From Sensitive to Formal Barbaric Systems Biology

Oded Maler Memorial

Alexandre Donzé

Decyphir SAS, Moirans, France

HSB'19, April 6th 2019

A Tribute To Oded?

Disclaimer: several ways this talk is (sort of) like one of Oded's

A Tribute To Oded?

Disclaimer: several ways this talk is (sort of) like one of Oded's

- Starts with some high level "French Cafe"

A Tribute To Oded?

Disclaimer: several ways this talk is (sort of) like one of Oded's

- Starts with some high level "French Cafe"
- Reuses bunch of un-organized old slides and recurrent examples (some also related to coffee)

A Tribute To Oded?

Disclaimer: several ways this talk is (sort of) like one of Oded's

- Starts with some high level "French Cafe"
- Reuses bunch of un-organized old slides and recurrent examples (some also related to coffee)
- Contains doses of cynism covering up for genuine curiosity and dedication to research and science

A Tribute To Oded?

Disclaimer: several ways this talk is (sort of) like one of Oded's

- Starts with some high level "French Cafe"
- Reuses bunch of un-organized old slides and recurrent examples (some also related to coffee)
- Contains doses of cynism covering up for genuine curiosity and dedication to research and science
- Eventually enthusiastically dives into unnecessary details, possibly forgeting about reality and losing audience for a while

A Tribute To Oded?

Disclaimer: several ways this talk is (sort of) like one of Oded's

- Starts with some high level "French Cafe"
- Reuses bunch of un-organized old slides and recurrent examples (some also related to coffee)
- Contains doses of cynism covering up for genuine curiosity and dedication to research and science
- Eventually enthusiastically dives into unnecessary details, possibly forgeting about reality and losing audience for a while
- Ends with time-invariant Future Works and Perspectives

Where Do We Come From?

Verimag, a computer scientists lab, specializing in programs verifying programs

Where Do We Come From?

Verimag, a computer scientists lab, specializing in programs verifying programs

Sometimes compromising with control engineers, embedded systems designers, circuit designers, etc

Where Do We Come From?

Verimag, a computer scientists lab, specializing in programs verifying programs

Sometimes compromising with control engineers, embedded systems designers, circuit designers, etc

Oded was certainly the boldest compromising himself with applied mathematicians, physicists and even biologists...

Why Systems Biology?

A summary of Oded's views (quotes from opening remarks of TSB'11)

Cynical: When you have a hammer, everything looks like a nail.

Fortunately, my hammer is universal.

Why Systems Biology?

A summary of Oded's views (quotes from opening remarks of TSB'11)

Cynical: When you have a hammer, everything looks like a nail.

Fortunately, my hammer is universal.

Arrogant: Biologists need *real* scientists to guide them (Math, CS, physicists)

Like monotheists converting the pagans, these merchants of abstract methodologies try to impress the poor savage with their logics and miracles

Why Systems Biology?

A summary of Oded's views (quotes from opening remarks of TSB'11)

Cynical: When you have a hammer, everything looks like a nail.

Fortunately, my hammer is universal.

Arrogant: Biologists need *real* scientists to guide them (Math, CS, physicists)

Like monotheists converting the pagans, these merchants of abstract methodologies try to impress the poor savage with their logics and miracles

Humble: *Living systems are more mysterious and primordial than the prime numbers, the algebra of Boole or the free monoid*

We should be very happy and proud for doing, for once, something meaningful

Why Systems Biology?

A summary of Oded's views (quotes from opening remarks of TSB'11)

Cynical: When you have a hammer, everything looks like a nail.

Fortunately, my hammer is universal.

Arrogant: Biologists need *real* scientists to guide them (Math, CS, physicists)

Like monotheists converting the pagans, these merchants of abstract methodologies try to impress the poor savage with their logics and miracles

Humble: *Living systems are more mysterious and primordial than the prime numbers, the algebra of Boole or the free monoid*

We should be very happy and proud for doing, for once, something meaningful

Sober: Biology is dominated by data (omics)

- *Systems Biology is about seeking some clearer (conceptual and mathematical) models of dynamical systems at various levels of abstraction*
- *These models, if thoughtfully constructed, may help reducing the gap between cellular biochemistry and physiology*

Why Systems Biology? (subjective)

Looking back at my own motivations, I was mostly in the sober/humble view

Convinced that some dose of formal methods can and should help biology

But with less ambitious goal on the modeling part, focusing on more specific aspects: parameters, simulation, specifications,...

Next are a few introductory slides from a talk I gave to an unexpected audience in 2010...

Formal Verification

A domain taking its roots in early computer science theory (language and automata theory), discrete mathematics, logics, even philosophy

Its goal: to prove correctness

Growing in applicability/popularity steadily since the early 80s and the advent of Model Checking (Turing award of Clarke, Emerson and Sifakis in 2007)

Its popularity "benefited" from spectacular failure of simple testing and bug finding in the 90s (Pentium bug, Ariane 5 self-destruction due to a software bug)

Proving correctness?

Correctness is a subjective notion until it is defined *formally*.
For this we need:

- a description of the systems behaviors
- a specification language to describe *desired* (good) and *unwanted* (bad) properties

Proving correctness?

Correctness is a subjective notion until it is defined *formally*.
For this we need:

- a description of the systems behaviors
- a specification language to describe *desired* (good) and *unwanted* (bad) properties

Coffee machine example

- a good property is: *if I insert a coin and push 'coffee', I get coffee*
- a bad one: *I get a tea (and no change)*

Proving correctness?

Correctness is a subjective notion until it is defined *formally*.
For this we need:

- a description of the systems behaviors
- a specification language to describe *desired* (good) and *unwanted* (bad) properties

Coffee machine example

- a good property is: *if I insert a coin and push 'coffee', I get coffee*
- a bad one: *I get a tea (and no change)*

The system is declared correct iff

all the behaviors of the system satisfies *all* the good properties and *none* of the bad ones

Reactive Systems and Temporal Logics

A key issue is the appropriate choice of language to describe properties:

- Enough expressivity
- Ease of writing specification

Reactive Systems and Temporal Logics

A key issue is the appropriate choice of language to describe properties:

- Enough expressivity
- Ease of writing specification

Temporal logics popularized in 1978 by Amir Pnueli when programs shifted from simple input-output relations to reactive programs,

A typical reactive program is an operating system:

- a good property is *always when the mouse is moved, the cursors moves*
- a bad one: *always eventually a blue screen appears and nothing happens*

Reactive Systems and Temporal Logics

A key issue is the appropriate choice of language to describe properties:

- Enough expressivity
- Ease of writing specification

Temporal logics popularized in 1978 by Amir Pnueli when programs shifted from simple input-output relations to reactive programs,

A typical reactive program is an operating system:

- a good property is *always when the mouse is moved, the cursors moves*
- a bad one: *always eventually a blue screen appears and nothing happens*

A good property such as the one above is a *liveness property*. Living systems are typically reactive programs..

From Verification to Synthesis

Verification of mis-conceived systems can be tedious and frustrating. Rather than chasing bugs, can't we prevent them from happening in the first place ?

Synthesis is the ultimate goal of Formal Verification:

Building correct-by-construction systems directly from specifications

For synthesized systems, verification is unnecessary.

Synthesis in the Wild

Synthesis is a difficult problem: decades of research, actually applied for hardly a couple of years to produce small digital circuits

Attempts to apply synthesis in even more challenging context: software, analog circuits , control engineering, biology, etc

Synthesis in the Wild

Synthesis is a difficult problem: decades of research, actually applied for hardly a couple of years to produce small digital circuits

Attempts to apply synthesis in even more challenging context: software, analog circuits , control engineering, biology, etc

Is this reasonnable/useful ? In most cases, no. A common syndrome:

When you have a hammer, everything looks like a nail

Synthesis in the Wild

Synthesis is a difficult problem: decades of research, actually applied for hardly a couple of years to produce small digital circuits

Attempts to apply synthesis in even more challenging context: software, analog circuits , control engineering, biology, etc

Is this reasonnable/useful ? In most cases, no. A common syndrome:

When you have a hammer, everything looks like a nail

Still, genuine belief that diffusing formal methods to other, more primitive scientific domains, if done in an humble and intelligent way, can do some good

1. Parameter Synthesis
   - Parametric Systems
   - Sensitive Systematic (aka Barbaric) Simulation

2. Parameter Synthesis with Formal Specifications
   - Signal Temporal Logic
   - Property parameters
   - Model parameters

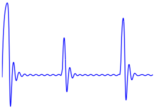
3. Some Results and Concluding Remarks

Parametric Systems

Definition (Parametric System)

An object mapping a finite set of values (parameters) to a set of signals

$$\mathbf{p} = (p_1, \cdots, p_n) \longrightarrow \boxed{\text{System } \mathcal{S}} \longrightarrow \mathbf{x}[t] :$$

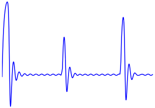


Parametric Systems

Definition (Parametric System)

An object mapping a finite set of values (parameters) to a set of signals

$$\mathbf{p} = (p_1, \cdots, p_n) \longrightarrow \boxed{\text{System } \mathcal{S}} \longrightarrow \mathbf{x}[t] :$$



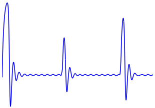
- $\mathbf{p}$, t, $\mathbf{x}[t]$ in $\mathbb{R}$ domain, $t \mapsto \mathbf{x}[t]$ continuous "almost everywhere"

