The Abstract Syntax as Ontology

Krasimir Angelov

Chalmers University of Technology

August 25, 2009
1 Introduction

2 Types, Dependent Types and Reasoning

3 Semantic Web

4 Ratatui’s Ontology

5 Conclusion
1. Introduction

2. Types, Dependent Types and Reasoning

3. Semantic Web

4. Ratatui’s Ontology

5. Conclusion
Ontology is the philosophical study of the nature of being, existence or reality in general, as well as of the basic categories of being and their relations. Traditionally listed as a part of the major branch of philosophy known as metaphysics, ontology deals with questions concerning what entities exist or can be said to exist, and how such entities can be grouped, related within a hierarchy, and subdivided according to similarities and differences.
Ontologies in C++

The Object Oriented Modeling is building an Ontologies for some concrete domain.

```cpp
class Shape { ... }
class Triangle : Shape { ... }
class Rectangle : Shape { ... }
class Square : Rectangle { ... }
```

The inheritance graph is also called Taxonomy
Classes have properties, members, fields they define relations between entities in the ontology.

```cpp
class Rectangle : Shape { Point topleft, bottomright; }
class Point { int x, y; }
```
The database schema is an ontology defined using relational algebra.
Ontologies in Semantic Web

Object models could be defined in a family of languages RDF, RDFS, OWL Lite, OWL Full

```xml
<shapes:Rectangle rdf:about="#R123">
  <shapes:topleft>
    <shapes:Point>
      <shapes:x>10</shapes:x>
      <shapes:y>10</shapes:y>
    </shapes:Point>
  </shapes:topleft>
  <shapes:bottomright>
    <shapes:Point>
      <shapes:x>100</shapes:x>
      <shapes:y>100</shapes:y>
    </shapes:Point>
  </shapes:bottomright>
</shapes:Rectangle>
```

OWL allows light inference also: inverse properties, transitivity etc.
Ontologies in First-Order Logic

Suggested Upper Merged Ontology (SUMO)

- The biggest open source ontology - 20000 concepts and 70000 axioms
- Mapping to WordNet
- Language generation templates for Hindi, Chinese, Italian, German, Czech and English
- Partial export to OWL
Ontologies in Type Theory

Dependently Types Languages - GF, Agda, Epigram, Coq …

\textbf{cat} \ Shape;\\
  \ Triangle;\\
  \ Rectangle;\\
  \ Point;\\

\textbf{fun} \ triangle : Point \rightarrow\rightarrow\rightarrow Triangle;\\
  \ rectangle : Point \rightarrow\rightarrow Rectangle;\\

\hspace{1cm} \textit{shapeTri} : \textit{Triangle} \rightarrow \textit{Shape};\\
\hspace{1cm} \textit{shapeRect} : \textit{Rectangle} \rightarrow \textit{Shape};
Why Ontology Matters in Linguistics?

- **WordNet**
  - every synset is a semantic concept
  - hierarchy of hypernyms and hyponyms
  - other relations - meronym/holonym

- **VerbNet**
  - primary valency dictionary
  - . . . but also sortal restrictions

- **FrameNet**
  - knowledge about situations
Ontology vs Knowledge Base

The ontology is the schema - definition of classes and properties
The knowledge base is a set of instances, related by properties

In Grammatical Framework:
- Abstract Syntax $\equiv$ Ontology
- Expressions $\equiv$ Knowledge Base
The simple types are well known from every programming language

**Types in Syntax**

- `cat NP, VP, S;`
- `fun everyone_NP : NP;`
- `someone_NP : NP;`

**Types in Semantics**

- `cat Human, Company;`
- `fun john_H : Human;`
- `google_H : Company;`

*Note: In GF there is not firm separation between syntax and semantics*
Dependent Types

The dependent types are types indexed by some value. The dependency could be used to enforce semantic conditions.

