Searching for the Holy Grail

Ian Horrocks
<ian.horrocks@comlab.ox.ac.uk>
Information Systems Group
Oxford University Computing Laboratory
Background and Motivation

• Medicine has a large and complex vocabulary
• Long history of “formalising” and codifying medical vocabulary
  – Numerous medical “controlled vocabularies” of various types
• Large size of static coding schemes makes them difficult to build and maintain
  – Many terminologies specific to purpose (statistical analysis, bibliographic retrieval), specialty (epidemiology, pathology) or even database
  – Ad hoc terms frequently added to cover fine detail required for clinical care
GALEN Project

Goals of the project were:

• Design/select an appropriate (for medical terminology) modelling language: **GRAIL\(^1\)**

• Develop tools to support conceptual modelling in this language: **GRAIL classifier** (amongst others)

• Use these tools to develop a suitable model of medical terminology: **GALEN terminology** (aka ontology)

\(^1\)GALEN Representation And Integration Language
Problems

• Recognised:
  – Classifier too slow
    • over 24 hours to classify ontology

• Unrecognised:
  – Vague semantics
    • no formal specification or mapping to (description) logic
  – Language lacked many features
    • cardinality restrictions (other than functional roles)
    • negation and disjunction (not even disjointness)
  – Reasoning via ad hoc structural approach
    • incorrect w.r.t. any reasonable semantics
Why Not Use a Description Logic?

• Advantages:
  – Formalise semantics via mapping to DL
  – Algorithms relatively simple and clearly described
  – Already some work on implementation & optimisation (KRIS)

• Disadvantages:
  – Only relatively simple DLs had so far been implemented
  – GALEN used transitive and functional roles, role hierarchy and “General Concept Inclusion axioms” (GCIs)

Idea: extend Baader/Sattler transitive orbits to (transitive and functional) role hierarchy, and internalise GCIs
Optimising (Tableau) Reasoners

- Reasoner based on published algorithms fails to complete a single GALEN subsumption test
- Performance problems mainly caused by GCIs
  - standard “theoretical” technique is to use internalisation:
    \[ C \subseteq D \iff \top \subseteq (D \sqcup \neg C) \], and
    \[ (D \sqcup \neg C) \] applied to every individual using a “universal role”
  - convenient for proofs, but hopelessly inefficient in practice
    - over 1,200 GCIs in GALEN ontology

\textbf{Lesson}: Theory \( \neq \) practice!
Optimising (Tableau) Reasoners

Idea: suggested by structure of GALEN KB
- GCIs all of the form $C_1 \cap \ldots \cap C_n \subseteq D$
- can be rewritten as $C_1 \subseteq D \cup \neg(C_2 \cap \ldots \cap C_n)$
- and “absorbed” into primitive “definition” axiom for $C_1$
- resulting TBox is “definitorial”
  - no GCIs
  - dealt with via lazy unfolding

Result: close, but no cigar
- search space still too large
- effective non-termination
Optimising (Tableau) Reasoners

Idea: Investigate other optimisations, e.g., from SAT
- simplifications (e.g., Boolean Constraint Propagation)
- semantic branching
- caching
- heuristics
- smart backtracking

Result: (qualified) success!
- “FaCT” reasoner classified GALEN core in <400s
Qualifications

• Only works for GALEN “core”
  – full ontology is much larger & couldn’t be classified by FaCT

• No support for complex roles
  – GRAIL allows for axioms of form \((r \circ s) \sqsubseteq r\)

• Weak (cheating?) semantics for inverse roles
  – GRAIL treats them as pre-processing macros:
    \((r \circ s) \sqsubseteq r \Leftrightarrow (s^- \circ r^-) \sqsubseteq r^-\)

Result: progress, but still searching for the Holy Grail
Extending the Logic

- Qualified Cardinality Restrictions ($\mathcal{Q}$)
- Inverse roles ($\mathcal{I}$)
  - loss of finite model property
  - requires new “double blocking” technique
- Nominals ($\mathcal{O}$)
  - interaction with $\mathcal{QI} \rightsquigarrow$ new nominal introduction rule
  - complexity increases to NExpTime-complete
- Complex role inclusions ($\mathcal{R}$)
  - roles treated as automata
  - Complexity increases to 2NExpTime
New Algorithms and Optimisations

- HyperTableau algorithm
- Caching and individual reuse
- Exploiting constructed models
- Optimised “KP” classification
- Optimised blocking
- ...

Result:
- ✓ SROIQ can (easily) capture Grail
- ✓ Performance greatly improved (in general)
- ❌ But still can’t classify GALEN
- ❌ Some other ontologies still problematical
Scalability Issues

Problems with very large and/or cyclical ontologies

- E.g. SNOMED defines 100s of thousands of terms
  - individual tests trivial, but huge number needed for classification
- E.g., cycles in GALEN lead to construction of very large models

LeftSide \subseteq \exists hasComponent.AorticValve

LeftSide \subseteq \exists hasComponent.MitralValve

AorticValve \subseteq \exists hasConnection.LeftVentricle

MitralValve \subseteq \exists hasConnection.LeftVentricle

LeftVentricle \subseteq \exists isDivisionOf.LeftSide
Solutions?

