Time-Series Constraints

Ekaterina Arafailova, Nicolas Beldiceanu, Mats Carlsson, Rémi Douence, Pierre Flener, M. Andreína Francisco R., Justin Pearson, and Helmut Simonis

8th December 2016

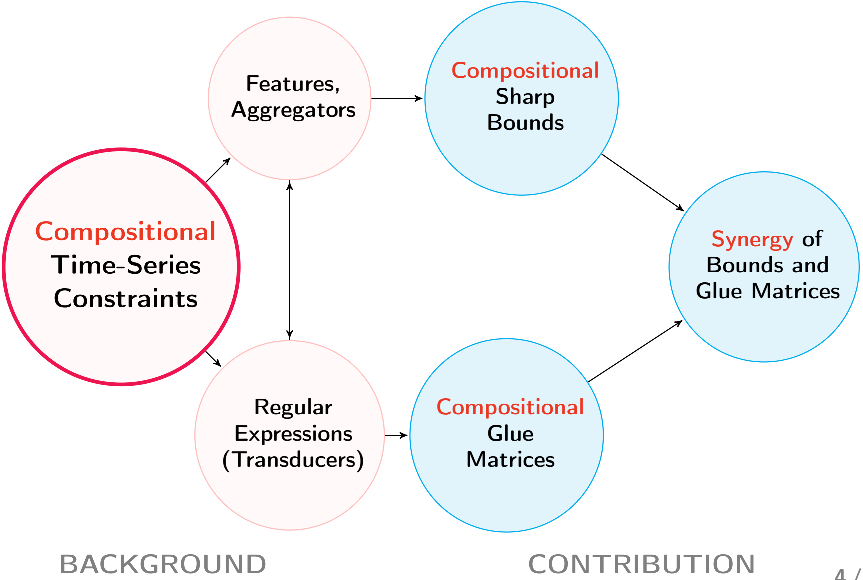
About Me

- Graduated from M2 ORO in 2015.
- Internship at EMN under supervision of N. Beldiceanu (EMN) and M. Minoux (LIP6).
- Started a PhD on the 1st of September, 2015.
- Thesis Director: N. Beldiceanu, Co-Supervisor: R. Douence.
- GRACeFUL project.

Collaboration

1. GRACeFUL people.
2. Helmut Simonis (University College Cork).
3. Pierre Flener, M. Andreína Francisco R., Justin Pearson (The University of Uppsala).
4. Mats Carlsson (SICS)

Contents

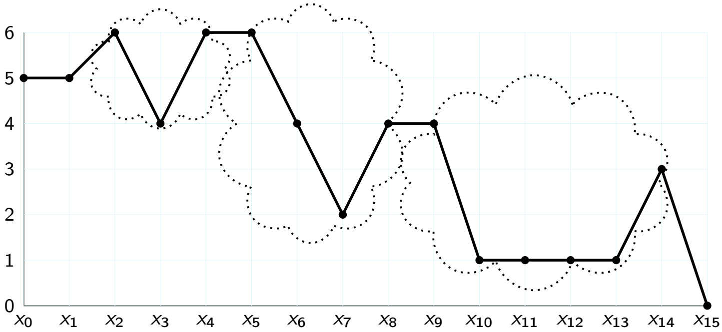

Features,
Aggregators
Compositional
Sharp
Bounds
Compositional
Time-Series
Constraints
Synergy of
Bounds and
Glue Matrices
Regular
Expressions
(Transducers)
Compositional
Glue
Matrices
BACKGROUND
CONTRIBUTION

Example of a Time-Series Constraint

Constrain the maximum of the minima of the valleys in the time series $\langle 5, 5, 6, 4, 6, 6, 4, 2, 4, 4, 1, 1, 1, 1, 3, 0 \rangle$.

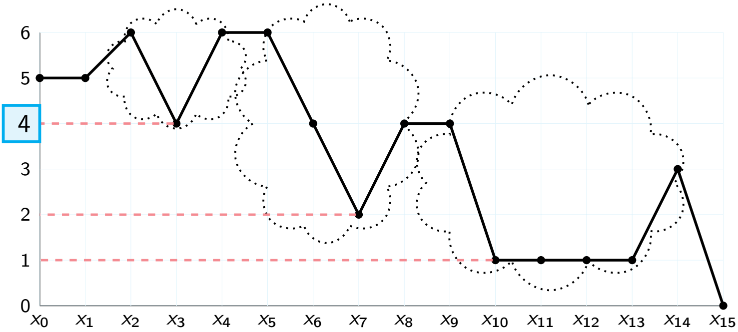
Example of a Time-Series Constraint

Constrain the maximum of the minima of the valleys in the time series $\langle 5, 5, 6, 4, 6, 6, 4, 2, 4, 4, 1, 1, 1, 1, 3, 0 \rangle$.



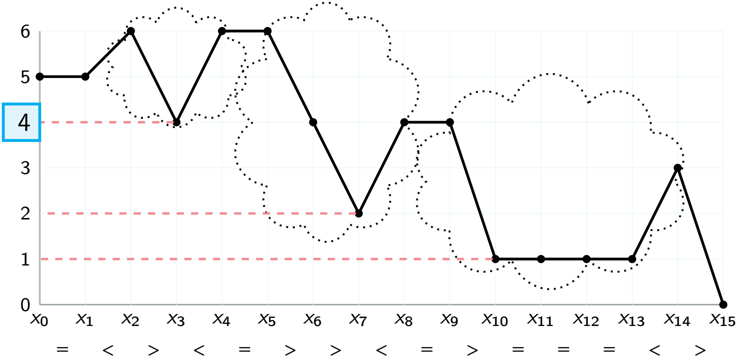
Example of a Time-Series Constraint

Constrain the maximum of the minima of the valleys in the time series $\langle 5, 5, 6, 4, 6, 6, 4, 2, 4, 4, 1, 1, 1, 1, 3, 0 \rangle$.