Example:

**Simply Typed**

```plaintext
cat Array;
fun plus : Array → Array → Array
```

**Dependently Typed**

```plaintext
cat Array Int;
fun plus : (k : Int) → Array k → Array k → Array k
```
Knowledge about the world could be encoded in the abstract syntax

**Travels Map**

```plaintext
<cat> City;
    Route City City;

<fun> gothenburg_C : City
    stockholm_C : City
    london_C : City

<fun> got2stk_R : Route gothenburg_C stockholm_C
    stk2lon_R : Route stockholm_C london_C
```
We need one more function to make transfers

Travels Map

```
fun join : (c₁, c₂, c₃ : City)
  → Route c₁ c₂
  → Route c₂ c₃
  → Route c₁ c₃
```

Travels Map

```
join gothenburg_C stockholm_C london_C got2stk_R stk2lon_R
  : Route gothenburg_C london_C
```
Every route has a distance which could be computed

Defining the distance function:

```plaintext
fun dist : (c₁, c₂ : City) → Route c₁ c₂ → Int;
def dist c₁ c₂ got2stk_R = 110;
dist c₁ c₂ stk2lon_R = 420;
dist c₁ c₂ (join c₁ c₃ c₂ r₁₃ r₃₂) = dist c₁ c₃ r₁₃ + dist c₃ c₂ r₃₂;
```
1. Introduction

2. Types, Dependent Types and Reasoning

3. Semantic Web

4. Ratatui’s Ontology

5. Conclusion
RDF/RDFS in five sentences

- RDF/RDFS describes **resources**
- Every resource belongs to some **class**
- The descriptions are composed of **statements** which are the atomic units
- Every statement is a triple of subject (resource), predicate (**property**) and object (**value**).
- Every class, statement, and property is also a resource.
RDF/RDFS in four sentences

**cat** Resource \((c : Class)\);

\(Class\);

\(Statement\);

\(Property\ (domain, range : Class)\);

**fun** class : \(Class \rightarrow Resource\) \(class\_C\);

\(statement : Statement \rightarrow Resource\) \(statement\_C\);

\(property : (d, r : Class) \rightarrow Property\) \(d \ r \rightarrow Resource\) \(property\_C\);
Example Resources

Abstract Syntax

```agda
fun organization_\text{C}, sector_\text{C} : \text{Class}
  digitalgrammars_\text{R} : \text{Resource} \ organization_\text{C} ;
  grammars_sector_\text{R} : \text{Resource} \ sector_\text{C} ;
  activeInSector_\text{P} : \text{Property} \ organization_\text{C} \ sector_\text{C}
```

Concrete Syntax

```plaintext
lincat Resource, Class \equiv Str ;
  lin organization_\text{C} = "http://\ldots/protont\#Organization" ;
  sector_\text{C} = "http://\ldots/protonu\#IndustrySector" ;
  digitalgrammars_\text{R} = "http://www.digitalgrammars.com" ;
```

The statements are also known as assertions in logic:

\[
\text{fun assert : (d, r : Class) } \rightarrow \\
\text{ Resource d } \rightarrow \text{ Property d r } \rightarrow \text{ Value r } \rightarrow \text{ Statement;}
\]

here the value category is needed because the object could be a literal also:

\[
\text{cat Value (c : Class);} \\
\text{fun res : (c : Class) } \rightarrow \text{ Resource c } \rightarrow \text{ Value c;} \\
\text{lit : Literal } \rightarrow \text{ Value literal_C;}
\]
English

Digital Grammars is active in sector grammars.

