Computational Methods in Systems and Synthetic Biology

François Fages
Constraint Programming Group
INRIA Paris-Rocquencourt

mailto:Francois.Fages@inria.fr
http://contraintes.inria.fr

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA ROCQUENCOURT

Overview of the Lectures

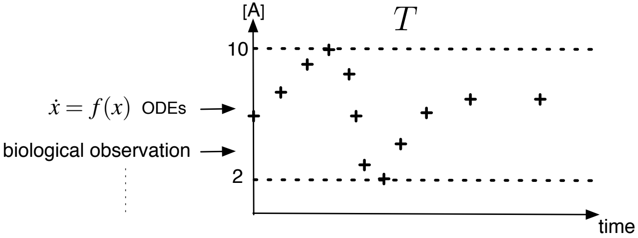
1. Formal molecules and reaction models in BIOCHAM
2. Kinetics
3. Qualitative properties formalized in temporal logic CTL
4. Quantitative properties formalized in LTL(R) and pLTL(R)
5. Reaction hypergraphs and influence graphs
6. Hierarchy of semantics and typing for systems biology by abstract interpretation
7. Learning parameters from temporal logic properties
   - From model-checking to constraint solving
   - QFLTL constraint solving
   - Continuous valuation of QFLTL formulae
   - Parameter optimization by randomized search
8. Robustness analysis

A Logical Paradigm for Systems Biology

$$\begin{gathered}
\textit{Biological Model} = \textit{(Quantitative) State Transition System } K \\
\textit{Biological Properties} = \textit{Temporal Logic Formulae } \phi \\
\textit{Automatic Validation} = \textit{Model-checking } K \models \phi \\
\textit{Model Inference} = \textit{Constraint Solving } K' \models \phi
\end{gathered}$$

- Verification of **high-level specifications** on state transition systems
  - Introduced by [Pnueli 77, Clarke 80] for program/circuit verification
  - Model-checking can be efficient on large complex systems
  - Temporal logic with numerical constraints can deal with continuous time models (ODE or CTMC, hybrid systems)
- Applications of Temporal Logics in Systems Biology:
  - **query language of large reaction networks** [Eker et al. PSB 02, Chabrier Fages CMSB 03, Batt et al. Bioinformatics 05]
  - **analysis of experimental data time series** [Fages Rizk CMSB 07]
  - **parameter search** [Bernot et al. JTB 04] [Calzone et al. TCSB 06] [Rizk et al. 08 CMSB]
  - **robustness analysis** [Batt et al. 07] [Rizk et al. 09 ISMB]
  - **model coupling** [De Maria Soliman Fages 09 CMSB]

Linear Time Logic



- $\mathbf{F}\phi$ (*finally*) : ϕ is true at some time point in the future;
- $\mathbf{G}\phi$ (*globally*) : ϕ is true at all time points in the future;
- $\phi_1\mathbf{U}\phi_2$ (*until*) : ϕ_1 is true until ϕ_2 becomes true.
- $\mathbf{X}\phi$ (*next*) : ϕ is true at the next time point;

Examples of LTL($\mathbb{R}$) Formulae

- **F**([A]>10) : the concentration of A eventually gets above 10.
- **FG**([A]>10) : the concentration of A eventually reaches and remains above value 10.
- **F**(Time=t1∧[A]=v1 ∧ **F**(.... ∧ **F**(Time=tN∧[A]=vN)...)) Numerical data time series (e.g. experimental curves)
- **G**([A]+[B]<[C]) : the concentration of C is always greater than the sum of the concentrations of A and B.
- **F**((d[M]/dt > 0) ∧ **F**((d[M]/dt < 0) ∧ **F**((d[M]/dt > 0)))) : change of sign of the derivative of M.
- oscillations, period constraints, etc.

True/False valuation of temporal logic formulae

The **True/False** valuation of temporal logic formulae is **not well adapted** to several problems :

- parameter search, optimization and control of continuous models
- quantitative estimation of robustness
- sensitivity analyses

True/False valuation of temporal logic formulae

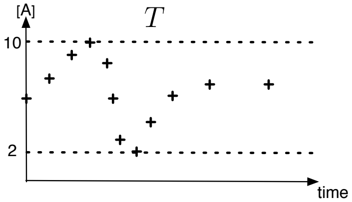
The **True/False** valuation of temporal logic formulae is **not well adapted** to several problems :

- parameter search, optimization and control of continuous models
- quantitative estimation of robustness
- sensitivity analyses

$\rightarrow$ need for a continuous degree of satisfaction of temporal logic formulae

How far is the system from verifying the specification ?

