

Defeasibility and Deontic

OASIS LegalRuleML Technical Committee
Guido Governatori

RuleML 2013, Seattle, 13 July 2013

Part I

Defeasibility

Why/What Defeasibility

body1 \Rightarrow *head*

body2 \Rightarrow \neg *head*

Why/What Defeasibility

body1 \Rightarrow *head*

body1, exception_conditions $\Rightarrow \neg$ *head*

Why/What Defeasibility

body1 \Rightarrow *head*

exception_conditions $\Rightarrow \neg$ *head*

What is a rule?

What is a rule?

A rule is a binary relation

$$\textit{body} \times \textit{head}$$

What is a rule?

A rule is a binary relation

$$\textit{body} \times \textit{head}$$

What are the relationships between the *body* and the *head*?

Strength of a rule

Head/body relationship	Notation	Strength
body always head	$body \rightarrow head$	strict
body sometimes head	$body \Rightarrow head$	defeasible
body not complement head	$body \rightsquigarrow head$	defeater
body no relationship head		
body always complement head	$body \rightarrow \neg head$	strict
body sometimes complement head	$body \Rightarrow \neg head$	defeasible
body not head	$body \rightsquigarrow \neg head$	defeater

Defeasibility and strength of rules in LegalRuleML

Option 1, Context block:

```
<lrml:Context key="ruleInfo2">  
  <lrml:appliesStrength  
    iri="&defeasible-ontology;#defeasible2"/>  
  <lrml:toStatement keyref="#cs1"/>  
</lrml:Context>
```

Defeasibility and strength of rules in LegalRuleML

Option 2, directly inside a rule:

```
<ruleml:Rule key=":key1">  
  <lrml:hasStrength>  
    <lrml:Defeater key="str4"/>  
  </lrml:hasStrength>  
  ...  
</ruleml:Rule>
```

Example

Complaint

means an expression of dissatisfaction made to a Supplier in relation to its Telecommunications Products or the complaints handling process itself, where a response or Resolution is explicitly or implicitly expected by the Consumer.

An initial call to a provider to request a service or information or to request support is not necessarily a Complaint. An initial call to report a fault or service difficulty is not a Complaint. However, if a Customer advises that they want this initial call treated as a Complaint, the Supplier will also treat this initial call as a Complaint.

If a Supplier is uncertain, a Supplier must ask a Customer if they wish to make a Complaint and must rely on the Customers response.

Example

$r1 : \text{expressionDisatisfaction} \Rightarrow \text{complaint}$

Example

$r1 : \text{expressionDisatisfaction} \Rightarrow \text{complaint}$

$r2 : \text{initialCall} \Rightarrow \neg \text{complaint}$

Example

$r1 : \text{expressionDisatisfaction} \Rightarrow \text{complaint}$

$r2 : \text{initialCall} \Rightarrow \neg \text{complaint}$

$r3 : \text{initialCall}, \text{adviseCompliant} \Rightarrow \text{complaint}$

Example

```
<ruleml:Rule key=":rule1">
  <lrml:hasStrength>
    <lrml:Defeasible key=":str1"/>
  </lrml:hasStrength>
  <ruleml:if>
    <ruleml:Atom>
      <ruleml:Var>X</ruleml:Var>
      <ruleml:Rel>is an expression of dissatisfaction ...</ruleml:Rel>
    </ruleml:Atom>
  </ruleml:if>
  <ruleml:then>
    <ruleml:Atom>
      <ruleml:Var>X</ruleml:Var>
      <ruleml:Rel>is a complaint</ruleml:Rel>
    </ruleml:Atom>
  </ruleml:then>
</Rule>
```


Part II

Deontic

Prescriptive vs Constitutive Rules

Section 29 of the Australian “National Consumer Credit Protection Act 2009” (Act No. 134 of 2009).

(1) A person must not engage in a credit activity if the person does not hold a licence authorising the person to engage in the credit activity.

Constitutive and Prescriptive Rules

- ▶ A constitutive rule defines a term
- ▶ A prescriptive rule asserts a deontic effect (e.g., obligation, permission, prohibition)

Prescriptive Rules

$body \Rightarrow [D_1]formula_1, [D_2]formula_2, \dots, [D_n]formula_n$

Prescriptive Rules

$$body \Rightarrow \underbrace{[D_1]formula_1, [D_2]formula_2, \dots, [D_n]formula_n}_{\text{suborder list}}$$

Prescriptive Rules in LegalRuleML

```
<lrml:PrescriptiveStatement key="ps1">
  <ruleml:Rule key=":key1">
    <lrml:hasStrength>
      strength of the rule
    </lrml:hasStrength>
    <ruleml:if>
      set of deontic formulas and formulas
    </ruleml:if>
    <ruleml:then>
      <lrml:SuborderList>
        list of deontic formulas
      </lrml:SuborderList>
    </ruleml:then>
  </ruleml:Rule>
</lrml:PrescriptiveStatement>
```

Example

Section 29 of the Australian “National Consumer Credit Protection Act 2009” (Act No. 134 of 2009).

(1) A person must not engage in a credit activity if the person does not hold a licence authorising the person to engage in the credit activity.

Civil penalty: 2,000 penalty units.

[...]

Criminal penalty: 200 penalty units, or 2 years imprisonment, or both.

Penalty and Reparation

A penalty is just a subordr list

$$[D_1]formula_1, [D_2]formula_2, \dots, [D_n]formula_n$$

Penalty and Reparation

A penalty is just a subordr list

$$[D_1]formula_1, [D_2]formula_2, \dots, [D_n]formula_n$$

A reparation connect a norm (rule) with the penalties compensating the violation of the norm

Penalty and Reparation in LegalRuleML

```
<lrml:PenaltyStatement key="pen2">  
  <lrml:SuborderList>  
    list of deontic formulas  
  </lrml:SuborderList>  
</lrml:PenaltyStatement>
```

Penalty and Reparation in LegalRuleML

```
<lrml:PenaltyStatement key="pen2">  
  <lrml:SuborderList>  
    list of deontic formulas  
  </lrml:SuborderList>  
</lrml:PenaltyStatement>
```

```
<lrml:ReparationStatement key="rep1">  
  <lrml:hasTemplate>  
    <lrml:Reparation key="assoc1">  
      <lrml:appliesPenalty keyref="#pen1"/>  
      <lrml:toTarget keyref="#ps1"/>  
    </lrml:Reparation>  
  </lrml:hasTemplate>  
</lrml:ReparationStetement>
```

Example

Section 29 of the Australian “National Consumer Credit Protection Act 2009” (Act No. 134 of 2009).

(1) A person must not engage in a credit activity if the person does not hold a licence authorising the person to engage in the credit activity.

Civil penalty: 2,000 penalty units.

[...]

Criminal penalty: 200 penalty units, or 2 years imprisonment, or both.

Example

$ps1: Person(x) \Rightarrow [FORB]EngageCreditActivity(x)$

$ps2: HasLicence(x) \Rightarrow [PERM]EngageCreditActivity(x)$

$ps2 > ps1$

$pen1: [OBL]PayCivilUnits(x, 2000)$

$pen2: [OBL]PayPenalUnits(x, 200),$

$[OBL]Imprisonment(x, 2y),$

$[OBL]PayPenaltyUnitsAndImprisonment(x, 200, 2y)$

$rep1: [Violation]ps1, pen1$

$rep2: [Violation]ps1, pen2$

Thank You

Questions?