ f_j \text{ for } l_j \Rightarrow r_j \}_{j=1,\ldots,n}$ given with rate functions f_j

one associates the Ordinary Differential Equations (ODE) over $\{A_1, \ldots, A_m\}$

$$dA_i/dt = \Sigma^n_{j=1} f_j \,.\, (r_j(A_i) - l_j(A_i)) = \Sigma^n_{j=1} f_j \,.\, v_j(A_i)$$

where
- $l_j(A_i)$ is the stoichiometric coefficient of A_i in l_j
- $r_j(A_i)$ is the stoichiometric coefficient of A_i in r_j
- $v_j = r_i - l_j$ is the net stoichiometric change vector of reaction j
- f_j is the rate function of dimension volume^{-1} time^{-1}

In matrix form: $\dot{A} = V.f(A)$

$$\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ & \ddots & \\ \vdots & & \\ v_{m1} & & v_{mn} \end{pmatrix} . \begin{pmatrix} f_1 \\ \cdots \\ f_n \end{pmatrix}$$

Variable Elimination by Conservation Laws

$E+S \xrightarrow{k1} C \xrightarrow{k3} E+P$

$E+S \xleftarrow{k2} C$

$dE/dt = -k1.E.S+(k2+k3).C$

$dS/dt = -k1.E.S+k2.C$

$dC/dt = k1.E.S-(k3+k2).C$

$dP/dt = k3.C$

A conservation law is a set of species {Mi} than remains with same total amount

i.e. a *Petri net place invariant*, or equivalently a *structural ODE invariant* $\Sigma^{n}_{i=1} dMi/dt = 0$

Here two invariants: $E+C=E_0+C_0$,

$S+C+P=S_0+C_0+P_0$

We can thus eliminate variables **$E= E_0+C_0–C$** and $P= S_0+C_0+P_0-C-P$

and get the algebra-differential system **$E= E_0–C$** assuming $C_0=0$, $P_0=0$,

$dS/dt = -k1.(E_0 - C).S + k2.C$

$dC/dt = k1.E_0.S - (k1.S+k2+k3).C$

Model Reduction by Quasi-Steady State Approximation (QSSA)

After short time assume $dC/dt \simeq 0 \simeq k1E_0S-(k1S+k2+k3)C$

Then $C = k1E_0S/(k1S+k2+k3)$

$= E_0S/(S+(k2+k3)/k1)$

$= E_0S/(K_m+S)$ with $K_m=(k2+k3)/k1$

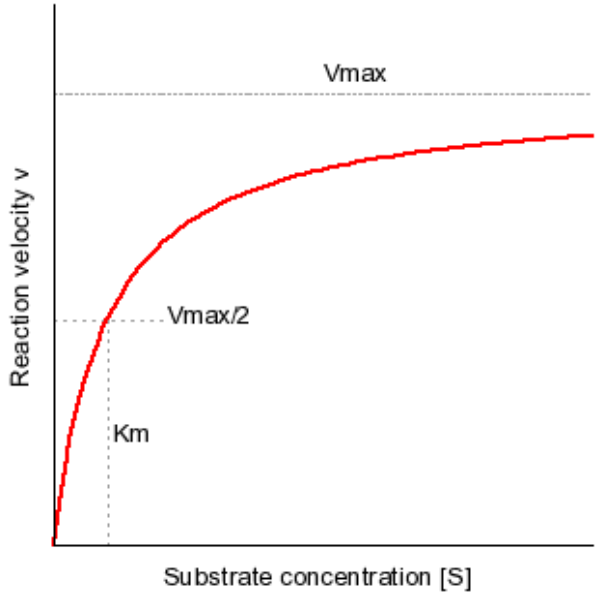
K_m *is substrate concentration with half maximum velocity*

We get $dP/dt = -dS/dt = -k1(E_0-C)S+k2C$

$= -k1E_0S + (k1S+k2)\ E_0\ S\ /\ (K_m+S)$

$= V_m\ S\ /\ (K_m+S)$ where $V_m= k3E_0$

V_m *is maximum velocity at saturing substrate concentration*



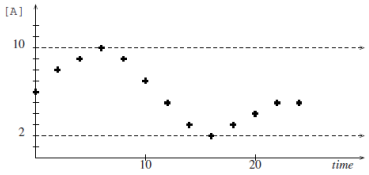
Michaelis-Menten kinetics: **$V_m\ S/(K_m+S)$ for S => P** *justified when* ***E<<S***