RDF Abstract Syntax

assert organization_C sector_C digitalgrammars_R activeInSector_P grammars_sector_R

RDF Concrete Syntax

⟨http://www.digitalgrammars.com⟩
⟨http://.../protonu#activeInSector⟩
⟨http://www.digitalgrammars.com/grammars⟩
Inheritance and Type Casting

\textbf{cat} \textit{SubClass Class Class;}
\textit{Inheritance Class Class;}

\textbf{fun} \textit{trans} : (c_1, c_2, c_3 : \textit{Class})
\rightarrow \textit{SubClass} c_1 c_2
\rightarrow \textit{Inheritance} c_2 c_3
\rightarrow \textit{Inheritance} c_1 c_3

\textbf{fun} \textit{upcast} : (c_1, c_2 : \textit{Class})
\rightarrow \textit{Inheritance} c_1 c_2
\rightarrow \textit{Resource} c_1
\rightarrow \textit{Resource} c_2
INGREDIENTS

- 1/4 cup finely chopped onion
- 1 tablespoon butter or margarine
- 1/4 teaspoon dried basil
- 1/4 teaspoon paprika
- 1/8 teaspoon garlic powder
- 1 can condensed tomato soup
- 1 cup milk

DIRECTIONS

Saute onion in butter until tender. Add basil, paprika and garlic powder. Stir in soup and milk until well blended. Cook over medium heat for 6 minutes or until heated through.
INGREDIENTS
1/4 cup finely chopped onion
1 tablespoon butter or margarine
1/4 teaspoon dried basil
1/4 teaspoon paprika
1/8 teaspoon garlic powder
1 can condensed tomato soup
1 cup milk

DIRECTIONS
Saute onion in butter until tender. Add basil, paprika and garlic powder. Stir in soup and milk until well blended. Cook over medium heat for 6 minutes or until heated through.
The Ontology

\textbf{cat} \textit{Quantity};

\textbf{fun} \textit{whole} : \textit{Int} \rightarrow \textit{Quantity}; 
\hspace{5cm} - - 1

\hspace{1cm} \textit{fraction} : \textit{Int} \rightarrow \textit{Int} \rightarrow \textit{Quantity}; 
\hspace{5cm} - - \frac{1}{2}

\hspace{1cm} \textit{whole_plus} : \textit{Int} \rightarrow \textit{Int} \rightarrow \textit{Int} \rightarrow \textit{Quantity}; 
\hspace{5cm} - - 1 \frac{1}{2}

\textbf{cat} \textit{Unit};

\textbf{fun} \textit{cup, tablespoon, teaspoon, can} : \textit{Unit};

\textbf{cat} \textit{Measure};

\textbf{fun} \textit{measure} : \textit{Quantity} \rightarrow \textit{Unit} \rightarrow \textit{Measure};
The Receipe

**INGREDIENTS**
- 1/4 cup finely chopped onion
- 1 tablespoon butter or margarine
- 1/4 teaspoon dried basil
- 1/4 teaspoon paprika
- 1/8 teaspoon garlic powder
- 1 can condensed tomato soup
- 1 cup milk

**DIRECTIONS**
Saute onion in butter until tender. Add basil, paprika and garlic powder. Stir in soup and milk until well blended. Cook over medium heat for 6 minutes or until heated through.
cat Product;
fun onion, butter, margarine,
  basil, paprika, milk, powder, soup : Product;
INGREDIENTS
1/4 cup finely chopped onion
1 tablespoon butter or margarine
1/4 teaspoon dried basil
1/4 teaspoon paprika
1/8 teaspoon garlic powder
1 can condensed tomato soup
1 cup milk

DIRECTIONS
Saute onion in butter until tender. Add basil, paprika and garlic powder. Stir in soup and milk until well blended. Cook over medium heat for 6 minutes or until heated through.
The Ontology

cat Product;

fun onion, butter, margarine, basil, paprika, milk, garlic, tomato : Product;

fun powder, soup : Product → Product;
INGREDIENTS
1/4 cup finely chopped onion
1 tablespoon butter or margarine
1/4 teaspoon dried basil
1/4 teaspoon paprika
1/8 teaspoon garlic powder
1 can condensed tomato soup
1 cup milk