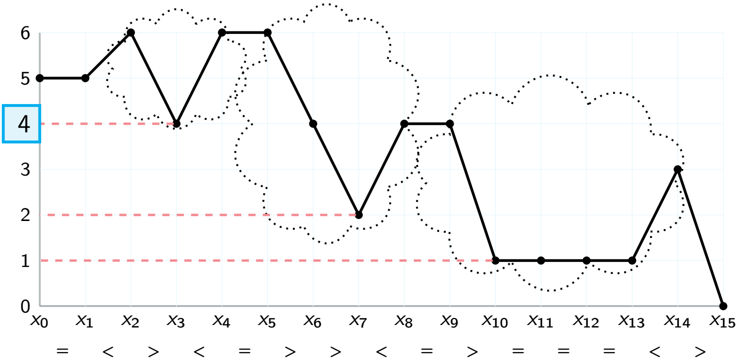
Example of a Time-Series Constraint

Constrain the maximum of the minima of the valleys in the time series $\langle 5, 5, 6, 4, 6, 6, 4, 2, 4, 4, 1, 1, 1, 1, 3, 0 \rangle$.



Example of a Time-Series Constraint

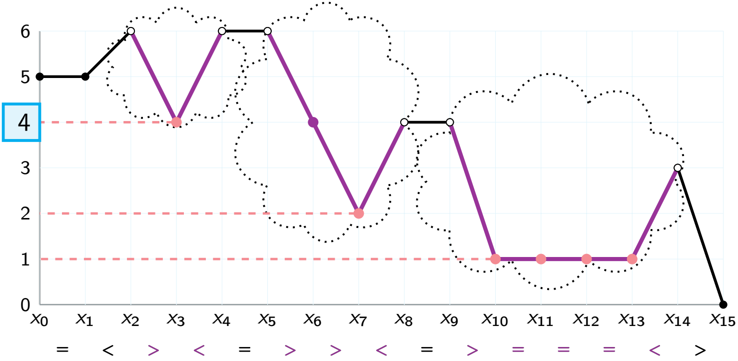
Constrain the maximum of the minima of the valleys in the time series $\langle 5, 5, 6, 4, 6, 6, 4, 2, 4, 4, 1, 1, 1, 1, 3, 0 \rangle$.



A subsequence $\langle X_i, \ldots, X_j \rangle$ of $\langle X_0, \ldots, X_m \rangle$ is a **valley** if the signature of $\langle X_{i-1}, \ldots, X_{j+1} \rangle$ is a maximal word matching **'>(>|=)*(<|=)*<'**.

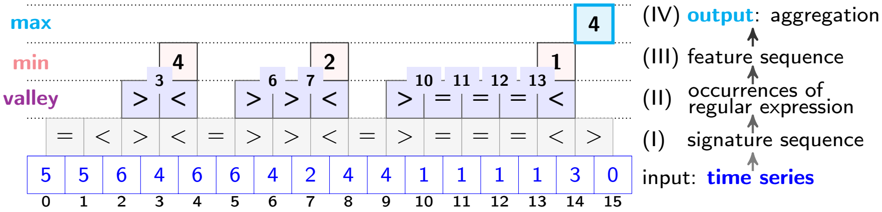
Example of a Time-Series Constraint

Constrain the maximum of the minima of the valleys in the time series $\langle 5, 5, 6, 4, 6, 6, 4, 2, 4, 4, 1, 1, 1, 1, 3, 0 \rangle$.



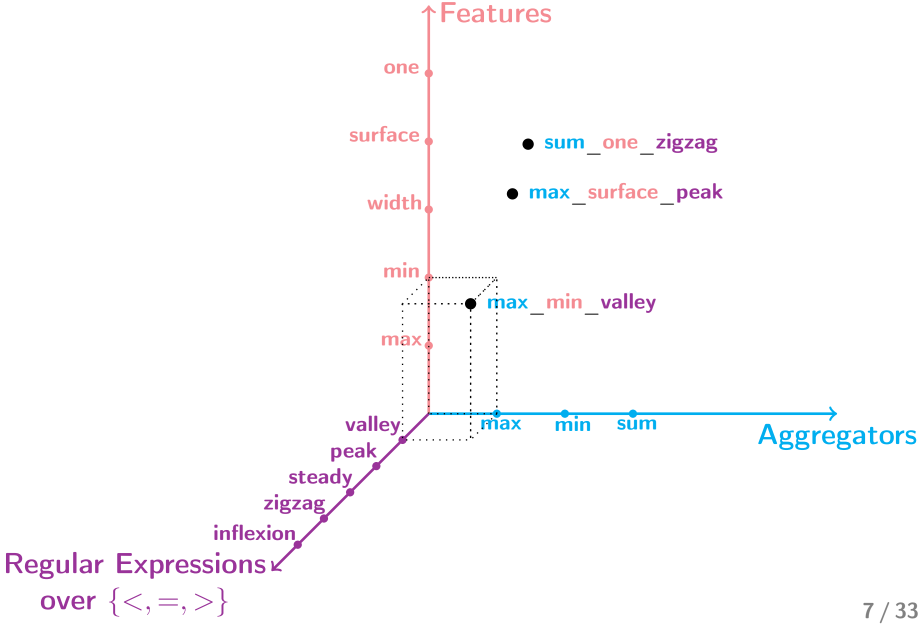
A subsequence $\langle X_i, \ldots, X_j \rangle$ of $\langle X_0, \ldots, X_m \rangle$ is a **valley** if the signature of $\langle X_{i-1}, \ldots, X_{j+1} \rangle$ is a maximal word matching **'>(>|=)*(<|=)*<'**.

Compositional Time-Series Definition by Multiple Layers of Functions

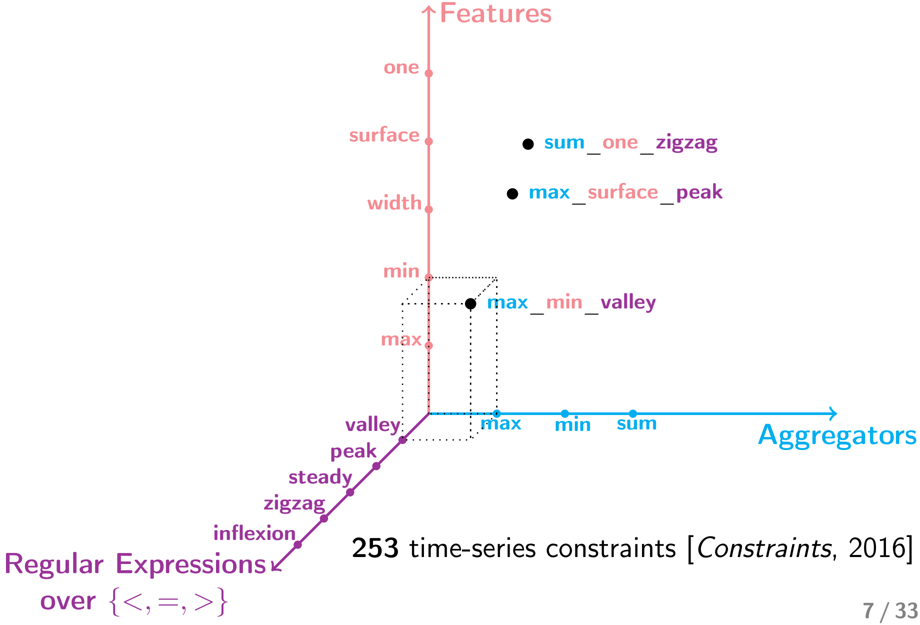


max_min_valley(⟨5, 5, 6, 4, 6, 6, 4, 2, 4, 4, 1, 1, 1, 1, 3, 0⟩, 4) holds

Space of Time-Series Constraints



Space of Time-Series Constraints



253 time-series constraints [*Constraints*, 2016]

