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

Background

Encoding Directed Acyclic Graphs

Encoding Directed Forests

Encoding Directed Trees

Conclusions
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

Background

Encoding Directed Acyclic Graphs

Encoding Directed Forests

Encoding Directed Trees

Conclusions
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 $(V \neq \emptyset)$

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
Consider nodes with labels $1, \ldots, n$

\[
\text{order}(X,Y,1..n) :- \text{pair}(X,Y), \text{not edge}(X,Y).
\]
\[
\text{order}(X,Y,N-1) :- \text{pair}(X,Y), \text{order}(Y,N), 1 < N.
\]
\[
\text{order}(X,N) :- \text{node}(X), \text{order}(X,Y,N) : \text{pair}(X,Y).
\]
\[
:- \text{node}(X), \text{not order}(X,1).
\]

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

<table>
<thead>
<tr>
<th>600 edge candidates</th>
<th>leaf-driven encoding</th>
<th>root-driven encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>non-tight</td>
<td>tight</td>
</tr>
<tr>
<td>clasp</td>
<td>0.31</td>
<td>80.26</td>
</tr>
<tr>
<td>clasp/sat</td>
<td>1.03</td>
<td>0.86</td>
</tr>
<tr>
<td>lingeling</td>
<td>2.62</td>
<td>1.78</td>
</tr>
<tr>
<td>z3</td>
<td>1.12</td>
<td>—</td>
</tr>
<tr>
<td>cplex</td>
<td>681.47</td>
<td>261.90</td>
</tr>
</tbody>
</table>

**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

Background

Encoding Directed Acyclic Graphs

Encoding Directed Forests

Encoding Directed Trees

Conclusions
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**

\[ \text{:- edge}(X,Z), \text{edge}(Y,Z), X < Y. \]

\( \mathcal{O}(|E| \times |V|) \)

**Cardinality constraint**

\[ \text{:- node}(Y), 2 \{ \text{edge}(X,Y) \}. \]

\( \mathcal{O}(|E|) \)
Linear "Normalized" At-Most-One Tests

Linear traversal

Bidirectional traversal

Tournament traversal
### Experimental Evaluation

<table>
<thead>
<tr>
<th>clasp (3.1.0)</th>
<th>leaf-driven encoding</th>
<th>root-driven encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>non-tight</td>
<td>tight</td>
</tr>
<tr>
<td>pairwise</td>
<td>5.93</td>
<td>8.18</td>
</tr>
<tr>
<td>cardinality</td>
<td>3.69</td>
<td>6.41</td>
</tr>
<tr>
<td>linear</td>
<td>7.29</td>
<td>8.47</td>
</tr>
<tr>
<td>bidirectional</td>
<td>4.68</td>
<td>6.44</td>
</tr>
<tr>
<td>tournament</td>
<td>12.36</td>
<td>8.56</td>
</tr>
</tbody>
</table>
Overview

Background

Encoding Directed Acyclic Graphs

Encoding Directed Forests

Encoding Directed Trees

Conclusions
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

<table>
<thead>
<tr>
<th>clasp (3.1.0)</th>
<th>leaf-driven encoding</th>
<th></th>
<th>root-driven encoding</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>non-tight</td>
<td>tight</td>
<td>non-tight</td>
<td>tight</td>
</tr>
<tr>
<td>pairwise</td>
<td>2.29</td>
<td>3.73</td>
<td>2.78</td>
<td>3.02</td>
</tr>
<tr>
<td>cardinality</td>
<td>2.14</td>
<td>4.09</td>
<td>2.77</td>
<td>5.38</td>
</tr>
<tr>
<td>linear</td>
<td>8.42</td>
<td>11.06</td>
<td>8.93</td>
<td>46.92</td>
</tr>
<tr>
<td>bidirectional</td>
<td>5.46</td>
<td>6.47</td>
<td>2.87</td>
<td>3.99</td>
</tr>
<tr>
<td>tournament</td>
<td>11.56</td>
<td>5.09</td>
<td>11.68</td>
<td>7.81</td>
</tr>
<tr>
<td>≥</td>
<td>V</td>
<td>−1 edges</td>
<td>10.16</td>
<td>28.25</td>
</tr>
<tr>
<td>=</td>
<td>V</td>
<td>−1 edges</td>
<td>12.81</td>
<td>16.43</td>
</tr>
<tr>
<td>connectedness</td>
<td>2.30</td>
<td>5.85</td>
<td>2.72</td>
<td>46.07</td>
</tr>
</tbody>
</table>
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