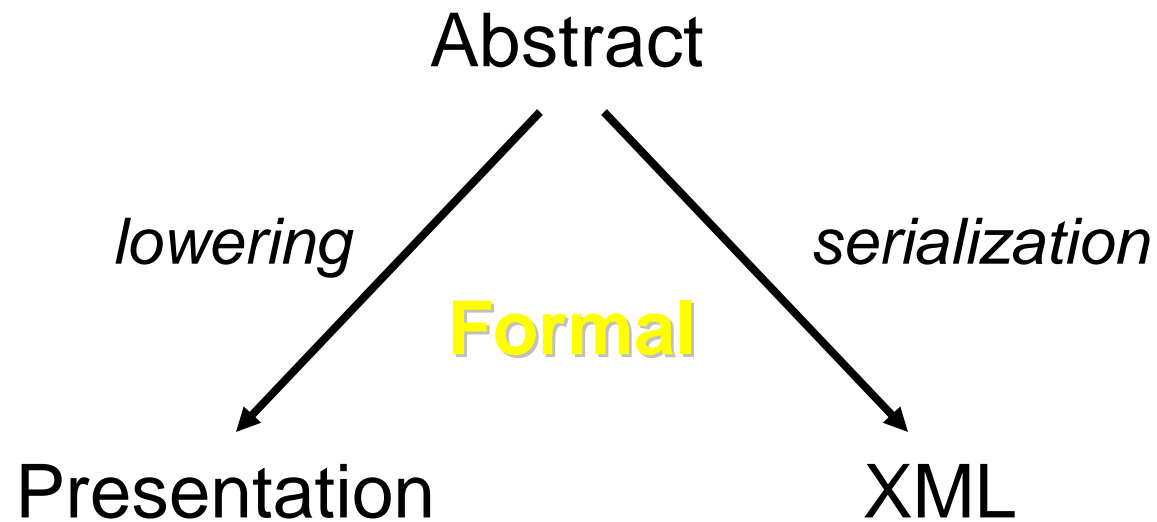# RIF BLD Syntaxes:
Presentation*, Abstract, XML
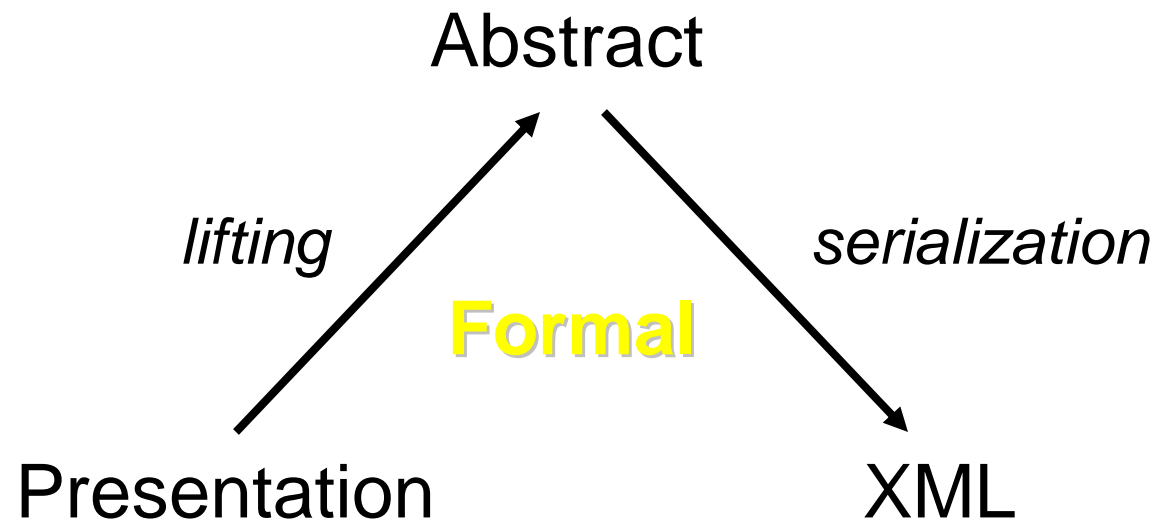
F2F7, Hawthorne, 27/28 Sept 2007