Model-Checking Generalized to Constraint Solving



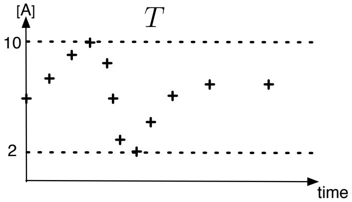
$LTL(\mathbb{R})$

$\Phi = F([A] \geq 7 \wedge F([A] \leq 0))$

Model-checking

the formula is false

Model-Checking Generalized to Constraint Solving



$LTL(\mathbb{R})$

Φ=F([A]≥7
∧F([A]≤0))

Model-checking

the formula is false

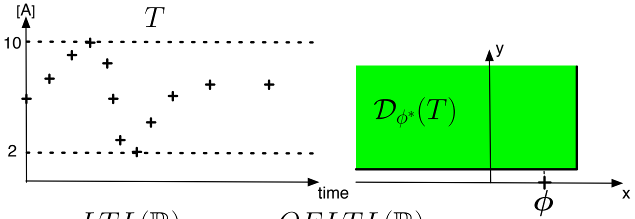
$QFLTL(\mathbb{R})$

Φ*=F([A]≥x
∧F([A]≤y))

Constraint solving

the formula is true for any
x≤10 ∧ y≥2

Model-Checking Generalized to Constraint Solving



$LTL(\mathbb{R})$

Φ=F([A]≥7
∧F([A]≤0))

Model-checking

the formula is false

$QFLTL(\mathbb{R})$

Φ*=F([A]≥x
∧F([A]≤y))

Constraint solving

the formula is true for any
x≤10 ∧ y≥2

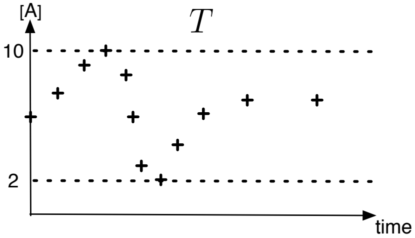
Validity domain $\mathcal{D}_{\phi^*}(T)$: set of values of the variables in a QFLTL formula making it true on a given trace T.

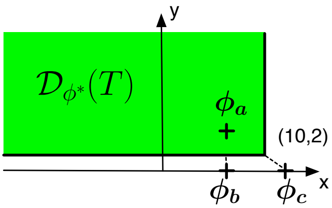
Violation degree of an LTL formula

Definition of violation degree $vd(T,\phi)$ and satisfaction degree $sd(T,\phi)$

In the variable space of ϕ^*, original formula ϕ is single point $var(\phi)$.

$vd(T,\phi) = min_{v \in D_{\phi^*}(T)} d(v, var(\phi))$ $\qquad$ $sd(T,\phi) = \frac{1}{1+vd(T,\phi)} \in [0,1]$





ϕ^*(x,y)= F([A]≥x ∧ F([A]≤y))

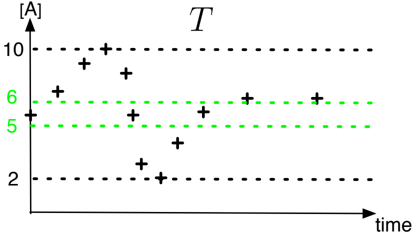
ϕ_a = F([A]≥6 ∧ F([A]≤5))	ϕ^*(6,5)	*vd*=0	(✓)
ϕ_b = F([A]≥6 ∧ F([A]≤0))	ϕ^*(6,0)	*vd*=2	(✗)
ϕ_c = F([A]≥12 ∧ F([A]≤0))	ϕ^*(12,0)	*vd*=2√2	(✗)

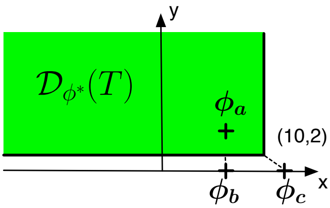
Violation degree of an LTL formula

Definition of violation degree $vd(T, \phi)$ and satisfaction degree $sd(T, \phi)$

In the variable space of ϕ^*, original formula ϕ is single point $var(\phi)$.

