

# 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



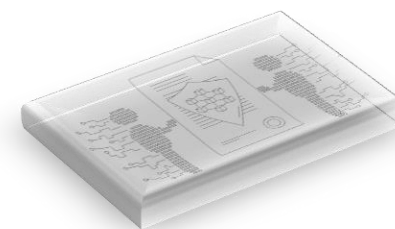
Marco Roveri

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



Alireza P. Mosaed

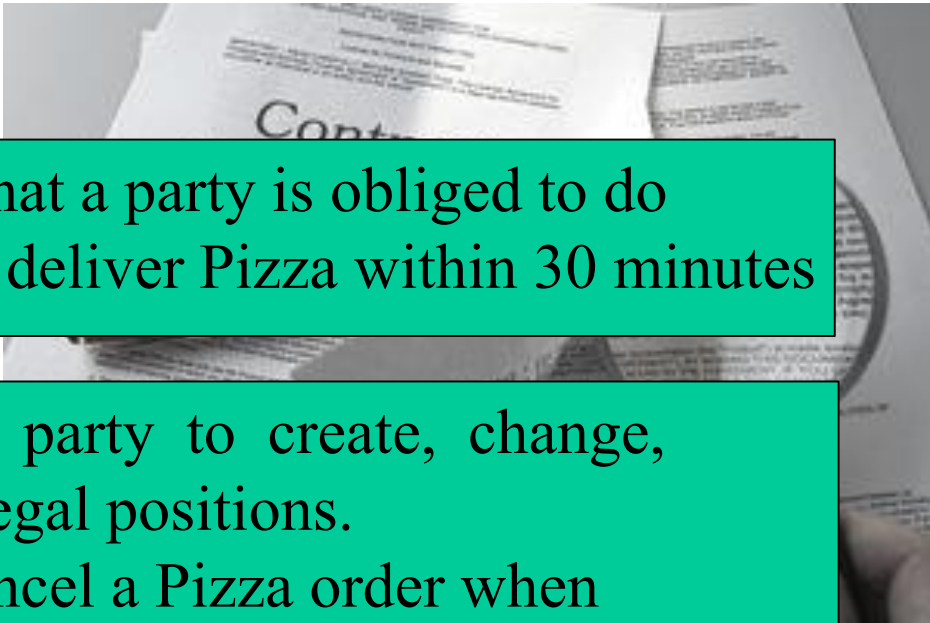
Ph.D. Candidate  
Computer Science, uOttawa



<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

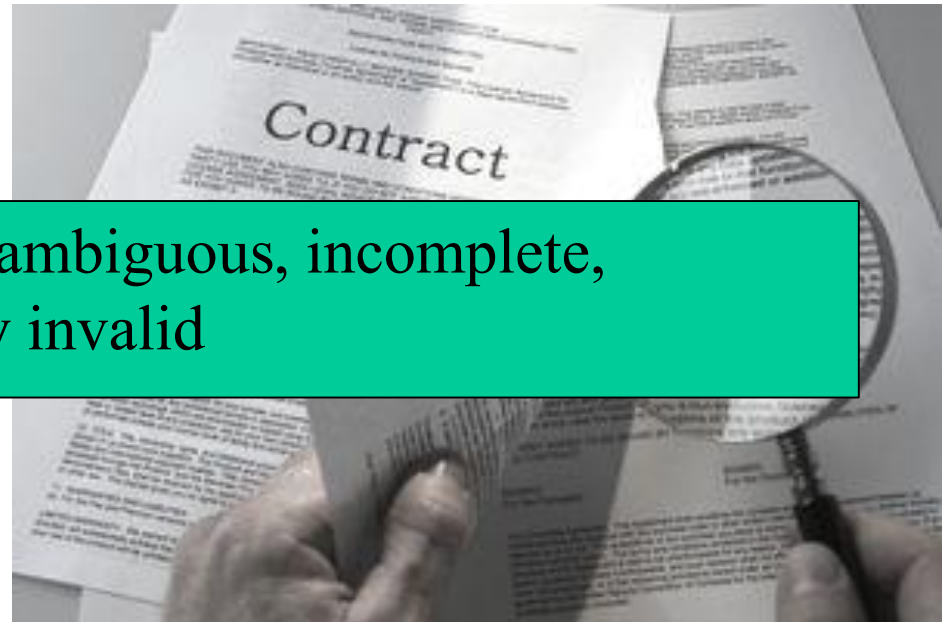
Power is the right of a party to create, change, suspend or extinguish legal positions.  
E.g., a customer may cancel a Pizza order when delivery takes longer than 30 minutes.

# Introduction: Legal Contract

What is a contract?

What is the problem?

