Modeling overlapping structures: Graphs and serializability

Yves Marcoux

Université de Montréal, Canada

Michael Sperberg-McQueen

Black Mesa Technologies

Claus Huitfeldt

University of Bergen, Norway

Overview of the talk

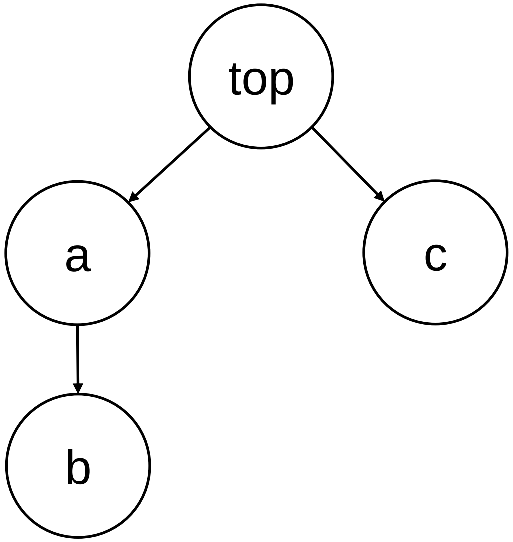
1. Graph representations of structured documents
   - Overlap, in markup and structure
   - Overlap-only-TexMECS
   - Child-arc-ordered DGs (CODGs)
2. 2008 / 2011 results and consequences
3. Solution to the 2011 thorn
4. Future work

1. Graph representations of structured documents

XML document = tree

```
<top>
  <a>
    <b/>
  </a>
  <c/>
</top>
```

⇔

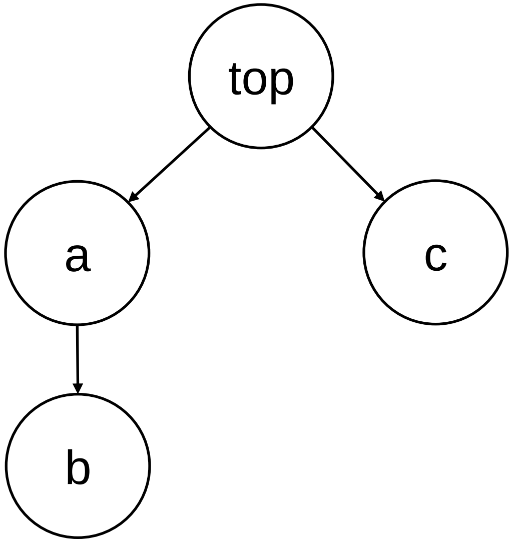


Embedding in markup ⇔ Ancestor-descendant in tree

Any tree ⇒ an XML document

```
<top>
  <a>
    <b/>
  </a>
  <c/>
</top>
```

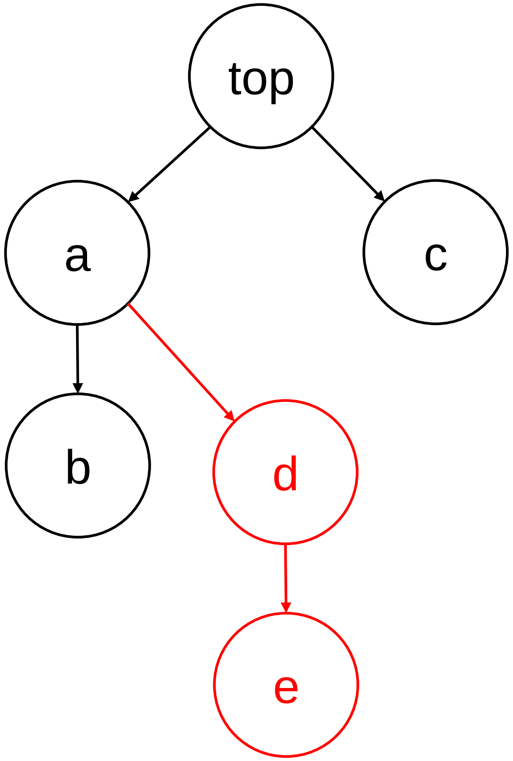
⇔



Any tree ⇒ an XML document

```
<top>
  <a>
    <b/>
    <d><e/></d>
  </a>
  <c/>
</top>
```

⇔



Perfect isomorphism !

