Aalto University
School of Science

Declarative Encodings of Acyclicity Properties

Martin Gebser, Tomi Janhunen, Jussi Rintanen

Finnish Centre of Excellence in Computational Inference Research (COIN)

Computational Logic Day, December 16, 2014

Overview

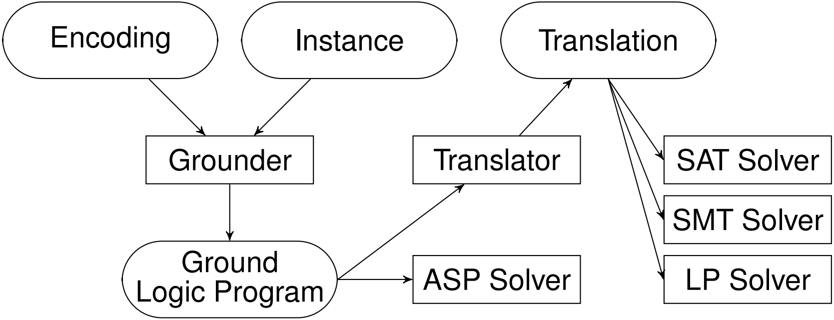
Motivation

- Numerous application problems involve the construction of acyclic or tree structures
  - Bayesian networks, Markov networks, Phylogenetic trees, ...
- Such structures are no basic primitives in common constraint-based representation formalisms, e.g.:
  - Answer Set Programming (ASP)
  - Boolean Satisfiability (SAT)
  - SAT Modulo Theories (SMT)
  - Mixed Integer Linear Programming (LP)

☞ Need for compact and efficient encodings

Approach

1. Uniform encoding in first-order ASP language
2. Automatic grounding w.r.t. instance data
3. Off-the-shelf solving in ASP or via translation to SAT / SMT / LP



Overview

Directed Graph Representation

- The following instance represents graphs with
  - 5 nodes
  - complete set of edge candidates

```
node(1). node(2). node(3). node(4). node(5).
pair(1,2).  pair(1,3).  pair(1,4).  pair(1,5).
pair(2,1).  pair(2,3).  pair(2,4).  pair(2,5).
pair(3,1).  pair(3,2).  pair(3,4).  pair(3,5).
pair(4,1).  pair(4,2).  pair(4,3).  pair(4,5).
pair(5,1).  pair(5,2).  pair(5,3).  pair(5,4).
```

- Edge generator rule:

```
{ edge(X,Y) } :- pair(X,Y).
```

Acyclicity Testing

Acyclicity test procedure

While there is a leaf $v \in V$:

$E := E \setminus \{\langle u, v\rangle \mid u \in V\}$

$V := V \setminus \{v\}$

Return $(V \stackrel{?}{=} \emptyset)$

Declarative ASP encoding

```
order(X,Y) :- pair(X,Y), not edge(X,Y).
order(X,Y) :- pair(X,Y), order(Y).
order(X)   :- node(X), order(X,Y) : pair(X,Y).
:- node(X), not order(X).
```

☞ Encoding is leaf-driven, non-tight, and linear

From Non-Tight to Tight (Leaf-Driven) Encoding

- Consider nodes with labels `1,...,n`

```
order(X,Y,1..n) :- pair(X,Y), not edge(X,Y).
order(X,Y,N-1)      :- pair(X,Y), order(Y,N), 1 < N.
order(X,N)          :- node(X), order(X,Y,N) : pair(X,Y).
:- node(X), not order(X,1).
```

☞ Encoding is leaf-driven, tight, and of size $\mathcal{O}(|E| \times |V|)$

Experimental Evaluation

600 edge candidates	leaf-driven encoding		root-driven encoding	
	non-tight	tight	non-tight	tight
clasp	**0.31**	80.26	**0.34**	**0.39**
clasp/sat	1.03	**0.86**	2.38	1.37
lingeling	2.62	1.78	1.82	1.70
z3	1.12	—	1.48	—
cplex	681.47	261.90	655.29	333.47