C and E are eliminated but well sometimes E can be re-injected as a slow variable in Vm …

k3*E*S /(Km+S) for S+E => E+P

Closed LTL($\mathbb{R}$) Formulae over Finite Traces

A trace (of experiment or simulation) gives concentration values at discrete time points:



State variables: concentrations $A, B, \ldots$, possibly real time *Time*
Arithmetic expressions over state variables (no free variable in closed formulae)
Temporal operators of LTL: **X** next, **F** finally, **G** globally, **U** until, **R** release.

Reachability of minimum value: $\mathbf{F}(A > 0.2)$
Global minimum value: $\mathbf{G}(A > 0.2)$
Reachability of global minimum value: $\mathbf{FG}(A > 0.2)$
Peak: $A < 0.4 \wedge \mathbf{F}(A > 0.4 \wedge \mathbf{F}(A < 0.4))$
Curve fitting: $\mathbf{F}(\mathit{Time} == 1 \wedge M == 8.1 \wedge \mathbf{F}(\mathit{Time} == 2 \wedge M == 9 \wedge \ldots$
$==$ provides equality between discrete time points by interpolation

Semantics of Closed LTL($\mathbb{R}$) over Infinite Traces

Completion of finite traces with an infinite loop on the last state.

$\pi \models \phi$ for a closed proposition ϕ if ϕ holds in the first state of π
$\pi \models \mathbf{X}\phi$ if $\pi^1 \models \phi$
$\pi \models \mathbf{F}\phi$ if $\exists k \geq 0\ \pi^k \models \phi$
$\pi \models \mathbf{G}\phi$ if $\forall k \geq 0\ \pi^k \models \phi$
$\pi \models \phi\ \mathbf{U}\ \psi$ if $\exists k \geq 0\ \pi^k \models \psi \wedge \forall j < k\ \pi^j \models \phi$
$\pi \models \phi\ \mathbf{R}\ \psi$ if $\forall k \geq 0\ \pi^k \models \psi \vee \exists j < k\ \pi^j \models \phi$
ϕ releases ψ if ψ is always true or until ϕ becomes true

Duality:
$\neg\mathbf{X}\phi = \mathbf{X}\neg\phi$,
$\neg\mathbf{F}\phi = \mathbf{G}\neg\phi$,
$\neg(\phi\ \mathbf{U}\ \psi) = \neg\phi\ \mathbf{R}\ \neg\psi$.

First-Order FO-LTL($\mathbb{R}_{lin}$) Constraints with Free Variables

- Free variables $x, y, \ldots$ in addition to state variables $A, B, \ldots$
- Linear constraints over free and state variables as atomic propositions
- Logical quantifiers $\forall x \ \exists y$
- Temporal operators: **X**, **F**, **G**, **U** , **R**

maximum(A,x): $\mathbf{G}(A <= x) \wedge \mathbf{F}(A >= x)$

local_maximum(A,x): $\mathbf{F}(A < x \wedge \mathbf{X}(A >= x \wedge \mathbf{X}(A <= x)))$

decrease(A): $\exists x \ A >= x \wedge \mathbf{X}(A < x)$

peak(A,x,t)): $A < x \wedge \mathbf{X}(A >= x \wedge \mathbf{X}(A <= x) \wedge Time = t)$

Minimal Set of CTL* Operators

Minimal set of operators:

- Logical connectives: $\vee$, $\neg$
- Path quantifier: **E** "exists"
- Temporal operators: **X** "next", **U** "until"

Other operators defined by abbreviations:

$\boldsymbol{\phi} \wedge \boldsymbol{\psi} = \neg(\boldsymbol{\phi} \vee \boldsymbol{\psi})$

$\boldsymbol{\phi} \Rightarrow \boldsymbol{\psi} = \neg\, \boldsymbol{\phi} \vee \boldsymbol{\psi}$

Aϕ **=** $\neg$ **E** $\neg$ ϕ "always"

