

# Towards a Formal Semantics for ODRL

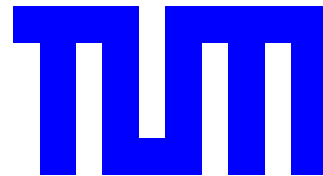
**Stefan Katzenbeisser**

**Markus Holzer**

**Christian Schallhart**

Institut für Informatik

Technische Universität München



# Why semantics?

- ▶ only a formal semantics provides a precise way to express the “meaning” of ODRL expressions
- ▶ important properties can be checked automatically
- ▶ it may enhance the readability of an expression

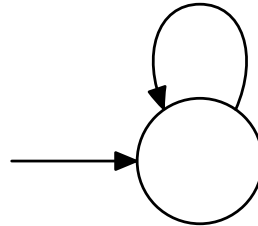
# A semantics based on automata

- ▶ modeled as automaton
- ▶ edge represents action
- ▶ states represent “state of the license”
- ▶ states are labeled with user names
- ▶ actions are atomic (constant time)
- ▶ time elapses with each action or  $\tau$
- ▶ each infinite trace through automaton describes sequence of actions that are allowed by the license

# A simple example

```
<permission>  
  <display/>  
  <print/>  
</permission>
```

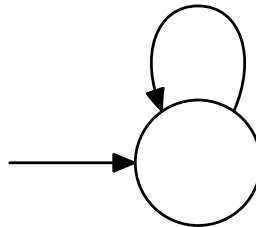
$\tau$ , print, display



# Requirements

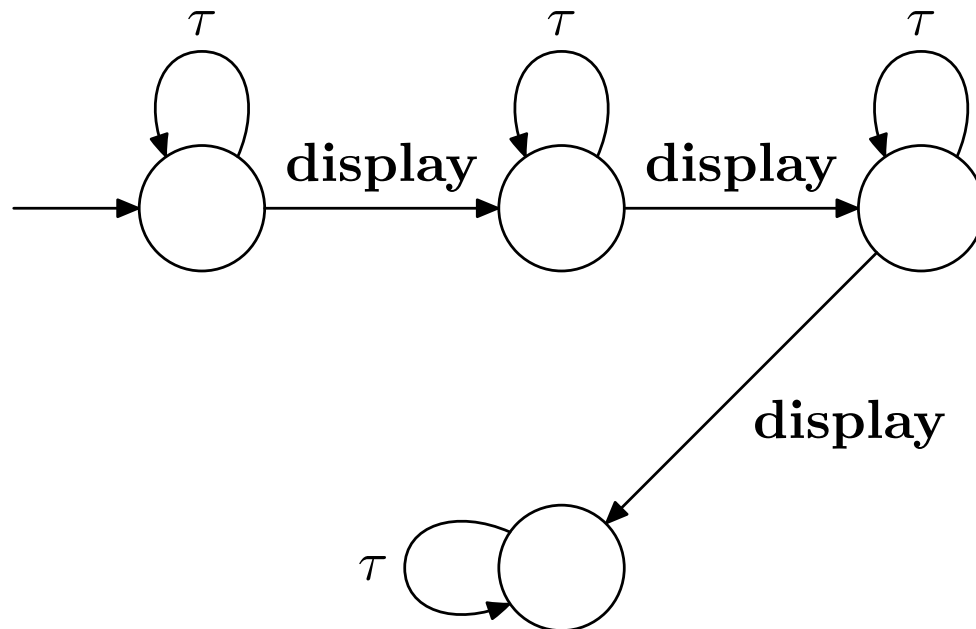
- ▶ ... are associated with edges
- ▶ edge can only be taken if requirement is fulfilled

$\tau$ , print, display [payment]



