Automatic Generation of Descriptions of Time-Series Constraints

M. Andreína Francisco Pierre Flener Justin Pearson

Department of Information Technology

Uppsala University

Sweden

November 6, 2017

Take-away points

- High-level way of describing constraints over sequences of variables.
- Automatically-synthesised constraint decompositions.
- New family of constraints for time-series.
- Applications in data analysis as well as optimisation.

Background: Combinatorial optimisation

Combinatorial optimisation consists of finding an object from a finite set of objects:

- The set of feasible solutions is discrete or can be discretised.
- The goal is to find a solution, or all solutions, or a best solution.
- Examples:
  - puzzles: sudoku, nonograms, etc.
  - the nurse scheduling problem.

Constraint programming (CP) is a set of techniques and tools for effectively modelling and efficiently solving hard combinatorial problems.

$$\text{CP solving} = \text{propagation} + \text{search}$$

Background: CP Modelling

Constraints form the vocabulary of a CP modelling language: they allow a modeller to express commonly occurring substructures.

Example

The AllDifferent(x, y, z) constraint, over the variables x, y, and z with domains $x \in \{1, 2\}$, $y \in \{1, 2\}$, and $z \in \{2, 3, 4, 5\}$, holds if and only if the variables x, y, and z take pairwise distinct values.

A constraint model is a conjunction of constraints.

Example

$$\text{AllDifferent}(x, y, z) \wedge x + y < z$$

Background: CP Propagation and Search

A constraint comes with a propagator, which removes impossible values from the domains of its variables.

Example

AllDifferent(x, y, z) with $x = \{1, 2\}, y = \{1, 2\}$, and $z = \{\cancel{2}, 3, 4, 5\}$

After the propagators have removed the values they can, the solver will begin a systematic search if need be:

- Select a variable
- Select a value (or a range of values)
- Propagate again on the domain of the variables

Example

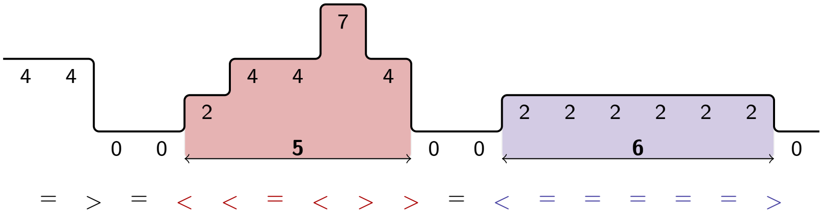
$x = 1$: AllDifferent(x, y, z), $x = \{1, \cancel{2}\}, y = \{\cancel{1}, 2\}$, and $z = \{3, 4, 5\}$
$x \neq 1$: AllDifferent(x, y, z), $x = \{\cancel{1}, 2\}, y = \{1, \cancel{2}\}$, and $z = \{3, 4, 5\}$

Background

Although modern CP solvers have many built-in constraints, often a constraint that one is looking for is not there. In such cases, the choices are:

- to reformulate the model without the needed constraint;
- to write a propagator for the new constraint;
- to decompose the constraint into a conjunction of constraints with already existing propagators. For example, the constraint AllDifferent(x, y, z) can be decomposed into $x \neq y \wedge x \neq z \wedge y \neq z$.

Example: $g_f_$Peak



Time-series constraint [CP'15]

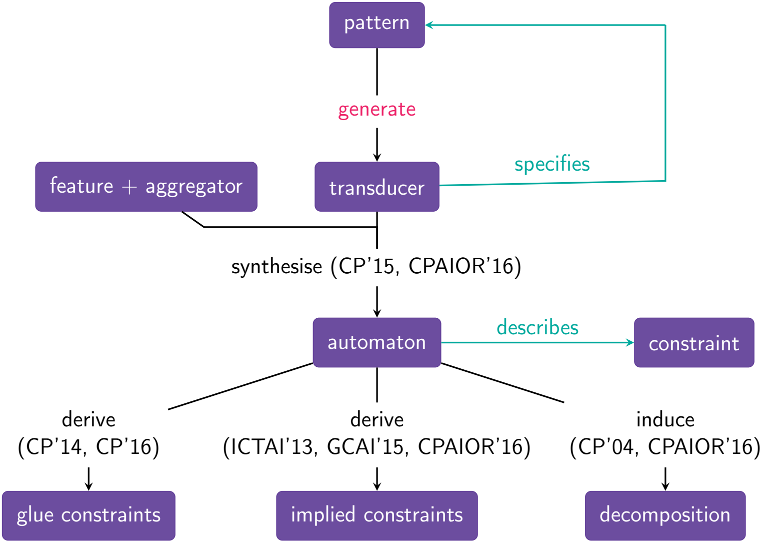
N. Beldiceanu, M. Carlsson, R. Douence, H. Simonis . "Using finite transducers for describing and synthesising structural time-series constraints"