Parametric Systems

Definition (Parametric System)

An object mapping a finite set of values (parameters) to a set of signals

$$\mathbf{p} = (p_1, \cdots, p_n) \longrightarrow \boxed{\text{System } \mathcal{S}} \longrightarrow \mathbf{x}[t] :$$



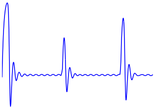
- $\mathbf{p}$, t, $\mathbf{x}[t]$ in $\mathbb{R}$ domain, $t \mapsto \mathbf{x}[t]$ continuous "almost everywhere"
- Typically (for us): $\mathcal{S}$ is a (hybrid) system of ordinary differential equations

Parametric Systems

Definition (Parametric System)

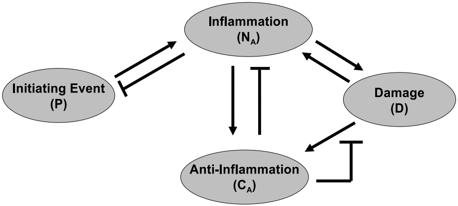
An object mapping a finite set of values (parameters) to a set of signals

$$\mathbf{p} = (p_1, \cdots, p_n) \longrightarrow \boxed{\text{System } \mathcal{S}} \longrightarrow \mathbf{x}[t] :$$



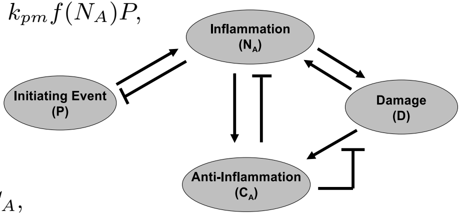
- $\mathbf{p}$, t, $\mathbf{x}[t]$ in $\mathbb{R}$ domain, $t \mapsto \mathbf{x}[t]$ continuous "almost everywhere"
- Typically (for us): $\mathcal{S}$ is a (hybrid) system of ordinary differential equations
- But most of what we do works for black box parametric systems

Example: acute inflamatory response to pathogen



Example: acute inflamatory response to pathogen

$$
\begin{aligned}
\frac{dP}{dt} &= k_{pg}P\big(1-\frac{P}{p_\infty}\big) - \frac{k_{pm}s_m P}{\mu_m + k_{mp}P} - k_{pm}f(N_A)P,\\
\frac{dN_A}{dt} &= \frac{s_{nr}R}{\mu_{nr}+R} - \mu_n N_A,\\
\frac{dD}{dt} &= k_{dn}f_s(f(N_A)) - \mu_d D,\\
\frac{dC_A}{dt} &= s_c + \frac{k_{cn}f(N_A + k_{cmd}D)}{1+f(N_A+k_{cmd}D)} - \mu_c C_A,
\end{aligned}
$$

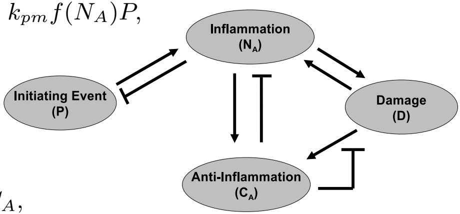


Parameters

- "Initial" conditions: $P(t=0), N_A(t=0), D(t=0), C_A(t=0)$.
- Others: $k_{pg}, p_\infty, k_{pm}, s_m, \mu_m, s_{nr}, \ldots$

Example: acute inflamatory response to pathogen

$$
\begin{aligned}
\frac{dP}{dt} &= k_{pg}P\big(1-\frac{P}{p_\infty}\big) - \frac{k_{pm}s_mP}{\mu_m + k_{mp}P} - k_{pm}f(N_A)P, \\
\frac{dN_A}{dt} &= \frac{s_{nr}R}{\mu_{nr}+R} - \mu_n N_A, \\
\frac{dD}{dt} &= k_{dn}f_s(f(N_A)) - \mu_d D, \\
\frac{dC_A}{dt} &= s_c + \frac{k_{cn}f(N_A + k_{cmd}D)}{1+f(N_A+k_{cmd}D)} - \mu_c C_A,
\end{aligned}
$$



Parameters

- "Initial" conditions: $P(t=0), N_A(t=0), D(t=0), C_A(t=0)$.
- Others: $k_{pg}, p_\infty, k_{pm}, s_m, \mu_m, s_{nr}, \ldots$

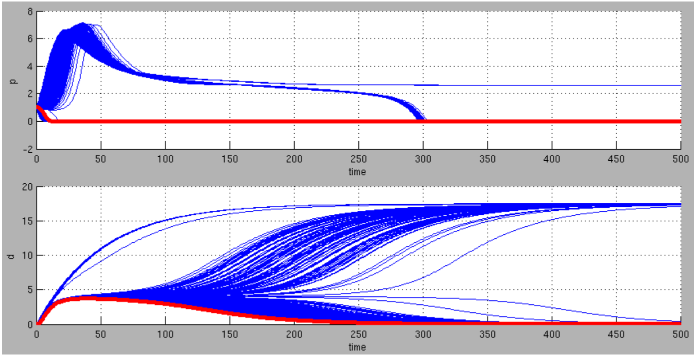
Depending on their values, three possible outcomes

- Health: pathogen and damage are driven to a low steady state
- Aseptic death: pathogen is eliminated but not tissue damage
- Septic death: tissue damage and pathogen remain high

Healthy outcome

Pathogen

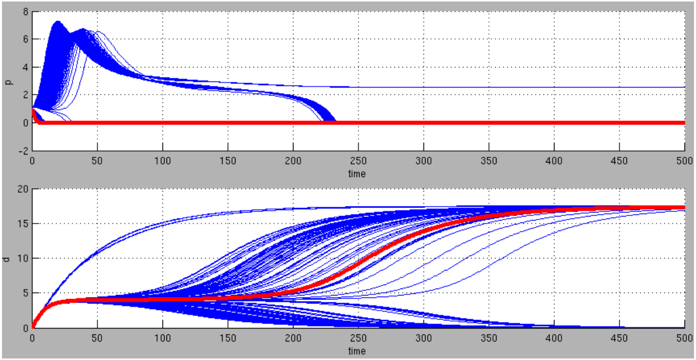
Damage



Aseptic death outcome

Pathogen

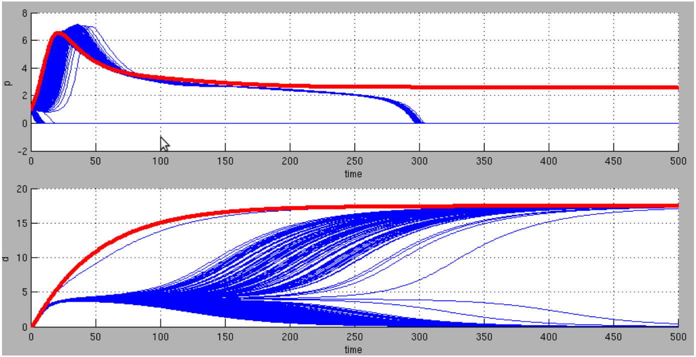
Damage



Septic death outcome

Pathogen

Damage



The problem with parameters

We don't know them.

The problem with parameters

We don't know them.

Traditional approach to solve this

- Calibration: Find $\mathbf{p}$ such that $\|S(\mathbf{p}) - \mathbf{x}_{\text{measured}}\|$ is minimized.
- Usually some optimization problem.

The problem with parameters

We don't know them.

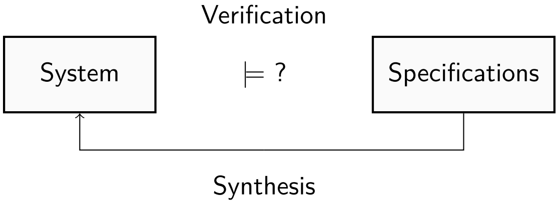
Traditional approach to solve this

- Calibration: Find $\mathbf{p}$ such that $\|S(\mathbf{p}) - \mathbf{x}_{\text{measured}}\|$ is minimized.
- Usually some optimization problem.

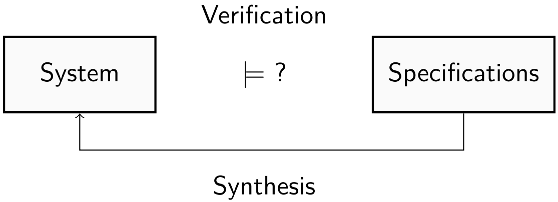
Validation (hard)

- $S(\mathbf{p})$ predicts $\mathbf{x}_{\text{measured}}$ *before* it's measured
- (and not just once by luck)
- Robustness: $S(\mathbf{p} + \epsilon)$ is not vastly different from $S(\mathbf{p})$
- ?

