Semantic Web
and
Natural Semantics

Adrian Pop

IDA/PELAB

adrpo@ida.liu.se

Outline

- Semantic Web
- Natural Semantics
- Relational Meta-Language (RML)
- Semantic Web Layers
- Tools
  - Ontology Checkers
  - Checking Semantics of Websites
- Conclusions

Semantic Web

- The information in the current web:
  - has meaning for human only
  - is not machine processable
- Semantic Web brings:
  - semi-structured information
  - means to add structure (semantics/constrains) on data, with the use of:
  - languages: XML, XMLSchema, RDF, RDFS, DAML+OIL, OWL

Natural Semantics

- Based on
  - Gordon Plotkin's Structural Operational Semantics (SOS)
  - Gentzen's Sequent Calculus for Natural Deduction.
- "Natural Semantics" (NS)
  - term by Gilles Kahn
  - formalism for specifications of:
    - type systems
    - programming languages

Natural Semantics - Syntax

$$\frac{H_1 \vdash T_1 : R_1 \quad \ldots \quad H_n \vdash T_n : R_n}{H \vdash T : R} \quad \textbf{if} \; \textbf{<cond>}$$

- Hi are hypotheses (environments with bindings)
- Ti are terms (pieces of abstract syntax)
- Ri are results (types, run-time values, changed environments)
- Hj |- Tj : Rj are sequents
- Premises or preconditions are above the line
- Conclusion is below the line
- Condition on the side if exists must be satisfied

Natural Semantics vs RML

- RML has the same visual syntax as NS

```
rule    RelName1(H1,T1) => R1 & ...
        RelNameN(Hn,Tn) => Rn &
        <cond>
        ------------------------------
        RelName(H, T) => R
```

Natural Semantics vs RML

- rules and propositions are grouped into relations

```
relation negate: bool => bool =
  axiom negate true => false
  axiom negate false => true
end
```

- axioms are rules with no premises

```
axiom negate true => false
rule
       ---------------------
       negate true => false
```

RML datatypes, types, tuples

- datatype declaration

```
datatype Exp = INT of int
             | NEG of Exp
             | ADD of Exp*Exp
```

- type declaration (aliases)

```
type Constant = int
type Identifier = string
```

