# Ontology Integration and Interoperability (OntoIOp) – Part 1: The Distributed Ontology Language (DOL)

## ISO WD 17347 (Draft 2)

Till Mossakowski, Oliver Kutz, Christoph Lange, Michael Grüninger

Universität Bremen, Germany
University of Toronto, Canada

2012-02-23

Universität Bremen

# Working Draft 2 (December/January)

- OntoIOp **Working Draft 2** circulated on 2011-12-24
- Release notes were:
  - abstract syntax specified
  - concrete syntax
    - **mostly complete text syntax** (example in this talk)
    - initial XML and RDF syntaxes
    - minimal RDF logic description vocabulary
  - **foundations of the semantics** (overview in this talk)
  - **semantics of the abstract syntax** (excerpt in this talk)
  - **high-level descriptions of use cases** (example in this talk)
  - revised and improved requirements and **conformance criteria** (overview in this talk)

Universität Bremen

# DOL Semantics Discussion (January/February)

- Two central issues:
  - Standard **semantics of the basic language** should not require institutions (perceived as a conceptual overhead)
  - Common Logic is the most expressive standard ontology language in OntoIOp ⇒ reuse existing **Common Logic semantics** for the rest of DOL
- Agreement of 2012-02-15:
  - **Three compatible semantics** for the basic language (details later in this talk):
    1. direct
    2. translational
    3. collapsed
  - Institutional semantics (compatible with the former) for extended language

Universität Bremen

# DOL Text Syntax of an Ontology

One **ontology** within a distributed ontology looks as follows:

```
ONTO ::= BASIC-ONTO               %% logic-specific, given in a conforming language
       | ONTO TRANSLATION     %% translate ONTO and/or rename symbols
       | ONTO RESTRICTION     %% hide symbols, or reveal hidden symbols
       | ONTO and CONS-STRENGTH? ONTO        %% union
       | ONTO then CONS-STRENGTH? BASIC-ONTO %% extension
       | ONTO-REF
       | logic LOGIC-REF : GROUP-ONTO %% switch the logic
       | combine ONTO-OR-INTPR-REF ,..., ONTO-OR-INTPR-REF REMOVE-IMPORTS?
CONS-STRENGTH  ::= %mcons | %ccons | %mono | %def | %implied
```

A **distributed ontology** comprises

- such ontologies
- and links between them (interpretations and alignments).

Universität Bremen

# Complete Example in Text Syntax (1)

## A heterogeneous ontology for mereology:

```
prefix       = <http://www.example.org/mereology#>
prefix owl   = <http://www.w3.org/2002/07/owl#>
prefix log   = <http://purl.net/dol/logic/> %% descriptions of logics ...
prefix trans = <http://purl.net/dol/translations/> %% ... and translations

distributed-ontology Mereology

logic log:Propositional                       %% syntax used: similar to OWL Manchester
ontology Taxonomy =          %% DOLCE's basic taxonomic information about mereology
  props PT %[ Particular  ]%, PD %[ Perdurant      ]%, T %[ TimeInterval ]%,
        S  %[ SpaceRegion ]%, AR %[ AbstractRegion ]%
  . S ∨ T ∨ AR ∨ PD ⟶ PT %% PT is the top concept
  . S ∧ T  ⟶ ⊥ %% PD, S, T, AR are pairwise disjoint
  . T ∧ AR ⟶ ⊥ %% ...
end

logic log:OWL                                  %% syntax: OWL Manchester serialization
ontology BasicParthood =                   %% Parthood in OWL DL, as far as expressible
  Class: ParticularCategory SubClassOf: PT     %% other class declarations omitted
    DisjointUnionOf: S, T, AR, PD             %% pairwise disjointness more compact
  ObjectProperty: isPartOf       Characteristics: Transitive
  ObjectProperty: isProperPartOf Characteristics: Transitive, Asymmetric
    SubPropertyOf: isPartOf
  Class: Atom EquivalentTo: inverse isProperPartOf only owl:Nothing
end                                                     %% an atom has no proper parts

interpretation TaxonomyToParthood : Taxonomy with logic trans:PropToOWL to BasicParthood
```