Formal methods and parameter synthesis

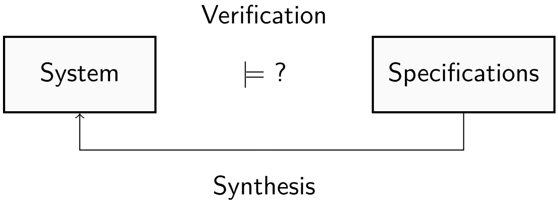


Formal methods and parameter synthesis



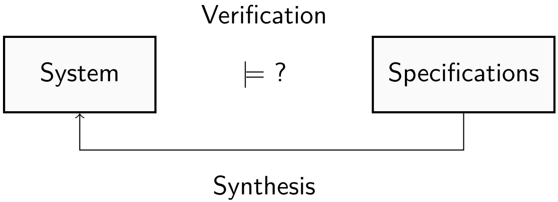
- Parameter synthesis reduces synthesis to finding “a few” valid values for parameters

Formal methods and parameter synthesis



- Parameter synthesis reduces synthesis to finding “a few” valid values for parameters
- We consider:
  - *System* parameters: for which values is the spec. satisfied ?
  - *Specification* parameters: what is the spec. actually satisfied ?

Formal methods and parameter synthesis



- Parameter synthesis reduces synthesis to finding "a few" valid values for parameters
- We consider:
  - *System* parameters: for which values is the spec. satisfied ?
  - *Specification* parameters: what is the spec. actually satisfied ?

In the following we focus on **reachability specifications**

Reachability analysis and systematic simulation

Reachable set

Note $\mathbf{x}(t, \mathbf{p})$ the simulation trace obtained using $\mathbf{p}$. We define

$$\text{Reach}(T, \mathcal{P}) = \{\mathbf{x}(t, \mathbf{p}) \text{ such that } t \leq T, \mathbf{p} \in \mathcal{P}\}$$

Lots of very sophisticated, non-scalable techniques developed to compute it using computer geometry, numerical and symbolic analysis, formal methods, etc.

Reachability analysis and systematic simulation

Reachable set

Note $\mathbf{x}(t, \mathbf{p})$ the simulation trace obtained using $\mathbf{p}$. We define

$$\text{Reach}(T, \mathcal{P}) = \{\mathbf{x}(t, \mathbf{p}) \text{ such that } t \leq T, \mathbf{p} \in \mathcal{P}\}$$

Lots of very sophisticated, non-scalable techniques developed to compute it using computer geometry, numerical and symbolic analysis, formal methods, etc.

Systematic simulation

- Estimates $\text{Reach}(T, \mathcal{P})$ by computing *bunch of* trajectories from $\mathcal{P}$

Reachability analysis and systematic simulation

Reachable set

Note $\mathbf{x}(t, \mathbf{p})$ the simulation trace obtained using $\mathbf{p}$. We define

$$\mathsf{Reach}(T, \mathcal{P}) = \{\mathbf{x}(t, \mathbf{p}) \text{ such that } t \leq T, \mathbf{p} \in \mathcal{P}\}$$

Lots of very sophisticated, non-scalable techniques developed to compute it using computer geometry, numerical and symbolic analysis, formal methods, etc.

Systematic simulation

- Estimates $\mathsf{Reach}(T, \mathcal{P})$ by computing *bunch of* trajectories from $\mathcal{P}$
- Also known as *Barbaric reachability*

Reachability analysis and systematic simulation

Reachable set

Note $\mathbf{x}(t, \mathbf{p})$ the simulation trace obtained using $\mathbf{p}$. We define

$$\text{Reach}(T, \mathcal{P}) = \{\mathbf{x}(t, \mathbf{p}) \text{ such that } t \leq T, \mathbf{p} \in \mathcal{P}\}$$

Lots of very sophisticated, non-scalable techniques developed to compute it using computer geometry, numerical and symbolic analysis, formal methods, etc.

Systematic simulation

- Estimates Reach$(T, \mathcal{P})$ by computing *bunch of* trajectories from $\mathcal{P}$
- Also known as *Barbaric reachability*
- It works by
  1. Sampling the parameter set $\mathcal{P}$.
  2. Computing and visualizing the simulation traces.

Sampling Parameter Sets

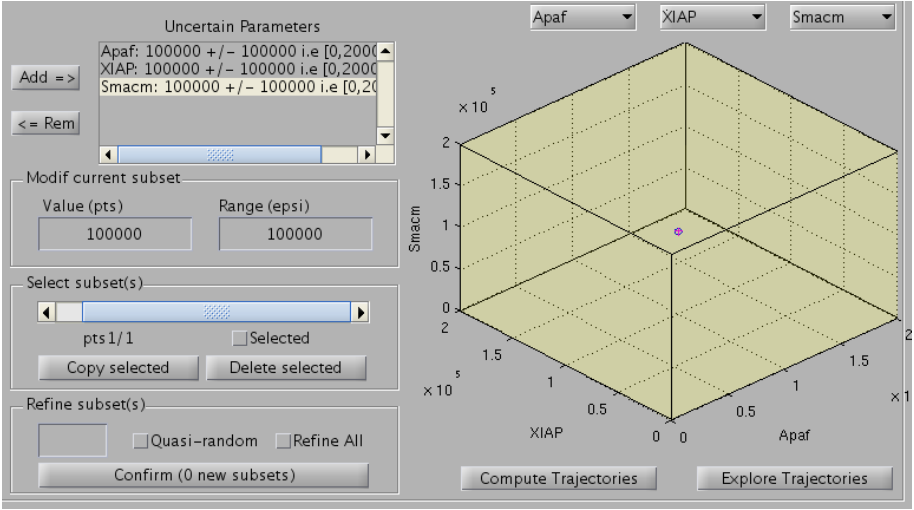
In Breach, parameter sets $\mathcal{P}$ are defined as boxes (hyper-rectangles)

A parameter set can be *refined* into subsets by

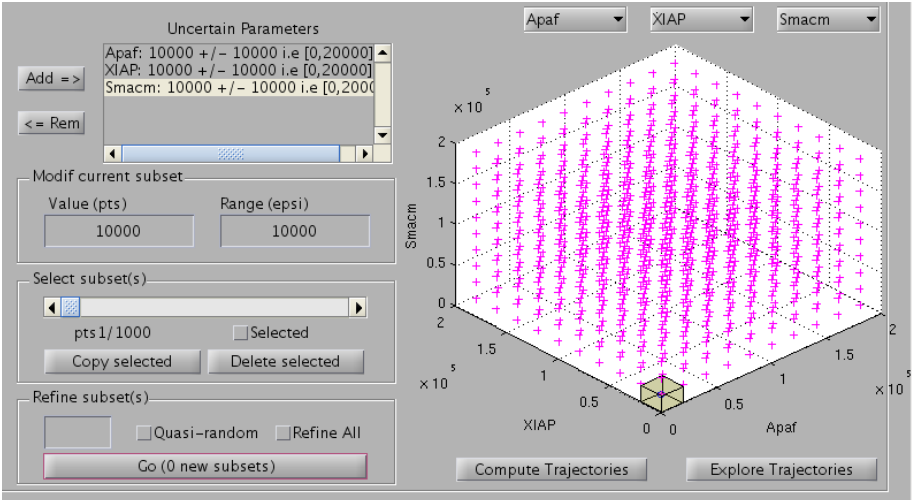
- grid refinement, usually if $\mathcal{P}$ is of low dimension
- quasi-random refinement if $\mathcal{P}$ is high-dimensional

Additionally, the GUI allows to change parameters interactively with automatic recomputation of trajectories