# Constraints

```
<display>  
  <constraint>  
    <count>3</count>  
  </constraint>  
</display>
```



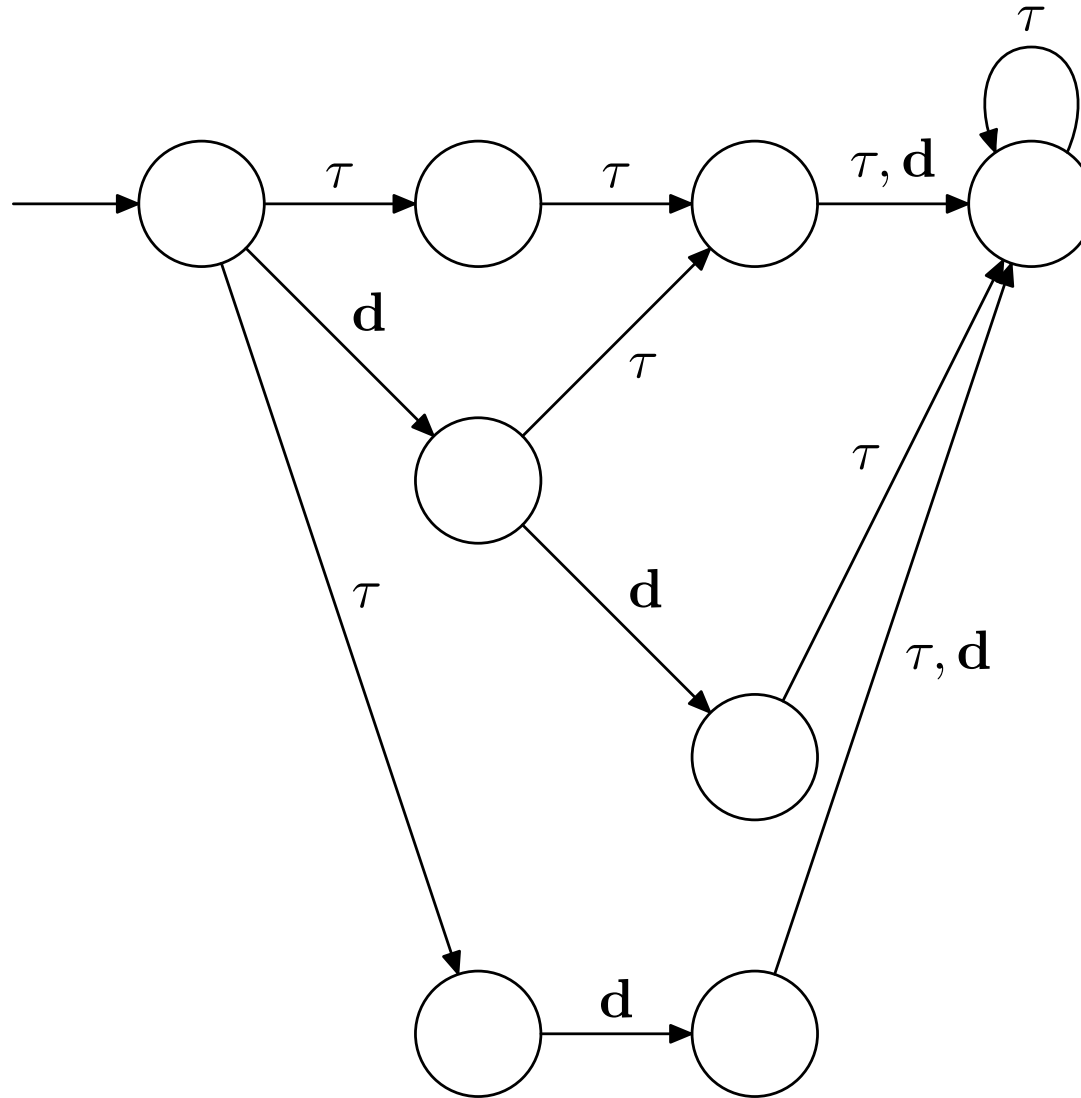
# Time constraints (1)

- ▶ ... states implicitly encode elapsed time
- ▶ it may be necessary to use vast number of states

## Example:

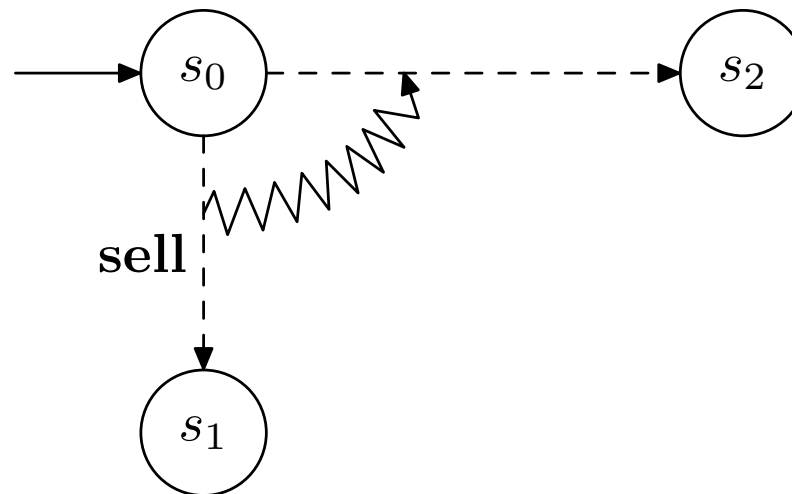
```
<display>  
  <constraint>  
    <interval>2-5</interval>  
    <constraint>  
      <count>2</count>  
    </constraint>  
  </constraint>  
</display>
```