Use tractable fragment such as $E\mathcal{L}++$

✓ PTime algorithm for classification

✓ Works well in practice

✓ Expressivity sufficient for some life science ontologies, including SNOMED

✗ Not expressive enough for GALEN

✗ Not clear that e.g. anatomy can be faithfully modelled

✗ Development and repair of ontologies tends to push them outside this fragment
Case Study: SNOMED

- **Kaiser Permanente** extending SNOMED to express, e.g.:
  - *non-viral pneumonia* (negation)
  - *infectious pneumonia* is caused by a *virus* or a *bacterium* (disjunction)
  - *double pneumonia* occurs in *two lungs* (cardinalities)

- This is easy in **SNOMED-OWL**
  - but reasoner failed to find expected subsumptions, e.g., that *bacterial pneumonia* is a kind of *non-viral pneumonia*

- Ontology highly under-constrained: need to add disjointness axioms (at least)
  - *virus* and *bacterium* must be disjoint
Case Study: SNOMED

• Adding disjointness led to *surprising results*
  – many classes become inconsistent, e.g., *percutanious embolization of hepatic artery using fluoroscopy guidance*

• One cause of *inconsistencies* identified as class *groin*
  – *groin* asserted to be subclass of both *abdomen* and *leg*
  – *abdomen* and *leg* are disjoint
  – modelling of *groin* (and other similar “junction” regions) identified as incorrect
Case Study: SNOMED

• Faithful modelling of groin is quite complex, e.g.:
  – groin has a part that is part of the abdomen, and has a part that is part of the leg (*inverse properties*)
    \[
    \text{Groin} \sqsubset \exists \text{hasPart}.(\exists \text{isPartOf}.\text{Abdomen})
    \]
    \[
    \text{Groin} \sqsubset \exists \text{hasPart}.(\exists \text{isPartOf}.\text{Leg})
    \]
    \[
    \text{hasPart} \equiv \text{isPartOf}^-
    \]
  – all parts of the groin are part of the abdomen or the leg (*disjunction*)
    \[
    \text{Groin} \sqsubset \forall \text{hasPart}.(\exists \text{isPartOf}.(\text{Abdomen} \sqcup \text{Leg}))
    \]
  – ...

Other Solutions?

Use PAYG “consequence-based” algorithm
✓ Deductive reasoning extending $\mathcal{EL}^{++}$ algorithm
✓ PTime when ontology inside relevant fragment
✓ Optimised implementation works well in practice for Horn−$SHIQ$ ontologies (CB reasoner)
✓ Encouraging early results even beyond Horn (ConDOR reasoner)
✓ Expressive enough for GALEN
## Preliminary Evaluation

<table>
<thead>
<tr>
<th></th>
<th>FaCT++</th>
<th>HermiT</th>
<th>Pellet</th>
<th>CEL</th>
<th>CB</th>
<th>ConDOR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GO</strong></td>
<td>15.2</td>
<td>199.5</td>
<td>72.0</td>
<td>1.8</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td><strong>NCI</strong></td>
<td>6.0</td>
<td>169.5</td>
<td>26.5</td>
<td>5.8</td>
<td>3.6</td>
<td></td>
</tr>
<tr>
<td><strong>SNOMED</strong></td>
<td>650.4</td>
<td>–</td>
<td>–</td>
<td>1185.7</td>
<td>51.8</td>
<td>40.4</td>
</tr>
<tr>
<td><strong>GALEN-Ł</strong></td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>4.6</td>
<td>4.9</td>
<td></td>
</tr>
<tr>
<td><strong>GALEN v.0</strong></td>
<td>465.4</td>
<td>45.7</td>
<td>–</td>
<td>n/a</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td><strong>GALEN v.7</strong></td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>n/a</td>
<td>9.6</td>
<td></td>
</tr>
<tr>
<td><strong>SAM+</strong></td>
<td>2324.1</td>
<td>–</td>
<td>–</td>
<td>n/a</td>
<td>n/a</td>
<td>88.9</td>
</tr>
<tr>
<td><strong>OBI</strong></td>
<td>153.8</td>
<td>2.5</td>
<td>11.8</td>
<td>n/a</td>
<td>n/a</td>
<td>0.6</td>
</tr>
<tr>
<td><strong>FMA-C</strong></td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>n/a</td>
<td>n/a</td>
<td>11.7</td>
</tr>
<tr>
<td><strong>Wine</strong></td>
<td>0.1</td>
<td>0.7</td>
<td>1.7</td>
<td>n/a</td>
<td>n/a</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Discussion

Not clear that $\mathcal{EL}++$ suffices for many applications
- Existing $\mathcal{EL}++$ ontologies may only be “historical accidents”
- Some form of counting needed in many applications
- Clear case for SNOMED to be extended beyond $\mathcal{EL}++$

$\mathcal{EL}++$ techniques can be lifted to Horn and beyond
- Extended algorithms still optimal on $\mathcal{EL}++$ fragment
- Perform very well on Horn-$\mathcal{SHIQ}$ ontologies
- Encouraging preliminary results for more expressive languages
Discussion

Lessons learned:

– **Deductive** algorithms highly effective (on some ontologies)
– **Optimisations** are still crucial
  • Some optimisations even feed back to tableau provers
– Extension to SROIQ seems **challenging**
  • But we are trying!
Discussion

Lessons learned:

– Deductive algorithms highly effective (on some ontologies)
– Optimisations are still crucial
  • Some optimisations even feed back to tableau provers
– Extension to SROIQ seems challenging
  • But we are trying!

Already found the holy grail, but we want to go further!
Thanks To

- Yevgeny Kazakov
- Boris Motik
- Rob Shearer
- Birte Glimm
Thank you for listening

Any questions?