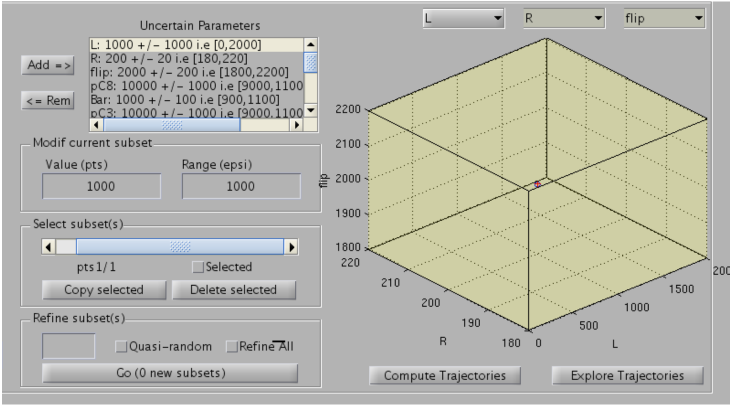
Grid Refinement



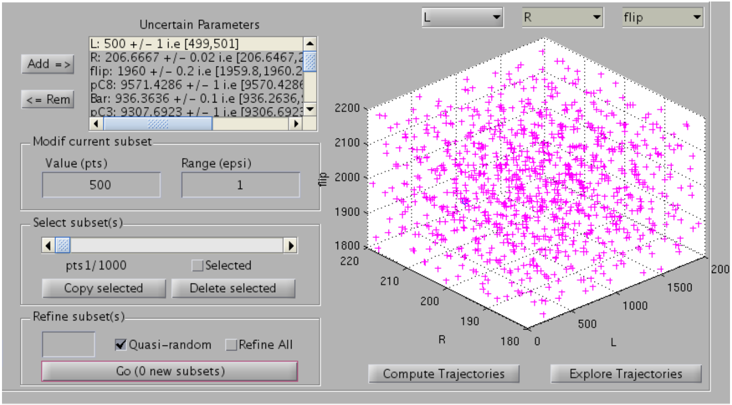
Grid Refinement



Quasi-random Refinement

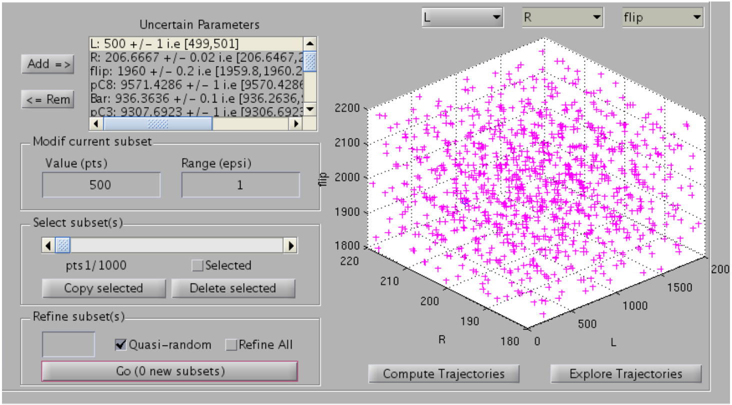


Quasi-random Refinement

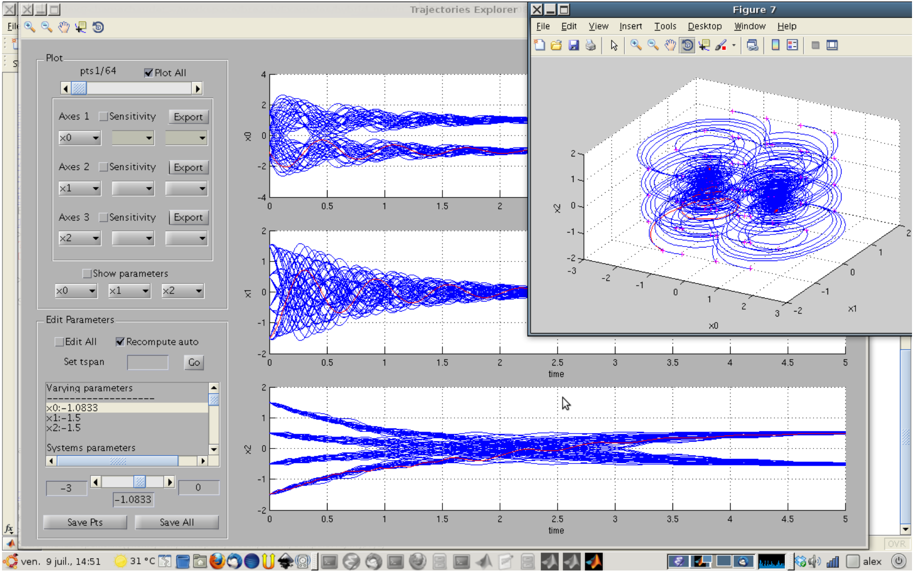


Quasi-random Refinement

Quasi-random provides better repartition than uniform-random sampling



Plotting simulation traces



Barbarians can be sensitive

Sensitivity functions

$s_{ij}(t) = \frac{\partial \mathbf{x}_i}{\partial \mathbf{p}_j}(t)$ can also be computed by CVodes solver

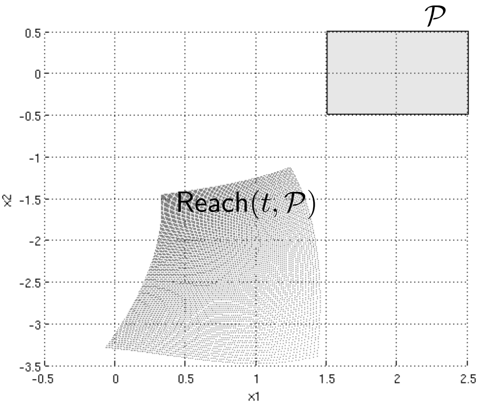
Note $S(t, \mathbf{p}) = (s_{ij}(t))_{i,j}$ is called the sensitivity matrix.

Provides for a cheap estimate of Reach$(t, \mathcal{P})$ by the affine transform of $\mathcal{P}$:[1]

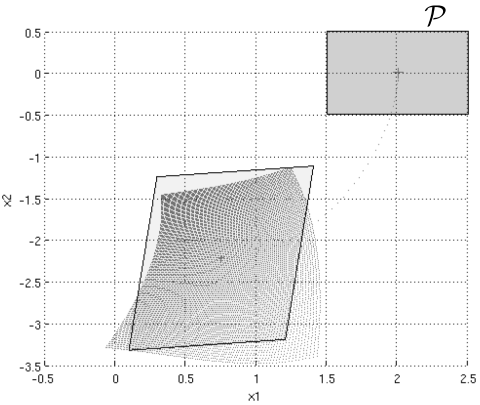
$$\text{Reach}(t, \mathcal{P}) \simeq \mathbf{x}(t, \mathbf{p}_0) + S(t, \mathbf{p}_0) \cdot (\mathcal{P} - \mathbf{p}_0)$$

[1] (*Systematic Simulation Using Sensitivity Analysis* Donzé, Maler, HSCC'07)

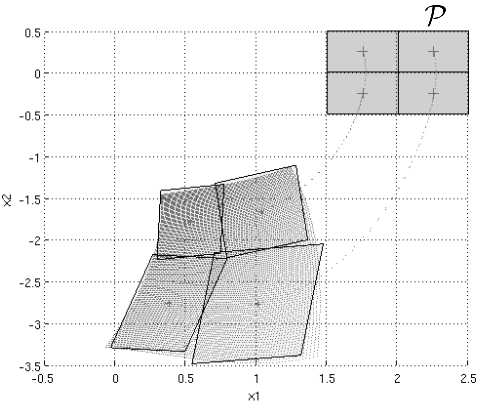
Reachability using sensitivity



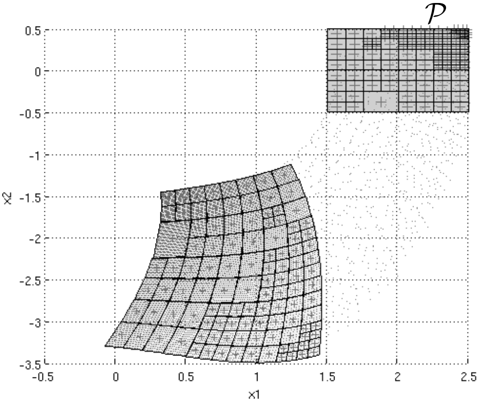
Reachability using sensitivity



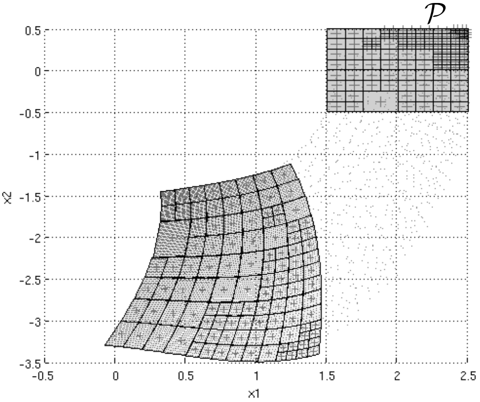
Reachability using sensitivity



Reachability using sensitivity



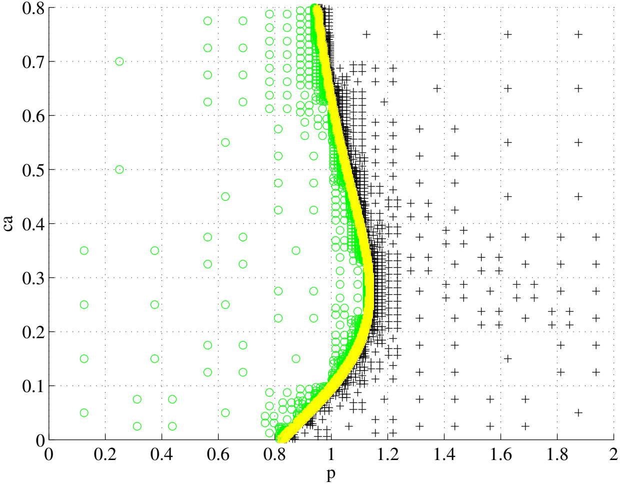
Reachability using sensitivity



- Works well for low dimensional $\mathcal{P}$
- Otherwise, averaging $s_{ij}(t)$ over samplings of $\mathcal{P}$ provides estimates of global sensitivity / robustness

Results on the acute inflammatory response model

Circles lead to health, crosses to death...



Considered parameters are the initial concentrations of pathogen and anti-inflammatory agents

1. Parameter Synthesis
   - Parametric Systems
   - Sensitive Systematic (aka Barbaric) Simulation

2. Parameter Synthesis with Formal Specifications
   - Signal Temporal Logic
   - Property parameters
   - Model parameters

3. Some Results and Concluding Remarks

Temporal logics in a nutshell

Temporal logics specify patterns that timed behaviors of systems may or may not satisfy.

The most intuitive is the Linear Temporal Logic (LTL), dealing with discrete sequences of states.

Based on logic operators ($\neg$, $\wedge$, $\vee$) and temporal operators: "next", "always" (alw), "eventually" (ev) and "until" ($\mathcal{U}$)

Linear Temporal Logic

An LTL formula φ is evaluated on a sequence, e.g., $w = aaabbaaa\ldots$

At each step of w, we can define a truth value of φ, noted $\chi^{\varphi}(w,i)$

LTL atoms are symbols: a, b:

$$\begin{array}{rccccccccc} i = & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \ldots \\ w = & a & a & a & b & b & a & a & a & \ldots \\ \chi^{a}(w,i) = & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & \ldots \\ \chi^{b}(w,i) = & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & \ldots \end{array}$$

LTL, temporal operators

$\bigcirc$ ("next"), alw ("globally"), ev ("eventually") and $\mathcal{U}$ ("until").