Contract terms may be ambiguous, incomplete, conflicting and possibly invalid



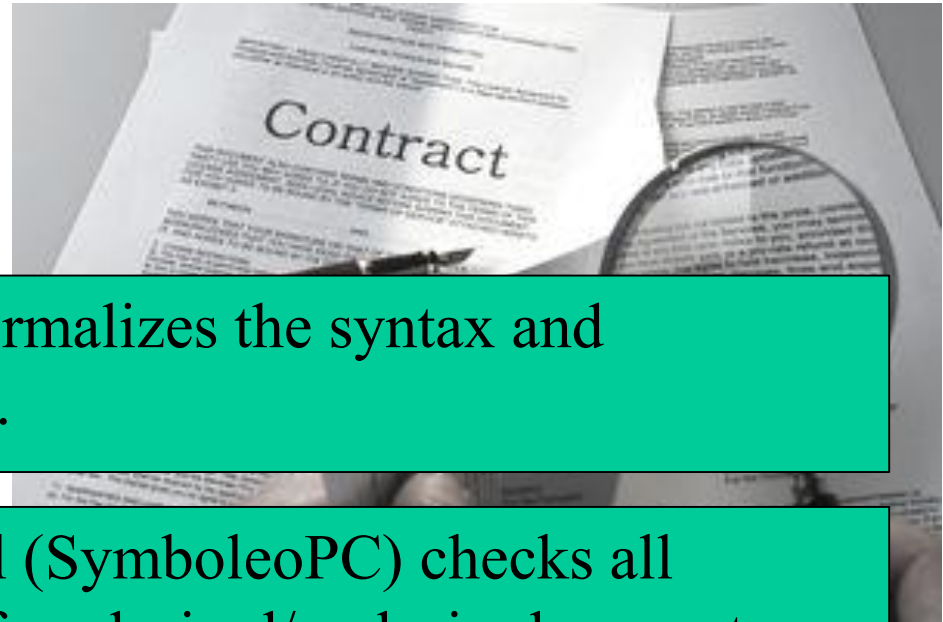


# Introduction: Legal Contract

What is a contract?

What is the problem?

What is the solution?

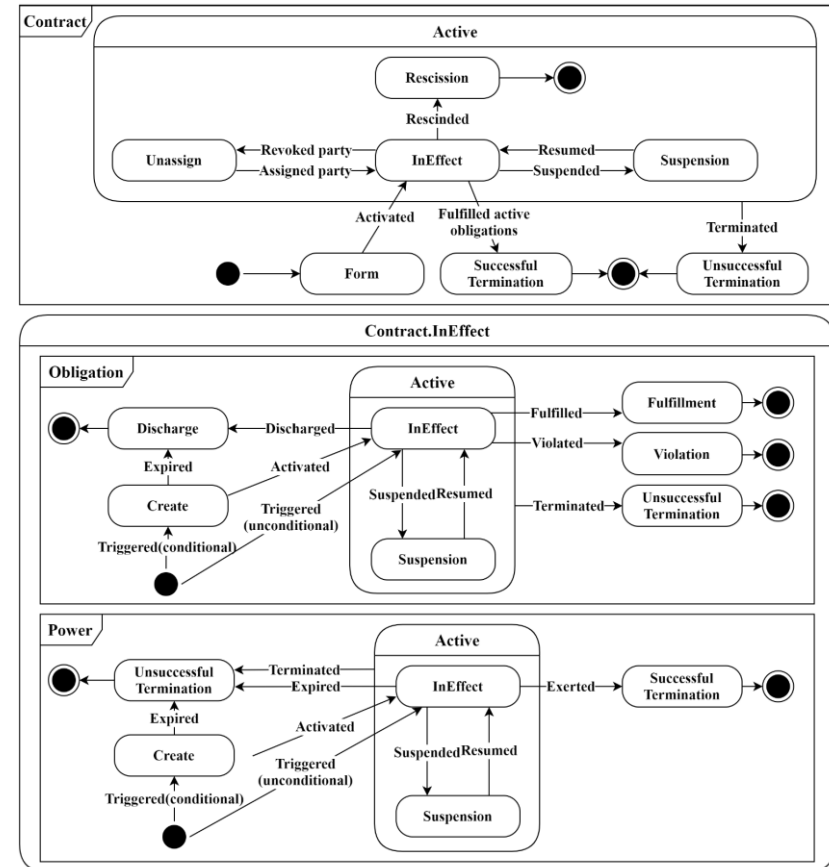


Symboleo language formalizes the syntax and semantics of contracts.

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

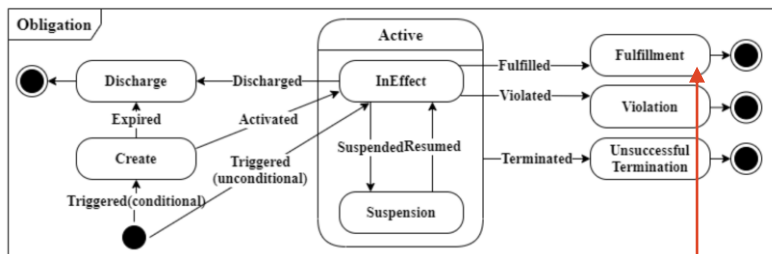
# Symboleo's Semantics

- 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\_termination, 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_termination & !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:**