Main Question

**How to go from a compositional constraint definition
to compositional combinatorial objects
that can be used in different contexts,
such as CP, MIP, local search, and data mining?**

Main Question

How to go from a compositional constraint definition
to compositional combinatorial objects
that can be used in different contexts,
such as CP, MIP, local search, and data mining?

Time-series constraints are very numerous:
we cannot afford to consider each of them separately.

Applications of Time-Series Constraints

- Analysis of output of electric power stations over multiple days. [*CP 2013*]
- Power management for large-scale-distributed systems. [*Computing*, 2016]
- Trace analysis for Internet Service Provider. (CeADAR)
- Anomaly detection and error correction in building data. (Campus21)

Synthesising Time-Series Constraints

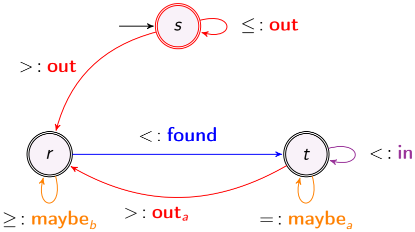
Use a transducer: [*Constraints*, 2016]

- Input sequence: Handle the **matching** aspect
- Output sequence: Handle the **computational** aspect

Synthesising Time-Series Constraints

Use a **transducer**: [*Constraints*, 2016]

- Input sequence: Handle the **matching** aspect
- Output sequence: Handle the **computational** aspect

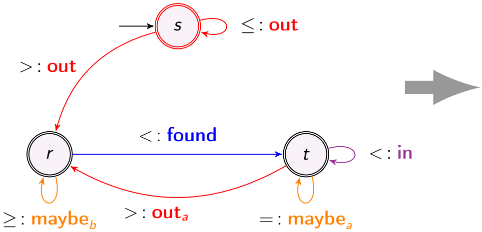


Transducer for **valley**

Synthesising Time-Series Constraints

Use a **transducer**: [*Constraints*, 2016]

- Input sequence: Handle the **matching** aspect
- Output sequence: Handle the **computational** aspect

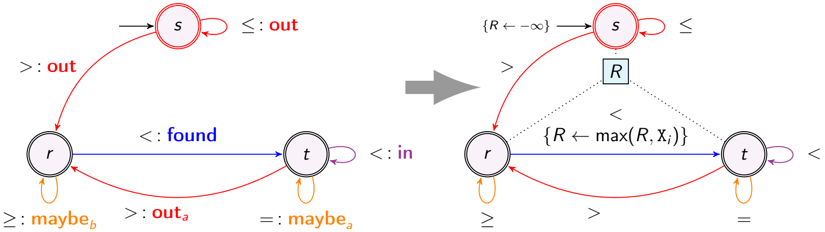


Transducer for **valley**

Synthesising Time-Series Constraints

Use a **transducer**: [*Constraints*, 2016]

- Input sequence: Handle the **matching** aspect
- Output sequence: Handle the **computational** aspect

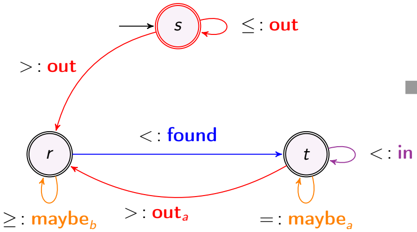


Transducer for **valley** Automaton for **max_min_valley**

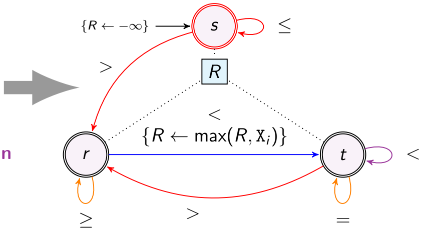
Synthesising Time-Series Constraints

Use a **transducer**: [*Constraints*, 2016]

- Input sequence: Handle the **matching** aspect
- Output sequence: Handle the **computational** aspect



Transducer for **valley**



Automaton for **max_min_valley**

5	$\xrightarrow{=}$	5	$\xrightarrow{<}$	6	$\xrightarrow{>}$	4	$\xrightarrow{<}$	6	$\xrightarrow{=}$	6	$\xrightarrow{>}$	4	$\xrightarrow{>}$	2	$\xrightarrow{<}$	4
s	out	s	out	s	out	r	found	t	maybe$_a$	t	out$_a$	r	maybe$_b$	r	found	t

Implementing Time-Series Constraints

- CP: **Reformulation of automata** with accumulators*
- MIP: **Linear reformulation** of automata with accumulators**

* [*Constraints*, 2005], ** [*CPAIOR 2016*]

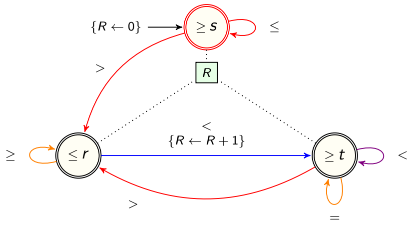
Goal: Address the combinatorial aspect of time-series constraints.

Method: Exploit the **compositional nature** of time-series constraints at the combinatorial level.

Input

- Time-series variables X_i with i in $[1, n]$ over their domains $[a_i, b_i]$
- An automaton with accumulators for a time-series constraint with
  - a set of states Q;
  - an input alphabet Σ;
  - an m-tuple of integer accumulators with their initial values $I = \langle I_1, \ldots, I_m \rangle$;
  - a transition function $\delta : Q \times Z^m \times \Sigma \to Q \times Z^m$.