They are evaluated at each step wrt the future of sequences

		$w =$	a	a	a	b	b	a	a	a	$\ldots$
$\bigcirc b$	(next)	$\chi^{\bigcirc b}(w,i) =$	0	0	1	1	0	0	0	$?$	$\ldots$
alw a	(always)	$\chi^{\mathsf{alw} a}(w,i) =$	0	0	0	0	0	$1?$	$1?$	$1?$	$\ldots$
ev b	(eventually)	$\chi^{\mathsf{ev} b}(w,i) =$	1	1	1	1	1	$0?$	$0?$	$0?$	$\ldots$
$a\,\mathcal{U}\,\lfloor$	(until)	$\chi^{a\mathcal{U}\lfloor}(w,i) =$	1	1	1	0	0	$0?$	$0?$	$0?$	$\ldots$

LTL, temporal operators

$\bigcirc$ ("next"), alw ("globally"), ev ("eventually") and $\mathcal{U}$ ("until").

They are evaluated at each step wrt the future of sequences

		$w =$	a	a	a	b	b	a	a	a	$\ldots$
$\bigcirc b$	(next)	$\chi^{\bigcirc b}(w,i) =$	0	0	1	1	0	0	0	?	$\ldots$
alw a	(always)	$\chi^{\mathsf{alw} a}(w,i) =$	0	0	0	0	0	1?	1?	1?	$\ldots$
ev b	(eventually)	$\chi^{\mathsf{ev} b}(w,i) =$	1	1	1	1	1	0?	0?	0?	$\ldots$
$a\,\mathcal{U}\,\lfloor$	(until)	$\chi^{a\mathcal{U}\lfloor}(w,i) =$	1	1	1	0	0	0?	0?	0?	$\ldots$

LTL, temporal operators

$\bigcirc$ ("next"), alw ("globally"), ev ("eventually") and $\mathcal{U}$ ("until").

They are evaluated at each step wrt the future of sequences

		$w =$	a	a	a	b	b	a	a	a	$\dots$
$\bigcirc b$	(next)	$\chi^{\bigcirc b}(w,i) =$	0	0	1	1	0	0	0	$0?$	$\dots$
alw a	(always)	$\chi^{\mathsf{alw} a}(w,i) =$	0	0	0	0	0	$1?$	$1?$	$1?$	$\dots$
ev b	(eventually)	$\chi^{\mathsf{ev} b}(w,i) =$	1	1	1	1	1	$0?$	$0?$	$0?$	$\dots$
$a\,\mathcal{U}\,\lfloor$	(until)	$\chi^{a\mathcal{U}\lfloor}(w,i) =$	1	1	1	0	0	$0?$	$0?$	$0?$	$\dots$

LTL, temporal operators

$\bigcirc$ ("next"), alw ("globally"), ev ("eventually") and $\mathcal{U}$ ("until").

They are evaluated at each step wrt the future of sequences

		$w =$	a	a	a	b	b	a	a	a	$\ldots$
$\bigcirc b$	(next)	$\chi^{\bigcirc b}(w,i) =$	0	0	1	1	0	0	0	$?$	$\ldots$
alw a	(always)	$\chi^{\mathsf{alw} a}(w,i) =$	0	0	0	0	0	$1?$	$1?$	$1?$	$\ldots$
ev b	(eventually)	$\chi^{\mathsf{ev} b}(w,i) =$	1	1	1	1	1	$0?$	$0?$	$0?$	$\ldots$
$a\,\mathcal{U}\,\lfloor$	(until)	$\chi^{a\mathcal{U}\lfloor}(w,i) =$	1	1	1	0	0	$0?$	$0?$	$0?$	$\ldots$

From LTL to STL

Extension of LTL with real-time and real-valued constraints

From LTL to STL

Extension of LTL with real-time and real-valued constraints

Ex: request-grant property

LTL G(r => F g)

Boolean predicates, discrete-time

From LTL to STL

Extension of LTL with real-time and real-valued constraints

Ex: request-grant property

LTL G(r => F g)
Boolean predicates, discrete-time

MTL G(r => $F_{[0,.5s]}$ g)
Boolean predicates, real-time

From LTL to STL

Extension of LTL with real-time and real-valued constraints

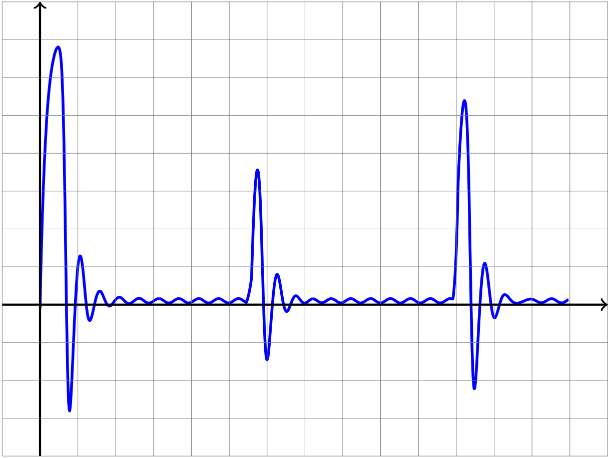
Ex: request-grant property

LTL G(r $=>$ F g)
Boolean predicates, discrete-time

MTL G(r $=>$ $\mathsf{F}_{[0,.5s]}$ g)
Boolean predicates, real-time

STL G($x[t] > 0 =>$ $\mathsf{F}_{[0,.5s]} y[t] > 0$)
Predicates over real values , real-time

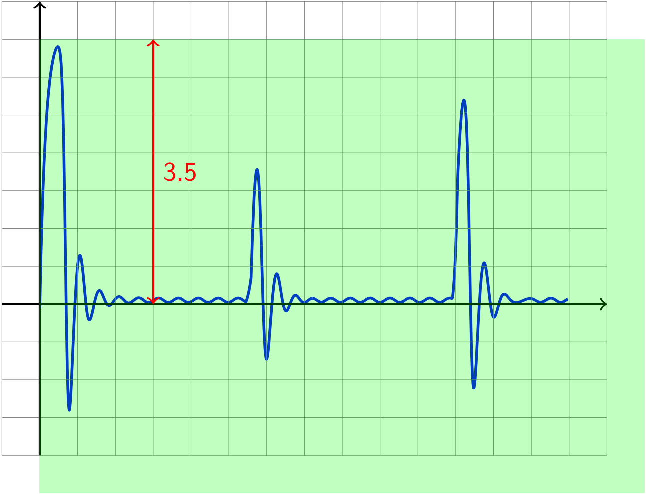
STL examples



STL examples

The signal is never above 3.5

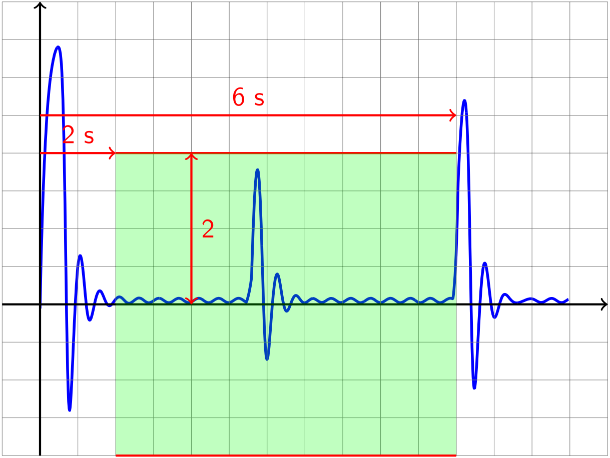
$$\varphi := \mathsf{alw}\ (x[t] < 3.5)$$



STL examples

Between 2s and 6s the signal is between -2 and 2

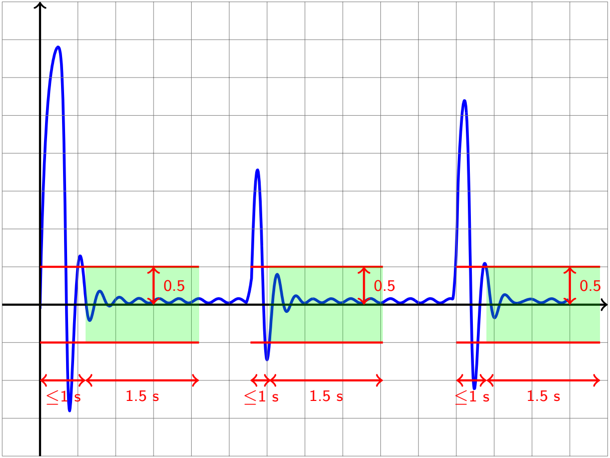
$$\varphi := \text{alw}_{[2,6]} \ (|x[t]| < 2)$$