- tuples declaration

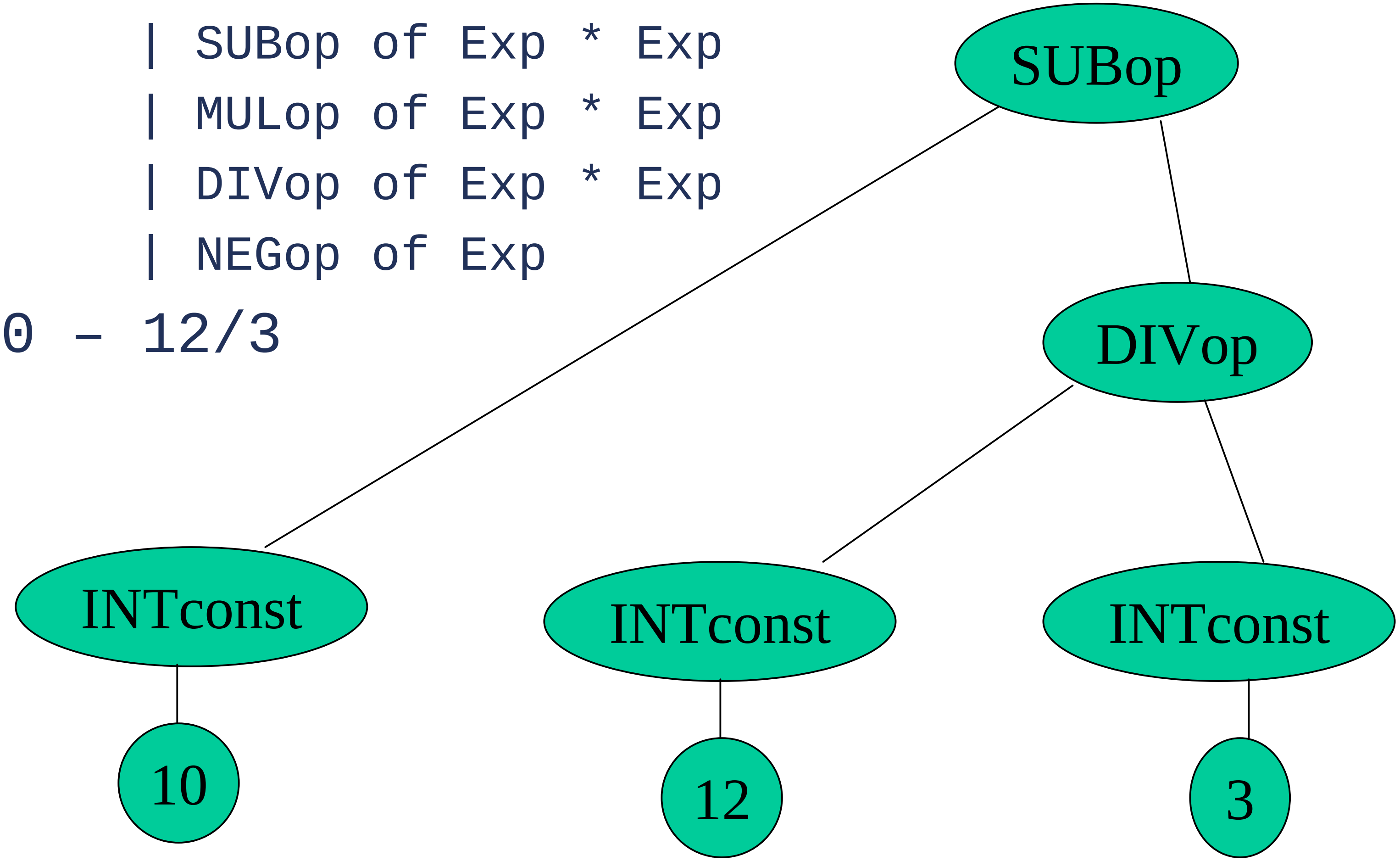
```
type Point = int * int
type Bag = string * int * Exp
```

RML example: the Exp language

- Abstract syntax

```
datatype Exp = INTconst of int
             | PLUSop of Exp * Exp
             | SUBop of Exp * Exp
             | MULop of Exp * Exp
             | DIVop of Exp * Exp
             | NEGop of Exp
```