ASP solvers: clasp (3.1.0)

SAT solvers: clasp/sat, lingeling (ayv-86bf266-140429)

SMT solvers: z3 (4.3.2)

LP solvers: cplex (12.6.0.0)

Overview

From Directed Acyclic Graphs to Forests

- Directed forests are directed acyclic graphs such that every node has at most one incoming edge

☞ Directed acyclic graph encodings can be augmented with respective test

Pairwise mutual exclusion

```
:- edge(X,Z), edge(Y,Z), X < Y.
```

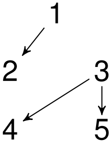
☞ Space complexity $\mathcal{O}(|E| \times |V|)$

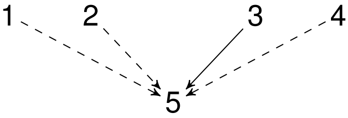
Cardinality constraint

```
:- node(Y), 2 { edge(X,Y) }.
```

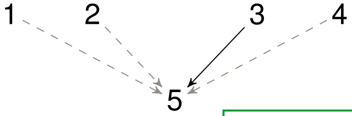
☞ Space complexity $\mathcal{O}(|E|)$

Linear "Normalized" At-Most-One Tests

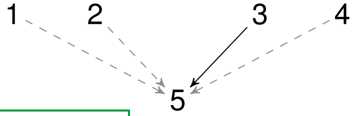




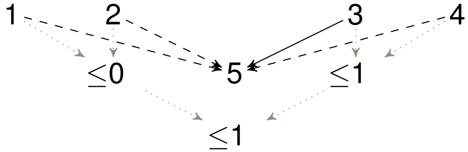
Linear traversal



Bidirectional traversal



Tournament traversal



Experimental Evaluation

clasp (3.1.0)	leaf-driven encoding non-tight	leaf-driven encoding tight	root-driven encoding non-tight	root-driven encoding tight
pairwise	5.93	8.18	7.44	8.95
cardinality	**3.69**	**6.41**	**5.30**	**5.11**
linear	7.29	8.47	8.50	7.08
bidirectional	4.68	6.44	6.14	6.94
tournament	12.36	8.56	8.41	8.61

Overview

From Directed Forests to Trees

- Directed trees are directed forests such that
  - there is (exactly) one root
  - there are (exactly) $|V|-1$ edges
  - all nodes are connected

☞ Directed forest encodings can be augmented with test(s)

One root

```
child(Y) :- edge(X,Y).
```

+ any at-most-one encoding over instances of '`not child(Y)`'

$|V|-1$ edges

```
:- not n-1 { edge(X,Y) } n-1.
```

Connectedness

```
reach(X) :- reach(Y), edge(X,Y).   reach(1).
reach(Y) :- reach(X), edge(X,Y). :- node(X), not reach(X).
```

Experimental Evaluation

clasp (3.1.0)	leaf-driven encoding non-tight	leaf-driven encoding tight	root-driven encoding non-tight	root-driven encoding tight
pairwise	2.29	**3.73**	2.78	**3.02**
cardinality	**2.14**	4.09	2.77	5.38
linear	8.42	11.06	8.93	46.92
bidirectional	5.46	6.47	2.87	3.99
tournament	11.56	5.09	11.68	7.81
$\geq \lvert V\rvert - 1$ edges	10.16	28.25	15.86	7.82
$= \lvert V\rvert - 1$ edges	12.81	16.43	12.18	38.47
connectedness	2.30	5.85	**2.72**	46.07

Conclusions

- Acyclic or tree structures are central in many applications
- Need for development and study of declarative encodings
- First-order ASP language (with recursion, cardinality and weight constraints, etc.) facilitates exploration of encodings
- Diverse formulations of acyclicity, forest, and tree conditions yield rich family of encoding variants to experiment with
- Grounding and automatic translations provide large variety of back-end solvers
- Given encodings furnish templates for corresponding tasks
- See paper for further study of logic-based characterizations of undirected forests and trees as well as chordal graphs