$vd(T, \phi) = min_{v \in D_{\phi^*}(T)} d(v, var(\phi))$ $\qquad sd(T, \phi) = \frac{1}{1+vd(T,\phi)} \in [0, 1]$





ϕ^*(x,y)= F([A]≥x ∧ F([A]≤y))

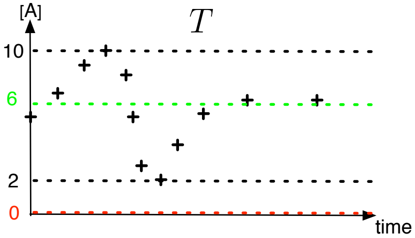
ϕ_a = F([A]≥6 ∧ F([A]≤5))	ϕ^*(6,5)	*vd*=0	(✓)
ϕ_b = F([A]≥6 ∧ F([A]≤0))	ϕ^*(6,0)	*vd*=2	(✕)
ϕ_c = F([A]≥12 ∧ F([A]≤0))	ϕ^*(12,0)	*vd*=2√2	(✕)

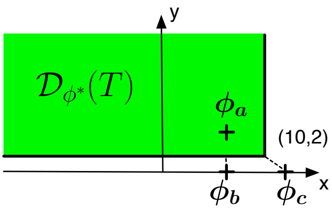
Violation degree of an LTL formula

Definition of violation degree $vd(T,\phi)$ and satisfaction degree $sd(T,\phi)$

In the variable space of ϕ^*, original formula ϕ is single point $var(\phi)$.

$vd(T,\phi) = min_{v \in D_{\phi^*}(T)} d(v, var(\phi))$ $\qquad$ $sd(T,\phi) = \frac{1}{1+vd(T,\phi)} \in [0,1]$





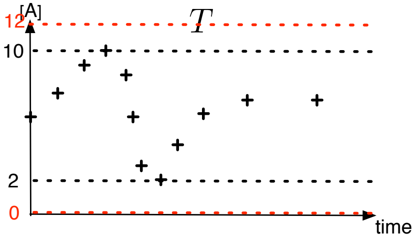
ϕ^*(x,y)= F([A]≥x ∧ F([A]≤y))

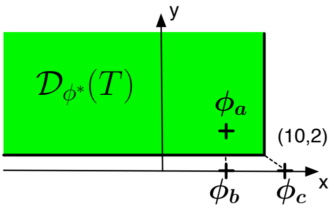
ϕ_a = F([A]≥6 ∧ F([A]≤5))	ϕ^*(6,5)	*vd*=0	(✓)
ϕ_b = F([A]≥6 ∧ F([A]≤0))	ϕ^*(6,0)	*vd*=2	(✗)
ϕ_c = F([A]≥12 ∧ F([A]≤0))	ϕ^*(12,0)	*vd*=2√2	(✗)

Violation degree of an LTL formula

Definition of violation degree $vd(T,\phi)$ and satisfaction degree $sd(T,\phi)$

In the variable space of ϕ^*, original formula ϕ is single point $var(\phi)$.
$vd(T,\phi) = min_{v \in D_{\phi^*}(T)} d(v, var(\phi))$ $sd(T,\phi) = \frac{1}{1+vd(T,\phi)} \in [0,1]$



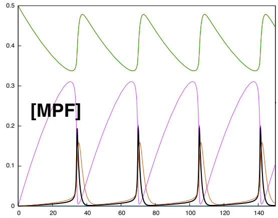


ϕ^*(x,y)= F([A]≥x ∧ F([A]≤y))

ϕ_a = F([A]≥6 ∧ F([A]≤5))	ϕ^*(6,5)	*vd*=0	(✓)
ϕ_b = F([A]≥6 ∧ F([A]≤0))	ϕ^*(6,0)	*vd*=2	(✗)
ϕ_c = F([A]≥12 ∧ F([A]≤0))	ϕ^*(12,0)	*vd*=2√2	(✗)

