Trace Simplications preserving Temporal Logic Formulae with Case Study in a Coupled Model of the Cell Cycle and the Circadian Clock

CMSB 2014

Pauline Traynard and Francois Fages and Sylvain Soliman
Modeling the coupling between the cell cycle and the circadian clock

**Context:** Optimizing cancer treatment with chronotherapy

Experimental observations on periods and phases suggest bidirectional influence between cell divisions and the autonomous cellular circadian clock.

What are the mechanisms behind these observations?

**Model building** assisted with formal methods (model calibration)

**Predictions:** mechanisms and perturbations, optimization

**Bidirectional coupled model** of the cell cycle and the circadian clock
Temporal logic specifications

- Dynamical behaviors for oscillatory systems:
  - period, amplitude, phase
  - oscillations regularity

- Formalised with temporal logic:
  Ex: \( \text{period} \quad \phi = \exists (t_1, t_2) \mid p = t_2 - t_1 \land t_1 < t_2 \)
  \[ \land F( \frac{dA}{dt} > 0 \land X( \frac{dA}{dt} \leq 0 \land \text{Time} = t_1 )) \]
  \[ \land F( \frac{dA}{dt} > 0 \land X( \frac{dA}{dt} \leq 0 \land \text{Time} = t_2 )) \]

- Applications:
  - Data analysis: extracting meaningful information from a trace
  - Model checking: verifying that a model satisfies some constraints
  - Model analysis: comparing how the properties of a model evolve when some parameters vary
  - Parameter inference: continuous satisfaction degree of a temporal logic formula, powerful optimization algorithm CMA-ES

Result:
- Possible values: \( p = 23 \mid p = 24.5 \)
- Satisfaction degree (with objective \( p = 24 \)): 0.1

http://lifeware.inria.fr/Biocham/
Validation domain computing algorithm

Generic algorithm:

• **Decomposition** of $\phi$ in **sub-formulas**

• For each constraint and each time point, computing of a **domain** of possible variables

• **Combination** of the subdomains with the logical operators:
  
  - Operator $F$ (finally) → union: $D_{s_i}^T, F_\phi = \bigcup_{j=i}^n D_{s_j}^T, \phi$

  - Operator $G$ (globally) → intersection: $D_{s_i}^T, G_\phi = \bigcap_{j=i}^n D_{s_j}^T, \phi$

  - Operator $X$ (next) → next domain if valid: $D_{s_i}^T, X_\phi = D_{s_{i+1}}^T, \phi$

  - Operator $U$ (until) → union of intersections: $D_{s_i}^T, \phi U \psi = \bigcup_{j=i}^n (D_{s_j}^T, \psi \cap \bigcap_{k=i}^{j-1} D_{s_k}^T, \phi)$

• Domain for $\phi =$ combinated domain for the **first point** of the trace

\[ \phi = F( [A] > s ) \]

\[ D_\phi = \{ s < \text{max}[A] \} \]

**Computational cost:** up to $O(n^v)$

($v =$ number of variables)

How to find a simplified trace that will keep the same validity domain?
Validation domain computing algorithm

Generic algorithm:

- **Decomposition** of $\phi$ in **sub-formulas**
- For each constraint and each time point, computing of a **domain** of possible variables
- **Combination** of the subdomains with the logical operators:
  - Operator $F$ (finally) $\rightarrow$ union: $D_{s_i}^T, F_\phi = \bigcup_{j=i}^{n} D_{s_j}^T, \phi$
  - Operator $G$ (globally) $\rightarrow$ intersection: $D_{s_i}^T, G_\phi = \bigcap_{j=i}^{n} D_{s_j}^T, \phi$
  - Operator $X$ (next) $\rightarrow$ next domain if valid: $D_{s_i}^T, X_\phi = D_{s_{i+1}}^T, \phi$
  - Operator $U$ (until) $\rightarrow$ union of intersections: $D_{s_i}^T, \phi U_\psi = \bigcup_{j=i}^{n} \left( D_{s_j}^T, \psi \cap \bigcap_{k=i}^{j-1} D_{s_k}^T, \phi \right)$
- Domain for $\phi =$ combined domain for the **first point** of the trace

$$\phi = F( [A] > s )$$

$$D_\phi = \{ s < \text{max}[A] \}$$

**Computational cost:** up to $O(n^v)$
($v =$ number of variables)

How to find a simplified trace that will keep the same validity domain?
Validation domain computing algorithm

