Consistency of Automaton-Induced Constraint Decompositions

M. A. Francisco Rodríguez, P. Flener, J. Pearson

May 14, 2012

Motivation

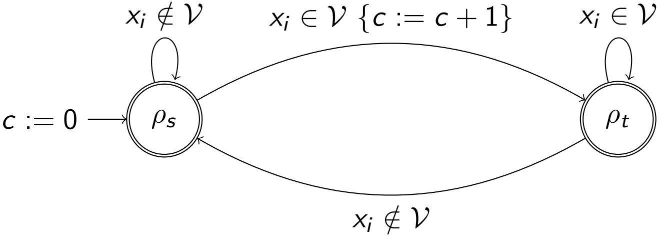
Why are decompositions important?

- Expressing constraints as an automaton is “easy”
- Most automaton-induced decompositions do not maintain HAC

Is it possible to derive implied constraints to maintain HAC?

The Group Constraint

A *group* is a subsequence of variables that take values from a given set $\mathcal{V}$.
The $n\_group(N, X, \mathcal{V})$ constraint holds when there are N groups in X. For example, the $n\_group(2, [a, a, b, c], \{a, c\})$ holds.

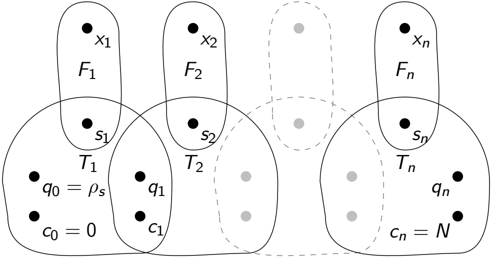


Signature constraints are $s_i = 0 \Leftrightarrow x_i \notin \mathcal{V}$ and $s_i = 1 \Leftrightarrow x_i \in \mathcal{V}$.

The Automaton-Induced Decomposition

α-acyclic constraint hypergraph induced by the counter automaton of the *n_group* constraint.

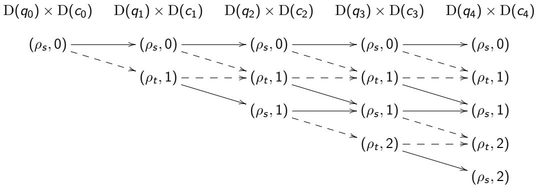
Variables q_i are state variables, and variables c_i are counter variables with $i = 0, \dots, n$



The Reachability Graph

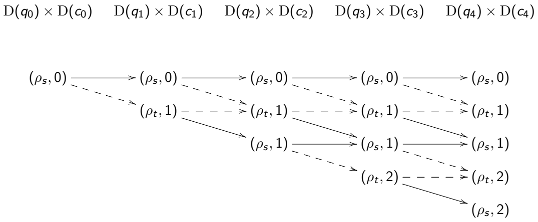
Reachability graph of the *n_group* constraint for $|V| = 4$.
Layer i is represented by $\mathrm{D}(q_i) \times \mathrm{D}(c_i)$.

Dashed arrows represent $s_i = 1$ and solid arrows represent $s_i = 0$.



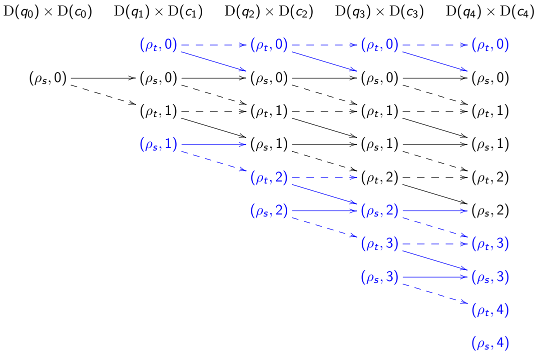
The Full Transition Graph

Full transition graph of the *n_group* constraint for $|V| = 4$.



The Full Transition Graph

Full transition graph of the *n_group* constraint for $|V| = 4$.



