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

What is ontology?

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.*

Wikipedia

Ontologies in C++

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

C++

class *Shape* { . . . }
class *Triangle* : *Shape* { . . . }
class *Rectangle* : *Shape* { . . . }
class *Square* : *Rectangle* { . . . }

The inheritance graph is also called Taxonomy

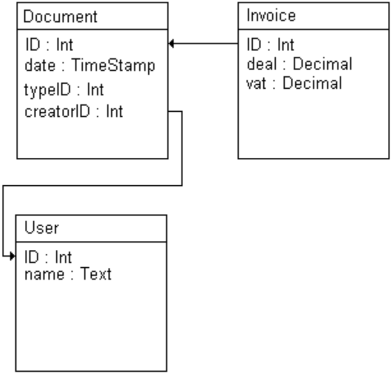
Ontological Relations

Classes have properties, members, fields they define relations between entities in the ontology.

C++

class *Rectangle* : *Shape* { *Point topleft*, *bottomright*; }
class *Point* { int x,y; }

The database schema is an ontology defined using relational algebra.



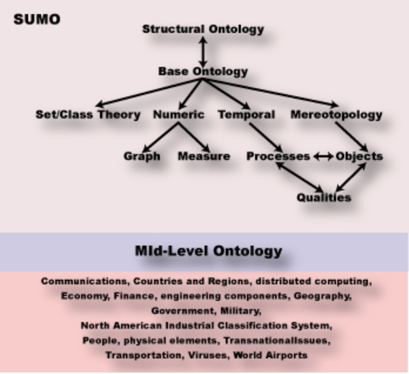
Object models could be defined in a family of languages RDF, RDFS, OWL Lite, OWL Full

```
<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.

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 . . .

$$
\begin{aligned}
&\textbf{cat}\ \ \mathit{Shape};\\
&\phantom{\textbf{cat}}\ \ \mathit{Triangle};\\
&\phantom{\textbf{cat}}\ \ \mathit{Rectangle};\\
&\phantom{\textbf{cat}}\ \ \mathit{Point};
\end{aligned}
$$

$$
\begin{aligned}
&\textbf{fun}\ \ \mathit{triangle} : \mathit{Point} \longrightarrow \mathit{Point} \longrightarrow \mathit{Point} \longrightarrow \mathit{Triangle};\\
&\phantom{\textbf{fun}}\ \ \mathit{rectangle} : \mathit{Point} \longrightarrow \mathit{Point} \longrightarrow \mathit{Rectangle};\\
&\\
&\phantom{\textbf{fun}}\ \ \mathit{shapeTri} : \mathit{Triangle} \longrightarrow \mathit{Shape};\\
&\phantom{\textbf{fun}}\ \ \mathit{shapeRect} : \mathit{Rectangle} \longrightarrow \mathit{Shape};
\end{aligned}
$$

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

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

Simple Types

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

$$\textbf{cat}\ \mathit{Array};$$

$$\textbf{fun}\ \mathit{plus} : \mathit{Array} \longrightarrow \mathit{Array} \longrightarrow \mathit{Array}$$

Dependently Typed

$$\textbf{cat}\ \mathit{Array}\ \mathit{Int};$$

$$\textbf{fun}\ \mathit{plus} : (k : \mathit{Int}) \longrightarrow \mathit{Array}\ k \longrightarrow \mathit{Array}\ k \longrightarrow \mathit{Array}\ k$$

Reasoning

Knowledge about the world could be encoded in the abstract syntax

Travels Map

$$\begin{aligned}
&\textbf{cat}\ City;\\
&\quad Route\ City\ City;\\
\\
&\textbf{fun}\ gothenburg_C : City\\
&\quad stockholm_C : City\\
&\quad london_C : City\\
\\
&\textbf{fun}\ got2stk_R : Route\ gothenburg_C\ stockholm_C\\
&\quad stk2lon_R : Route\ stockholm_C\ london_C
\end{aligned}$$

Reasoning

We need one more function to make transfers

Travels Map

$$\begin{aligned}\textbf{fun}\ \mathit{join} : (c_1, c_2, c_3 : \mathit{City}) \\ &\rightarrow \mathit{Route}\ c_1\ c_2 \\ &\rightarrow \mathit{Route}\ c_2\ c_3 \\ &\rightarrow \mathit{Route}\ c_1\ c_3\end{aligned}$$

Travels Map

$$\begin{aligned}&\mathit{join}\ \mathit{gothenburg\_C}\ \mathit{stockholm\_C}\ \mathit{london\_C}\ \mathit{got2stk\_R}\ \mathit{stk2lon\_R} \\ &\quad : \mathit{Route}\ \mathit{gothenburg\_C}\ \mathit{london\_C}\end{aligned}$$