STL examples

Always $|x| > 0.5 \Rightarrow$ *after 1 s,* $|x|$ *settles under 0.5 for 1.5 s*

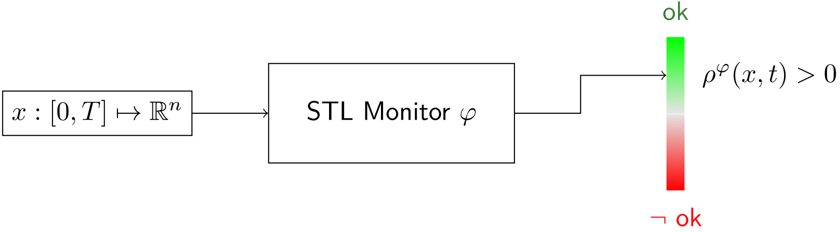
$$\varphi := \mathsf{alw}(x[t] > .5 \rightarrow \ \mathsf{ev}_{[0,.6]} \ \ (\ \mathsf{alw}_{[0,1.5]} \ x[t] < 0.5))$$



STL Robust Semantics

Given φ, x and t, the **quantitative satisfaction function** ρ is such that:

$$\rho^{\varphi}(x,t) > 0 \Rightarrow x,t \models \varphi$$
$$\rho^{\varphi}(x,t) < 0 \Rightarrow x,t \not\models \varphi$$



Quantitative Satisfaction, Example

Quantitative Satisfaction, Example

Between 2s and 6s the signal is between -2.5 and 2.5

$$\varphi := \mathsf{alw}_{[2,6]}\ (|x[t]| < 2.5)$$

Quantitative Satisfaction, Example

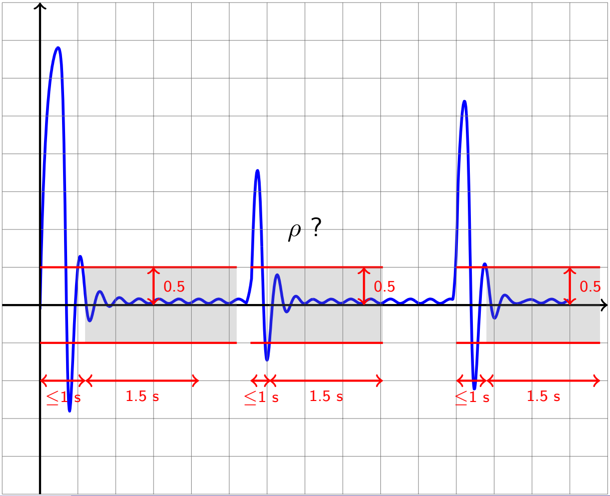
Between 2s and 6s the signal is between -1 and -1

$$\varphi := \mathsf{alw}_{[2,6]}\ (|x[t]| < 2.5)$$

Quantitative Satisfaction, Example

Always $|x|>0.5 \Rightarrow$ *after 1 s,* $|x|$ *settles under 0.5 for 1.5 s*

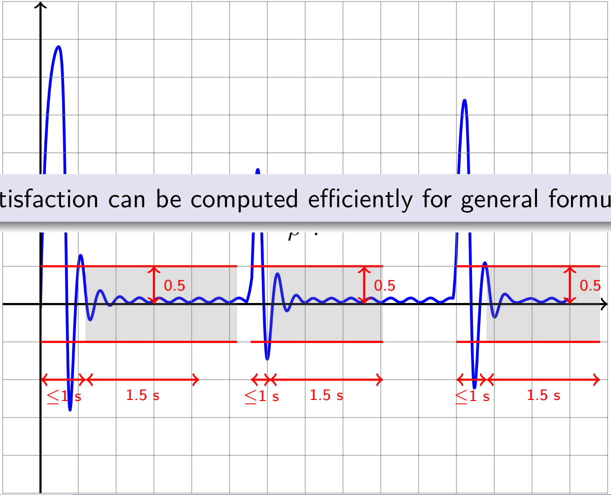
$$\varphi := \mathsf{alw}(x[t] > .5 \rightarrow \mathsf{ev}_{[0,1.]}\ (\mathsf{alw}_{[0,1.5]} x[t] < 0.5))$$



Quantitative Satisfaction, Example

Always $|x| > 0.5 \Rightarrow$ *after 1 s,* $|x|$ *settles under 0.5 for 1.5 s*

$$\varphi := \mathsf{alw}(x[t] > .5 \rightarrow \mathsf{ev}_{[0,1.]}\ (\mathsf{alw}_{[0,1.5]} x[t] < 0.5))$$



Robust satisfaction can be computed efficiently for general formulas

Computing the robust satisfaction function

(Donze, Ferrere, Maler, *Efficient Robust Monitoring of STL Formula*, CAV'13)

- The function $\rho^{\varphi}(x,t)$ is computed inductively on the structure of φ
  - linear time complexity in size of x is preserved
  - exponential worst case complexity in the size of φ
- Atomic transducers compute in linear time in the size of the input
  - Key idea is to exploit efficient streaming algorithm (Lemire's) computing the max and min over a moving window

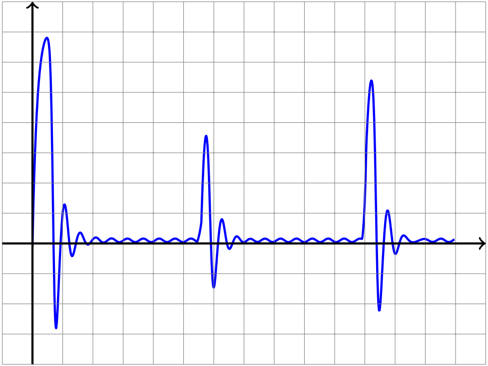
1. Parameter Synthesis
   - Parametric Systems
   - Sensitive Systematic (aka Barbaric) Simulation

2. Parameter Synthesis with Formal Specifications
   - Signal Temporal Logic
   - **Property parameters**
   - Model parameters

3. Some Results and Concluding Remarks

Parametric-STL Formulas

STL formula where numeric constants are left unspecified.

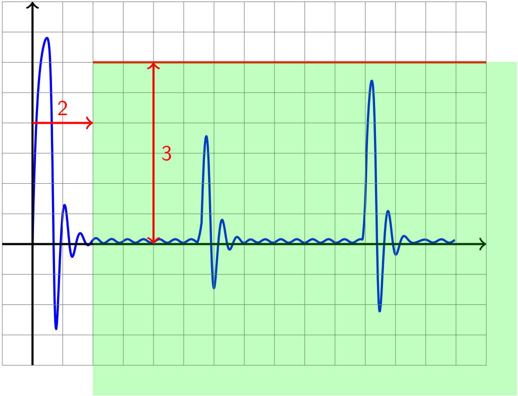


Parametric-STL Formulas

STL formula where numeric constants are left unspecified.

"After 2s, the signal is never above 3"

$$\varphi := \ \mathsf{ev}_{[2,\infty]} \ \ (x[t] < 3)$$

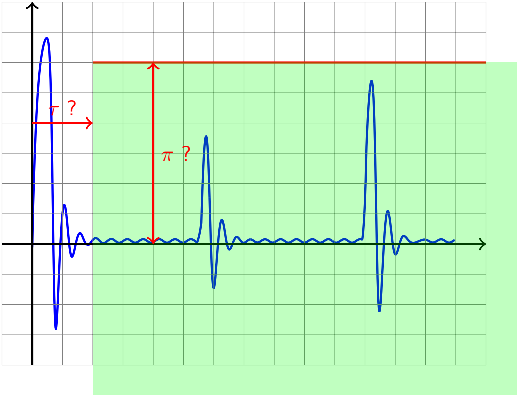


Parametric-STL Formulas

STL formula where numeric constants are left unspecified.

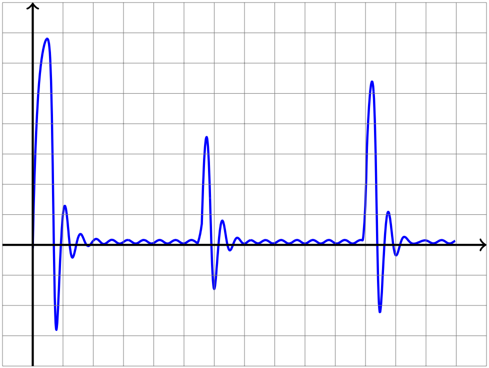
"After τ s, the signal is never above π"

$$\varphi := \text{alw}_{[\tau,\infty]} \ (x[t] < \pi)$$



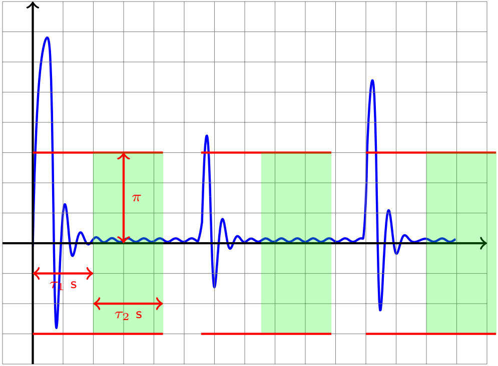
Parameter synthesis for PSTL

- In general, looking for "tight" valuations
- E.g., $\varphi := \mathsf{alw}\,\big(x[t] > \pi \rightarrow \ \mathsf{ev}_{[0,\tau_1]}\ \ (\ \mathsf{alw}_{[0,\tau_2]}\ x[t] < \pi)\big)$