A time-series constraint $g\_f\_\sigma(\langle X_1, \ldots, X_n \rangle, M)$ is defined by:

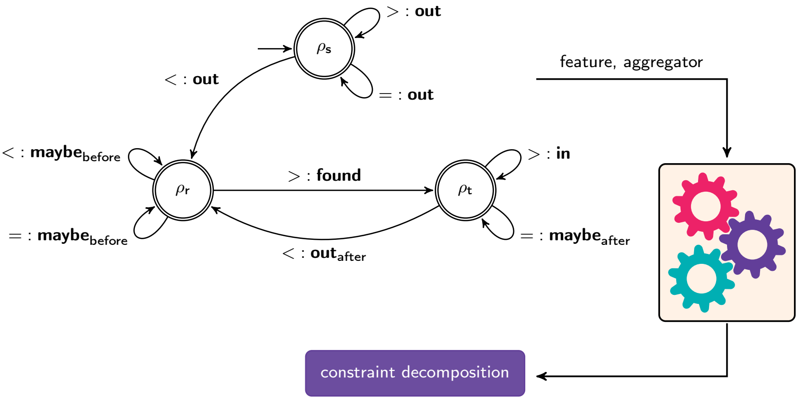
- A **pattern** σ is a regular expression over the alphabet $\{<, =, >\}$, e.g. `Peak` = '<(<|=)*(>|=)*>'.
  Only 22 patterns in the Time-Series Constraint Catalogue.
- A **feature** f is a function over a subseries:
  `one`, `max`, `min`, `surface`, `width`.
- An **aggregator** g is a function over the features: `Sum`, `Min`, `Max`.

where the variable sequence $\langle X_1, \ldots, X_n \rangle$ is a **time series** and variable M is the result of aggregating using g the feature values computed using f of all the maximal words matching σ in $\langle X_1, \ldots, X_n \rangle$.

Our research in context



Before



> : **out**
ρ_s
< : **out**
= : **out**
feature, aggregator
< : **maybe**before
> : **found**
> : **in**
ρ_r
ρ_t
= : **maybe**before
= : **maybe**after
< : **out**after
constraint decomposition

Pros and cons

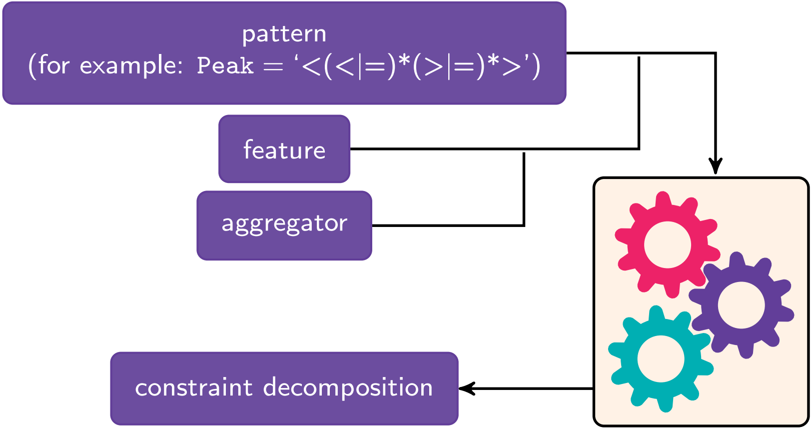
Pros

- Transducers, together with features and aggregators, are a convenient and high-level way for synthesising automata describing time-series constraints.
- Automatically-synthesised automaton-induced decompositions.

Cons

- Transducers need to be designed and verified by hand.
- Requires understanding the output alphabet of the transducers.
- Prone to errors.

Now


pattern
(for example: Peak = ‘<(<|=)*(>|=)*>’)
feature
aggregator
constraint decomposition

Results

- Our tool generates exactly the same transducers as in [CP'15].

- The obtained transducers are well-formed (correct).

- Now the Time-Series Constraint Catalogue can be extended at will.

The work on time-series constraints has been done in collaboration with / by:

- Ekaterina Arafailova
- Nicolas Beldiceanu
- Mats Carlsson
- Rémi Douence
- Helmut Simonis

The authors are supported by grants 2012-4908 and 2015-0491 of the Swedish Research Council (VR).