In a graph-based editor…

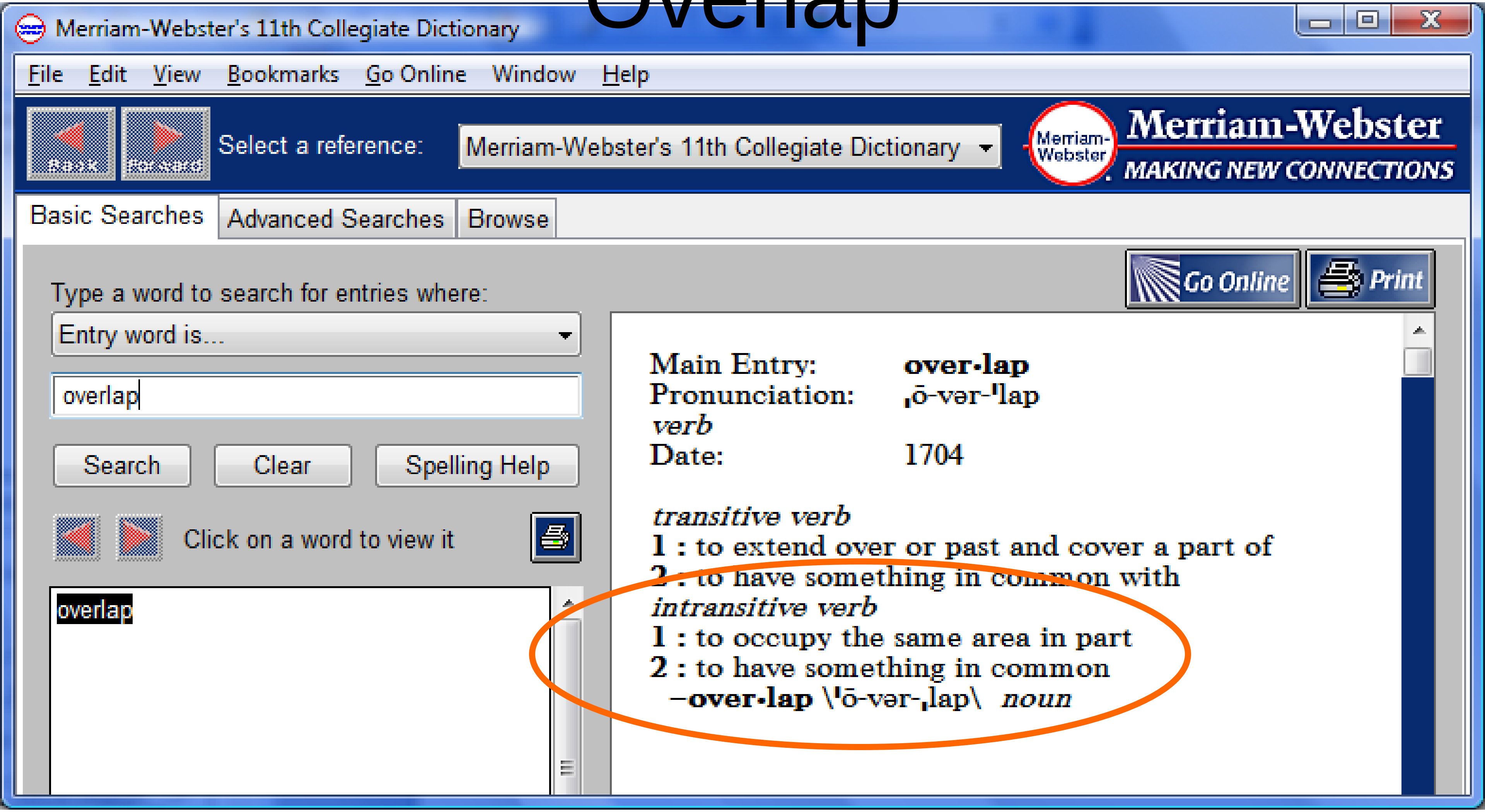
- It suffices to make sure that the graph remains a tree
  - Guarantees serializability as an XML document

Overlap, in markup and structure

Problem of overlap

- In real life (outside of XML documents!), information is often not purely hierarchical
- Classical examples:
  - verse structure *vs* sentence structure
  - speech structure *vs* line structure
- In general: multiple structures applied (at least in part) to same contents

Overlap



Type a word to search for entries where:

Entry word is...

overlap

overlap

Main Entry:	**over·lap**
Pronunciation:	ˌō-vər-ˈlap
verb	
Date:	1704

transitive verb

1 : to extend over or past and cover a part of

2 : to have something in common with

intransitive verb

1 : to occupy the same area in part

2 : to have something in common

–**over·lap** \ˈō-vər-ˌlap\ *noun*

Two views of overlap

- Geometric view: overlapping *markup*
- Common contents view:
  non-tree *graph structure*

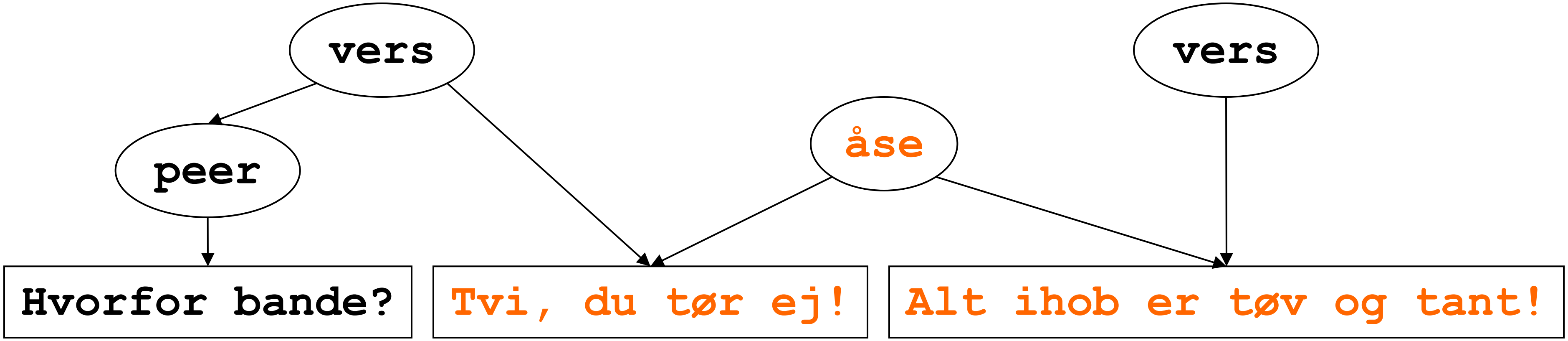
Example (markup)

(Peer) Hvorfor bande? (Åse) Tvi, du tør ej!
Alt ihob er tøv og tant!

```
<vers>
  <peer>Hvorfor bande?</peer><åse>Tvi, du tør ej!
</vers>
<vers>
  Alt ihob er tøv og tant!</åse>
</vers>
```

Example (graph)