# Time constraints (2)



# Selling objects (1)

- ▶ ... is modeled by a triggered transition
- ▶ dashed transition introduces new user



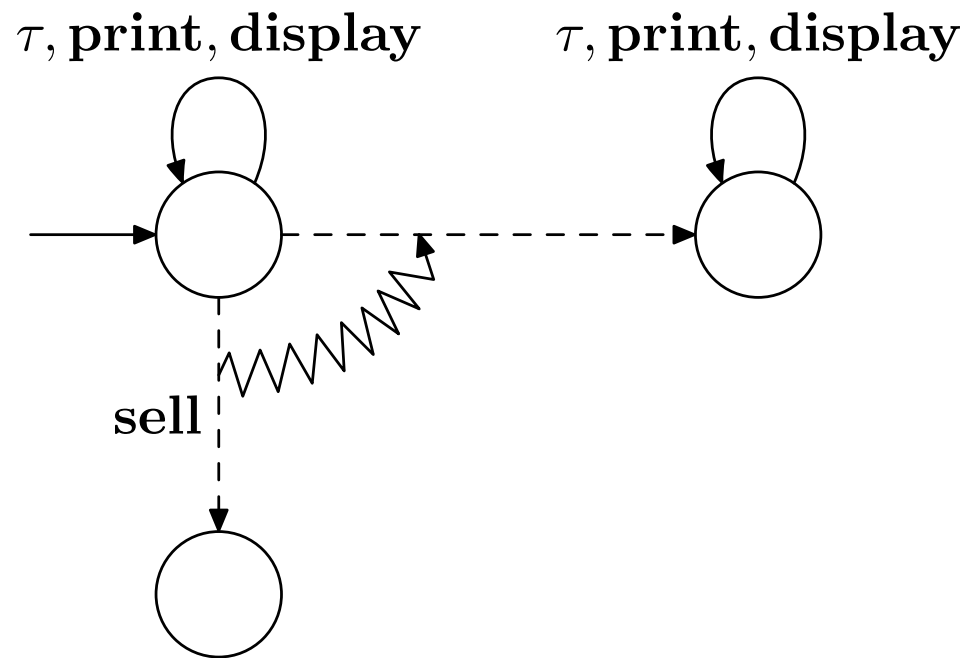
# Selling objects (2)

## Example:

```
<permission>
  <display>
  <print>
  <sell>
    <constraint>
      <transferPerm>
        <display>
        <print>
      </transferPerm>
    </constraint>
  </sell>
</permission>
```

