SymboleoPC: A Model-checking Tool for Legal Contract Verification

CSER21

The Contract Specification and Monitoring Lab (CSM Lab) | Presented by: Alireza Parvizimosaed
Contract Specification and Monitoring Lab @ uOttawa

John Mylopoulos
Professor
Computer Science, uOttawa

Daniel Amyot
Professor
Computer Science, uOttawa

Luigi Logrippo
Professor
Computer Science, UQO

Marco Roveri
Aggregate Professor
Department of Information Engineering and
Computer Science, University of Trento

Alireza P. Mosaed
PhD Candidate
Computer Science, Uottawa

https://sites.google.com/uottawa.ca/csmlab
Introduction: Legal Contract

What is a contract?

Obligation is an action that a party is obliged to do
E.g., a restaurant should deliver Pizza within 30 minutes

Power is the right of a party to create, change, suspend or extinguish legal positions.
E.g., a customer may cancel a Pizza order when delivery takes longer than 30 minutes.
Introduction: Legal Contract

What is a contract?
What is the problem?

Contract terms may be ambiguous, incomplete, conflicting and possibly invalid
Introduction: Legal Contract

What is a contract?
What is the problem?
What is the solution?

Symboleo language formalizes the syntax and semantics of contracts.

A model checking tool (SymboleoPC) checks all possible ways to satisfy a desired/undesired property.
Symboleo’s Semantics

- Semantics of contracts, obligations, powers, events and parties lifecycles are specified in terms of statecharts.
- 30 axioms formally specifies explicit and implicit semantics of transitions.
SymboleoPC: Encoding

- Encode Symboleo’s concepts in nuXmv: contract, obligation, power, event, party, etc.

```
MODULE Obligation(surviving, cnt_in_effect, cnt_untermination, fulfilled, triggered, violated, activated, expired, power_suspended, cnt_suspended, terminated, power_resumed, cnt_resumed, discharged, antecedent)

//removed some parts

ASSIGN
init(state) := not_created;
next(state) := case
  cnt_in_effect & state=not_created & triggered & !antecedent: create;
  cnt_in_effect & state=not_created & triggered & antecedent: inEffect;
  cnt_in_effect & state=create & antecedent: inEffect;
  cnt_in_effect & state=create & (expired | discharged): discharge;
  cnt_in_effect & state=inEffect & fulfilled: fulfillment;
  cnt_in_effect & state=inEffect & _suspended: suspension;
  cnt_in_effect & state=inEffect & violated: violation;
  cnt_in_effect & _active & terminated: unsTermination;
  cnt_untermination & !surviving & _active: unsTermination;
  sus_state=sus_by_contract & state=suspension & cnt_resumed: inEffect;
  sus_state=sus_by_power & state=suspension & power_resumed: inEffect;
TRUE : state;
esac;
```
SymboleoPC: Property

- SymboleoPC: Check desirable and undesirable properties
  - **Termination:**

<table>
<thead>
<tr>
<th>Number</th>
<th>Type</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>desirable-liveness</td>
<td>existence</td>
</tr>
</tbody>
</table>

**Description:** MeatSale contract eventually terminates.

**Property**

LTL $\text{LTL1} := F(sales\_cnt.contract\_state = s\text{Termination} \mid sales\_cnt.contract\_state = uns\text{Termination})$

**Result:** failed

**Explanation:** If payment is violated and seller suspends delivery by power while late payment is expired, then payment cannot be resumed. Thereafter, payment is always suspended and then the contract stays active.

- **Limited liability**

<table>
<thead>
<tr>
<th>Number</th>
<th>Type</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>undesirable-safety</td>
<td>absence</td>
</tr>
</tbody>
</table>

**Description:** In case of late payment, buyer cannot be penalized more than once.

**Property**

LTL $\text{LTL2} := G(sales\_cnt\_paid\_Late\_happened \& sales\_cnt\_paid\_Late\_performer = sales\_cnt\_Oipay\_debtor\_name \& sales\_cnt\_Oipay\_debtor\_is\_performer -> G !(sales\_cnt\_paid\_Late\_inactive))$

**Result:** succeed
SymboleoPC: Property

**Conformity to parties’ intentions**

<table>
<thead>
<tr>
<th>Number</th>
<th>Type</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>desirable-safety</td>
<td>precedence</td>
</tr>
</tbody>
</table>

**Description**

Delivery of goods always happens after payment.

**Property**


**Result: failed**

Explanation: The delivery obligation is independent of the payment obligation.

**Usefulness**

<table>
<thead>
<tr>
<th>Number</th>
<th>Type</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>desirable-safety</td>
<td>occurrence</td>
</tr>
</tbody>
</table>

**Description**

MeatSale is free of useless obligations or powers: all obligations and powers can be activated.

**Property**

CTLSPEC NAME CTL4_1 := EF(sales_cnt.PsusDel._active)
Scalability Analysis

- SymboleoPC supports up to 40 obligations and powers
- Property checking takes less than 3 seconds
Conclusion

• Symboleo is a specification language that streamlines legal contract analysis.
• SymboleoPC is an scalable property checker that verifies desirable and undesirable properties

• Future work
  – Autonomously translate Symboleo to SymboleoPC
  – Autonomously translate Symboleo to smart contract

Contact: aparv007@uottawa.ca
Published Papers

28th IEEE Requirements Engineering Conference (RE’20)

Symboleo: Towards a Specification Language for Legal Contracts

Sepehr Sharifi, Alireza Parvizimosaed, Daniel Amyot, Luigi Logrippo, John Mylopoulos
School of EECS, University of Ottawa, Ottawa, Canada
{sshari90, aparv007, damyot, logrippo, jmylopoou}@uottawa.ca

39th International Conference on Conceptual Modelling (ER’20)

Subcontracting, Assignment, and Substitution for Legal Contracts in Symboleo *

Alireza Parvizimosaed¹, Sepehr Sharifi¹, Daniel Amyot¹, Luigi Logrippo¹,², and John Mylopoulos¹

¹ School of EECS, University of Ottawa, Ottawa, Canada
{aparv007, sshari90, damyot, logrippo, jmylopoou}@uottawa.ca
² Université du Québec en Outaouais, Gatineau, Canada
Thank you for your attention

Question?