DIRECTIONS
Saute onion in butter until tender. Add basil, paprika and garlic powder. Stir in soup and milk until well blended. Cook over medium heat for 6 minutes or until heated through.
fun alternative, composite : Product → Product → Product;
**INGREDIENTS**
1/4 cup finely chopped onion
1 tablespoon butter or margarine
1/4 teaspoon dried basil
1/4 teaspoon paprika
1/8 teaspoon garlic powder
1 can condensed tomato soup
1 cup milk

**DIRECTIONS**
Saute onion in butter until tender. Add basil, paprika and garlic powder. Stir in soup and milk until well blended. Cook over medium heat for 6 minutes or until heated through.
The Ontology

\texttt{cat \ Preparation;}
\texttt{fun \ chopped, \ dried, \ condensed \ : \ Preparation;}

\texttt{fun \ preparation \ : \ Preparation \ \rightarrow \ Product \ \rightarrow \ Product;}

\texttt{fun \ preparation \ : \ Preparation \ \rightarrow \ Product \ \rightarrow \ Product;}

INGREDIENTS
1/4 cup finely chopped onion
1 tablespoon butter or margarine
1/4 teaspoon dried basil
1/4 teaspoon paprika
1/8 teaspoon garlic powder
1 can condensed tomato soup
1 cup milk

DIRECTIONS
Saute onion in butter until tender. Add basil, paprika and garlic powder. Stir in soup and milk until well blended. Cook over medium heat for 6 minutes or until heated through.
The Ontology

\textbf{cat} PreparationMod;
\textbf{fun} finely : PreparationMod;

\textbf{fun} preparation\_mod : PreparationMod \rightarrow Preparation \rightarrow Preparation;
**INGREDIENTS**
- 1/4 cup finely chopped onion
- 1 tablespoon butter or margarine
- 1/4 teaspoon dried basil
- 1/4 teaspoon paprika
- 1/8 teaspoon garlic powder
- 1 can condensed tomato soup
- 1 cup milk

**DIRECTIONS**
Saute onion in butter until tender. Add basil, paprika and garlic powder. Stir in soup and milk until well blended. Cook over medium heat for 6 minutes or until heated through.
The Ontology

\textbf{cat} Ingredient;

\textbf{fun} ingredient : Measure $\rightarrow$ Product $\rightarrow$ Ingredient;

\begin{align*}
ingredient & (measure \ (fraction \ 1 \ 4 \ teaspoon) \ (preparation \ dried \ basil)) & | & \text{1/4 teaspoon dried basil} \\
ingredient & (measure \ (fraction \ 1 \ 4 \ teaspoon) \ paprika) & | & \text{1/4 teaspoon paprika} \\
ingredient & (measure \ (fraction \ 1 \ 8 \ teaspoon) \ (powder \ garlic)) & | & \text{1/8 teaspoon garlic powder} \\
ingredient & (measure \ (whole \ 1 \ cup) \ milk) & | & \text{1 cup milk}
\end{align*}
INGREDIENTS
1/4 cup finely chopped onion
1 tablespoon butter or margarine
1/4 teaspoon dried basil
1/4 teaspoon paprika
1/8 teaspoon garlic powder
1 can condensed tomato soup
1 cup milk

DIRECTIONS
Saute onion in butter until tender. Add basil, paprika and garlic powder. Stir in soup and milk until well blended. Cook over medium heat for 6 minutes or until heated through.
The Ontology

Directions:

\textbf{cat} \ Direction;

\textbf{fun} \ action : Action \rightarrow \ Direction;

\hspace{1cm} \textit{operation} : Operation \rightarrow Duration \rightarrow Direction;
The Ontology