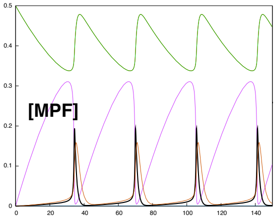
Learning kinetic parameter values from LTL specifications

- simple model of the yeast cell cycle from [Tyson PNAS 91]
- models Cdc2 and Cyclin interactions (6 variables, 8 kinetic parameters)



Learning kinetic parameter values from LTL specifications

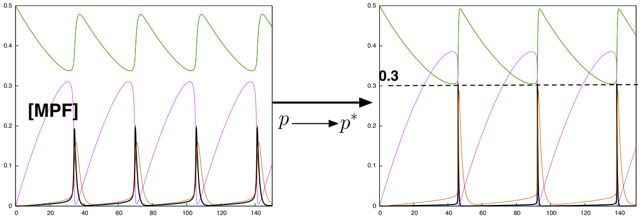
- simple model of the yeast cell cycle from [Tyson PNAS 91]
- models Cdc2 and Cyclin interactions (6 variables, 8 kinetic parameters)



- Pb : find values of 8 parameters such that amplitude is ≥ 0.3
  ϕ^*: **F**([A]>x $\wedge$ **F**([A]<y))
  amplitude z=x-y
  goal : z = 0.3

Learning kinetic parameter values from LTL specifications

- simple model of the yeast cell cycle from [Tyson PNAS 91]
- models Cdc2 and Cyclin interactions (6 variables, 8 kinetic parameters)



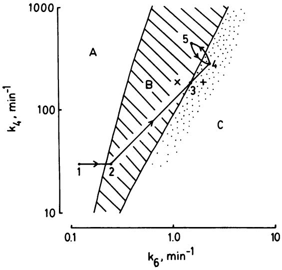
- Pb : find values of 8 parameters such that amplitude is ≥ 0.3
  ϕ^*: **F**([A]>x $\wedge$ **F**([A]<y))
  amplitude z=x-y
  goal : z = 0.3
- $\rightarrow$ solution found after 30s (100 calls to the fitness function)

LTL Continuous Satisfaction Diagram

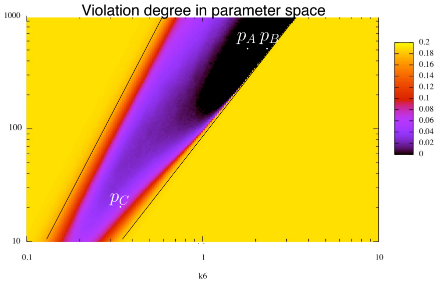
Example with :

- yeast cell cycle model [Tyson PNAS 91]
- oscillation of at least 0.3

ϕ^*: **F**([A]>x $\wedge$ **F**([A]<y)); amplitude x-y$\geq$0.3



Bifurcation diagram



LTL satisfaction diagram

Black-box Randomized Non-linear Optimization Method

- Use existing non-linear optimization toolbox for kinetic parameter search using satisfaction degree as fitness function

Black-box Randomized Non-linear Optimization Method

- Use existing non-linear optimization toolbox for kinetic parameter search using satisfaction degree as fitness function
- We use the state-of-the-art Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Hansen Osermeier 01, Hansen 08]

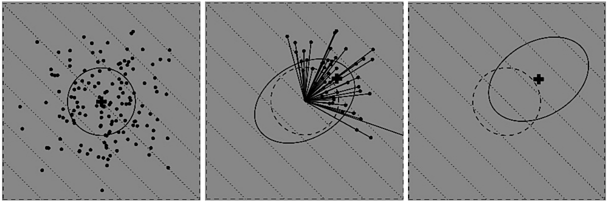
Black-box Randomized Non-linear Optimization Method

- Use existing non-linear optimization toolbox for kinetic parameter search using satisfaction degree as fitness function
- We use the state-of-the-art Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Hansen Osermeier 01, Hansen 08]
- CMA-ES maximizes an objective function in continuous domain in a black box scenario :

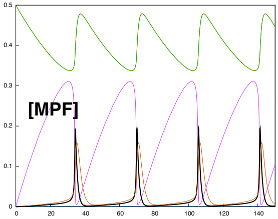
Black-box Randomized Non-linear Optimization Method