Exp: 10 – 12/3

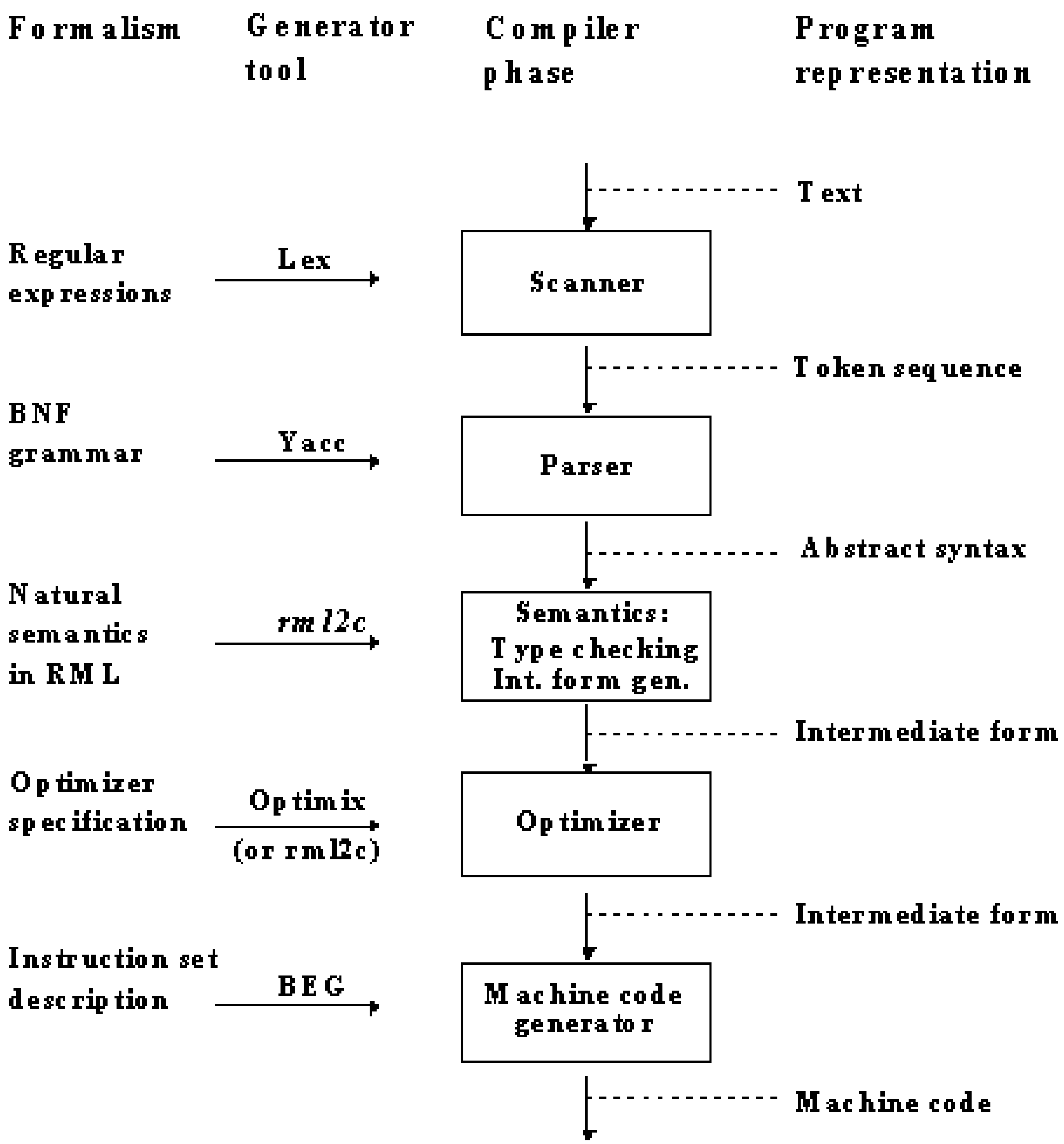


RML example: the Exp language

- Relation eval

```
relation eval: Exp => int =
 axiom eval(INTconst(ival)) => ival
 rule eval(e1) => v1 & eval(e2) => v2 & int_add(v1,v2) => v3
      -------------------------------------------------------
      eval (PLUSop(e1,e2)) => v3
 rule eval(e1) => v1 & eval(e2) => v2 & int_sub(v1,v2) => v3
      ------------------------------------------------------
      eval (SUBop(e1,e2)) => v3
 rule eval(e1) => v1 & eval(e2) => v2 & int_mul(v1,v2) => v3
      ------------------------------------------------------
      eval (MULop(e1,e2)) => v3
 rule eval(e1) => v1 & eval(e2) => v2 & int_div(v1,v2) => v3
      ------------------------------------------------------
      eval (DIVop(e1,e2)) => v3
 rule eval(e) => v1 & int_neg(v1) => v2
      ------------------------------------------------------
      eval (NEGop(e)) => v2
end
```

RML overview



Formalism

Generator tool

Compiler phase

Program representation

Text

Regular expressions

Lex

Scanner

Token sequence

BNF grammar

Yacc

Parser

Abstract syntax

Natural semantics in RML

rml2c

Semantics:
Type checking
Int. form gen.

Intermediate form

Optimizer specification

Optimix
(or rml2c)