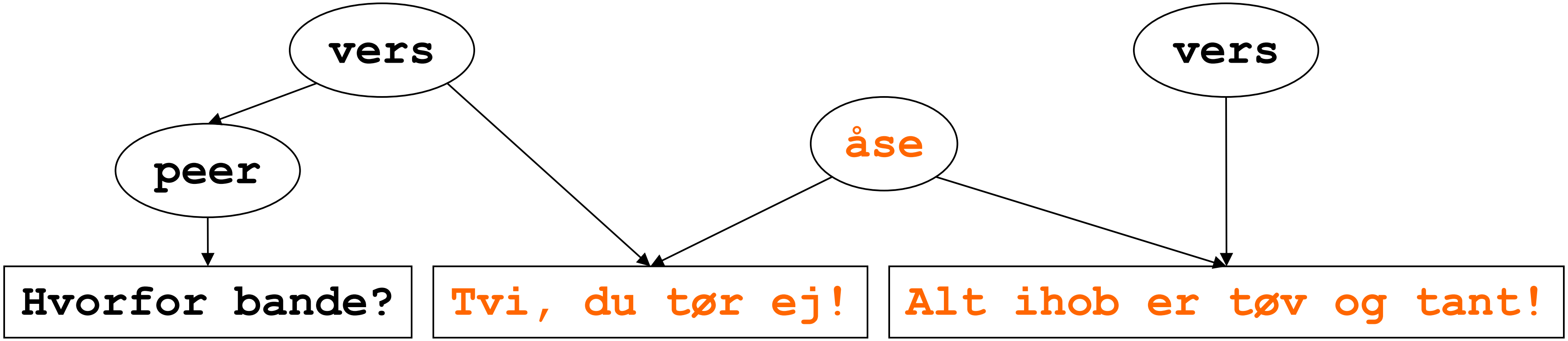
(Peer) Hvorfor bande? (Åse) Tvi, du tør ej!
Alt ihob er tøv og tant!



Document – graph correspondence ?

Embedding ⇔ Ancestor-descendant ? Yes, still true

```
<vers>
  <peer>Hvorfor bande?</peer><åse>Tvi, du tør ej!
</vers>
<vers>
  Alt ihob er tøv og tant!</åse>
</vers>
```

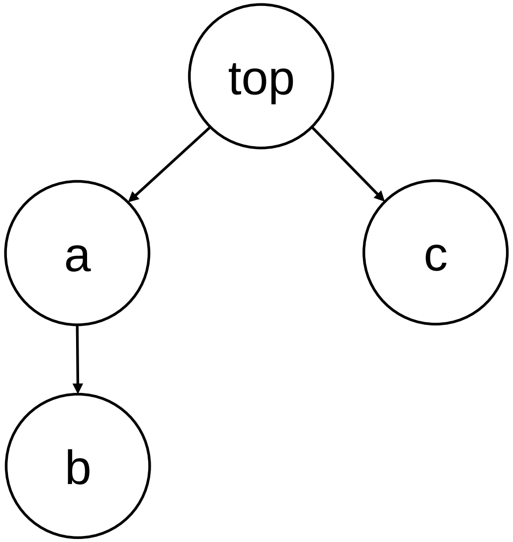


XML document = tree

FLASHBACK

```
<top>
  <a>
    <b/>
  </a>
  <c/>
</top>
```

⇔



Embedding in markup ⇔ Ancestor-descendant in tree

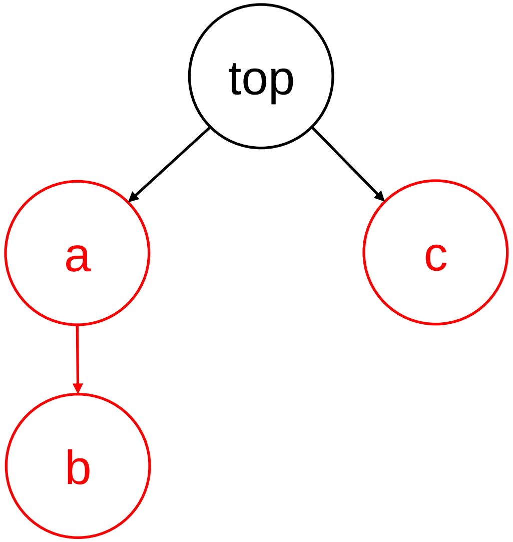
Since order matters….

- There was a (tacit) *additional rule* for a tree to correspond to a document:
  - Any node must correspond to a segment of the document that *precedes* the segment corresponding to any younger sibling
- … and since overl*happens* ***not*** in XML
  - “precedes” really means “ends before the other starts”

Example

```
<top>
  <c/>
  <a>
    <b/>
  </a>
</top>
```

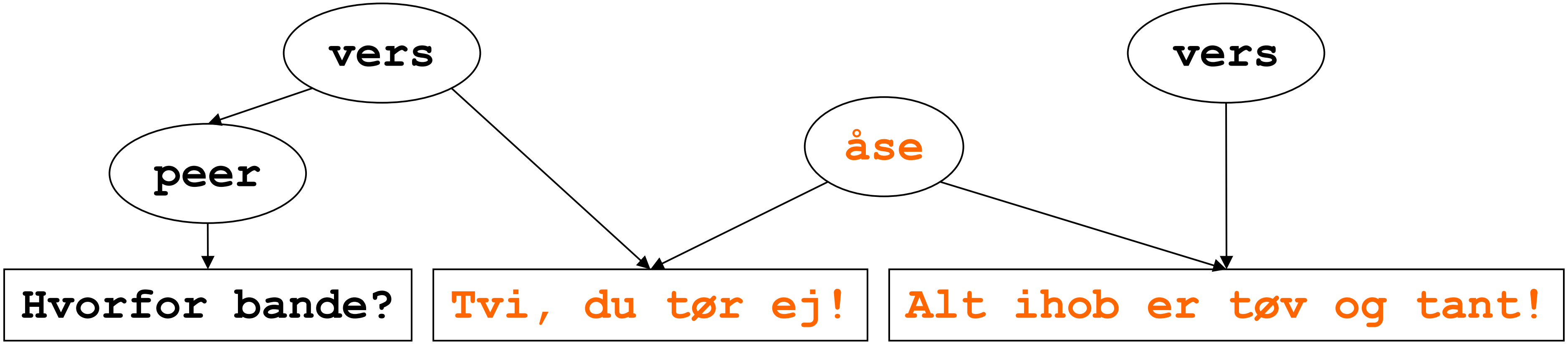
$\not\Leftrightarrow$



Need: additional rule for order

Any node "starts before" its younger siblings

```
<vers>
  <peer>Hvorfor bande?</peer><åse>Tvi, du tør ej!
</vers>
<vers>
  Alt ihob er tøv og tant!</åse>
</vers>
```