- Use existing non-linear optimization toolbox for kinetic parameter search using satisfaction degree as fitness function
- We use the state-of-the-art Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Hansen Osermeier 01, Hansen 08]
- CMA-ES maximizes an objective function in continuous domain in a black box scenario :

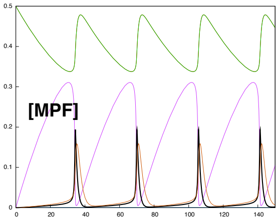
- CMA-ES uses a probabilistic neighborhood and updates information in covariance matrix at each move



Learning Parameter Values from Period Constraints in LTL

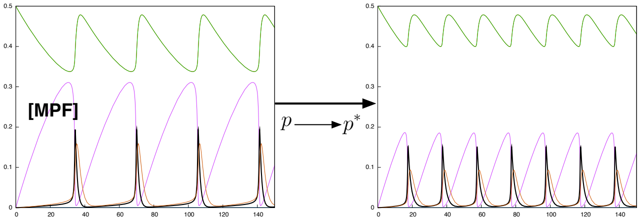


Learning Parameter Values from Period Constraints in LTL



- Pb : find values of 8 parameters such that period is 20
  ϕ^*:**F**(MPF$_{localmaximum}$ $\wedge$Time=t1$\wedge$ **F**(MPF$_{localmaximum}$ $\wedge$Time=t2))
  (with MPF$_{localmaximum}$: d([MPF])/dt>0 $\wedge$ **X**(d([MPF])/dt<0))
  period z=t2-t1
  goal z=20

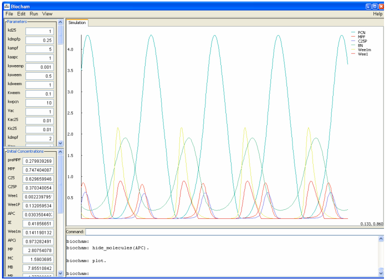
Learning Parameter Values from Period Constraints in LTL



- Pb : find values of 8 parameters such that period is 20
  ϕ^*:**F**(MPF$_{localmaximum}$ ∧Time=t1∧ **F**(MPF$_{localmaximum}$ ∧Time=t2))
  (with MPF$_{localmaximum}$: d([MPF])/dt>0 ∧ **X**(d([MPF])/dt<0))
  period z=t2-t1
  goal z=20
- → Solution found after 60s (200 calls to the fitness function)

Coupled Models of Cell Cycle, Circadian Clock, DNA repair

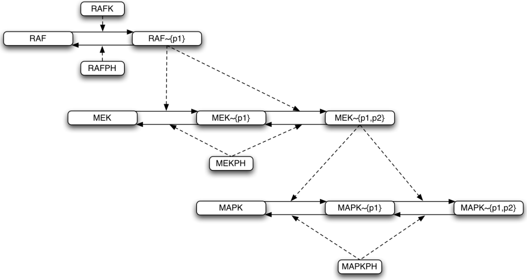
- Context of colorectal cancer chronotherapies
  EU project TEMPO, coord. F. Lévi INSERM Villejuif France
- Coupled model of the cell cycle [Tyson Novak 04] and the circadian clock [Leloup Goldbeter 99] with condition of entrainment in period [Calzone Soliman 06]



- Coupled model with DNA repair system p53/Mdm2 [Cilberto et al.04] and effect of irinotecan anticancer drug [De Maria Soliman Fages 09 CMSB]

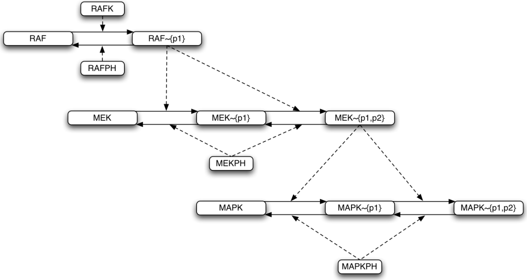
Oscillations in MAPK signal transduction cascade

- **MAPK** signaling model [Huang Ferrel PNAS 96]