Input: example



Automaton for nb_valley

$$\frac{\mathbf{5}}{s} \xrightarrow{=} \frac{\mathbf{5}}{s} \xrightarrow{<} \frac{\mathbf{6}}{s} \xrightarrow{>} \frac{\mathbf{4}}{r} \xrightarrow{<} \frac{\mathbf{6}}{t} \xrightarrow{=} \frac{\mathbf{6}}{t} \xrightarrow{>} \frac{\mathbf{4}}{r} \xrightarrow{>} \frac{\mathbf{2}}{r} \xrightarrow{<} \frac{\mathbf{4}}{t}$$

$$\frac{\mathbf{5}}{0} \xrightarrow{=} \frac{\mathbf{5}}{0} \xrightarrow{<} \frac{\mathbf{6}}{0} \xrightarrow{>} \frac{\mathbf{4}}{0} \xrightarrow{<} \frac{\mathbf{6}}{1} \xrightarrow{=} \frac{\mathbf{6}}{1} \xrightarrow{>} \frac{\mathbf{4}}{1} \xrightarrow{>} \frac{\mathbf{2}}{1} \xrightarrow{<} \frac{\mathbf{4}}{2}$$

Goal

Our Goal

A way to generate a model for an automaton with linear or linearisable accumulator updates, for example containing min and max.

Linear decomposition of automata without accumulators

Côté, M.C., Gendron, B., Rousseau, L.M.: Modeling the regular constraint with integer programming. In: CPAIOR 2007. LNCS, vol. 4510, pp. 29–43. Springer (2007)

Signature Constraints

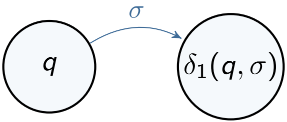
Introduced variables: S_i over Σ with $i \in [0, n-2]$.

What do the values of S_i mean ?

$$S_i = \text{'}{>}\text{'} \Leftrightarrow X_i > X_{i+1}, \forall i \in [0, n-2]$$
$$S_i = \text{'}{=}\text{'} \Leftrightarrow X_i = X_{i+1}, \forall i \in [0, n-2]$$
$$S_i = \text{'}{<}\text{'} \Leftrightarrow X_i < X_{i+1}, \forall i \in [0, n-2]$$

Transition Function Constraints

Introduced variables: Q_i over Q with $i \in [0, n-1]$; T_i over $Q \times \Sigma$ with $i \in [0, n-2]$



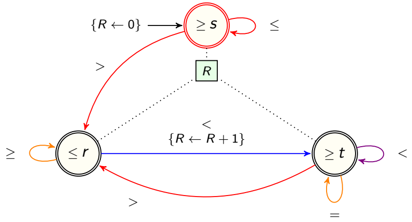
Each transition constraint has a form:

$Q_i = q \wedge S_i = \sigma \Leftrightarrow Q_{i+1} = \delta_1(q, \sigma) \wedge T_i = \langle q, \sigma \rangle,$
$\forall i \in [0, n-2],\ \forall q \in Q,\ \forall \sigma \in \Sigma$

Initial state is fixed:

$Q_0 = q_0$

Accumulator Updates



Accumulator updates

R_i over $[a, b]$ with i in $[0, n-1]$; T_i over $Q \times \Sigma$ with i in $[0, n-2]$.

- $R_0 = 0$
- $T_i = \langle r, > \rangle \Rightarrow \mathbf{R_{i+1}} = \mathbf{R_i} + \mathbf{1}, \forall i \in [0, n-2]$
- $T_i = \langle q, \sigma \rangle \Rightarrow \mathbf{R_{i+1}} = \mathbf{R_i}, \forall i \in [0, n-2], \forall \langle q, \sigma \rangle \in (Q \times \Sigma) \setminus \langle r, > \rangle$

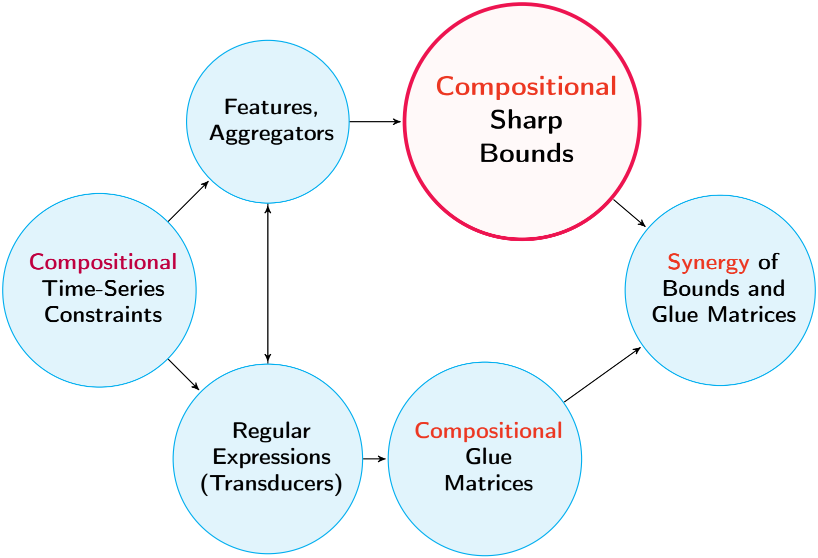
New Variables for the Linear Model

New variables

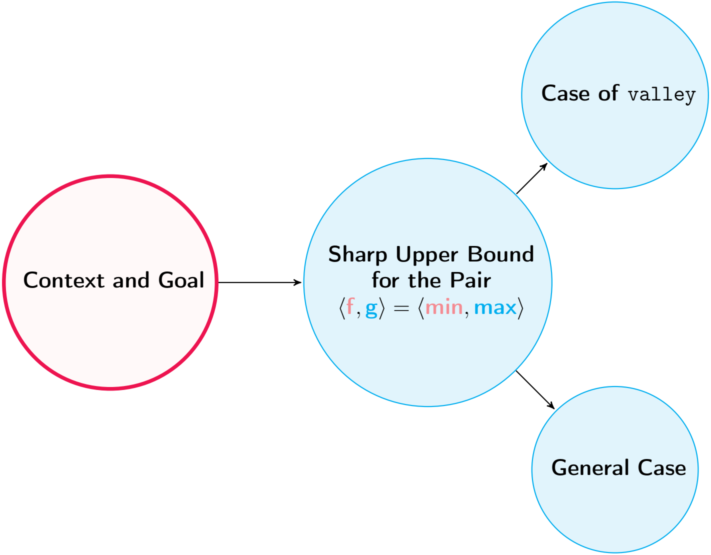
- Q_i is replaced by 0-1 variables Q_i^q for all q in Q.
  $Q_i^q = 1 \Leftrightarrow Q_i = q$
- New constraint: $\sum_{q \in Q} Q_i^q = 1, \forall i \in [0, \ldots, n-1]$
- The same procedure for T_i and S_i wrt their domains
- X_i and R_i remain integer variables!