Computation

Every route has a distance which could be computed

Distance

fun $\mathit{dist} : (c_1, c_2 : \mathit{City}) \to \mathit{Route}\ c_1\ c_2 \to \mathit{Int};$

def $\mathit{dist}\ c_1\ c_2\ \mathit{got2stk\_R} = 110;$

$\mathit{dist}\ c_1\ c_2\ \mathit{stk2lon\_R} = 420;$

$\mathit{dist}\ c_1\ c_2\ (\mathit{join}\ c_1\ c_3\ c_2\ r_{13}\ r_{32}) = \mathit{dist}\ c_1\ c_3\ r_{13} + \mathit{dist}\ c_3\ c_2\ r_{32};$

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 tripple of subject (resource), predicate (**property**) and object (**value**).
- Every class, statement, and property is also a resource.

RDF/RDFS in four sentences

$$
\begin{aligned}
&\textbf{cat}\ \mathit{Resource}\ (c : \mathit{Class});\\
&\quad \mathit{Class};\\
&\quad \mathit{Statement};\\
&\quad \mathit{Property}\ (\mathit{domain}, \mathit{range} : \mathit{Class});
\end{aligned}
$$

$$
\begin{aligned}
&\textbf{fun}\ \mathit{class} : \mathit{Class} \to \mathit{Resource}\ \mathit{class\_C};\\
&\quad \mathit{statement} : \mathit{Statement} \to \mathit{Resource}\ \mathit{statement\_C};\\
&\quad \mathit{property} : (d, r : \mathit{Class}) \to \mathit{Property}\ d\ r \to \mathit{Resource}\ \mathit{property\_C};
\end{aligned}
$$

Example Resources

Abstract Syntax

$$
\begin{aligned}
&\textbf{fun}\ \mathit{organization\_C}, \mathit{sector\_C} : \mathit{Class} \\
&\quad \mathit{digitalgrammars\_R} : \mathit{Resource}\ \mathit{organization\_C}; \\
&\quad \mathit{grammars\_sector\_R} : \mathit{Resource}\ \mathit{sector\_C}; \\
&\quad \mathit{activeInSector\_P} : \mathit{Property}\ \mathit{organization\_C}\ \mathit{sector\_C}
\end{aligned}
$$

Concrete Syntax

$$
\begin{aligned}
&\textbf{lincat}\ \mathit{Resource}, \mathit{Class} = \mathit{Str}; \\
&\quad \textbf{lin}\ \mathit{organization\_C} = \text{"http://.../protont\#Organization"}; \\
&\qquad \mathit{sector\_C} = \text{"http://.../protonu\#IndustrySector"}; \\
&\qquad \mathit{digitalgrammars\_R} = \text{"http://www.digitalgrammars.com"}; \\
&\quad \ldots
\end{aligned}
$$

The statements are also known as assertions in logic:

$$\textbf{fun } \mathit{assert} : (d, r : \mathit{Class}) \rightarrow \\ \mathit{Resource}\ d \rightarrow \mathit{Property}\ d\ r \rightarrow \mathit{Value}\ r \rightarrow \mathit{Statement};$$

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

$$\textbf{cat } \mathit{Value}\ (c : \mathit{Class});$$
$$\textbf{fun } \mathit{res} : (c : \mathit{Class}) \rightarrow \mathit{Resource}\ c \rightarrow \mathit{Value}\ c;$$
$$\mathit{lit} : \mathit{Literal} \rightarrow \mathit{Value}\ \mathit{literal\_C};$$

Example Statement

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⟩

Class Hierarchy in OWL

Inheritance and Type Casting

$$\begin{aligned}
\textbf{cat}\ & \mathit{SubClass}\ \mathit{Class}\ \mathit{Class}; \\
& \mathit{Inheritance}\ \mathit{Class}\ \mathit{Class};
\end{aligned}$$

$$\begin{aligned}
\textbf{fun}\ \mathit{trans} : (c_1, c_2, c_3 : \mathit{Class}) \\
\rightarrow\ & \mathit{SubClass}\ c_1\ c_2 \\
\rightarrow\ & \mathit{Inheritance}\ c_2\ c_3 \\
\rightarrow\ & \mathit{Inheritance}\ c_1\ c_3
\end{aligned}$$

$$\begin{aligned}