Oscillations in MAPK signal transduction cascade

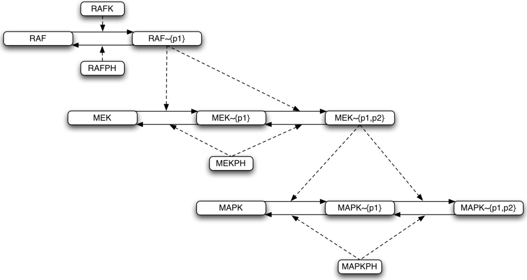
- **MAPK** signaling model [Huang Ferrel PNAS 96]



- **search for oscillations** in **37 dimensions** (30 parameters and 7 initial conditions)
  $\longrightarrow$ solution found after 3 min (200 calls to the fitness function)
  Oscillations already observed by simulation [Qiao et al. 07]

Oscillations in MAPK signal transduction cascade

- **MAPK** signaling model [Huang Ferrel PNAS 96]



- **search for oscillations** in **37 dimensions** (30 parameters and 7 initial conditions)
  $\longrightarrow$ solution found after 3 min (200 calls to the fitness function)
  Oscillations already observed by simulation [Qiao et al. 07]
- No negative feedback in the **reaction graph**, but negative circuits in the **influence graph** [Fages Soliman FMSB'08, CMSB'06]

LTL(R) formulae

State variables over the reals: time and $\mathbf{x}$, $\dot{\mathbf{x}}_i \in \mathbb{R}^m$ are vectors of state variable values and of their derivatives at given time.

Atomic propositions: arithmetic expressions with $<, \leq, =, \geq, >$ over the state variables (closed by negation)

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

Properties: $\mathbf{F}\phi = \text{true}\ \mathbf{U}\ \phi$, $\mathbf{G}\phi = \phi\ \mathbf{W}\ \text{false}$, $\phi\mathbf{W}\psi = \mathbf{G}\phi \vee (\phi\mathbf{U}(\phi \wedge \psi))$

Negation free formulae: expressed with $\wedge$, $\vee$, $\mathbf{F}$, $\mathbf{G}$, $\mathbf{U}$, $\mathbf{X}$ with negations eliminated down to atomic propositions.

LTL(R) formulae on finite traces

Finite trace $T = (s_0, s_1, \ldots, s_n)$ of timed states $s_i = (t_i, \mathbf{x}_i, \dot{\mathbf{x}}_i)$ where $t_i > t_{i-1}$

- $T \models \phi$ iff $T, s_0 \models \phi$,
- $T, s_i \models \pi$ iff $T, s_i \models_{\mathcal{R}} \pi(\mathbf{y})$,
- $T, s_i \models \phi \wedge \psi$ iff $T, s_i \models \phi$ and $T, s_i \models \psi$,
- $T, s_i \models \phi \vee \psi$ iff $T, s_i \models \phi$ or $T, s_i \models \psi$,
- $T, s_i \models \mathbf{F}\phi$ iff $\exists j \in [i, n]$ such that $T, s_j \models \phi$,
- $T, s_i \models \mathbf{G}\phi$ iff $\forall j \in [i, n]$, $T, s_j \models \phi$,
- $T, s_i \models \phi \mathbf{U} \psi$ iff $\exists j \in [i, n]$ s. t. $T, s_j \models \psi$ and $\forall k \in [i, j-1]$, $T, s_k \models \phi$.
- $T, s_i \models \mathbf{X}\phi$ iff $i < n$ and $T, s_{i+1} \models \phi$, or $i = n$ and $T, s_n \models \phi$,

Proposition

This interpretation of LTL formulae over finite traces is equivalent to the standard interpretation over infinite traces completed by a loop on the terminal state.

LTL(R) model-checking

Given a finite trace T and an LTL(R) formula ϕ

1. label each state with the atomic sub-formulae of ϕ that are true at this state;
2. add sub-formulae of the form $\phi_1\ U\ \phi_2$ to the states labeled by ϕ_2 and to the predecessors of states labeled with ϕ_2 as long as they are labeled by ϕ_1;
3. add sub-formulae of the form $\phi_1\ \mathbf{W}\ \phi_2$ to the last state if it is labeled by ϕ_1, and to the states labeled by ϕ_1 and ϕ_2, and to their predecessors as long as they are labeled by ϕ_1;
4. add sub-formulae of the form $\mathbf{X}\phi$ to the last state if it is labeled by ϕ and to the immediate predecessors of states labeled by ϕ;
5. return the vertices labeled by ϕ.