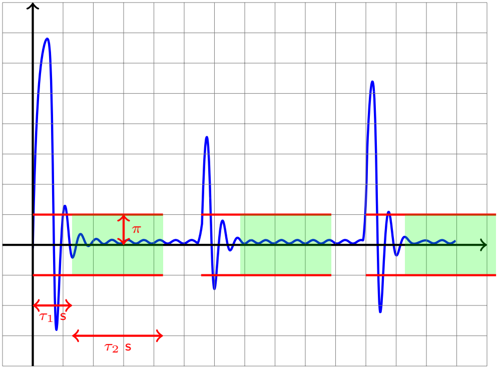
Parameter synthesis for PSTL

- In general, looking for "tight" valuations
- E.g., $\varphi := \mathsf{alw}\left(x[t] > \pi \rightarrow \mathsf{ev}_{[0,\tau_1]}\ \left(\mathsf{alw}_{[0,\tau_2]}\ x[t] < \pi\right)\right)$
  - Valuation 1: $\pi \leftarrow 1.5$, $\tau_1 \leftarrow 1\ s$, $\tau_2 \leftarrow 1.15\ s$



Parameter synthesis for PSTL

- In general, looking for "tight" valuations
- E.g., $\varphi := \text{alw}\left(x[t] > \pi \rightarrow \text{ev}_{[0,\tau_1]}\ \left(\text{alw}_{[0,\tau_2]}\ x[t] < \pi\right)\right)$
  - Valuation 1: $\pi \leftarrow 1.5$, $\tau_1 \leftarrow 1\ s$, $\tau_2 \leftarrow 1.15\ s$
  - Valuation 2 (tight): $\pi \leftarrow .5$, $\tau_1 \leftarrow 0.65\ s$, $\tau_2 \leftarrow 2\ s$



Parameter synthesis for PSTL

Challenges

- Multiple solutions: which one to chose ?
- Tightness implies to "optimize" the valuation $v(p_i)$ for each p_i

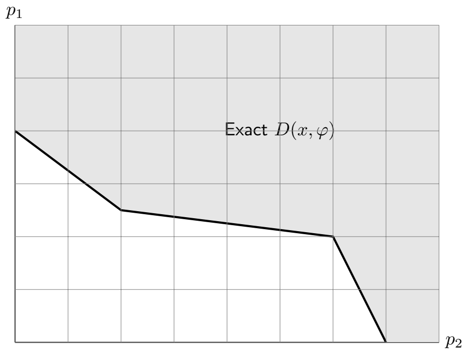
The problem can be simplified if the formula is *monotonic* in each p_i, i.e.,

- If the formula holds for p_i, then it will hold for $p_i' > p_i$, or
- if the formula holds for p_i, then it will hold for $p_i' < p_i$

If the formula is not monotonic, parameters can be treated as a system parameters (next section).

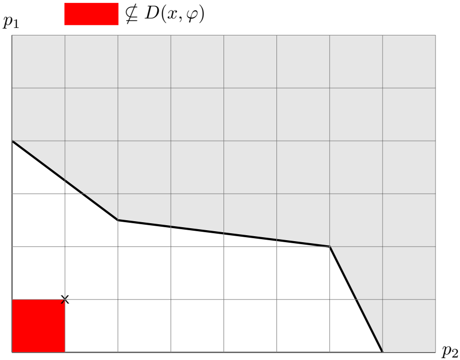
Monotonic validity domains

- The validity domain D of φ and x is the set of valuations v s.t. $x \models \varphi(v)$
- A tight valuation is a valuation in D close to its boundary ∂D
- In case of monoticity, ∂D has the structure of a Pareto front which can be estimated with generalized binary search heuristics



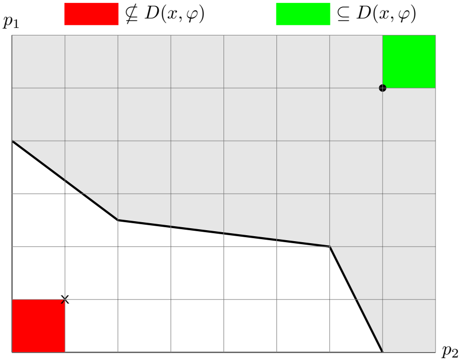
Monotonic validity domains

- The validity domain D of φ and x is the set of valuations v s.t. $x \models \varphi(v)$
- A tight valuation is a valuation in D close to its boundary ∂D
- In case of monoticity, ∂D has the structure of a Pareto front which can be estimated with generalized binary search heuristics



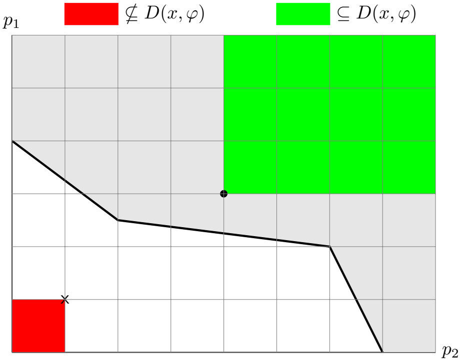
Monotonic validity domains

- The validity domain D of φ and x is the set of valuations v s.t. $x \models \varphi(v)$
- A tight valuation is a valuation in D close to its boundary ∂D
- In case of monoticity, ∂D has the structure of a Pareto front which can be estimated with generalized binary search heuristics



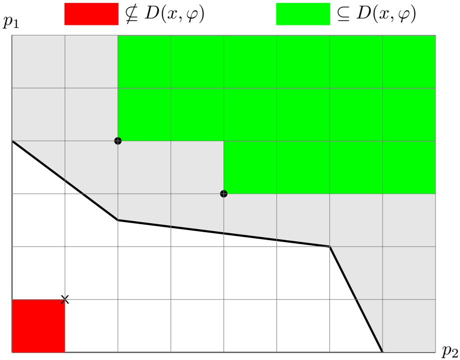
Monotonic validity domains

- The validity domain D of φ and x is the set of valuations v s.t. $x \models \varphi(v)$
- A tight valuation is a valuation in D close to its boundary ∂D
- In case of monoticity, ∂D has the structure of a Pareto front which can be estimated with generalized binary search heuristics



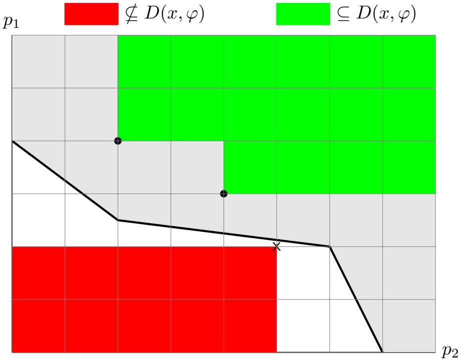
Monotonic validity domains

- The validity domain D of φ and x is the set of valuations v s.t. $x \models \varphi(v)$
- A tight valuation is a valuation in D close to its boundary ∂D
- In case of monoticity, ∂D has the structure of a Pareto front which can be estimated with generalized binary search heuristics



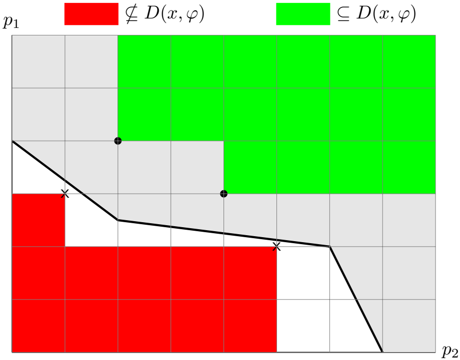
Monotonic validity domains

- The validity domain D of φ and x is the set of valuations v s.t. $x \models \varphi(v)$
- A tight valuation is a valuation in D close to its boundary ∂D
- In case of monoticity, ∂D has the structure of a Pareto front which can be estimated with generalized binary search heuristics



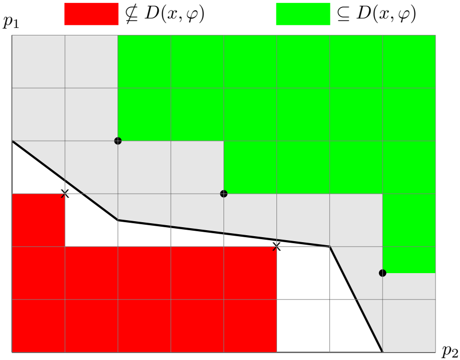
Monotonic validity domains

- The validity domain D of φ and x is the set of valuations v s.t. $x \models \varphi(v)$
- A tight valuation is a valuation in D close to its boundary ∂D
- In case of monoticity, ∂D has the structure of a Pareto front which can be estimated with generalized binary search heuristics



Monotonic validity domains

- The validity domain D of φ and x is the set of valuations v s.t. $x \models \varphi(v)$
- A tight valuation is a valuation in D close to its boundary ∂D
- In case of monoticity, ∂D has the structure of a Pareto front which can be estimated with generalized binary search heuristics



Parameter synthesis problem

Problem

Given the system:

$$p \longrightarrow \boxed{\textit{System } \mathcal{S}} \longrightarrow \mathcal{S}(u(t), p)$$

Find $p \in \mathcal{P}$ *such that* $S(u(t), p), 0 \models \varphi$

Parameter synthesis problem

Problem

Given the system:

$$p \longrightarrow \boxed{\textit{System } \mathcal{S}} \longrightarrow \mathcal{S}(u(t), p)$$