Number	Type	Pattern
1	desirable-liveness	existence
<b>Description</b>		
MeatSale contract eventually terminates.		
<b>Property</b>		
<b>LTLSPEC NAME</b> LTL1 := $F(\text{sales\_cnt.contract.state} = \text{sTermination} \mid \text{sales\_cnt.contract.state} = \text{unsTermination})$		
<b>Result:failed</b>		
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**

Number	Type	Pattern
2	undesirable-safety	absence
<b>Description</b>		
In case of late payment, buyer cannot be penalized more than once.		
<b>Property</b>		
<b>LTLSPEC NAME</b> LTL2 := $G(\text{sales\_cnt.paidLate\_happened} \ \& \ \text{sales\_cnt.paidLate.performer} = \text{sales\_cnt.Olpay\_debtor\_name} \ \& \ \text{sales\_cnt.Olpay\_debtor\_is\_performer} \rightarrow G!(\text{sales\_cnt.paidLate\_inactive}))$		
<b>Result:succeed</b>		



# SymboleoPC: Property

- Conformity to parties' intentions

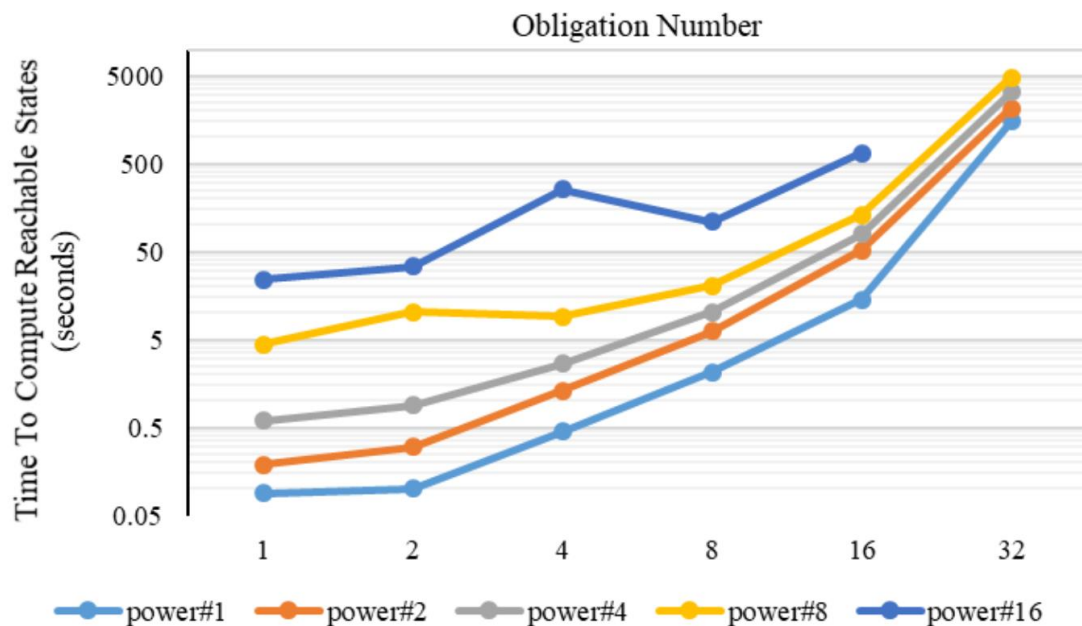
Number	Type	Pattern
3	desirable-safety	precedence
<b>Description</b>		
Delivery of goods always happens after payment.		
<b>Property</b>		
<b>LTLSPEC NAME</b> LTL3 := !(sales_cnt.delivered._happened & sales_cnt.delivered.performer = sales_cnt.Odel_debtor._name & sales_cnt.Odel_debtor._is_performer) <b>U</b> (sales_cnt.paid._happened & sales_cnt.paid.performer = sales_cnt.Opay_debtor._name & sales_cnt.Odel_debtor._is_performer)		
<b>Result:failed</b>		
Explanation: The delivery obligation is independent of the payment obligation.		

- Usefulness

Number	Type	Pattern
4	desirable-safety	occurrence
<b>Description</b>		
MeatSale is free of useless obligations or powers: all obligations and powers can be activated.		
<b>Property</b>		
<b>CTLSPEC NAME</b> CTL4_1 := <b>EF</b> (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](mailto: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*  
{sshari190, aparv007, damyot, logrippo, jmylopou}@uottawa.ca

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

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

Alireza Parvizimosaed<sup>1</sup>, Sepehr Sharifi<sup>1</sup>, Daniel Amyot<sup>1</sup>, Luigi Logrippo<sup>1,2</sup>,  
and John Mylopoulos<sup>1</sup>

<sup>1</sup> School of EECS, University of Ottawa, Ottawa, Canada  
{aparv007, sshari190, damyot, logrippo, jmylopou}@uottawa.ca

<sup>2</sup> Université du Québec en Outaouais, Gatineau, Canada

**Thank you for your attention  
Question?**