# Selling objects (3)

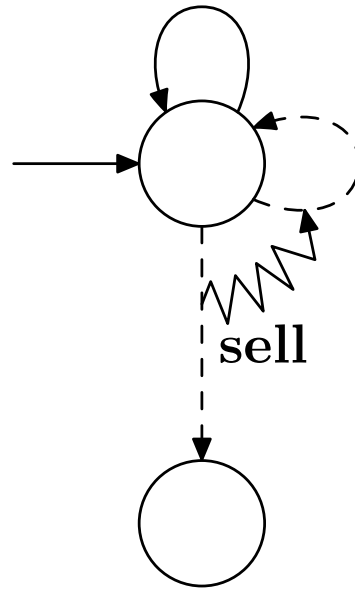
**Example, continued:**



# Selling objects (4)

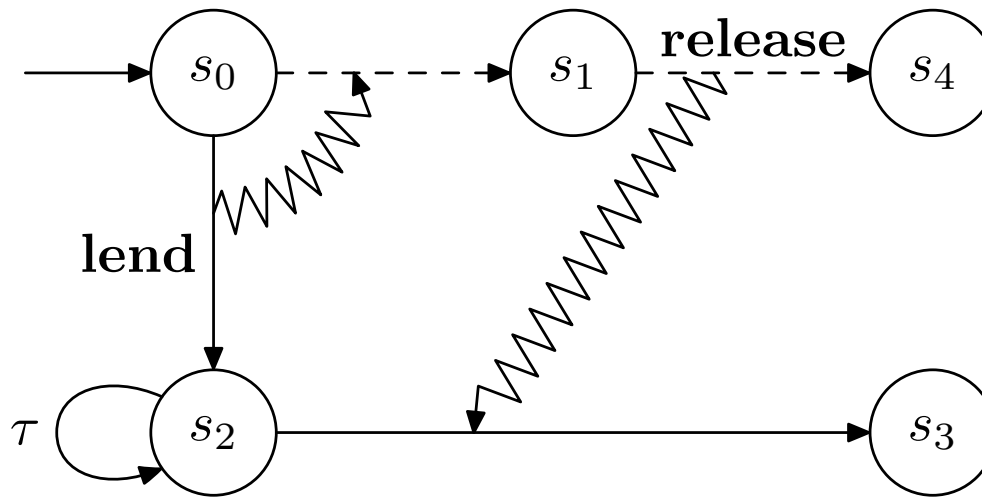
- ▶ infinite chains of sell actions possible

$\tau$ , print, display



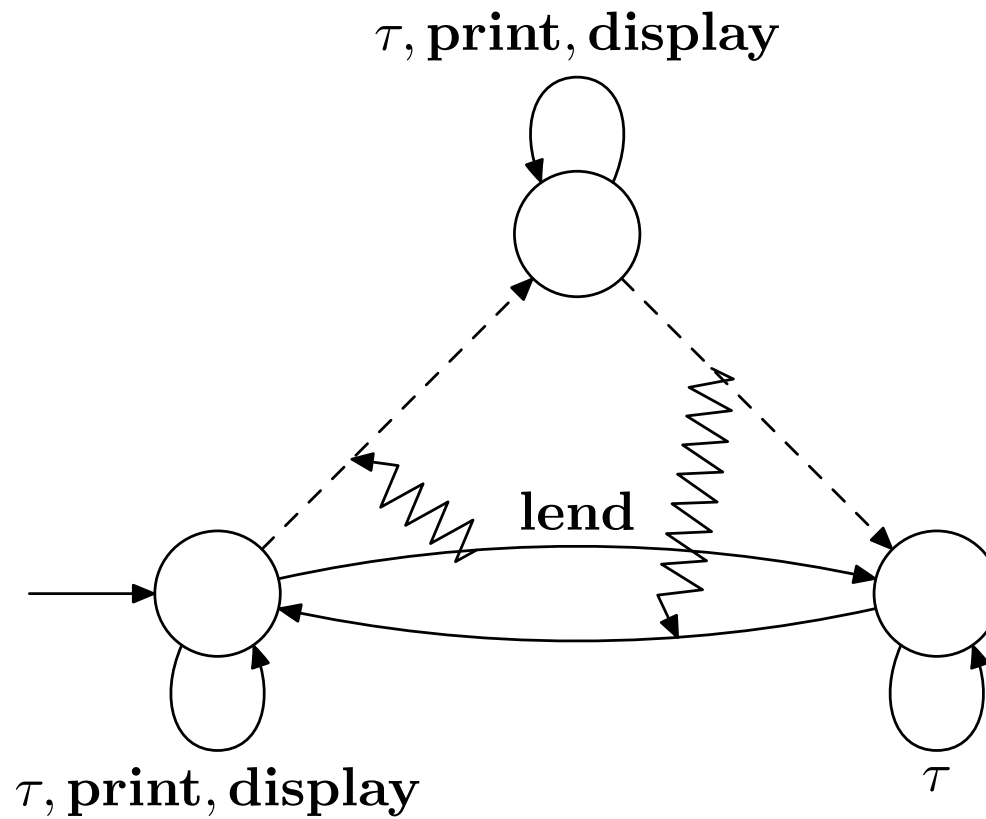
# Lending objects (1)

- ▶ user enters waiting state, triggers transition
- ▶ release action “wakes up” lender



# Lending objects (2)

Example:



# Conclusions

- ▶ automata can be used to formalize the semantics of rights expression languages
- ▶ semantics intuitively capture permissions
- ▶ selling of objects can be expressed by triggered transitions
- ▶ lending objects is the most complex operation
- ▶ infinite chains of lend operations need more expressive formalism