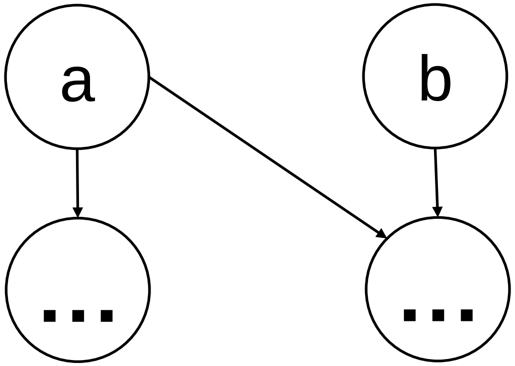


Please, observe that…

- Any node will not only start before, but also *end before* all of its younger siblings, because
- Otherwise, it would be a *parent* and not an older sibling
  - By virtue of the “embedding ⇔ ancestor-descendant” rule

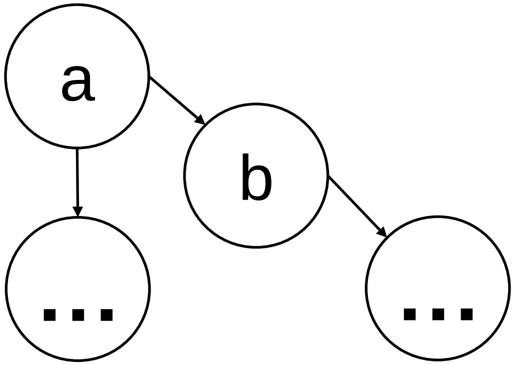
Visually… (1/2)

`<a>...<b>...</a></b>`



Visually… (2/2)

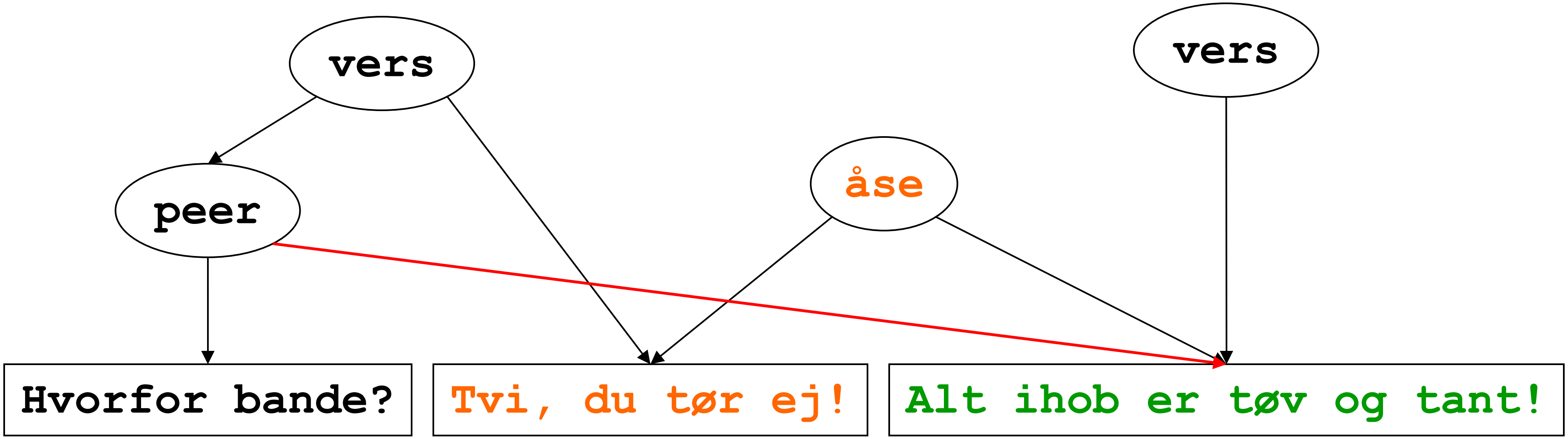
`<a>...<b>...</b></a>`



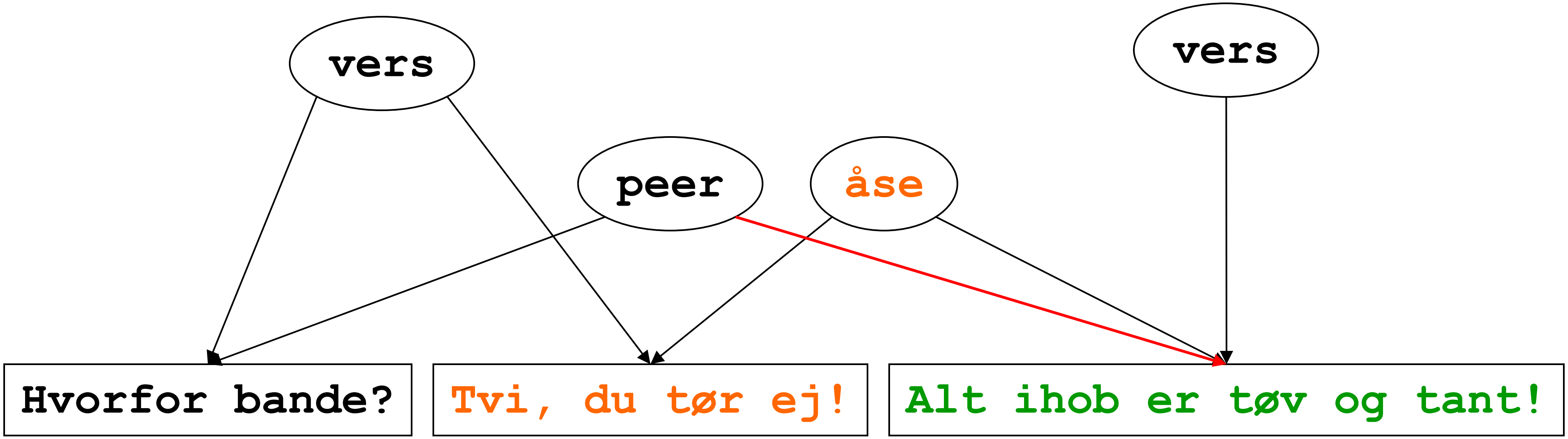
Still perfect isomorphism?