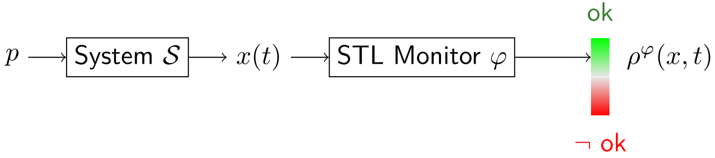
Find $p \in \mathcal{P}$ *such that* $S(u(t), p), 0 \models \varphi$

Main idea

Guide the search of a solution using the quantitative measure of satisfaction of φ

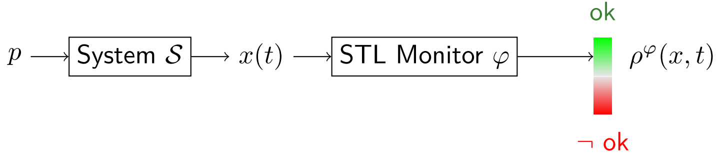
Parameter synthesis with quantitative satisfaction

Given a formula φ, a signal x and a time t, we can compute $\rho^{\varphi}(\mathbf{x}(p), t)$



Parameter synthesis with quantitative satisfaction

Given a formula φ, a signal x and a time t, we can compute $\rho^{\varphi}(\mathbf{x}(p), t)$



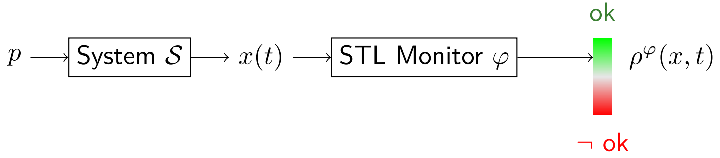
The synthesis problem can be reduced to solving

$$\rho^* = \max_{p \in \mathcal{P}} \rho^{\varphi}(x, 0), \text{ with } p^* = \arg\max_{p \in \mathcal{P}} \rho^{\varphi}(x, 0)$$

If $\rho^* > 0$, we found a parameter value, "maximally robust".

Parameter synthesis with quantitative satisfaction

Given a formula φ, a signal x and a time t, we can compute $\rho^{\varphi}(\mathbf{x}(p), t)$



The synthesis problem can be reduced to solving

$$\rho^* = \max_{p \in \mathcal{P}} \rho^{\varphi}(x, 0), \text{ with } p^* = \arg\max_{p \in \mathcal{P}} \rho^{\varphi}(x, 0)$$

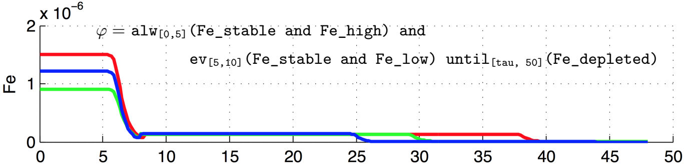
If $\rho^* > 0$, we found a parameter value, "maximally robust".

Actual robustness can be further assessed by

- Explore a neighborhood of p^*
- Compute different local and global sensitivity analysis (work by Mobilia, Fanchon et al, applied to iron homeostasis

Example [1]: modeling iron homeostasis

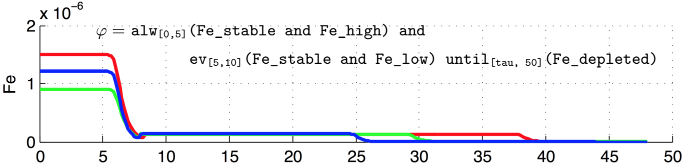
- Specifications
  Qualitative knowledge, quantitative measurements, partially formalizable



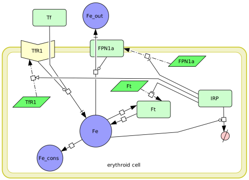
[1](joint work with N. Mobilia, E. Fanchon, J-M Moulis et al)

Example [1]: modeling iron homeostasis

- Specifications
  Qualitative knowledge, quantitative measurements, partially formalizable



- Model

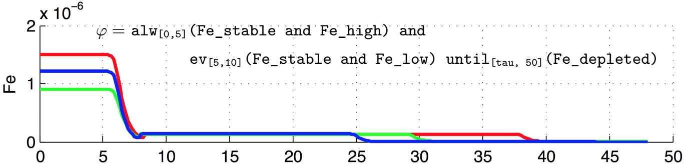


$$\frac{d}{dt}Fe = k_1\, TfR1\ Tf - k_2 Fe\ FPN1a$$
$$+k_3 Fe$$

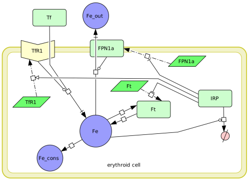
[1] (joint work with N. Mobilia, E. Fanchon, J-M Moulis et al)

Example [1]: modeling iron homeostasis

- Specifications
  Qualitative knowledge, quantitative measurements, partially formalizable



- Model

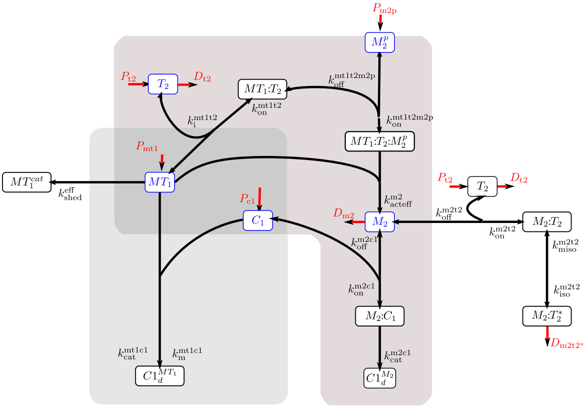


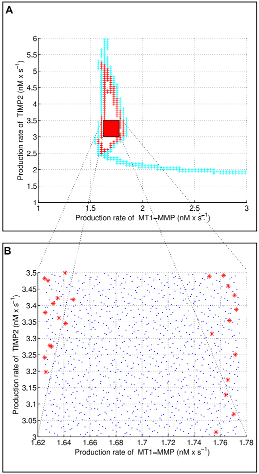
$$\frac{d}{dt}Fe = k_1\, TfR1\ Tf - k_2 Fe\ FPN1a + k_3 Fe$$

Problem: values for k_1, k_2, k_3, etc

[1] (joint work with N. Mobilia, E. Fanchon, J-M Moulis et al)

Angiogenesis[2]

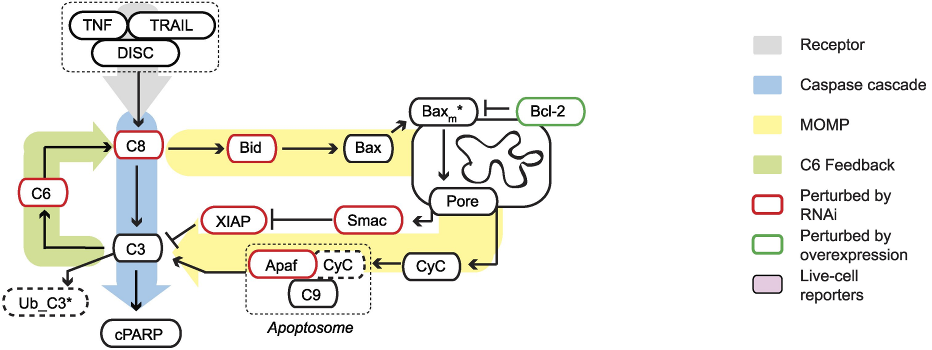




- Found values for protein production rates leading to oscillations

[2] (Donzé,Fanchon,Gattepaille,Maler,Tracqui, PloS One, 2011)

Apoptosis [3]



- Transient "race" conditions between direct and mito. path define cell type
- Formalized three definitions (e.g. "(not dead) until (MOMP)")
- Found contradiction between model prediction and experiments, tuned parameters to fix consistency

[3] (Stoma, Donzé, Maler, Bertaux, Batt, Plos Comp. Bio. 2013)

Conclusion and future work

- Advocated early simulation and visualization
- Simple, manual, coarse sampling and manipulation of parameter values can often provide quickly great deal of information
  - Breach was designed for this
- Then, harness the "right" optimization function with the "right" optimization algorithm
- Quantitative satisfaction of STL formulas is an appealing idea but
  - Difficult optimization problem in general: non-linear, non-smooth
  - Actual robustness of the obtained solution is not easy to estimate either

Concluding Remarks (From Oded's opening of TSB 2011)

Towards a Paper without the word *Towards* in the Title

O. Maler and A. Pnueli

Abstract. With the advent of post-genomic buzzword-driven big science a need was felt to moderate the level of hype and optimistic false promises in scientific papers and research proposals. In this paper we do not provide a comprehensive and complete solution to the problem mentioned in the title but nevertheless make a promising step towards the fulfillment of the goal.

- The word towards indicates that we are not there
- But where is there?

Concluding Remarks: Where is There?

Goal (say): handing a tool to biologists allowing to probe systems simulation or data with intuitive, biologically relevant requirements

But

- Modeling is still a huge problem
- Even when modeling is (somewhat) figured out:
  - Specification language standard?
  - Training users?
- More collaborations are needed...

Concluding Remarks: Where is There?

Far.

Concluding Remarks: Where is There?

Far.

But to some significant extent, Oded showed the way.

Inter-(disciplinary/domain) cooperation, wet/data biologists need modeling, maths, physics, and CS tools

Open-mindness and ability to gather people around original projects using cynical views if necessary is key and one of Oded greatest contribution to the field in my opinion