When do we get HAC for free?

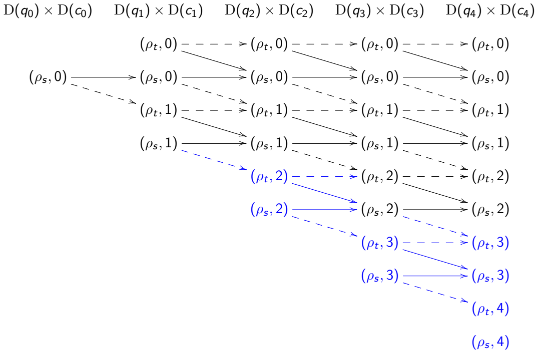
- When the reachability graph and the full transition graph are the same.

What can we do when we don't get HAC for free?

- We can add implied constraints.

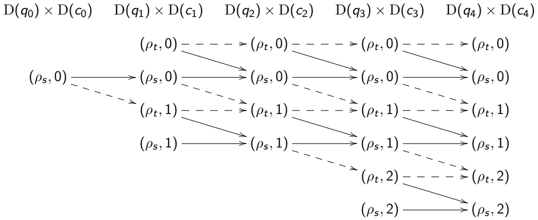
The Implied Constraints

The lower part can be eliminated by adding the constraints $c_{i+1} \leq (c_{i-1} + 1)$



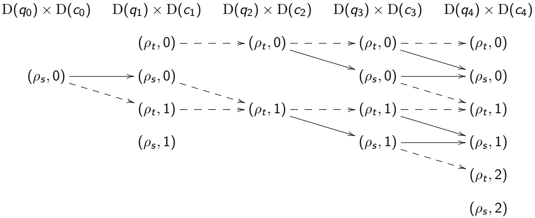
The Implied Constraints

The lower part can be eliminated by adding the constraints $c_{i+1} \leq (c_{i-1} + 1)$



When is HAC Lost Again?

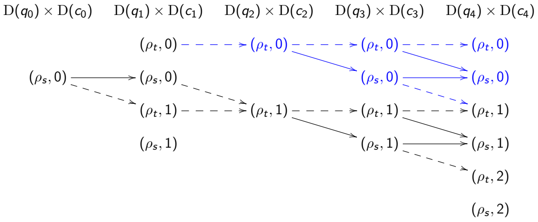
The *n_group* constraint after the assignment $s_2 := 1$ with the implied constraints $c_{i+1} \leq (c_{i-1} + 1)$.



Note that the upper part is partially disconnected

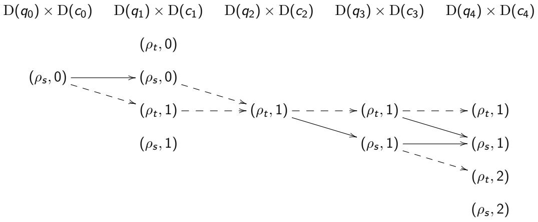
When is HAC Lost Again? (cont.)

We can eliminate the upper part by adding the constraint
$(q_{i-1} = \rho_s \wedge s_{i+1} = 1) \rightarrow (c_{i+1} > c_{i-1})$



When is HAC Lost Again? (cont.)

We can eliminate the upper part by adding the constraint
$(q_{i-1} = \rho_s \wedge s_{i+1} = 1) \rightarrow (c_{i+1} > c_{i-1})$



Example Summary

We derived the implied constraints:

- $c_{i+1} \leq (c_{i-1} + 1)$
- $(q_{i-1} = \rho_s \wedge s_{i+1} = 1) \rightarrow (c_{i+1} > c_{i-1})$

We maintain HAC in the whole decomposition

Conclusions and Future Work

Conclusions:

- Some decompositions do maintain HAC
- We can derive implied constraints and maintain HAC in other cases.
- We have done this for other constraints.

Future Work

- Can these be done automatically?

Questions?