- Not for graphs in general... cycles !
- Maybe for *acyclic* graphs ?
- Let's try more examples...

What if the last verse is said in chorus by Peer & Åse ?



Last verse in chorus (alt.)



Either way: acyclic, but… *no corresponding OO-document !*

So, *imperfect* isomorphism

- Some acyclic graphs have corresponding OO-documents
- Some (apparently) don't...
- Manipulations of the graph (DOM) may leave it non-OO-serializable!

- *Which graphs have corresponding OO-documents?*

Overlap-only-TexMECS

TexMECS

- A particular proposal to address the overlap problem *with overlapping markup*
- MECS (Huitfeldt 1992-1996)
  - Multi-element code system
- TexMECS (Huitfeldt & SMcQ 2003)
  - "Trivially extended MECS"
- Markup Languages for Complex Documents (MLCD) project

Overlap-only TexMECS

- TexMECS allows overlapping markup...
- but also much more:
  - virtual elements, interrupted elements, etc.
- OO-TexMECS 101
  - Start-tags: **`<a|`**
  - End-tags: **`|a>`**
  - Overlapping elements allowed
  - Natural notion of well-formedness

Child-arc-ordered DGs (CODGs)

Content ordering

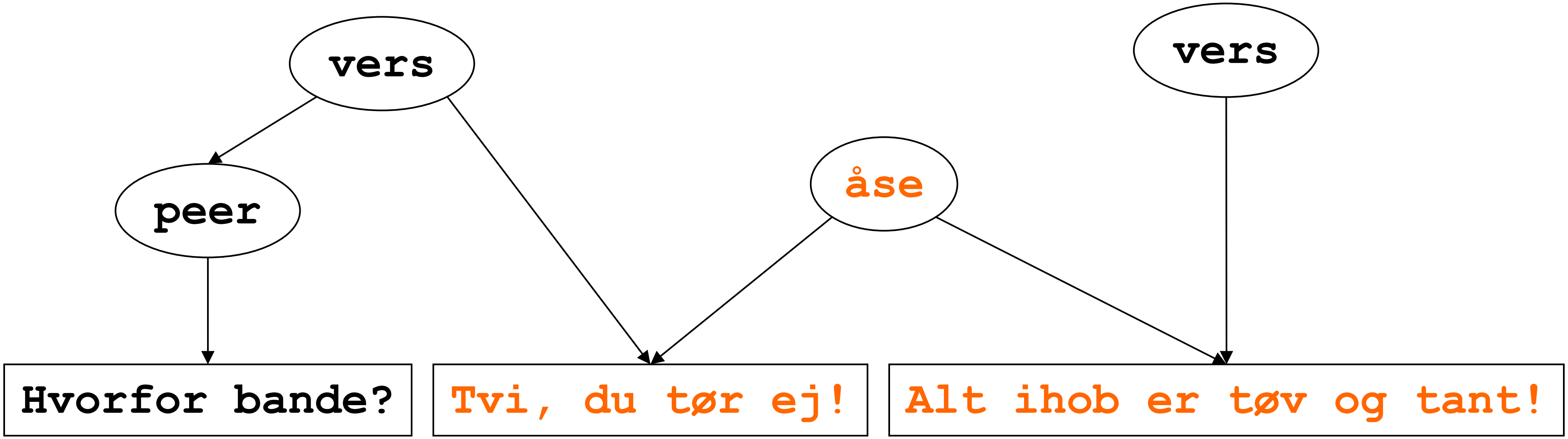
- In a serialized document, content appears in some order
- The order is often significant
  - Procedure steps, verses in a poem, etc.
- Thus, the order must be present in graph representations
  - XML: children ordered

Child-arc-ordered DGs

- CODGs (pronounced “codger”)
- Essentially a DG with outgoing arcs (“child-arcs”) ordered

CODG example 1

(Peer) Hvorfor bande? (Åse) Tvi, du tør ej!¶
Alt ihob er tøv og tant!¶

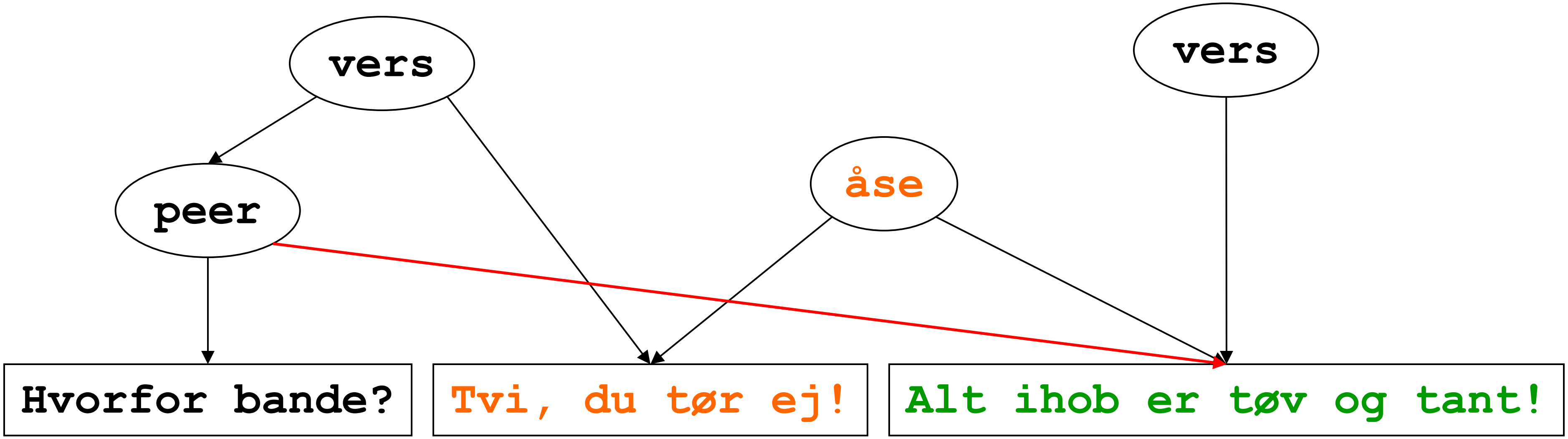


vers
peer
vers
åse
vers

Hvorfor bande?
Tvi, du tør ej!
Alt ihob er tøv og tant!