Universität Bremen

# Complete Example in Text Syntax (2)

## A heterogeneous ontology for mereology (cont'd.):

```
logic log:CommonLogic                        %% syntax: CLIF dialect of Common Logic
ontology ClassicalExtensionalParthood =
  BasicParthood then {       %% import OWL ontology from above, translate it to CL
  . (forall (X) (if (or (= X S) (= X T) (= X AR) (= X PD))
                  (forall (x y z) (if (and (X x) (X y) (X z))
                                      (and           %% now list all the axioms
      (if (and (isPartOf x y) (isPartOf y x)) (= x y))             %% antisymmetry
      (iff (overlaps x y) (exists (pt) (and (isPartOf pt x) (isPartOf pt y))))
      (iff (isAtomicPartOf x y) (and (isPartOf x y) (Atom x)))
      (iff (sum z x y)
           (forall (w) (iff (overlaps w z) (and (overlaps w x) (overlaps w y)))))
      (exists (s) (sum s x y)) %% existence of the sum
      )))))
  . (forall (Set a) (iff (fusion Set a)                      %% definition of fusion
           (forall (b) (iff (overlaps b a)
                            (exists (c) (and (Set c) (overlaps c a)))))))
  }
```

# Namespaces for Identifiers (1)

Rationale for Namespace Prefixes

- Requirement: support **Web-scalable ontologies**
- Easy solution: **all names are URIs** (actually IRIs)
  - In DOL, this includes names of distributed ontologies, links, basic ontologies, symbols in ontologies, etc.
  - always the case in RDF and OWL; Common Logic at least *allows* it
  - where basic ontologies to not use IRIs, synthesize them
- Negative consequence: names are long,
  human authors/readers **need abbreviations**
- Solution: **Namespaces**
  two alternative approaches (next slide):
  - syntactic namespaces
  - semantic namespaces

Universität Bremen

# Namespaces for Identifiers (2)

- DOL will use **syntactic namespaces** exactly like RDF and OWL.
  With `prefix:` bound to `http://iso.org/ontology#`,
  `prefix:name` expands to `http://iso.org/ontology#name`
- pure syntactic sugar, doesn't prevent semantic nonsense, e.g.:
  - binding `prefix:` to `http://iso.org/ont`, then using `prefix:ology#name`
  - declaring a symbol `http://iso.org/ontology#sym` in an ontology `http://foo.com/ontology`
  - RDF/linked data approach: conventions and best practices

Universität Bremen

# Namespaces for Identifiers (3)

Syntactic namespaces are easy – maybe too easy?

Alternative: **semantic namespaces**

- Possible DOL approach: three levels, concatenate IRIs
  - Distributed ontologies identified by IRI *d*
  - Ontologies identified by local names *o* within *d* $\leadsto$ d?o
  - Symbols identified by local names *s* within *o* $\leadsto$ d?o?s
- Downsides
  - overhead for semantics specification and application conformance
  - no existing standard to reuse
- Our pragmatic approach:
  - Stay syntactic, just don't accidentally rule out a possible future semantic namespace *extension*
  - will discuss issue with Common Logic community

Universität Bremen

# Overview of the Semantics

Three Semantics for the Basic Language:

1. **Direct Set-Theoretic Semantics**: reusing existing ontology language semantics, translations, meta level in semiformal textbook math
   - plus an **insitutional semantics** for the extended language
2. **Translational Semantics**: ontology languages and ontology language translations expressed in Common Logic, meta level still semiformal
3. **Collapsed Semantics**: ontologies, translations, and meta language in Common Logic, interpreted in Common Logic semantics

These are all compatible!

Details:

http://interop.cim3.net/file/pub/OntolOp/Publications/FOIS_2012/

# Institutional Semantics for Ext'd Language (1)

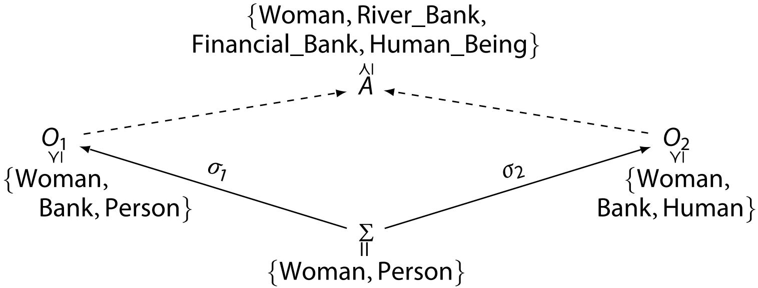
- ontology alignment and matching community works with *symbol mappings*

  (example on next slide)
- what is semantics of alignments and combinations
  - e.g.: "is this alignment a relative interpretation?"
  - e.g.: compute this combination
- logics need to be equipped with signature morphisms
  - institution theory then provides semantics
- main semantics of DOL are: direct/translational/collapsed
  - signature morphisms form an orthogonal extension

Universität Bremen

# Institutional Semantics for Ext'd Language (2)

V-alignment example:

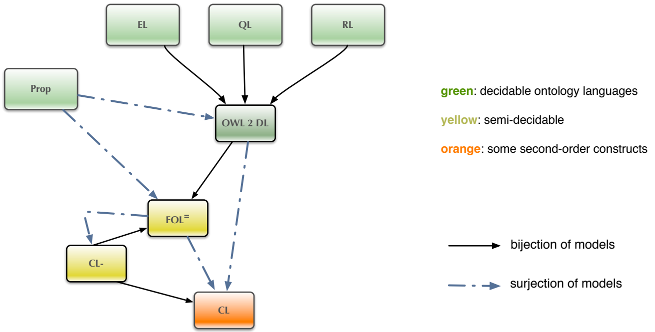



**interpretation** $\sigma_1$ : $\Sigma$ **to** $O_1$
**interpretation** $\sigma_2$ : $\Sigma$ **to** $O_2$ **with** Person $\mapsto$ Human
**ontology** A = **combine** $O_1$ $O_2$

Universität Bremen

# Ontology Languages and Translations (1): Graph

A subset of the Ontology Language and Translation Graph




- There can be multiple alternative translations (next slide)
- We want to specify a set of **composable default translations.**

Universität Bremen

# Translations with Theory Infrastructure (1)

## Example (Why are Theory Infrastructure Axioms useful?)

- CL has no “pairwise disjointness of predicates” built in
- it’s possible to define it from scratch

```
(forall (p) (mutually-disjoint p))
(forall (p q ...) (iff (mutually-disjoint p q ...)
  (and (forall (...x) (not (and (p ...x) (q ...x))))
       (mutually-disjoint p ...)
       (mutually-disjoint q ...))))
```

  - infrastructure axiomatization needs sequence markers!
  - . . . but direct translation of any concrete occurrence doesn’t
- but it would be much more convenient to simply reuse OWL’s DisjointObjectProperties/DisjointClasses!
  - but this requires an infrastructure axiom like the above to be available in the OWL→CL translation!

# Translations with Theory Infrastructure (2)

- too much junk that's not needed (see next slide)
- extra care needed to get translations right
  - interpreting DOLCE Lite in DOLCE FOL

    interpretation i:
    $\text{OWL} \rightarrow \text{CL}(\text{DOLCE}_{\text{Lite}})$ to $\text{DOLCE}_{\text{FOL}}$
  - DolceFOL does not have all of the OWL infrastructure (e.g. owl:inverseOf), so the interpretation would no longer work

Universität Bremen

# Translations with Theory Infrastructure (2)

- too much junk that's not needed (see next slide)
- extra care needed to get translations right
  - interpreting DOLCE Lite in DOLCE FOL

    interpretation i:

    OWL→CL($DOLCE_{Lite}$) to $DOLCE_{FOL}$ and $\underbrace{\text{OWL}\rightarrow\text{CL}(\{\})}_{\text{OWL infrastructure}}$
  - DolceFOL does not have all of the OWL infrastructure (e.g. owl:inverseOf), so the interpretation would no longer work
- pragmatic problems ("convenience") vs. fundamental logical problems

Universität Bremen

# Translations with Theory Infrastructure (3)

- some translations (e.g. FOL to CL) do not have infrastructure
- others (e.g. OWL to CL, see above) need it:
  1. Translation with minimal, absolutely necessary infrastructure
     - OWL individuals and datatypes both mapped to CL individuals
     - need to keep them apart (need infrastructure axiom that makes them disjoint)
     - (same for e.g. classes vs. properties vs. individuals in non-segregated dialects)
  2. Translation with further convenience infrastructure (e.g. DisjointObjectProperties/DisjointClasses)
  3. Want to have both translations – to be discussed/experimented: Which one should be default?
- Which OWL→CL to build on?
  - OWL 1 by Pat Hayes
  - OWL 1 by Chris Menzel
  - OWL 2 very rough draft by Fabian Neuhaus

Universität Bremen

# Open Tasks

- Identification of Sublogics without exploding the graph
  - OWL has useful **profiles**, specified as proper languages: EL, QL, RL
  - How about Common Logic? “Common Logic without sequence markers”, “Common Logic without quantification over predicates”, more . . . ?
- Importing ontologies formulated in a richer language into a poorer language
  - . . . i.e. specifying proper projections
  - common use case: TBox in OWL, ABox in RDF
- Approach: Resolve both issues with Common Logic community

Universität Bremen

# Use Cases

Use cases identified so far:

- Generating multilingual labels for menus in a user interface
- Connecting devices of differing complexity in an Ambient Assisted Living setting
- OWL↔FOL interpretations:
  - OWL re-formalization of the DOLCE foundational ontology → original DOLCE in FOL
  - OWL-Time → its FOL re-formalization (more comprehensive coverage of time)
  - OWL-S web service ontology → a FOL re-formalization (compare earlier SWSO/FLOWS approach)
- Metadata in COLORE (Common Logic Repository)

  details and demo below
- Extending OWL with datatypes defined in CASL

Universität Bremen

# COLORE (1): Relative Interpretations

Work in progress: modeling COLORE's metatheoretical relationships (e.g. one ontology has a relative interpretation in another ontology)

## Example (approximate_point ↔ interval_meeting)

1. in COLORE style (with the mapping in a theory of its own):
   delta = ∀ x,y . finer(x,y) ≡ starts(x,y) ∨ during(x,y) ∨ finishes(x,y)
   *interval_meeting* ∪ *delta* ⊨ *approximate_point*
2. in DOL: **interpretation** i : approximate_point **to** { interval_meeting **then** %def delta }
3. alternative, possibly more straightforward syntax? (direct interpretation, without a "mapping theory")

**interpretation** i : approximate_point **to** interval_meeting =
$finer(x, y) \mapsto starts(x, y) \vee during(x, y) \vee finishes(x, y)$
**end**

# COLORE (2): Faithful Interpretations

Second example: one ontology has a faithful interpretation in another ontology

## Example (approximate_point ↔ interval_meeting once more)

. . . is not just a relative but a *faithful* interpretation (preserves not just theorems but also satisfiability):
**interpretation** i2 : **%cons** approximate_point **to**
{ interval_meeting **then** %def delta }

Universität Bremen

# COLORE (3): Validation with Hets (Demo)

(demo given by Michael and Till)

Some Hets features:

- parsing CLIF
- access to various first-order provers (e.g. Vampire)
- access to higher-order provers (for ontologies involving sequence markers), e.g. Isabelle/HOL
- access to first-order model finders (e.g. darwin)
- verification of interpretations between CL theories (as can be found e.g. in COLORE)
- elimination of modules
- translation of OWL 2 to CL
- translation of propositional logic to CL

Universität Bremen

# OntoIOp for OOR

- OOR: Open Ontology Repository, design is work in progress
- should be OntoIOp-aware
  - Design of existing engines heavily influenced by individual logics (e.g. OWL in BioPortal)
- Bremen student project in March:
  - basic Web application that supports generic ontologies (= a bag of symbols and axioms), links, metadata, and an extensible supply of ontology languages and translations
  - features: browsing, validation via Hets web service, upload, search, maybe editing
  - detailed wish list to be clarified with OOR people; we're in touch

Universität Bremen

# Further Roadmap (as agreed on 2011-10-06)

- **2012-04-15**: Third (last) Working Draft
- **2012-06-21** to 2012-06-25 (one day): ISO/TC 37/SC 3 meeting in Madrid
- **2012-08-15**: Committee Draft
  then: 3 months review/ballot period
- What is the project team/the whole OntoIOp team expected to do?
  - in terms of work on the standard
  - administratively

Universität Bremen

# Standard Document Structure Overview

General questions:

- What should be in the **standard body**?
- What should be in a **normative annex**?
- What should be in an **informative annex**?
- What should not be in the standard, but within the "*infrastructure*" defined by the standard (compare "registry" approach of other ISO standards)?
- What should not be in the standard at all?

To be discussed for: syntax, semantics, ontology languages, translations, use cases (→ following slides)

Universität Bremen

# DOL syntaxes

Current approach:

- **Abstract syntax** in the standard body
- In normative annexes:
  - **Text syntax**: for human authors
  - **XML syntax**: for exchange with tools
  - **RDF syntax**: also for exchange with tools. . .
    - . . . but particularly in a linked data style on the Web
    - The vocabulary for describing logics and translations is a subset of the RDF syntax!
- Do we need **additional syntaxes** (via conformance criteria)?

Universität Bremen

# Semantics

Current approach:

- **Basic language**: in standard body
  - direct semantics
  - translational semantics
- **Extended language**: in standard body
  - direct semantics
  - translational semantics
- Collapsed Semantics
  - “can in principle be done” → informative annex?