- Every constraint of the logical model is made linear by applying some standard techniques
- The linear model has $O(n)$ variables and $O(n)$ constraints

Contents


Features, Aggregators
Compositional Sharp Bounds
Compositional Time-Series Constraints
Synergy of Bounds and Glue Matrices
Regular Expressions (Transducers)
Compositional Glue Matrices

Compositional Sharp Bounds: Contents



Context and Goal

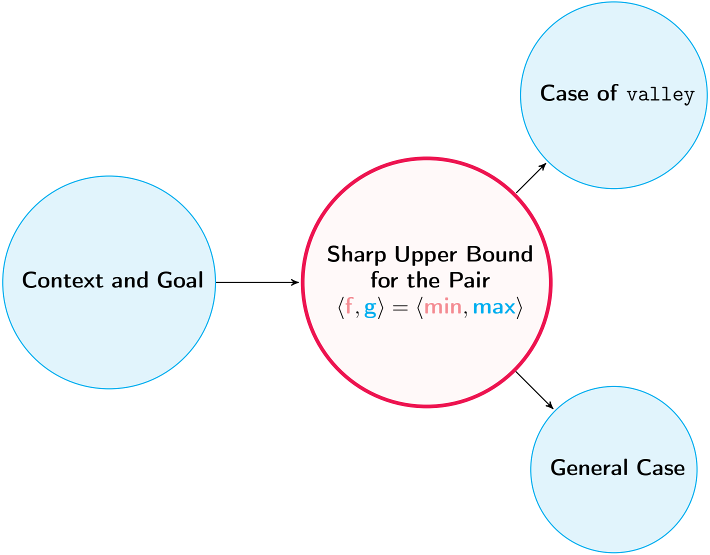
Consider a $\mathtt{g}\_\mathtt{f}\_\sigma(\langle X_0, \dots, X_m \rangle, N)$ time-series constraint, for a regular expression σ, a feature $\mathtt{f}$, and an aggregator $\mathtt{g}$, with every X_i ranging over **the same** integer interval $[a, b]$.

Context and Goal

Consider a $\mathtt{g\_f\_}\sigma(\langle X_0, \dots, X_m \rangle, N)$ time-series constraint, for a regular expression σ, a feature $\mathtt{f}$, and an aggregator $\mathtt{g}$, with every X_i ranging over **the same** integer interval $[a, b]$.

Goal: Derive formulae $\beta^{\mathbf{u}}_{\mathtt{f},\mathtt{g}}$ and $\beta^{\ell}_{\mathtt{f},\mathtt{g}}$ that are **parametrised by σ** and yield respectively sharp upper and lower bounds on the value of N.

Compositional Sharp Bounds: Contents



Computing a Sharp Upper Bound for the pair $\langle \mathbf{min}, \mathbf{max} \rangle$

Consider a $\mathbf{max\_min\_}\sigma(\langle X_0, \ldots, X_m \rangle, N)$ time-series constraint with every X_i ranging over $[a, b]$.

Goal: Compute a sharp upper bound on the value of N.

Computing a Sharp Upper Bound for the pair $\langle \mathbf{min}, \mathbf{max} \rangle$

Consider a max_min_$\sigma(\langle X_0, \ldots, X_m \rangle, N)$ time-series constraint with every X_i ranging over $[a, b]$.

Goal: Compute a sharp upper bound on the value of N.

Method:

1. Introduce the **characteristics** of a regular expression σ: *the minimal difference between the domain upper bound b and the minima of the subseries*, which is called the **shift** of σ and denoted by Δ_σ.

Computing a Sharp Upper Bound for the pair $\langle \mathbf{min}, \mathbf{max} \rangle$

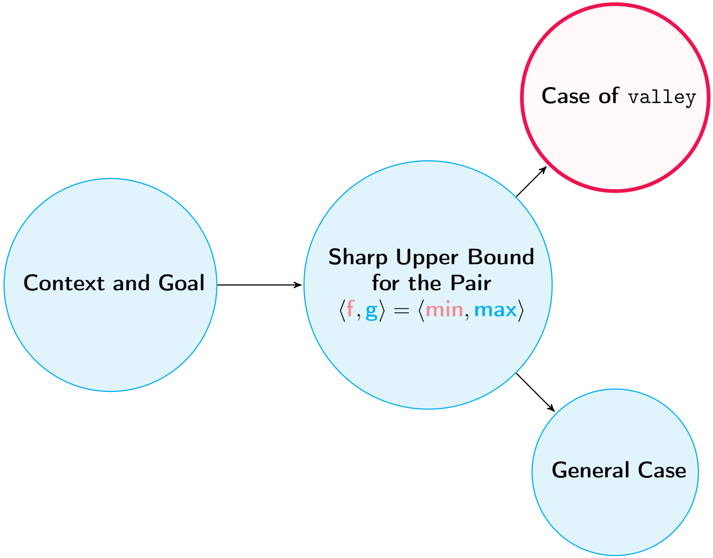
Consider a max_min_$\sigma(\langle X_0, \ldots, X_m \rangle, N)$ time-series constraint with every X_i ranging over $[a, b]$.

Goal: Compute a sharp upper bound on the value of N.

Method:

1. Introduce the **characteristics** of a regular expression σ: *the minimal difference between the domain upper bound b and the minima of the subseries*, which is called the **shift** of σ and denoted by Δ_σ.
2. Explore the connection between this characteristics and the sharp upper bound on the value of N.

Compositional Sharp Bounds: Contents



Computing a Sharp Upper Bound for max_min_valley

Consider the max_min_valley$(\langle X_0, \ldots, X_8 \rangle, N)$ time-series constraints with every X_i ranging over $[a, b] = [3, 6]$.

Shift of valley and Its Connection to a Sharp Upper Bound

- The shift of valley is the smallest possible difference between b and the minima of valleys, namely 1 and is reached, for example, for valleys whose signature is ‘$><$’.