Actions:

```haskell
cat Action;
fun add : Product → Action;
```
The Ontology

Operations:

```
cat Operation;
fun soute : Product → Product → Operation;
    stir_in : Product → Operation;
    cook : Temperature → Operation;

cat Temperature;
fun celsius : Int → Temperature;

fun medium_heat : Temperature;
def medium_heat = celsius 80;
```
INGREDIENTS
1/4 cup finely chopped onion
1 tablespoon butter or margarine
1/4 teaspoon dried basil
1/4 teaspoon paprika
1/8 teaspoon garlic powder
1 can condensed tomato soup
1 cup milk

DIRECTIONS
Saute onion in butter until tender. Add basil, paprika and garlic powder. Stir in soup and milk until well blended. Cook over medium heat for 6 minutes or until heated through.
The Ontology

cat Duration;

fun until : Condition → Duration;

fun for_time : Int → TimeUnit → Duration;

cat Condition;

fun tender, blended, heated : Condition;

cat TimeUnit;

fun second, minute, hour : TimeUnit;
INGREDIENTS
1/4 cup finely chopped onion
1 tablespoon butter or margarine
1/4 teaspoon dried basil
1/4 teaspoon paprika
1/8 teaspoon garlic powder
1 can condensed tomato soup
1 cup milk

DIRECTIONS
Saute onion in butter until tender. Add basil, paprika and garlic powder. Stir in soup and milk until well blended. Cook over medium heat for 6 minutes or until heated through.
The Ontology

fun either : Duration → Duration → Duration;
**INGREDIENTS**
- 1/4 cup finely chopped onion
- 1 tablespoon butter or margarine
- 1/4 teaspoon dried basil
- 1/4 teaspoon paprika
- 1/8 teaspoon garlic powder
- 1 can condensed tomato soup
- 1 cup milk

**DIRECTIONS**
Saute onion in butter until tender. Add basil, paprika and garlic powder. Stir in soup and milk until well blended. Cook over medium heat for 6 minutes or until heated through.
The Ontology

cat Receipe;

fun receipe : [Ingredient] → [Direction] → Receipe;
The ontology permits some expressions which are syntactically correct but not semantically consistent.

\[\text{soute onion garlic} \quad \text{soute onion in garlic}\]
\[\text{stir\_in paprika} \quad \text{stir in paprika}\]

Solution: Use \textbf{Dependent Types}
cat Kind;

fun liquid, oil, firm : Kind;

cat Product Kind;

fun alternative, composite : (k : Kind) → Product k → Product k → Product k;

fun add : (k : Kind) → Product k → Direction;

fun soute : Product firm → Product oil → Operation;
  stir_in : Product liquid → Operation;
Some product could be mentioned in the description only if it is also declared in the ingredients.

The total quantity of every product in the description should be equal to the quantity in the ingredients.
RDF Concrete Syntax

\[
gf2rdf : \text{Resource measure}_C \rightarrow \text{Measure} \rightarrow [\text{Statement}]
\]

\[
gf2rdf \ id \ (\text{whole} \ v) = \[
\[\text{assert} \ ? \ ? \ id \ \text{hasValue} \ (\text{lit} \ (\text{int} \ v))\]
\]
\]

\[
gf2rdf \ id \ (\text{fraction} \ m \ n) = \[
\[\text{assert} \ ? \ ? \ id \ \text{hasNomValue} \ (\text{lit} \ (\text{int} \ m))\]
, \text{assert} \ ? \ ? \ id \ \text{hasDenomValue} \ (\text{lit} \ (\text{int} \ n))\]
\]
\]

\[
gf2rdf \ id \ (\text{whole}_\text{plus} \ v \ m \ n) = \[
\[\text{assert} \ ? \ ? \ id \ \text{hasValue} \ (\text{lit} \ (\text{int} \ n))\]
, \text{assert} \ ? \ ? \ id \ \text{hasNomValue} \ (\text{lit} \ (\text{int} \ m))\]
, \text{assert} \ ? \ ? \ id \ \text{hasDenomValue} \ (\text{lit} \ (\text{int} \ n))\]
\]
\]
Thank You and Have Fun !!!