Generic algorithm:

- **Decomposition** of $\phi$ in **sub-formulas**
- For each constraint and each time point, computing of a **domain** of possible variables
- **Combination** of the subdomains with the logical operators:
  - Operator $F$ (finally) $\rightarrow$ union: $\mathcal{D}^T_{s_i, F\phi} = \bigcup_{j=i}^{n} \mathcal{D}^T_{s_j, \phi}$
  - Operator $G$ (globally) $\rightarrow$ intersection: $\mathcal{D}^T_{s_i, G\phi} = \bigcap_{j=i}^{n} \mathcal{D}^T_{s_j, \phi}$
  - Operator $X$ (next) $\rightarrow$ next domain if valid: $\mathcal{D}^T_{s_i, X\phi} = \mathcal{D}^T_{s_{i+1}, \phi}$
  - Operator $U$ (until) $\rightarrow$ union of intersections: $\mathcal{D}^T_{s_i, \phi U \psi} = \bigcup_{j=i}^{n} (\mathcal{D}^T_{s_j, \psi} \cap \bigcap_{k=i}^{j-1} \mathcal{D}^T_{s_k, \phi})$
- Domain for $\phi = $ combined domain for the **first point** of the trace

\[ \phi = F( [A] > s ) \]

\[ D_\phi = \{ s < \text{max}[A] \} \]

**Computational cost:** up to $O(n^v)$

$(v = \text{number of variables})$

How to find a simplified trace that will keep the same validity domain?
Validation domain computing algorithm

Generic algorithm:

• **Decomposition** of \( \phi \) in **sub-formulas**

• For each constraint and each time point, computing of a **domain** of possible variables

• **Combination** of the subdomains with the logical operators:
  
  - Operator \( F \) (finally) \( \rightarrow \) union:  
    \[ D_{s_i}^T, F \phi = \bigcup_{j=i}^n D_{s_j}^T, \phi \]

  - Operator \( G \) (globally) \( \rightarrow \) intersection:  
    \[ D_{s_i}^T, G \phi = \bigcap_{j=i}^n D_{s_j}^T, \phi \]

  - Operator \( X \) (next) \( \rightarrow \) next domain if valid:  
    \[ D_{s_i}^T, X \phi = D_{s_i+1}^T, \phi \]

  - Operator \( U \) (until) \( \rightarrow \) union of intersections:  
    \[ D_{s_i}^T, \phi U \psi = \bigcup_{j=i}^n (D_{s_j}^T, \psi \cap \bigcap_{k=i}^{j-1} D_{s_k}^T, \phi) \]

• Domain for \( \phi = \) combined domain for the **first point** of the trace

\[ \phi = F( [A] > s ) \]

\[ D_{\phi} = \{ s < \text{max}[A] \} \]

**Computational cost:** up to \( O(n^v) \)

\((v = \text{number of variables})\)

**How to find a simplified trace that will keep the same validity domain?**
Validation domain computing algorithm

Generic algorithm:
- **Decomposition** of $\phi$ in **sub-formulas**
- For each constraint and each time point, computing of a **domain** of possible variables
- **Combination** of the subdomains with the logical operators:
  - Operator $F$ (finally) $\rightarrow$ union: $D_s^T, F_\phi = \bigcup_{j=i}^n D_s^T, \phi$
  - Operator $G$ (globally) $\rightarrow$ intersection: $D_s^T, G_\phi = \bigcap_{j=i}^n D_s^T, \phi$
  - Operator $X$ (next) $\rightarrow$ next domain if valid: $D_s^T, X_\phi = D_{s_{i+1}}^T, \phi$
  - Operator $U$ (until) $\rightarrow$ union of intersections: $D_s^T, \phi U_\psi = \bigcup_{j=i}^n (D_{s_j}^T, \psi \cap \bigcap_{k=i}^{j-1} D_{s_k}^T, \phi)$
- Domain for $\phi = \text{combined domain}$ for the **first point** of the trace