Optimizer

Intermediate form

Instruction set description

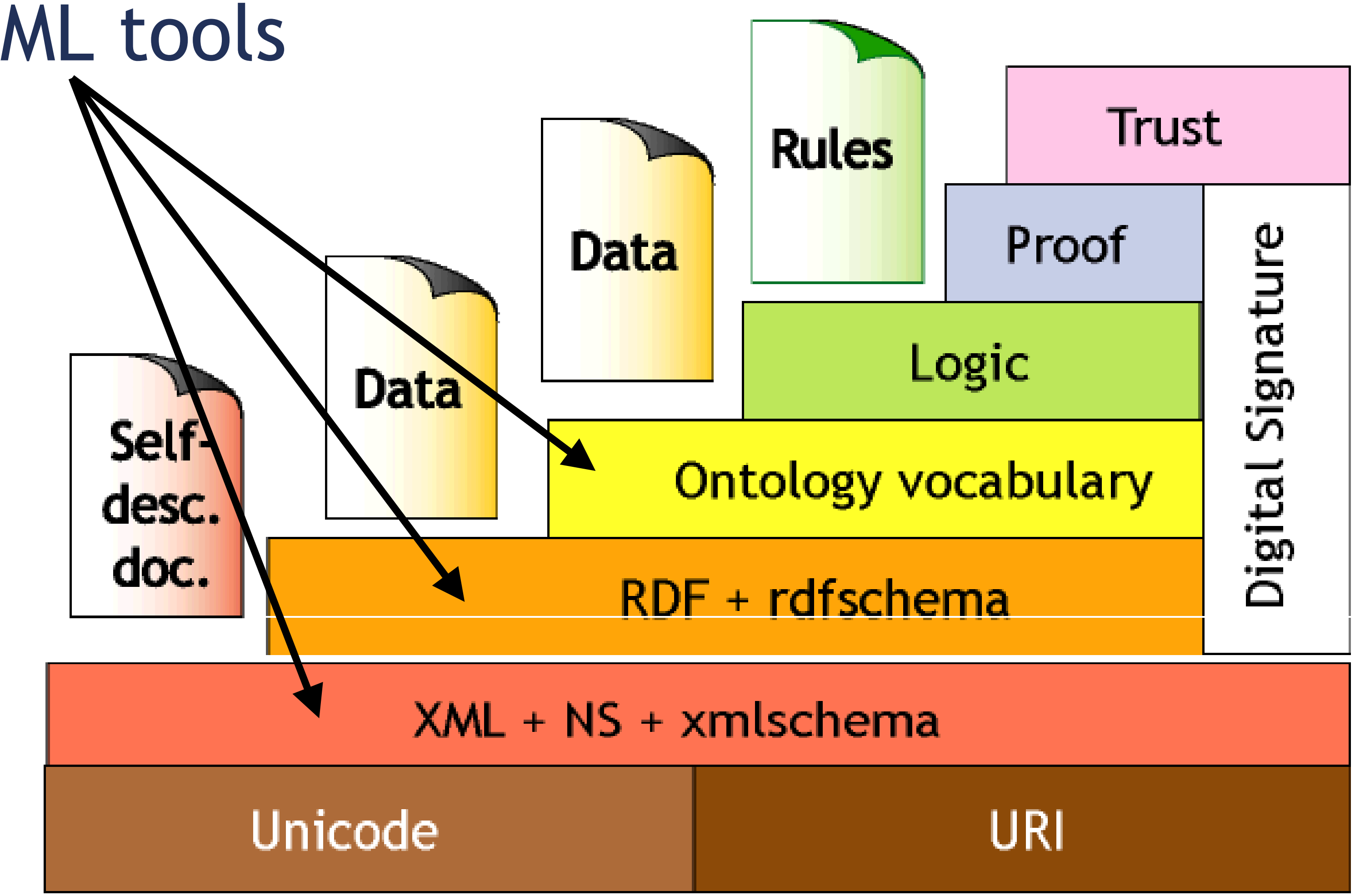
BEG

Machine code generator

Machine code

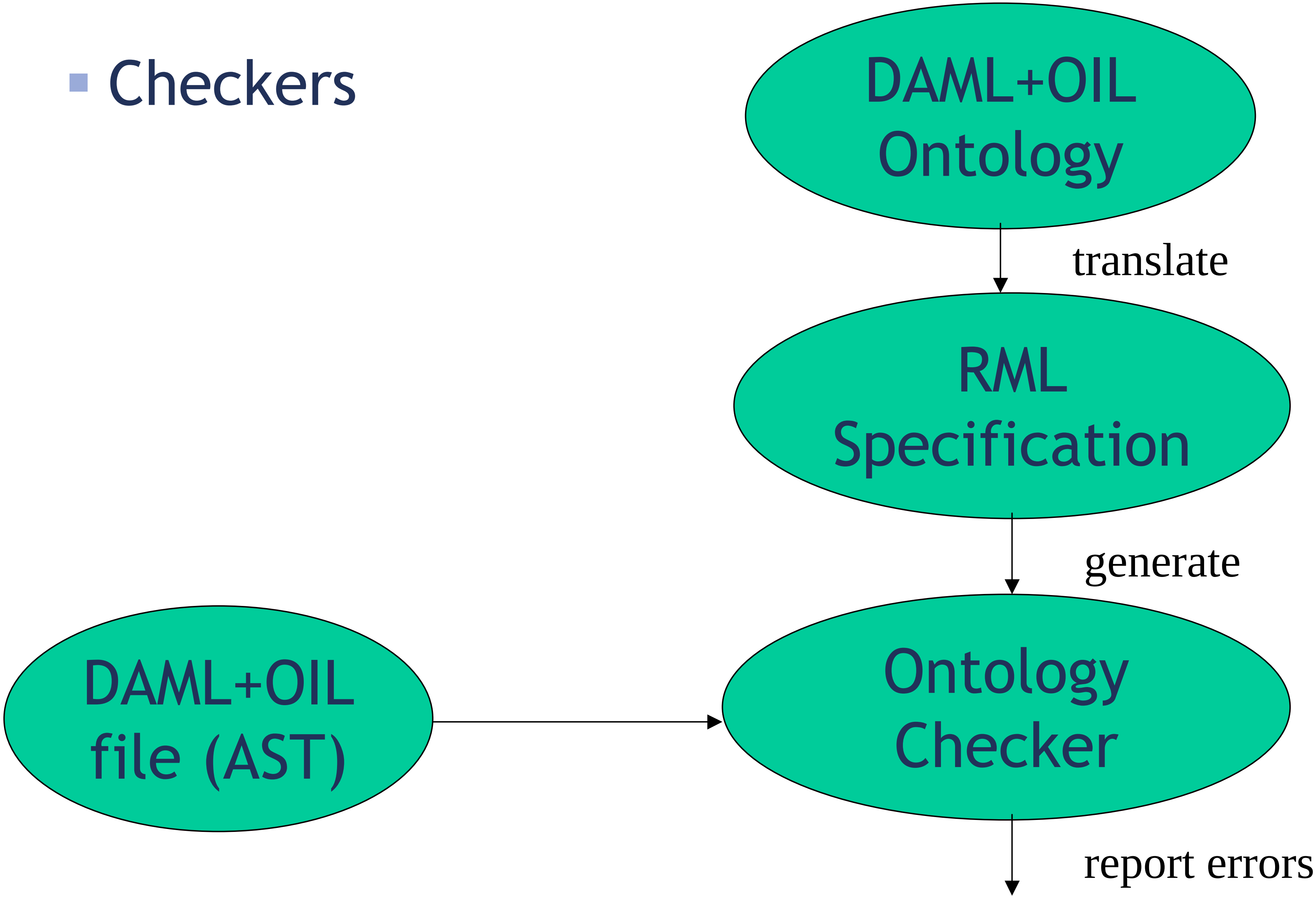
RML Tools for Semantic Web Layer

- RML tools



Ontology Checkers

- Checkers



Ontology Checkers – directions & problems

- Directions
  - building the DAML2RML translator
    - finding a mapping between DAML and RML can be problematic
- Problems
  - RML is not user friendly
  - The internal RML library is not very powerful
  - An RML debugger exists but has to be improved

Related Work - Checking Semantics of Websites

- **Thierry Despeyroux - Brigitte Trousse**
- http://www-rocq.inria.fr/~tdespeyr/papers/riao2000-2
- Applications of NS to Markup Language Based Documents
  - specifying the semantics of new XML languages with Natural Semantics
  - translation from DTD to a form of Abstract Syntax Tree

- Maintaining of Web Sites – Examples
  - **Specifying a Thematic Directory**
  - **Specifying the Coherence of an Institutional Site**

Future Work

- Applying RML to check ontologies
- RML vs RuleML (differences & similarities)

Conclusions

- Natural Semantics is a powerful formalism to used to specify semantics of programming languages
- RML (Relational Meta-Language) can generate executabile specifications
- Building a paralel between programming languages and Semantic Web languages one could find places where RML can prove useful.