Proposition

In trace $T = (s_1, \ldots, s_n)$, state s_i is labeled by ϕ if and only if $T, s_i \models \phi$.

QFLTL(R) formulae

Quantifier free LTL formulae, noted $\phi(\mathbf{y})$, add free variables $\mathbf{y}$ to state variables

The *satisfaction domain* of $\phi(\mathbf{y})$ in a trace T is the set of $\mathbf{y}$ values for which $\phi(\mathbf{y})$ holds:

$$\mathcal{D}_{T,\phi(\mathbf{y})} = \{\mathbf{y} \in \mathbb{R}^q \mid T \models \phi(\mathbf{y})\} \tag{1}$$

For linear constraints over R, satisfaction domains can be computed with polyhedral libraries.

Without loss of generality, let us consider negation free QFLTL formulae.

QFLTL(R) constraint solving

The satisfaction domains of QFLTL formulae satisfy the equations:

- $\mathcal{D}_{T,\phi(\mathbf{y})} = \mathcal{D}_{s_0,\phi(\mathbf{y})}$,
- $\mathcal{D}_{s_i,\pi(\mathbf{y})} = \{\mathbf{y} \in \mathbb{R}^m \mid s_i \models_{\mathcal{R}} \pi(\mathbf{y})\}$,
- $\mathcal{D}_{s_i,\phi(\mathbf{y})\wedge\psi(\mathbf{y})} = \mathcal{D}_{s_i,\phi(\mathbf{y})} \cap \mathcal{D}_{s_i,\psi(\mathbf{y})}$,
- $\mathcal{D}_{s_i,\phi(\mathbf{y})\vee\psi(\mathbf{y})} = \mathcal{D}_{s_i,\phi(\mathbf{y})} \cup \mathcal{D}_{s_i,\psi(\mathbf{y})}$,
- $\mathcal{D}_{s_i,\mathbf{F}\phi(\mathbf{y})} = \cup_{j\in[i,n]}\mathcal{D}_{s_j,\phi(\mathbf{y})}$,
- $\mathcal{D}_{s_i,\mathbf{G}\phi(\mathbf{y})} = \cap_{j\in[i,n]}\mathcal{D}_{s_j,\phi(\mathbf{y})}$,
- $\mathcal{D}_{s_i,\phi(\mathbf{y})\mathbf{U}\psi(\mathbf{y})} = \cup_{j\in[i,n]}\big(\mathcal{D}_{s_j,\psi(\mathbf{y})} \cap \cap_{k\in[i,j-1]}\mathcal{D}_{s_k,\phi(\mathbf{y})}\big)$,
- $\mathcal{D}_{s_i,\mathbf{X}\phi(\mathbf{y})} = \begin{cases} \mathcal{D}_{s_{i+1},\phi(\mathbf{y})}, & \text{if } i < n, \\ \mathcal{D}_{s_i,\phi(\mathbf{y})}, & \text{if } i = n, \end{cases}$

Proposition

The satisfaction domains of a QFLTL formula ϕ in a trace T can be computed with these equations following the increasing subformula ordering.

Complexity with bound constraints $x > b$, $x < b$

Bound constraints define boxes $\mathcal{R}_i \in \mathbb{R}^v$. Let the size of a union of boxes be the least integer k such that $\mathcal{D} = \bigcup_{i=1}^{k} \mathcal{R}_i$.

Proposition (complexity of the solution domain)

The validity domain of a QFLTL formula of size f containing v variables on a trace of length n is a union of boxes of size less than $(nf)^{2v}$.

