Computational models for contracts

John J. Camilleri
UiO talk, 2014-05-13
About me

- No background in law
- PhD student in Computer Science at Chalmers / University of Gothenburg
- Formal methods & language technology
REMU

Reliable Multilingual Digital Communication: Methods and Applications

http://remu.grammaticalframework.org/
1. Hybrid machine translation
   ○ Grammars + statistics
2. Formal methods for grammars
   ○ Ambiguity detection
3. Reasoning
What is a contract?

- **Document containing norms** prescribing procedures, behaviours, rights

- **Examples**
  - requirement specification
  - privacy policy
  - terms of service
  - service-level agreement
Motivating example

Application procedure

1. Applications may be submitted between 1st–31st May.
2. The secretary must process each application within 5 days.
3. The secretary should not process any applications after the period has closed.
Motivating example

Application ✓ Processing ✓ Application ✓ Processing ? Application ✗
Potential conflicts

- Is there something wrong with this contract?
- Should it be changed?

- That’s for a human to decide
- Computer: highlight potential conflicts for us
Desired tasks: static

● Detect conflicts
  ○ While writing (author)
  ○ Before accepting (party)

● Query
  ○ Am I allowed to …? 
  ○ What happens if …?

● Simulate
  ○ Discover undesirable possibilities, loopholes
Desired tasks: runtime

- Detect violations
- Enact reparations
- Logs, without interference

- Only computer-mediated transactions
The dream

Static
- Original contract
  - Contract model
    - Query Language
    - Properties
    - Counter-example analysis
    - Counter-example analysis
    - Violation analyser
    - Log

Runtime
- Monitor
- Violation analyser
E-contracts and models

- Computers cannot process raw text
- Use formal **model** to represent our real-world contract
Originals & models

Natural Language

Original contract

Formalism

\[ P(su) \land [\text{cl}]F_{\infty}(\text{pr.co}) \oplus O_{\perp}(xy \land (w \lor z)) \]

\[
\begin{align*}
\text{<contract>} \\
\text{<clauses>} \\
\text{<clause>[a3](P(a1))</clause>} \\
\text{<clause>(P(a2.a3))</clause>} \\
\text{</clauses>} \\
\text{<concurrentActions>} \\
\text{<action>a3#a4</action>} \\
\text{</concurrentActions>} \\
\text{</contract>}
\]
The formalism: Contract Logic (CL)

- A formal language for specifying “contracts”
- Action-based
  - Simple and complex actions
  - and, exclusive or, sequence
- Main modalities
  - Obligation
  - Permission
  - Prohibition
- Reparations
Examples of CL (1)

Student is **allowed** to submit an application

```
P(submit)
```

Secretary **must** process application and then send the reply

```
O(process . reply)
```
Examples of CL (2)

1. Student is allowed to submit an application
2. If the period is closed, the secretary should not process any application

\[ P(\text{submit}) \land \lbrack \text{close}\rbrack F(\text{process}) \]
Conflicts

- What is a conflict
  - $O(a)$ and $F(a)$
  - $P(a)$ and $F(a)$, etc.
- Only easy in tiny examples

$$P(\text{submit}) \land F(\text{submit} \& \text{process})$$
Conflicts: big example

- Quickly becomes difficult

\[ [open]P(\text{submit}) \land [\text{submit}]O(\text{process}) \land [\text{close}]F(\text{process}) \]

- CL Analyser “CLAN”
- Finds conflicts, gives counter-example
Counter-example

- Trace of actions which lead to conflict
  - 1. open
  - 2. submit
  - 3. close & submit

- May not be minimal!
What could be simpler?