Shift of valley and Its Connection to a Sharp Upper Bound

- The shift of valley is the smallest possible difference between b and the minima of valleys, namely 1 and is reached, for example, for valleys whose signature is '$><$'.
- The minimum of any valley is at least $\Delta_{\texttt{valley}}$, which is 1, away from b, which is 6, thus $N \leq b - \Delta_{\texttt{valley}} = 6 - 1 = 5$.

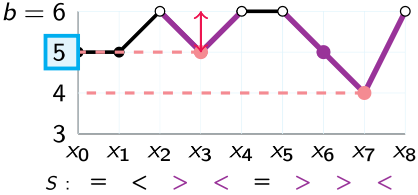
Shift of valley and Its Connection to a Sharp Upper Bound

- The shift of valley is the smallest possible difference between b and the minima of valleys, namely 1 and is reached, for example, for valleys whose signature is '$><$'.
- The minimum of any valley is at least $\Delta_{\texttt{valley}}$, which is 1, away from b, which is 6, thus $N \leq b - \Delta_{\texttt{valley}} = 6 - 1 = 5$.
- This bound is reached for some ground time series.

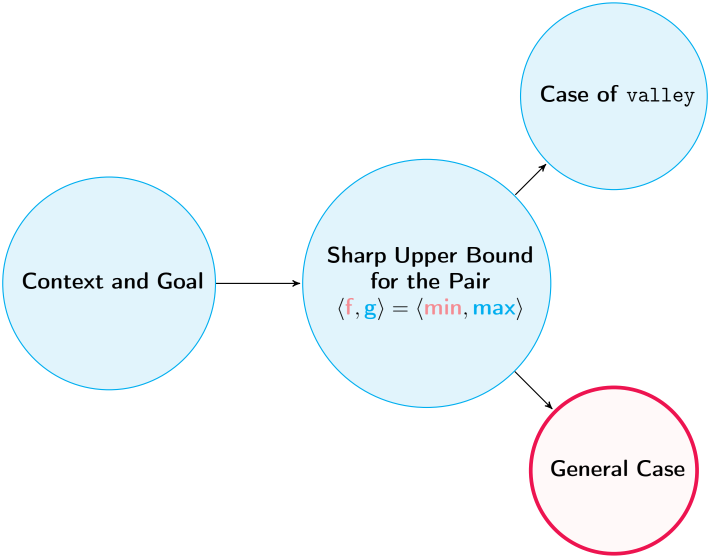
Shift of valley and Its Connection to a Sharp Upper Bound

- The shift of valley is the smallest possible difference between b and the minima of valleys, namely 1 and is reached, for example, for valleys whose signature is ‘$><$’.
- The minimum of any valley is at least $\Delta_{\texttt{valley}}$, which is 1, away from b, which is 6, thus $N \leq b - \Delta_{\texttt{valley}} = 6 - 1 = 5$.
- This bound is reached for some ground time series.

$$N = b - \Delta_{\texttt{valley}} = 6 - 1 = 5$$



Compositional Sharp Bounds: Contents



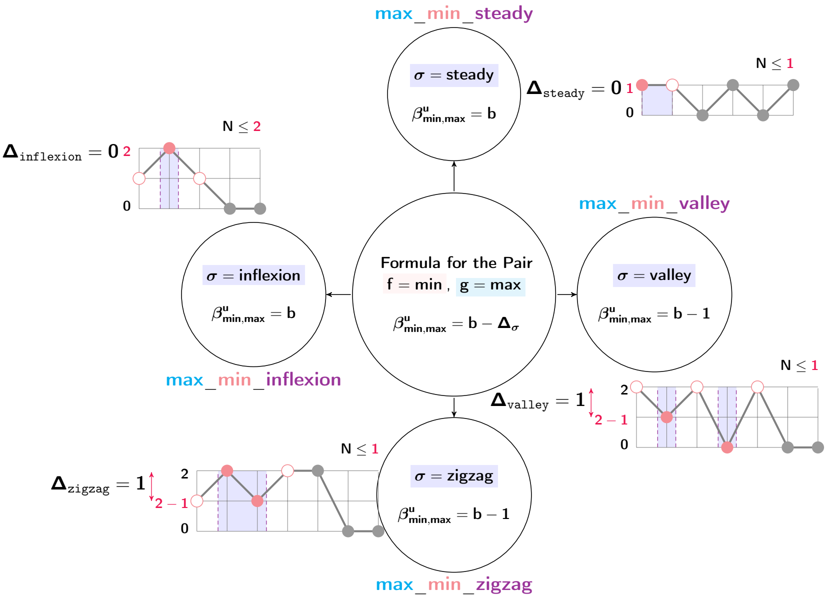
Sharp Upper Bound for the General Case

Theorem

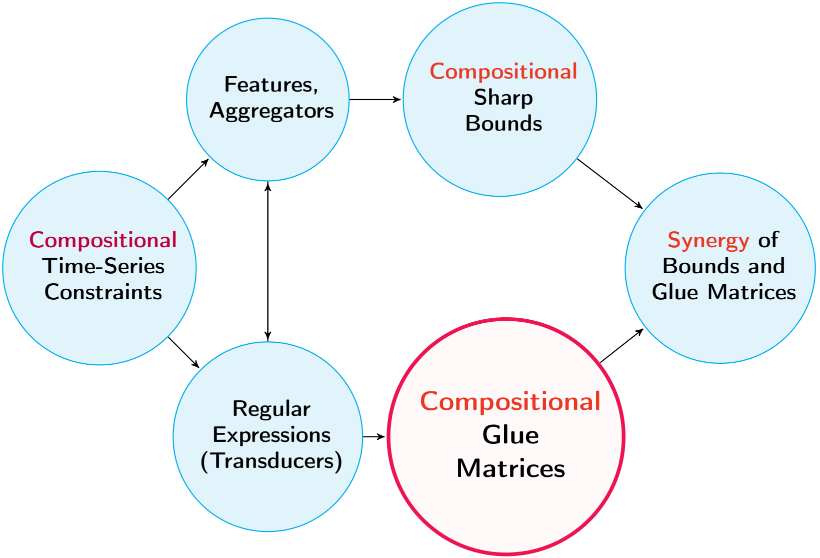
Consider a max_min_$\sigma(\langle X_0, \ldots, X_m \rangle, N)$ time-series constraint, where σ is one of the 22 regular expressions and every X_i ranges over $[a, b]$, then the sharp upper bound on the value of N is

$$\beta^{\mathbf{u}}_{\mathtt{min},\mathtt{max}} = b - \Delta_{\sigma}$$

Compositional Bounds (Summary)



Contents


Features, Aggregators
Compositional Sharp Bounds
Compositional Time-Series Constraints
Synergy of Bounds and Glue Matrices
Regular Expressions (Transducers)
Compositional Glue Matrices

Linking the Prefixes and Suffixes of a Sequence

For the ngroup$(\langle X_0, \ldots, X_m \rangle, \mathcal{S}, N)$ constraint, N is the number of maximal subsequences of $\langle X_0, \ldots, X_m \rangle$ whose values are all in $\mathcal{S}$.