Computational cost: up to $O(n^v)$
($v =$ number of variables)

How to find a simplified trace that will keep the same validity domain?
Validation domain computing algorithm

Generic algorithm:

- **Decomposition** of $\phi$ in sub-formulas
- For each constraint and each time point, computing of a **domain** of possible variables
- **Combination** of the subdomains with the logical operators:
  - Operator $F$ (finally) → union: $\mathcal{D}_{s_i}^T, F\phi = \bigcup_{j=i}^{n} \mathcal{D}_{s_j}^T, \phi$
  - Operator $G$ (globally) → intersection: $\mathcal{D}_{s_i}^T, G\phi = \bigcap_{j=i}^{n} \mathcal{D}_{s_j}^T, \phi$
  - Operator $X$ (next) → next domain if valid: $\mathcal{D}_{s_i}^T, X\phi = \mathcal{D}_{s_i+1}^T, \phi$
  - Operator $U$ (until) → union of intersections: $\mathcal{D}_{s_i}^T, \phi U \psi = \bigcup_{j=i}^{n} (\mathcal{D}_{s_j}^T, \psi \cap \bigcap_{k=i}^{j-1} \mathcal{D}_{s_k}^T, \phi)$
- Domain for $\phi =$ combined domain for the **first point** of the trace

**Computational cost:** up to $O(n^v)$
($v =$ number of variables)

How to find a simplified trace that will keep the same validity domain?
Generic algorithm:

- **Decomposition** of $\phi$ in sub-formulas
- For each constraint and each time point, computing of a **domain** of possible variables
- **Combination** of the subdomains with the logical operators:
  - Operator $F$ (finally) → union: $\mathcal{D}^T_{s_i}, F_\phi = \bigcup_{j=i}^n \mathcal{D}^T_{s_j}, \phi$
  - Operator $G$ (globally) → intersection: $\mathcal{D}^T_{s_i}, G_\phi = \bigcap_{j=i}^n \mathcal{D}^T_{s_j}, \phi$
  - Operator $X$ (next) → next domain if valid: $\mathcal{D}^T_{s_i}, X_\phi = \mathcal{D}^T_{s_i+1}, \phi$
  - Operator $U$ (until) → union of intersections: $\mathcal{D}^T_{s_i}, \phi U_\psi = \bigcup_{j=i}^n \left( \mathcal{D}^T_{s_j}, \psi \cap \bigcap_{k=i}^{j-1} \mathcal{D}^T_{s_k}, \phi \right)$
- Domain for $\phi =$ combinated domain for the **first point** of the trace

**Validation domain computing algorithm**

\[ \phi = F( [A] > s ) \]

\[ D_\phi = \{ s < \text{max}[A] \} \]

**Computational cost:** up to $O(n^v)$  
($v = \text{number of variables}$)

How to find a simplified trace that will keep the same validity domain?
Validation domain computing algorithm

Dedicated solvers:
- Specific function for a dynamical behavior
- Direct computing of the validity domain on the trace

**Specification:**
\[
\phi = \exists(t_1, t_2) \mid p = t_2 - t_1 \\
\wedge \mathbf{F}(\frac{dA}{dt}) > 0 \wedge \mathbf{X}(\frac{dA}{dt}) \leq 0 \wedge \text{Time} = t_1 \\
\wedge (\frac{dA}{dt} \leq 0) \mathbf{U}(\frac{dA}{dt} > 0 \\
\wedge ((\frac{dA}{dt} > 0) \mathbf{U}(\frac{dA}{dt} \leq 0 \wedge \text{Time} = t_2))))
\]

**Result:**
\[
p = 23 \mid p = 24.5
\]

**Computational cost:**
\[
O(n^2) \quad \text{vs} \quad O(n)
\]

---

Trace simplification

For the case when there is no dedicated solver, how to make the generic algorithm more efficient?