- Nearly impossible to do manually (without making mistakes)
  - Writing long formulas (model)
  - Analysing counter-examples

- Goal
  - Bridge the gap between formal model and natural language
The divide

Natural Language

Original contract

Formalism

Model

Analysis tools
The divide

Natural Language
- Original contract

Controlled Natural Language
- CNL model

Formalism
- Model

Analysis tools
Controlled Natural Language

- Simplified version of NL
- Restricted syntax, vocabulary
- Formal language (described precisely by grammar)
- Parse-able
- Unambiguous (ideally)
- Still human-friendly
Grammatical Framework (GF)

- Functional programming language for multilingual grammars
- Language-independent semantic interlingua
- Generation and parsing from a single grammar

http://www.grammaticalframework.org
Grammatical Framework (GF)

- Same grammar for both directions
- Only one grammar per language (no pairs)
Live demo: AnaCon

- Workflow
  - Input
  - Result
  - Intermediate files
Limitations (1)

- **Interface**
  - Low-level (terminal)
  - No help in composing

- **Formalism**
  - No separation of agent
  - No subject / object
  - No concept of time (only sequence of actions)
Limitations (2)

- Conflict detection
  - Counter-examples are not minimal
  - Analysis is a hard problem
- Querying?
Contract-Oriented diagrams

- Visual, modular
- Naming of clauses
- Separation of agent
- Real timing restrictions
- Guard conditions
- Cross-references
Running example
Intro to automata

- States/nodes
- Edges/transitions

- Examples
  - 0110 ✓
  - 01100 ❌
  - Even number of 0’s
Conversion to NTA

Translation function from diagrams into networks of timed automata
6 automata!
Model checking

What can we do with automata?

- Simulation
- Model checking
  - Test properties in temporal logic
    - “it is never the case that x and y”
      - $A \square ! (x \land y)$
- Querying by model checking
UPPAAL Demo

- Simulation
- Property checking
Lost in the details

- This gets pretty low level!
- We need to work at higher abstraction levels
- Separate tasks
  - Building models
  - Generating properties
  - Interpreting their result
One model, different views

Original contract

Controlled Natural Language

Diagrams

Model as XML

Spreadsheet

Analysis tools
Visual diagram editor

- Point and click diagram editor
- Web-based
- Automatic validation
- Export
(Another) CNL

- Similar to CNL in AnaCon
- More kinds of expressions
- Includes large-scale dictionary
- Focus on tools
### Starting place for auto-generated partial output

#### Spreadsheet view

<table>
<thead>
<tr>
<th>Original</th>
<th>Number</th>
<th>Credition</th>
<th>Time</th>
<th>Majority</th>
<th>Subject</th>
<th>Year</th>
<th>Object</th>
<th>Children</th>
<th>Reopenations</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40</td>
<td></td>
<td>D</td>
<td>syllabus</td>
<td>specific</td>
<td>content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>41</td>
<td></td>
<td>D</td>
<td>syllabus</td>
<td>specific</td>
<td>content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>42</td>
<td></td>
<td>D</td>
<td>syllabus</td>
<td>specific</td>
<td>content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>43</td>
<td></td>
<td>D</td>
<td>syllabus</td>
<td>specific</td>
<td>content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>44</td>
<td></td>
<td>D</td>
<td>syllabus</td>
<td>specific</td>
<td>content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>45</td>
<td></td>
<td>D</td>
<td>syllabus</td>
<td>specific</td>
<td>content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>46</td>
<td></td>
<td>D</td>
<td>syllabus</td>
<td>specific</td>
<td>content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>47</td>
<td></td>
<td>D</td>
<td>syllabus</td>
<td>specific</td>
<td>content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- Any changes saved in Drive.
- All changes saved in Drive.
Demo

- CNL editor
  - Example of CNL in text editor
  - Load in CNL editor
    - highlighting of labels, folding
    - snippets, autocompletion
    - validation

- Export to visual editor
The dream (revisited)
Problems: NLP

Natural language processing is hard

- CNL aims to bridge the gap
- Requires more time to sound less mechanical
- More work on editing tools
Problems: NLP (another solution)

- Machine Learning techniques for generating partial model
  - identification of modality
  - entity recognition for agents
- Use of spreadsheet format
Problems: formalism

Finding suitable formalism

- Expressivity, features, generality
- Validating translation to target formalism
- Formal semantics
Problems: query language

Properties and counter-examples

- We want queries and answers in NL
  a. NL query $\rightarrow$ property in temp. logic
  b. Manipulation of properties to find answer
  c. “answer” $\rightarrow$ NL response
• All of this is done in a closed environment
• Realising “the dream” requires collaboration with stakeholders outside our field
• That’s really why we’re here!

Hope you enjoyed listening, very happy to hear your comments and suggestions.
Thanks
AnaCon

CL

CLAN

C-O Diagrams