Linking the Prefixes and Suffixes of a Sequence

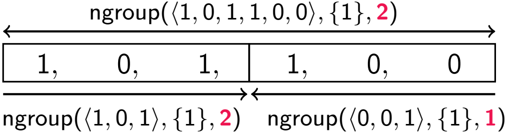
For the ngroup($\langle X_0, \ldots, X_m \rangle, \mathcal{S}, N$) constraint, N is the number of maximal subsequences of $\langle X_0, \ldots, X_m \rangle$ whose values are all in $\mathcal{S}$.

The ngroup($\langle 1, 0, 1, 1, 0, 0 \rangle, \{1\}, 2$) constraint holds.

Linking the Prefixes and Suffixes of a Sequence

For the ngroup($\langle X_0, \ldots, X_m \rangle, \mathcal{S}, N$) constraint, N is the number of maximal subsequences of $\langle X_0, \ldots, X_m \rangle$ whose values are all in $\mathcal{S}$.

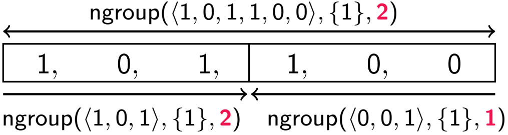
The ngroup($\langle 1, 0, 1, 1, 0, 0 \rangle, \{1\}, 2$) constraint holds.



Linking the Prefixes and Suffixes of a Sequence

For the ngroup($\langle X_0, \ldots, X_m \rangle, \mathcal{S}, N$) constraint, N is the number of maximal subsequences of $\langle X_0, \ldots, X_m \rangle$ whose values are all in $\mathcal{S}$.

The ngroup($\langle 1, 0, 1, 1, 0, 0 \rangle, \{1\}, 2$) constraint holds.

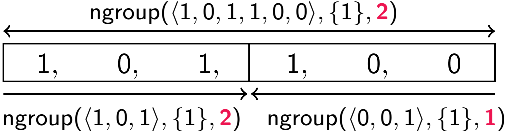


not just a sum ($2 \neq 2 + 1$)...

Linking the Prefixes and Suffixes of a Sequence

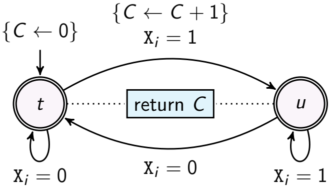
For the $\text{ngroup}(\langle X_0, \ldots, X_m \rangle, \mathcal{S}, N)$ constraint, N is the number of maximal subsequences of $\langle X_0, \ldots, X_m \rangle$ whose values are all in $\mathcal{S}$.

The $\text{ngroup}(\langle 1, 0, 1, 1, 0, 0 \rangle, \{1\}, 2)$ constraint holds.



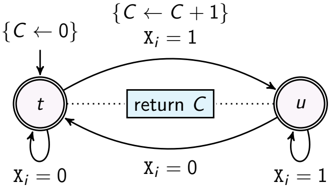
not just a sum $(2 \neq 2 + 1)$...how to compute the correction term -1?

Glue Matrix for the ngroup Constraint



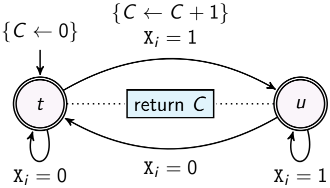
Automaton for **ngroup**

Glue Matrix for the ngroup Constraint



Automaton for **ngroup**

Glue Matrix for the ngroup Constraint

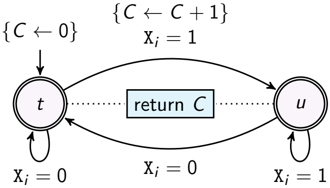


Automaton for **ngroup**

	t	**u**
t	0	0
u	0	-1

Glue Matrix for **ngroup**

Glue Matrix for the ngroup Constraint



Automaton for **ngroup**

	t	**u**
t	0	0
u	0	**−1**

Glue Matrix for **ngroup**

ngroup$(\langle 1,0,1,1,0,0\rangle, \{1\}, 2)$

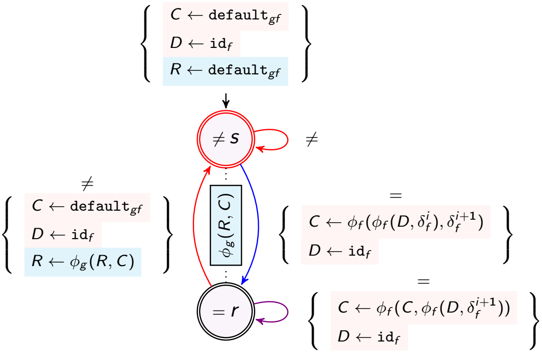
	1,	0,	1,		1,	0,	0
t,	u,	t,	**u**,	**u**,	t,	t,	t

ngroup$(\langle 1,0,1\rangle, \{1\}, 2)$ ngroup$(\langle 0,0,1\rangle, \{1\}, 1)$

Compositional Glue Matrices for Time-Series Constraints

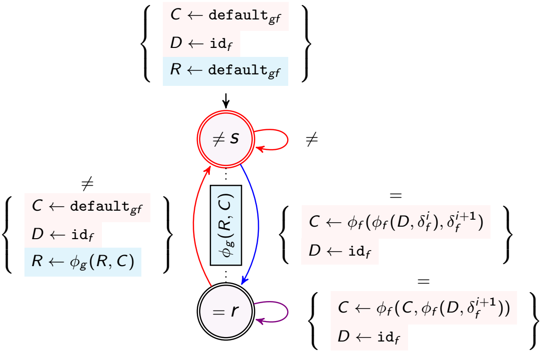
Goal: For a transducer, derive a glue matrix
that is parametrised by a feature and an aggregator.

