

# Qualitative Soft Constraints

Alexander Schiendorfer · Alexander Knapp ·  
Wolfgang Reif

Received: date / Accepted: date

**Abstract** Insert your abstract here. Include keywords, PACS and mathematical subject classification numbers as needed.

**Keywords** First keyword · Second keyword · More

## 1 Introduction

Your text comes here. Separate text sections with We started working on [1]

## 2 Foundations

## 3 Implementation

```
% Library predicate:  
% -----  
% Implements single predecessor  
% dominance on sets of constraints  
5 % (upper smyth ordering with partial  
%   order inversion)  
% -----  
include "alldifferent_except_0.mzn";
```

---

This research is partly sponsored by the German Research Foundation (DFG) in the project “OC-Trust” (FOR 1085).

---

Alexander Schiendorfer  
University of Augsburg  
E-mail: schiendorfer@isse.de

Alexander Knapp  
University of Augsburg  
E-mail: knapp@isse.de

Wolfgang Reif  
University of Augsburg  
E-mail: reif@isse.de

```

10 predicate spd_better(var set of int: lhs, var set of int: rhs,
    set of int: softConstraints,
    array[int, 1..2] of int: edges
) :: promise_total =
let {
15   int: le = min(index_set_lof2(edges));
   int: ue = max(index_set_lof2(edges));

   var set of int: lSymDiff = lhs diff rhs;
   var set of int: rSymDiff = rhs diff lhs;
20   set of int: softConstraints0 = {0} union softConstraints;
   % 0 represents noVal for constraints not in the right-hand-side

   var set of int: rUndefined = softConstraints diff rSymDiff;

25   % I need to make the dominance explicit by a function
   array[softConstraints] of var softConstraints0: witness;

   % collect all predecessors such that succ in lessThans[pred]
   % iff succ less than pred
30   array[softConstraints] of set of softConstraints: lessThans =
     [ {succ | succ in softConstraints where exists(e in le..ue)
        (edges[e,1] = pred /\ edges[e,2] = succ)} | pred in softConstraints];

   in (
35     lhs != rhs /\
     alldifferent_except_0(witness) /\
     forall(s in ub(rUndefined)) (s in rUndefined -> witness[s] = 0) /\
     forall(s in ub(rSymDiff)) (s in rSymDiff -> (witness[s] in lSymDiff /\
40        witness[s] in lessThans[s]))
);

```

## 4 Evaluation

## 5 Conclusion

## References

1. Schiendorfer, A., Steghöfer, J.P., Knapp, A., Nafz, F., Reif, W.: Constraint Relationships for Soft Constraints. In: M. Bramer, M. Petridis (eds.) Proc. 33<sup>rd</sup> SGAI Int. Conf. Innovative Techniques and Applications of Artificial Intelligence (AI'13), pp. 241–255. Springer (2013)