\textbf{fun}\ \mathit{upcast} : (c_1, c_2 : \mathit{Class}) \\
\rightarrow\ & \mathit{Inheritance}\ c_1\ c_2 \\
\rightarrow\ & \mathit{Resource}\ c_1 \\
\rightarrow\ & \mathit{Resource}\ c_2
\end{aligned}$$

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.

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.

The Ontology

cat $Quantity$;

fun $whole : Int \rightarrow Quantity$; -- 1

$fraction : Int \rightarrow Int \rightarrow Quantity$; -- 1/2

$whole\_plus : Int \rightarrow Int \rightarrow Int \rightarrow Quantity$; -- 1 1/2

cat $Unit$;

fun $cup, tablespoon, teaspoon, can : Unit$;

cat $Measure$;

fun $measure : Quantity \rightarrow Unit \rightarrow Measure$;

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.

The Ontology

cat *Product*;

fun *onion*, *butter*, *margarine*,
basil, *paprika*, *milk*, *powder*, *soup* : *Product*;

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.

The Ontology

$$
\begin{array}{ll}
\textbf{cat} & \mathit{Product}; \\
\textbf{fun} & \mathit{onion}, \mathit{butter}, \mathit{margarine}, \\
 & \mathit{basil}, \mathit{paprika}, \mathit{milk}, \mathit{garlic}, \mathit{tomato} : \mathit{Product}; \\
\textbf{fun} & \mathit{powder}, \mathit{soup} : \mathit{Product} \longrightarrow \mathit{Product};
\end{array}
$$

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.

$$\textbf{fun}\ \mathit{alternative}, \mathit{composite} : \mathit{Product} \rightarrow \mathit{Product} \rightarrow \mathit{Product};$$

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.

The Ontology

$\mathbf{cat}\ \ \mathit{Preparation};$

$\mathbf{fun}\ \ \mathit{chopped}, \mathit{dried}, \mathit{condensed} : \mathit{Preparation};$

$\mathbf{fun}\ \ \mathit{preparation} : \mathit{Preparation} \longrightarrow \mathit{Product} \longrightarrow \mathit{Product};$

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.

The Ontology

$$\textbf{cat}\ \textit{PreparationMod};$$

$$\textbf{fun}\ \textit{finely} : \textit{PreparationMod};$$

$$\textbf{fun}\ \textit{preparation\_mod} : \textit{PreparationMod} \rightarrow \textit{Preparation} \rightarrow \textit{Preparation};$$

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.

The Ontology

$$\textbf{cat}\ \mathit{Ingredient};$$

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

ingredient (*measure* (*fraction* 1 4) *teaspoon*) (*preparation dried basil*)	1/4 teaspoon dried basil
ingredient (*measure* (*fraction* 1 4) *teaspoon*) *paprika*	1/4 teaspoon paprika
ingredient (*measure* (*fraction* 1 8) *teaspoon*) (*powder garlic*)	1/8 teaspoon garlic powder
ingredient (*measure* (*whole* 1) *cup*) *milk*	1 cup milk

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.

The Ontology

Directions:

$$\begin{aligned}
&\textbf{cat}\ \ \mathit{Direction};\\
&\textbf{fun}\ \ \mathit{action} : \mathit{Action} \rightarrow \mathit{Direction};\\
&\qquad\ \ \mathit{operation} : \mathit{Operation} \rightarrow \mathit{Duration} \rightarrow \mathit{Direction};
\end{aligned}$$

The Ontology

Actions:

$$\textbf{cat}\ Action;$$
$$\textbf{fun}\ add : Product \rightarrow Action;$$

Operations:

$$\textbf{cat}\ \mathit{Operation};$$

$$\begin{aligned}