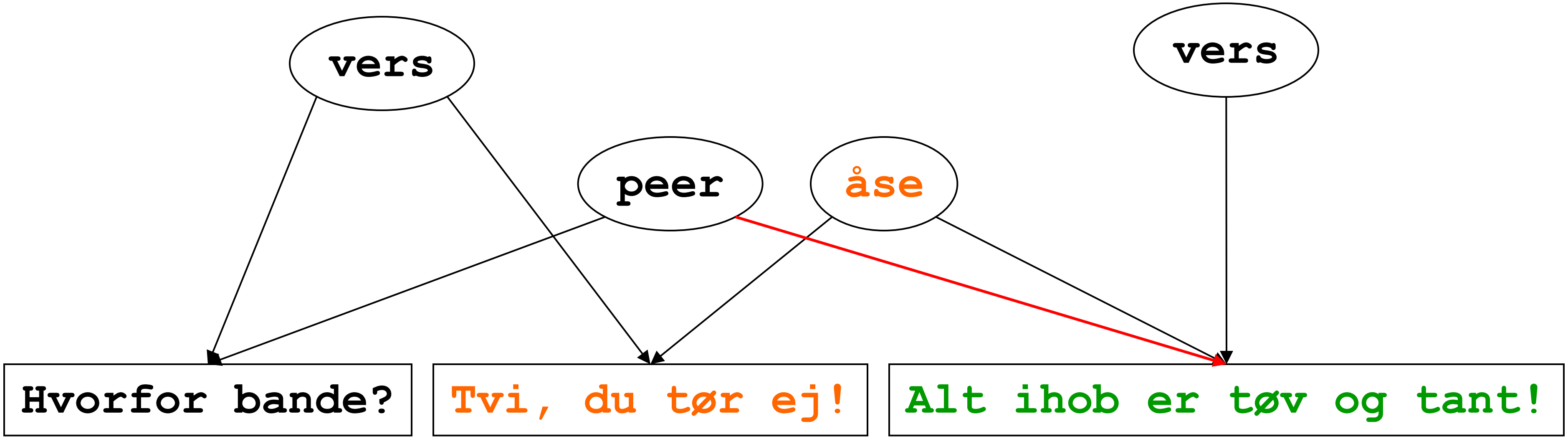
CODG example 2

(last verse in chorus)



CODG example 3

(last verse in chorus, alt.)



vers

vers

peer

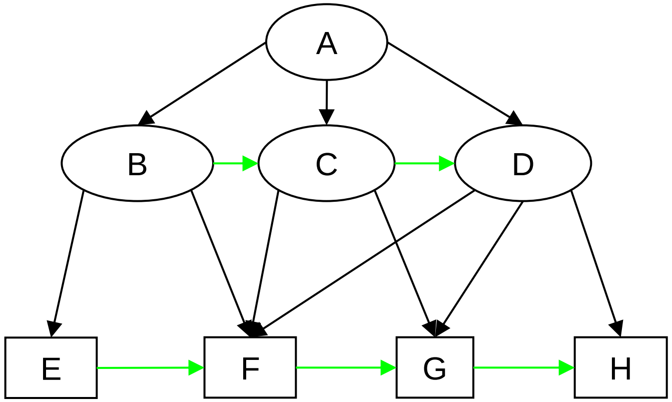
åse

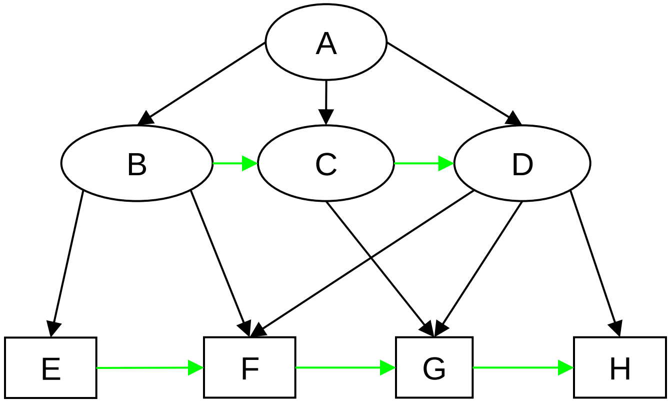
Hvorfor bande?

Tvi, du tør ej!

Alt ihob er tøv og tant!

CODG examples 4 and 5





2. 2008 / 2011 results and consequences

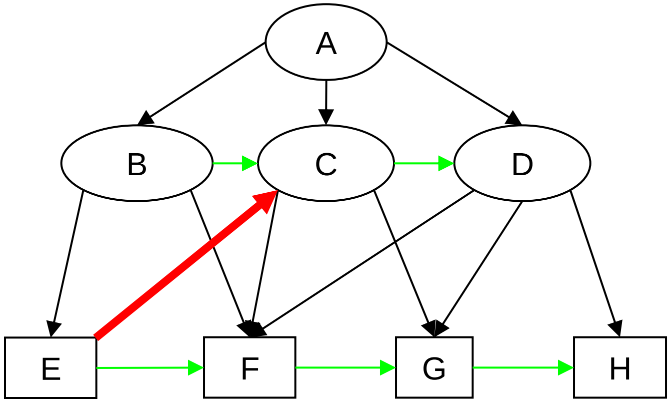
Remember the question?