* Directly corresponds to Formal Syntax

Syntaxes: Top-Down Order

Abstract

lowering   *serialization*

Formal

Presentation   XML

Syntaxes: Didactic Order

Abstract

lifting   *serialization*

Formal

Presentation   XML

Presentation Syntax: Given Ruleset

```
{

    Forall

      ?x

      (



        discount(?x 15)



          :-



        premium(?x)



      )
```

```
  Forall

    (

      premium(John)

    )



  Forall

    (

      premium(Mary)

    )
}
```

Abstract Syntax': Given Ruleset

Ruleset(

  formula ->

    Forall(

      declare -> Var(**x**)

      formula ->

       Implies(

       then ->
        Uniterm(
         op -> Const(discount)
         arg -> Var(**x**)
         arg -> Const(15))

       if ->
        Uniterm(
         op -> Const(premium)
         arg -> Var(**x**))

)
)

formula ->

  Forall(

    formula ->

      Uniterm(
        op -> Const(premium)
        arg -> Const(**John**))
)

formula ->

  Forall(

    formula ->

      Uniterm(
        op -> Const(premium)
        arg -> Const(**Mary**))
)
)

Abstract Syntax: Given Ruleset

Ruleset(

  formula ->

    Forall(

      declare -> Var(**x**)

      formula ->

       Implies(

        if ->
         Uniterm(
          op -> Const(premium)
          arg -> Var(**x**))

        then ->
         Uniterm(
          op -> Const(discount)
          arg -> Var(**x**)
          arg -> Const(15))

)
)

formula ->

  Forall(

    formula ->

     Uniterm(
        op -> Const(premium)
        arg -> Const(**John**))
)

formula ->

  Forall(

    formula ->

     Uniterm(
        op -> Const(premium)
        arg -> Const(**Mary**))
)
)

XML Syntax: Given Ruleset

```xml
<Ruleset>

 <formula>
  <Forall>
   <declare>
    <Var>x</Var>
   </declare>
   <formula>
    <Implies>
     <if>
      <Uniterm>
       <op><Const>premium</Const></op>
       <arg><Var>x</Var></arg>
      </Uniterm>
     </if>
     <then>
      <Uniterm>
       <op><Const>discount</Const></op>
       <arg><Var>x</Var></arg>
       <arg><Const>15</Const></arg>
      </Uniterm>
     </then>
    </Implies>
   </formula>
  </Forall>
 </formula>

 <formula>
  <Forall>
   <formula>
    <Uniterm>
     <op><Const>premium</Const></op>
     <arg><Const>John</Const></arg>
    </Uniterm>
   </formula>
  </Forall>
 </formula>

 <formula>
  <Forall>
   <formula>
    <Uniterm>
     <op><Const>premium</Const></op>
     <arg><Const>Mary</Const></arg>
    </Uniterm>
   </formula>
  </Forall>
 </formula>

</Ruleset>
```

Presentation Syntax: Derived Ruleset

{

Forall

(

discount(**John** 15)

)

Forall

(

discount(**Mary** 15)

)

}

Abstract Syntax: Derived Ruleset

Ruleset(

  formula ->

    Forall(

      formula ->

        Uniterm(
            op -> Const(discount)
            arg -> Const(**John**)
            arg -> Const(15))
)

  formula ->

    Forall(

      formula ->

        Uniterm(
            op -> Const(discount)
            arg -> Const(**Mary**)
            arg -> Const(15))
)

)

XML Syntax: Derived Ruleset

```
<Ruleset>

 <formula>
  <Forall>
   <formula>
    <Uniterm>
     <op><Const>discount</Const></op>
     <arg><Const>John</Const></arg>
     <arg><Const>15</Const></arg>
    </Uniterm>
   </formula>
  </Forall>
 </formula>

 <formula>
  <Forall>
   <formula>
    <Uniterm>
     <op><Const>discount</Const></op>
     <arg><Const>Mary</Const></arg>
     <arg><Const>15</Const></arg>
    </Uniterm>
   </formula>
  </Forall>
 </formula>

</Ruleset>
```