Trace simplification: local extrema

Under which condition on the constraints is it safe to use this simplification?
Proof of validity: peak and period

Peak

**Formula:** \( \phi = F(\frac{dA}{dt} > 0 \land X(\frac{dA}{dt} \leq 0 \land Time = t)) \).

\[ \mathcal{D}_{T, \phi} = \mathcal{D}_{s_0, \phi}^{T} \]

\[ = \bigcup_{i=0}^{n} (\mathcal{D}_{s_i}^{T}, \frac{dA}{dt} > 0) \cap (\mathcal{D}_{s_{i+1}}^{T}, \frac{dA}{dt} \leq 0) \cap (\mathcal{D}_{s_{i+1}}^{T}, Time = t) \]

\[ = \bigcup_{i \in \{0, \ldots, n\}} (\mathcal{D}_{s_i+1}^{T}, Time = t) \]

\[ i \in \{0, \ldots, n\} \land (\frac{dA}{dt})_{s_i} > 0 \land (\frac{dA}{dt})_{s_{i+1}} \leq 0 \]

\[ = \bigcup_{i \in \{0, \ldots, n\}} \{Time_{s_{i+1}}\} \]

\[ i \in \{0, \ldots, n\} \land (\frac{dA}{dt})_{s_i} > 0 \land (\frac{dA}{dt})_{s_{i+1}} \leq 0 \]

Period

**Formula:** \( \phi = \exists(t_1, t_2) \mid p = t_2 - t_1 \land t_1 < t_2 \)

\[ \land F(\frac{dA}{dt} > 0 \land X(\frac{dA}{dt} \leq 0 \land Time = t_1)) \]

\[ \land F(\frac{dA}{dt} > 0 \land X(\frac{dA}{dt} \leq 0 \land Time = t_2)) \]

\[ = t_3 \land \neg \exists t_3 \mid t_1 < t_3 < t_2 \land F(\frac{dA}{dt} > 0 \land X(\frac{dA}{dt} \leq 0 \land Time = \text{t3})) \]

Trace simplification:
The optimal trace simplification is \( T_J \) with \( J = \{i, i + 1 \in \{0, \ldots, n\} \mid \frac{dA}{dt}_{s_i} > 0 \land \frac{dA}{dt}_{s_{i+1}} \leq 0\} \)  

\( T^e_A \) is a simplification of \( T \) for \( \phi \).
Proof of validity: peak and period

**Peak**

Formula: \( \phi = F\left( \frac{dA}{dt} > 0 \land X\left( \frac{dA}{dt} \leq 0 \land Time = t \right) \right) \)

Trace simplification:
The optimal trace simplification is \( T_J \) with \( J = \{ i, i + 1 \in \{ 0, \ldots, n \} \mid \frac{dA}{dt}_{s_i} > 0 \land \frac{dA}{dt}_{s_{i+1}} \leq 0 \} \)

\( T^e_A \) is a simplification of \( T \) for \( \phi \).

**Period**

Formula: \( \phi = \exists(t_1, t_2) \mid p = t_2 - t_1 \land t_1 < t_2 \)

\( \land F\left( \frac{dA}{dt} > 0 \land X\left( \frac{dA}{dt} \leq 0 \land Time = t_1 \right) \right) \)

\( \land F\left( \frac{dA}{dt} > 0 \land X\left( \frac{dA}{dt} \leq 0 \land Time = t_2 \right) \right) \)

\( = t_3 \)) \land \neg \exists t_3 \mid t_1 < t_3 < t_2 \land F\left( \frac{dA}{dt} > 0 \land X\left( \frac{dA}{dt} \leq 0 \land Time = t_2 \right) \right) \)
Proof of validity: phase and amplitude

Peak

Formula:

\[ \phi = \exists (t_1, t_2) \mid p = t_2 - t_1 \land t_1 < t_2 \]

\[ \land F \left( \frac{dA}{dt} > 0 \land X \left( \frac{dA}{dt} \leq 0 \land \text{Time} = t_1 \right) \right) \]

\[ \land F \left( \frac{dB}{dt} > 0 \land X \left( \frac{dB}{dt} \leq 0 \land \text{Time} = t_2 \right) \right) \]

Trace simplification:
The optimal trace simplification is \( T_J \) with \( J = \{i, i+1 \in \{0, \ldots, n\} \mid \frac{dA}{dt} s_i > 0 \land \frac{dA}{dt} s_{i+1} \leq 0 \} \)

\( T_{e A, B} \) is a simplification of \( T \) for \( \phi \).