The maximum number of bounds for a variable x is $n \times f$ (which is is attained in e.g; $F([A] = u \vee [A] + 1 = u \vee \cdots \vee [A] + f = u))$.
If $\mathcal{B}_v(\phi)$ is the set of possible bounds for variable x in ϕ, and if ϕ_1 and ϕ_2 are subformulae of ϕ, we have $\mathcal{B}_v(\phi_1 \vee \phi_2) \subset \mathcal{B}_v(\phi)$ and
$\mathcal{B}_v(\phi_1 \wedge \phi_2) \subset \mathcal{B}_v(\phi)$.
As a box is a cartesian product of intervals defined by two bounds for each variable. the size of the solution domain is less than $(nf)^{2v}$.
$F([A_1] = X_1 \vee [A_1] + 1 = X_1 \vee ... \vee [A_1] + f = X_1) \;\wedge\; ...$
$\wedge\; F([A_v] = X_v \vee [A_v] + 1 = X_v \vee ... \vee [A_v] + f = X_v)$
has a solution domain of size $(nf)^v$ on a trace of n values with $[A_i] + k$ all different for $1 \leq i \leq v$, $0 \leq k \leq f$.

LTL formulae as points in QFLTL formula space $\mathbb{R}^v$

An LTL formula can be seen as an instance of a QFLTL formula obtained by abstracting the constants appearing in the formula by new variables $\mathbf{y} \in \mathbb{R}^q$.

For example, to $\phi_1 = \mathbf{F}(\texttt{[A]} > 7 \wedge \mathbf{F}\ \texttt{[A]} < 3)$
we associate the formula $\phi(\mathbf{y}) = \phi(y_1, y_2) = \mathbf{F}(\texttt{[A]} > y_1 \wedge \mathbf{F}\ \texttt{[A]} < y_2)$.
Then we have $\phi_1 = \phi(7, 3)$.

This variable abstraction/instantiation process allows us to view a LTL formula as a point in the QFLTL *formula space* $\mathbb{R}^q$, where q is the number of constants appearing in ϕ (or the number of constants that are replaced by variables, if not all constants are abstracted away).

Continuous valuation of QFLTL formulae in $[0, 1]$

The *violation degree* $vd(T, \phi)$ of a formula ϕ w.r.t. trace T is the *distance* between the actual specification and validity domain $\mathcal{D}_{T,\phi(\mathbf{y})}$ of the QFLTL formula $\phi(\mathbf{y})$ obtained by variable abstraction:

$$vd(T, \phi) = dist(\phi, \mathcal{D}_{T,\phi(\mathbf{y})}).$$

Abstracting constants by variables in temporal logic formulae is a means to define a *metric* on the set of formulae. All set operations and distance computations are made in the corresponding metric space, known as the formula space.

The continuous *satisfaction degree* of a formula w.r.t. a trace T:

$$sd(T, \phi) = \frac{1}{1 + vd(T, \phi)} \in [0, 1], \qquad (2)$$

Conclusion

Definition of a continuous degree of satisfaction of LTL($\mathbb{R}$) formulae which can be computed by LTL($\mathbb{R}$) constraint solving algorithm [Fages Rizk CMSB'07, CP'09] Shown useful for :

- measuring the satisfaction of high level specifications
- optimizing kinetic parameters and initial conditions w.r.t. temporal specifications (37 parameters for MAPK)
- optimizing control laws
- measuring the robustness of a model w.r.t temporal logic specifications

Related work :

- probabilistic/statistical model checking [Kwiatkowska et al. SIGMETRICS 08, Clarke et al. CMSB 08]
- alternative quantitative interpretation of TL [Fainekos and Pappas FORMATS 07]

On-going work

Computational methods (implementation in BIOCHAM and MathLab)

- parallelization on clusters of 100-10000 processors
- multi-trace LTL specifications (e.g. different initial cond., mutations)
- evaluation on larger models with rich biological data (e.g. Chen et al. cell cycle model validation w.r.t. 130 mutants)
- generalization to non-deterministic quantitative transition systems [Fages Rizk CP'09]

Use in systems biology

- development of new models of GPCR-receptor activation (collab. INRA France)
- development of coupled models of mammalian cell cycle, circadian rythm, DNA damage repair systems and anticancer drugs (collab. INSERM France, EU Tempo) [De Maria et al. 09 CMSB]

Use in synthetic biology

- model of perturbation for transcriptional cascade among E. Coli cells (collab. of Greg Batt with Ron Weiss, Princeton)
- integration of robustness as a parameter optimization criterion