Generic Glue Matrix for steady_sequence = '$=^+$'



Generic Automaton for
g_f_steady_sequence

Generic Glue Matrix for steady_sequence = '$=^+$'

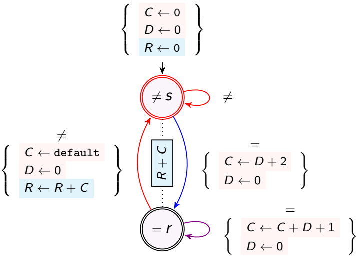


Generic Automaton for g_f_steady_sequence

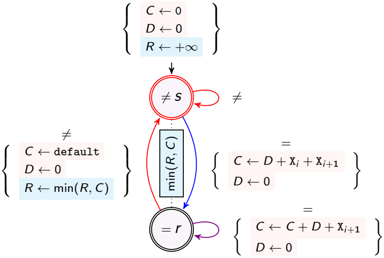
	s	r
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^i)$

Generic Glue Matrix for g_f_steady_sequence

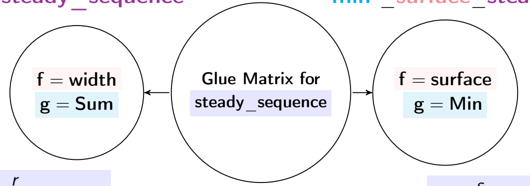
Corresponding Compositional Glue Matrices



sum _width_steady_sequence



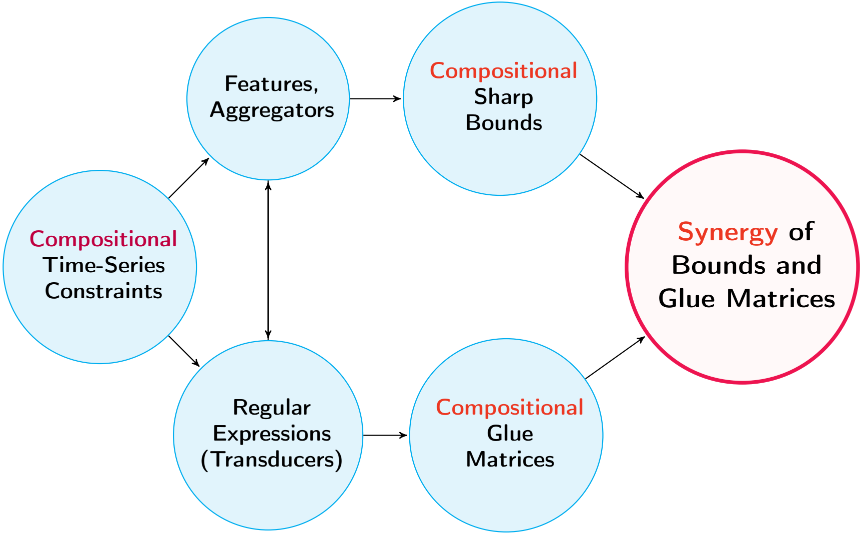
min _surface_steady_sequence



	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + X_i$

Contents



Features, Aggregators
Compositional Sharp Bounds
Compositional Time-Series Constraints
Synergy of Bounds and Glue Matrices
Regular Expressions (Transducers)
Compositional Glue Matrices

Synergy of Sharp Bounds and Glue Matrices

Consider a g_f_$\sigma(\langle X_0, \ldots, X_m \rangle, N)$ time-series constraint where all X_i range over the same domain.

- Add **sharp bounds** for every prefix and every reversed suffix of $X_0, \ldots, X_m$.
- Impose **glue-matrix implied constraints** on every prefix and every reversed suffix of $X_0, \ldots, X_m$.

The combination of sharp bounds and glue matrices improves propagation.

Conclusion: clear separation between three levels

- **Language**: compositional description of constraints with aggregator g, feature f, regular expression σ

- Abstract Combinatorial Objects (*transducers and formulae*)
  - Transducers for each regular expression σ (22)
  - Parametrised bounds for each pair g,f (20)
  - Parametrised glue matrices for each regular expression σ (22)

MANUAL

- Multiple usages (*synthesised code:* LaTeX(154100), Prolog (135267))
  - **CP**: enhanced propagation (*bounds, glue matrices*)
  - **Local search**: constant time probing (*glue matrices*)
  - **Data mining**: range of variation of feature (*bounds*)

SYNTHESISED

**Separate definition from usage,
be compositional at all levels**

Publications on Time-Series Constraints

- N. Beldiceanu, R. Douence, M. Carlsson, H. Simonis. "Using Finite Transducers for Describing and Synthesising Structural Time-Series Constraints". Constraints'16.
- E. Arafailova, N. Beldiceanu, R. Douence, M. A. F. Rodriguez, P. Flener, J. Pearson, H. Simonis. "Time-series constraints: Improvements and application in CP and MIP contexts". CPAIOR'16.
- E. Arafailova, N. Beldiceanu, M. Carlsson, M. A. F. Rodriguez, P. Flener, J. Pearson, H. Simonis. "Systematic Derivation of Bounds and Glue Constraints for Time-Series Constraints". CP'16.
- E. Arafailova, N. Beldiceanu, R. Douence, M. Carlsson, M. A. F. Rodriguez, P. Flener, J. Pearson, H. Simonis. "Global Constraint Catalog, Volume II, Time-Series Constraints". CoRR.

Arafailova, Ekaterina et al. (2016). "Time-series constraints: Improvements and application in CP and MIP contexts". In: *CPAIOR 2016*. Ed. by Claude-Guy Quimper. Vol. 9676. LNCS. Springer, pp. 18–34.

Beldiceanu, Nicolas et al. (2005). "Reformulation of global constraints based on constraints checkers". In: *Constraints* 10.4, pp. 339–362.

Beldiceanu, Nicolas et al. (2013). "Describing and generating solutions for the EDF unit commitment problem with the ModelSeeker". In: *CP 2013*. Ed. by Christian Schulte. Vol. 8124. LNCS. Springer, pp. 733–748.

Beldiceanu, Nicolas et al. (2016a). "Towards energy-proportional Clouds partially powered by renewable energy". In: *Computing*, p. 20. url: https://hal.inria.fr/hal-01340318.

Beldiceanu, Nicolas et al. (2016b). "Using finite transducers for describing and synthesising structural time-series constraints". In: *Constraints* 21.1, pp. 22–40.