Minimal amplitude

Formula: \( \phi = \exists v \mid F(A < v) \land F(A > v + a) \)

Validity Domain:

\[ D_{T, \phi} = \bigcap_{a} \left( D_{s_0}^{T} F(A < v) \cap D_{s_0}^{T} F(A > v + a) \right) \]

\[ = \bigcap_{a} \left( \left( \bigcup_{i=0}^{n} D_{s_i}^{T} A < v \right) \cap \left( \bigcup_{i=0}^{n} D_{s_j}^{T} A > v + a \right) \right) \]

\[ = \bigcap_{a} \left( D_{s_{\min} A}^{T} A < v \cap D_{s_{\max} A}^{T} A > v + a \right) \]

Trace simplification:
The optimal trace simplification is \( T_J \) where \( J = \{\min A, \max A\} \).

\( T_{e A} \) is a simplification of \( T \) for \( \phi \).
**First theorem:** If a simplification trace is correct for \( \phi \) and \( \psi \) then it is correct for the logical combinations of \( \phi \) and \( \psi \).

Proof: \[
D_{s_i, \phi \land \psi}^T = D_{s_i, \phi}^T \cap D_{s_i, \psi}^T = D_{s_j, \phi}^T \cap D_{s_j, \psi}^T = D_{s_j, \phi \land \psi}^T
\]

**Second theorem:**

If a subtrace contains extreme domains, it is a simplification for \( F \).

Proof: \[
D^T_T = \bigcup_i D_{s_i, \phi} \cap \bigcup_j D_{s_j, \phi}
\]

Similar result for \( G \): A simplification trace of \( G\phi \) is the set of points \( s_j \) whose \( D_{s_j, \phi} \) is contained in all the \( D_{s_i, \phi} \)

**Corollary:** A simplified trace on \( T \) for \( F(c \land \phi) \) can be found by discarding all the points where \( c \) is false, if this defines a simplified trace on \( T \) for \( \phi \).
Second theorem:
If a subtrace contains extreme domains, it is a simplification for $F$.

**Proof:** $D^T_\phi = \bigcup_i D_{sj,\phi} \subset \bigcup_j D_{sj,\phi}$

Similar result for $G$: A simplification trace of $G\phi$ is the set of points $s_j$ whose $D_{sj,\phi}$ is contained in all the $D_{si,\phi}$.
Second theorem:
If a subtrace contains extreme domains, it is a simplification for F.

Proof: $D^T_\phi = \bigcup_i D_{s_j,\phi} \subset \bigcup_j D_{s_j,\phi}$

Similar result for G: A simplification trace of $G\phi$ is the set of points $s_j$ whose $D_{s_j,\phi}$ is contained in all the $D_{s_i,\phi}$.
**Corollary:** A simplified trace on $T$ for $F(c \land \phi)$ can be found by discarding all the points where $c$ is false, if this defines a simplified trace on $T$ for $\phi$.

**Formula:** $\phi = F(Time > 20 \land A < v)$

**Validity Domain:** $D_{T,\phi} = D_{s_0,F(Time > 20 \land A < v)}$

$$= \bigcup_{i=0}^{n} D_{s_i,Time > 20 \land A < v}^T$$

$$= \bigcup_{i=0}^{n} (D_{s_i,Time > 20}^T \cap D_{s_i,A < v}^T)$$

$$= \bigcup \{D_{s_i,A < v}^T \mid Time_{s_i} > 20\}$$

$$= D_{s_{\text{min}_A > 20},A < v}^T$$

**Threshold**

**Trace simplification:**
The single point $s_{\text{min}_A > 20}$ defines an optimal trace simplification of $T$ for $\phi$. $T^e_A$ is not a simplification of $T$ for $\phi$ unless it does contain a local minimum such that $\text{Time} > 20$. 
Corollary: A simplified trace on $T$ for $F(c \land \phi)$ can be found by discarding all the points where $c$ is false, if this defines a simplified trace on $T$ for $\phi$.