\textbf{fun}\ & \mathit{soute} : \mathit{Product} \rightarrow \mathit{Product} \rightarrow \mathit{Operation}; \\
& \mathit{stir_in} : \mathit{Product} \rightarrow \mathit{Operation}; \\
& \mathit{cook} : \mathit{Temperature} \rightarrow \mathit{Operation};
\end{aligned}$$

$$\textbf{cat}\ \mathit{Temperature};$$

$$\textbf{fun}\ \mathit{celsius} : \mathit{Int} \rightarrow \mathit{Temperature};$$

$$\textbf{fun}\ \mathit{medium\_heat} : \mathit{Temperature};$$

$$\textbf{def}\ \mathit{medium\_heat} = \mathit{celsius}\ 80;$$

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.

The Ontology

cat *Duration*;

fun $until : Condition \rightarrow Duration$;

$for\_time : Int \rightarrow TimeUnit \rightarrow Duration$;

cat *Condition*;

fun $tender, blended, heated : Condition$;

cat *TimeUnit*;

fun $second, minute, hour : TimeUnit$;

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.

The Ontology

$$\textbf{fun}\ \mathit{either} : \mathit{Duration} \to \mathit{Duration} \to \mathit{Duration};$$

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.

The Ontology

$$\textbf{cat } \mathit{Receipe};$$

$$\textbf{fun } \mathit{receipe} : [\mathit{Ingredient}] \rightarrow [\mathit{Direction}] \rightarrow \mathit{Receipe};$$

Defects in Ratatui

The ontology permits some expressions which are syntactically correct but not semantically consistent.

soute onion garlic	soute onion in garlic
stir_in paprika	stir in paprika

Solution: Use **Dependent Types**

Ratatui with Dependent Types

cat Kind;

fun $\mathit{liquid}, \mathit{oil}, \mathit{firm} : \mathit{Kind}$;

cat $\mathit{Product}\ \mathit{Kind}$;

fun $\mathit{alternative}$,

$\quad \mathit{composite} : (k : \mathit{Kind}) \to \mathit{Product}\ k \to \mathit{Product}\ k \to \mathit{Product}\ k$;

fun $\mathit{add} : (k : \mathit{Kind}) \to \mathit{Product}\ k \to \mathit{Direction}$;

fun $\mathit{soute} : \mathit{Product}\ \mathit{firm} \to \mathit{Product}\ \mathit{oil} \to \mathit{Operation}$;

$\quad \mathit{stir\_in} : \mathit{Product}\ \mathit{liquid} \to \mathit{Operation}$;

Things that are not covered

- 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

Ratatui to RDF

RDF Concrete Syntax

$$
\begin{array}{l}
\mathit{gf2rdf} : \mathit{Resource\ measure\_C} \rightarrow \mathit{Measure} \rightarrow [\mathit{Statement}] \\
\mathit{gf2rdf}\ \mathit{id}\ (\mathit{whole}\ v) = \\
\quad [\ \mathit{assert}\ ?\ ?\ \mathit{id}\ \mathit{hasValue}\ (\mathit{lit}\ (\mathit{int}\ v)) \\
\quad] \\
\mathit{gf2rdf}\ \mathit{id}\ (\mathit{fraction}\ m\ n) = \\
\quad [\ \mathit{assert}\ ?\ ?\ \mathit{id}\ \mathit{hasNomValue}\ (\mathit{lit}\ (\mathit{int}\ m)) \\
\quad ,\ \mathit{assert}\ ?\ ?\ \mathit{id}\ \mathit{hasDenomValue}\ (\mathit{lit}\ (\mathit{int}\ n)) \\
\quad] \\
\mathit{gf2rdf}\ \mathit{id}\ (\mathit{whole\_plus}\ v\ m\ n) = \\
\quad [\ \mathit{assert}\ ?\ ?\ \mathit{id}\ \mathit{hasValue}\ (\mathit{lit}\ (\mathit{int}\ n)) \\
\quad ,\ \mathit{assert}\ ?\ ?\ \mathit{id}\ \mathit{hasNomValue}\ (\mathit{lit}\ (\mathit{int}\ m)) \\
\quad ,\ \mathit{assert}\ ?\ ?\ \mathit{id}\ \mathit{hasDenomValue}\ (\mathit{lit}\ (\mathit{int}\ n)) \\
\quad]
\end{array}
$$

Thank You

Thank You and Have Fun !!!