- *Which graphs have corresponding OO-documents?*
- That is:
  - Which CODGs have a corresponding OO-TexMECS document?
- Or (equivalently):
  - Which CODGs are serializable in OO-TexMECS?

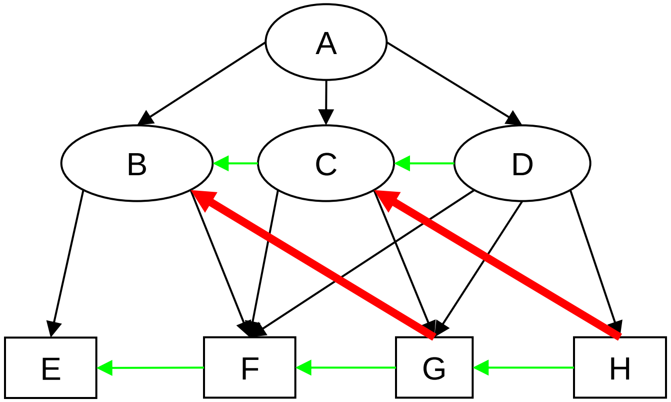
Answer: completions

- Intuition: combination of parent-child relationships & child-arc-ordering dictates constraints on the relative positioning of certain tags in any *eventual* corresponding OO-TexMECS document
- When contradictory constraints are observed: the graph is not OO-TexMECS serializable

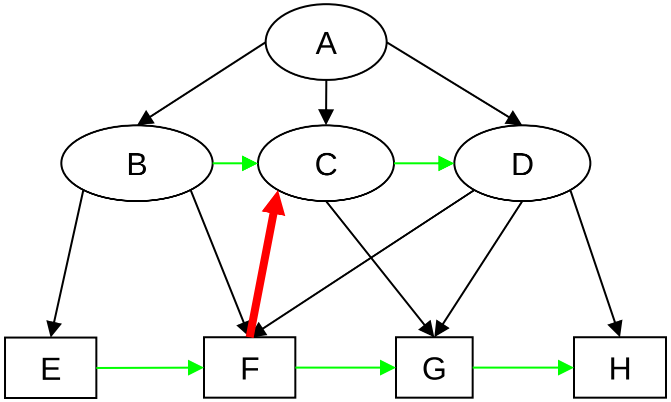
"starts-before" completion



"ends-after" completion

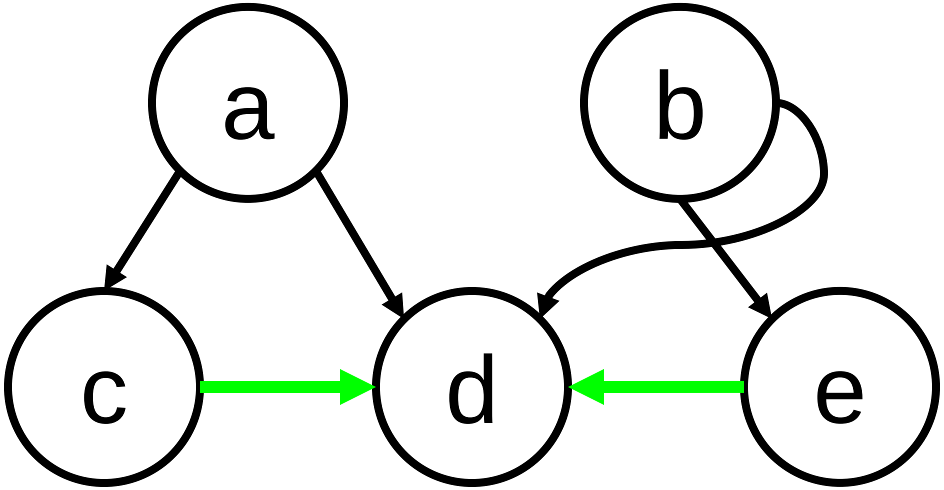


Cycle = contradiction !



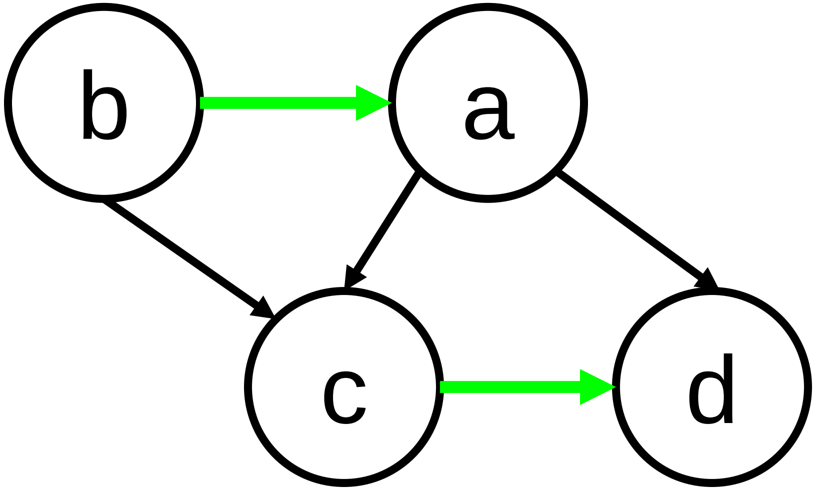
Completion-acyclic ≡ each completion is acyclic

But, if more than one root



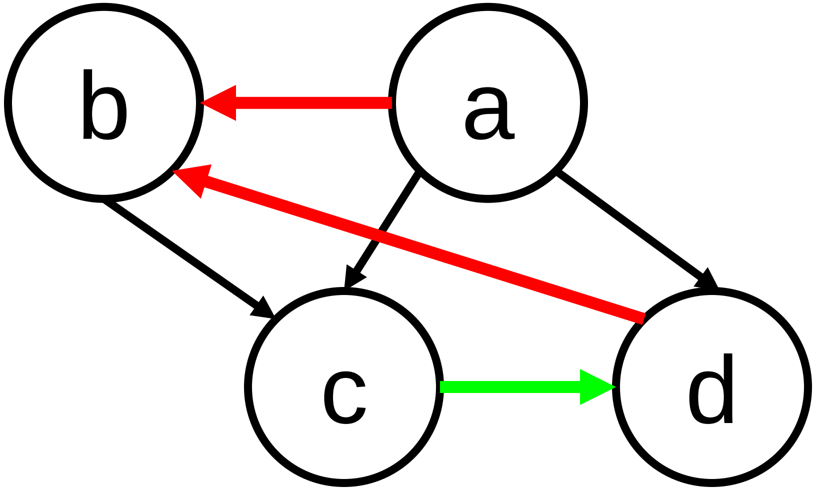
- Completion-acyclic, yet not OO-TexMECS serializable
- Reason: unordered roots
  - – If ordered (either way): completion-*cyclic*

But how we order roots matters…



- OK: no completion is cyclic

But how we order roots matters…



- SB-completion is cyclic !