**Formula:** $\phi = F(\text{Time} > 20 \land A < v)$

**Validity Domain:** $D_{T,\phi} = D_{s_0, F(\text{Time} > 20 \land A < v)}$

$$= \bigcup_{i=0}^{n} D^{T}_{s_i, \text{Time} > 20 \land A < v}$$

$$= \bigcup_{i=0}^{n} (D^{T}_{s_i, \text{Time} > 20} \cap D^{T}_{s_i, A < v})$$

$$= \bigcup \{D^{T}_{s_i, A < v} | \text{Time}_{s_i} > 20 \}$$

$$= D^{T}_{s_{\text{min}A > 20}, A < v}$$

**Threshold**

**Trace simplification:**

The single point $s_{\text{min}A > 20}$ defines an optimal trace simplification of $T$ for $\phi$. $T^e_A$ is not a simplification of $T$ for $\phi$ unless it does contain a local minimum such that $\text{Time} > 20$. 
**Corollary:** A simplified trace on $T$ for $F(c \land \phi)$ can be found by discarding all the points where $c$ is false, if this defines a simplified trace on $T$ for $\phi$.

**Threshold**

**Formula:** $\phi = F(\text{Time} > 20 \land A < v)$

**Validity Domain:** $D_{T,\phi} = D_{s_0,F(\text{Time} > 20 \land A < v)}$

\[
= \bigcup_{i=0}^{n} D_{s_i,\text{Time} > 20 \land A < v}
\]

\[
= \bigcup_{i=0}^{n} (D_{s_i,\text{Time} > 20} \cap D_{s_i,A < v})
\]

\[
= \bigcup \{ D_{s_i,A < v} \mid \text{Time}_{s_i} > 20 \}
\]

\[
= D_{s_{\min A > 20},A < v}
\]

**Trace simplification:**

The single point $s_{\min A > 20}$ defines an optimal trace simplification of $T$ for $\phi$. $T^e_A$ is not a simplification of $T$ for $\varphi$ unless it does contain a local minimum such that $\text{Time} > 20$. 
**Corollary**: A simplified trace on $T$ for $F(c \land \phi)$ can be found by discarding all the points where $c$ is false, if this defines a simplified trace on $T$ for $\phi$.

**Threshold**

**Formula**: $\phi = F(Time > 20 \land A < v)$

**Validity Domain**: $D_{T,\phi} = D_{s_0,F(Time > 20 \land A < v)}$

\[ D_{T,\phi} = \bigcup_{i=0}^{n} D_{s_i,Time > 20 \land A < v} \]

\[ = \bigcup_{i=0}^{n} (D_{s_i,Time > 20} \cap D_{s_i,A < v}) \]

\[ = \bigcup_{\{i | Time_{s_i} > 20\}} D_{s_i,A < v} \]

\[ = D_{s_{\min A > 20},A < v} \]

**Trace simplification**: The single point $s_{\min A > 20}$ defines an optimal trace simplification of $T$ for $\phi$. $T_{s_A}$ is not a simplification of $T$ for $\phi$ unless it does contain a local minimum such that $Time > 20$. 
Crossing

**Formula:** \( \phi = F(A > B \land X(A \leq B \land Time = t)) \)

**Validity Domain:**

\[
\mathcal{D}_{T, \phi} = \bigcup_{i=0}^{n} (\mathcal{D}_{s_i}^{T} A_{s_i} > B_{s_i} \cap (\mathcal{D}_{s_{i+1}}^{T} A_{s_{i+1}} \leq B_{s_{i+1}} \cap \mathcal{D}_{s_{i+1}}^{T} Time = t)) = \bigcup_{i \in \{0, \ldots, n\} | A_{s_i} > B_{s_i} \land A_{s_{i+1}} \leq B_{s_{i+1}}} \{Time_{s_{i+1}}\}
\]

Here \( T_{e, A, B} \) is NOT a simplification of \( T \) for \( \phi \).

A simplification trace is defined by the points in:

\[
J = \{i, i + 1 \in \{0, \ldots, n\} | A_{s_i} > B_{s_i} \land A_{s_{i+1}} \leq B_{s_{i+1}}\}
\]
Crossing

**Formula:** $\phi = F(A > B \land X(A \leq B \land Time = t))$

**Validity Domain:**

$$D_{T,\phi} = \bigcup_{i=0}^{n} (D_{s_i}^T, A_{s_i} > B_{s_i} \cap (D_{s_i+1}^T, A_{s_i} \leq B_{s_i} \cap D_{s_i+1}^T, Time = t))$$

$$= \bigcup_{i \in \{0, \ldots, n\}} \{Time_{s_i+1}\}$$

Here $T_{A,B}^{e}$ is **NOT** a simplification of $T$ for $\phi$.