Fϕ **=** true **U** ϕ "finally"

Gϕ **=** $\neg$ **F** $\neg$ ϕ "globally"

ϕ_1 **R** ϕ_2 = $\neg$ **(** $\neg$ ϕ_1 **U** $\neg$ ϕ_2**)** "release"

CTL Fragment of CTL*

In CTL fragment, each temporal operator must be preceded by a path quantifier

Basis of three operators: **EX, EG, EU**

- **EF** ϕ = **E**(true **U** ϕ) $s \models \mathbf{EF}\ \phi$ if $\exists\ \pi$ from s $\exists\ k \geq 0$ $\pi^k \models \phi$
- **AX** $\phi = \neg\ \mathbf{EX} \neg\ \phi$ $s \models \mathbf{AX}\ \phi$ if $\forall\ \pi$ from s $\pi^1 \models \phi$
- **AF** $\phi = \neg\ \mathbf{EG} \neg\ \phi$ $s \models \mathbf{AF}\ \phi$ if $\forall\ \pi$ from s $\exists\ k \geq 0$ $\pi^k \models \phi$
- **AG** $\phi = \neg\ \mathbf{EF} \neg\ \phi$ $s \models \mathbf{AG}\ \phi$ if $\forall\ \pi$ from s $\forall\ k \geq 0$, $\pi^k \models \phi$
- Etc…

Any CTL formula is thus a state formula
and can be identified to the set of states that satisfy it

$$\phi \simeq \{s \in S : s \models \phi\}$$ [Emerson 90]

LTL Fragment of CTL*

Linear Time Logic (LTL) formulae are of the form **A**ϕ (noted just ϕ without the **A**) where ϕ contains no path quantifier, only temporal operators: **X, U** and their duals

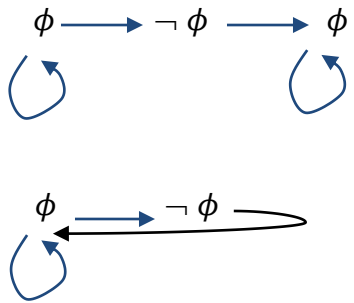
- The LTL formula **FG** ϕ is not expressible in CTL

  Stronger CTL formula ? ***AF(AG ϕ)***

  false on $\phi \rightarrow \neg\phi \rightarrow \phi$

  Weaker CTL formula ? ***AF(EG ϕ)***

  true on $\phi \rightarrow \neg\phi$



- The CTL formula **EF**(**AG** ϕ) **AF**(**AG** ϕ) are not expressible in LTL
- LTL and CTL are strict fragments of CTL*

Biochemical Reachability Properties in CTL (from some initial state)

Initial state = initial biological conditions = molecules present / absent (/ undetermined)

- Can the cell produce some protein P (from initial state) ?
  - **EF(**P**)** ≜ **reachable(**P**)**
- Can the cell produce P, Q and not R?
  - **reachable(**P^Q^¬R**)**

About *pathways*:

- Can the cell reach a given set s of states while passing by another set of states s_2?
  - **EF(**s_2^**EF**s**)**
- Is it possible to produce P without Q before ?
  - **E(**¬Q **U** P**)**
- If not, this gives a *phenomenological non-causal notion of checkpoint*
  - ¬**E(**¬s_2 **U** s**)** ≜ **checkpoint(**s_2,s**)**

Cum hoc sed non propter
Correlation is not causality

Biochemical Reachability Properties in CTL (from some initial state)

- Is a given set of states `s` a stable state set *(infinite loop with no escaping possibility)*?
  - **`stable(s)` ≜ `AG(s)`**
- Is `s` a steady state *(infinite loop with escaping possibility)* ?
  - **`steady(s)` ≜ `EG(s)`**

- Can the cell reach a given stable state `s`?
  - **`reachable(stable(s))`**

  alternance of path quantifiers **`EF AG`** ϕ *(not expressible in LTL)*

- Must the cell reach a given stable state `s`?
  - **`AF(stable(s))`**

- What are the stable states?
  - *Not expressible in CTL.*

  needs to combine CTL with enumeration, see Biocham `generate_ctl(stable(s))`