Full-completion-acyclicity

- If *at least one* root-ordering gives rise to acyclic completions, we say the CODG is *fully completion-acyclic*

Results and consequences (1/3)

- A CODG is serializable in OO-TexMECS iff it is FCA
- Any OO-TexMECS well-formed document can be obtained by serializing some FCA CODG
- You don't gain any expressivity by allowing ordering over and above child-arc-ordering

Results and consequences (2/3)

- We now know what to check in a CODG to maintain OO-serializability (FCA)
- A graph-based editor is *complete*
- Round-tripping is possible between FCA CODGs and OO-TexMECS
- Results also apply to similar formalisms
  - TexMECS with more features *except* virtual and interrupted elements

Results and consequences (3/3)

- *If you need to express more complex structures than FCA CODGs, you must extend XML with more than overlap*
  - or, of course, use *ad hoc* layers of semantics

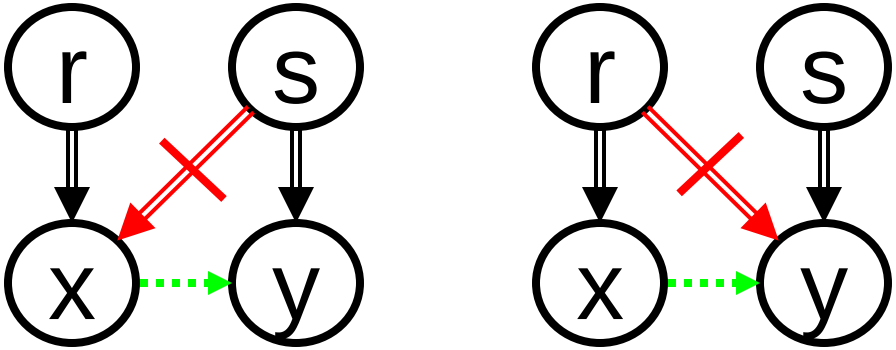
“2011 thorn”

- How do we determine whether *some* ordering of the roots of a CODG gives rise to acyclic completions?
  - We can try them all…
  - But for n roots, there are n! orderings
    - That’s a lot…
- How do we *find* such a root-ordering?

3. Solution to the 2011 thorn

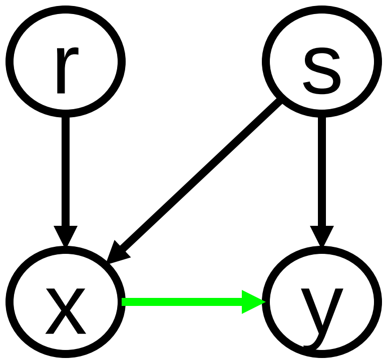
Take pairs of roots

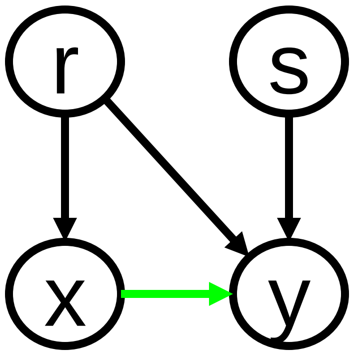
- There are “only” n^2 pairs
- Look for one of these patterns:

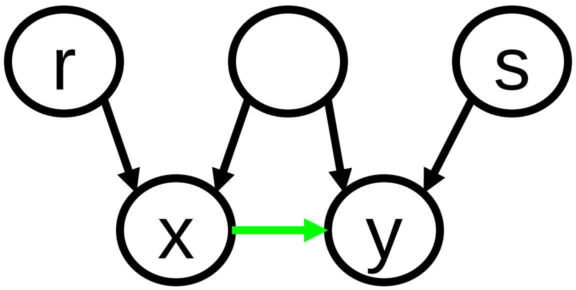


- If found, conclude: r *must start before* s

Examples





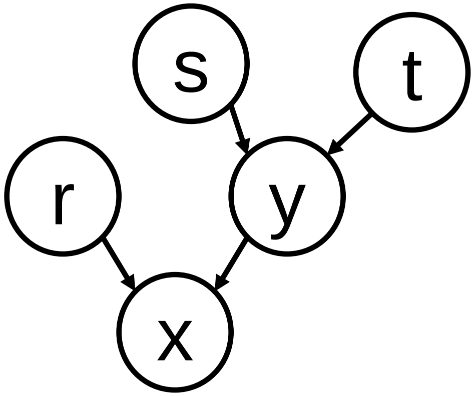


Key point

- After processing all pairs of roots
- if no cycle has been found
- then, the CODG is FCA

Last point

- Some pairs may remain unordered



- They can be ordered “almost” randomly to get a complete root-ordering
  - and, hence, an OO-TexMECS serialization

Results

- FCA can be determined in poly-time
- An OO-TexMECS serialization of an FCA CODG can be computed in poly-time

4. Future work

- Optimal algorithm for verification of FCA
- Optimal serialization algorithm
- Exact relationships with GODDAGs, in particular, *restricted* GODDAGs (SMcQ & Huitfeldt 2004)

Thank you !

Questions ?

<ymarcoux@gmail.com>