A simplification trace is defined by the points in:

$$J = \{i, i + 1 \in \{0, \ldots, n\} | A_{s_i} > B_{s_i} \land A_{s_i+1} \leq B_{s_i+1}\}$$
Evaluation on Oscillation Constraints between the Cell Cycle and Circadian Clock

• The cell cycle and the circadian clock: two coupled oscillators involving:
  – qualitative properties: oscillations, stability
  – quantitative properties: period of each oscillator, phase

• Constraints on one molecule:
  – Minimum amplitude
  – Distance between successive peaks
  – Regularity of the distances between peaks
  – Regularity of the peak amplitudes

• Constraints on two molecules:
  – Phase

Cell cycle: MPF, Wee1
Circadian clock: Bmal1, PerCry, Rev-erbα
Evaluation on Oscillation Constraints between the Cell Cycle and Circadian Clock

**Trace simplification:**
- **Extrema subtrace** implemented in BIOCHAM
- Computing times:
  - Rosenbrock’s variable step-size simulation: 8-16 ms
  - 4th order Runge-Kutta fixed step-size simulation: 160-250 ms

**Validity domain computing time (in ms):**

<table>
<thead>
<tr>
<th>Formula</th>
<th>Nb of points</th>
<th>Solver</th>
<th>First trace</th>
<th></th>
<th></th>
<th>Second trace</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>variable Bef.</td>
<td>Aft.</td>
<td>fixed Bef.</td>
<td>Aft.</td>
<td>variable Bef.</td>
</tr>
<tr>
<td>Reachability of PerCry</td>
<td></td>
<td></td>
<td>971</td>
<td>18</td>
<td>20002</td>
<td>18</td>
<td>1047</td>
</tr>
<tr>
<td>Minimum amplitude of PerCry</td>
<td></td>
<td></td>
<td>12</td>
<td>0</td>
<td>260</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>Local maxima of PerCry</td>
<td></td>
<td></td>
<td>132</td>
<td>0</td>
<td>2728</td>
<td>0</td>
<td>132</td>
</tr>
<tr>
<td>Distance betw. PerCry peaks</td>
<td></td>
<td></td>
<td>64</td>
<td>0</td>
<td>1308</td>
<td>4</td>
<td>72</td>
</tr>
<tr>
<td>Distance betw. succ. PerCry peaks</td>
<td></td>
<td></td>
<td>512</td>
<td>12</td>
<td>9584</td>
<td>12</td>
<td>708</td>
</tr>
<tr>
<td>Regularity of PerCry peaks</td>
<td></td>
<td></td>
<td>532</td>
<td>12</td>
<td>10980</td>
<td>12</td>
<td>1188</td>
</tr>
<tr>
<td>Phase betw. PerCry and MPF</td>
<td></td>
<td></td>
<td>1700</td>
<td>32</td>
<td>34818</td>
<td>32</td>
<td>3056</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>456</td>
<td>16</td>
<td>9332</td>
<td>16</td>
<td>496</td>
</tr>
</tbody>
</table>

- Extrema subtrace implemented in BIOCHAM
- Computing times:
  - Rosenbrock’s variable step-size simulation: 8-16 ms
  - 4th order Runge-Kutta fixed step-size simulation: 160-250 ms
Conclusion

• Temporal logic patterns provide an elegant way to extract meaningful information on the periods and phases from numerical traces use these formulae as constraints for parameter search

• Simplifying the trace prior to the solving makes the generic solving algorithm more efficient

• Under some general conditions on the syntax of the formulae given as theorems it is correct to keep in the trace only the time points corresponding to
  • the local extrema of the molecules
  • or the crossing points between molecular concentrations

• On simulation traces, the speedup obtained in computation time was by several orders of magnitude: up to 1000 fold.

• The trace simplifications described in this paper are implemented in Biocham release 3.6.