Universität Bremen

# Ontology Languages

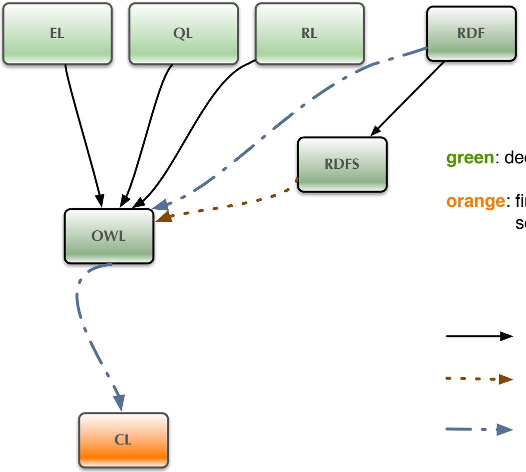
- Conformance criteria for DOL-conforming languages/logics
- Standards: RDF(S), OWL (both W3C), Common Logic (ISO)

  → normative annexes
- No standards but important (libraries, tools exist) → informative or normative annexes?
  - Propositional logic (de-facto standard library satlib)
  - First-order logic with equality (tools exist! De-facto standard syntax TPTP, IFIP standard CASL)
- No standards but reasonable applications in the DOL universe: F-logic, UML class diagrams (with one particular ontological semantics), OBO (with OWL semantics), DDL, $\mathcal{E}$-connections, Relational schemas, HOL (THF)

  → informative annexes?
- Infrastructure/registry for further/future languages?

Universität Bremen

# Translations and Defaults (1)




green: decidable ontology languages

orange: first-order with some second-order constructs

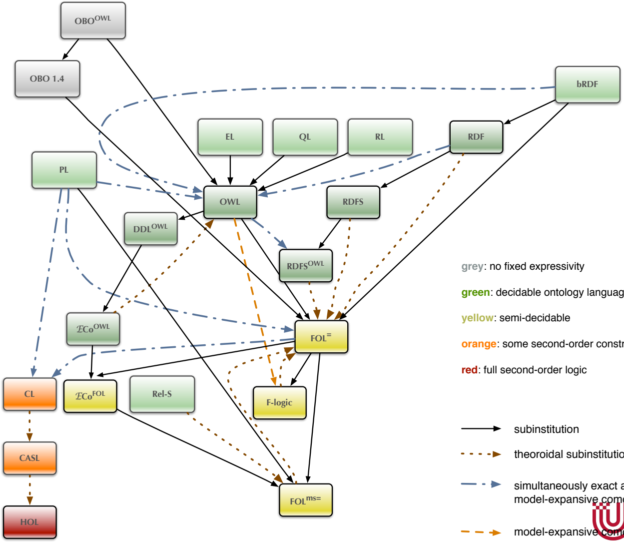
subinstitution

theoroidal subinstitution

simultaneously exact and model-expansive comorphisms

Universität Bremen

# Translations and Defaults (2)




grey: no fixed expressivity

green: decidable ontology languages

yellow: semi-decidable

orange: some second-order constructs

red: full second-order logic

subinstitution

theoroidal subinstitution

simultaneously exact and model-expansive comorphisms

model-expansive comorphisms

# Use cases

- Current approach
  - **short descriptions** of actual and potential use cases in an **informative annex**
  - classification criteria:
    - **status** (existing?, already based on DOL?)
    - DOL **features** employed
    - How does DOL **improve** the situation?
- Does this make sense, or should we rather refer to a community homepage that collects use cases?

Universität Bremen

# Conformance Criteria and Extensibility

We specify the **conformance** of the following with DOL:

- **logics** (= set-theoretical or institutional semantics)
- **serializations** . . .
  - of a conforming ontology language (XML, RDF, text, . . . )
  - for DOL (allow many concrete syntaxes!)
- **documents** ("Is this document a syntactically valid distributed ontology?")
- **applications** (produce conforming documents)

We envision a **division of labor** between:

- **core project team** (plus interested experts)
- **community** (possibly represented by some experts)

Universität Bremen

# Conformance Criteria and Extensibility

We specify the conformance of the following with DOL:

- logics
- serializations
- documents
- applications

We envision a **division of labor**:

- **project team** works on general **syntax**, **semantics**, and **conformance** criteria
- **community** populates the annexes, i.e.
  - **establish conformance** of “their favorite ontology languages”
  - **provide translations** between logics

Universität Bremen

# Organization

- Nomination of Carla Freericks
- Involvement of experts besides the project team
- Christoph Lange leaving his position

# Direct Semantics of an Ontology (1)

$$\boxed{sem(\Gamma, L, O) = (L', \Sigma, \mathcal{M})}$$

In the context of a **global environment** $\Gamma$ and the **current logic** $L$, an ontology $O$ is interpreted as a signature $\Sigma = sig(\Gamma, L, O)$ in some logic $L' = logic(\Gamma, L, O)$ and a class of models $\mathcal{M} = Mod(\Gamma, L, O)$ over that signature. We combine this into

$$sem(\Gamma, L, O) = (logic(\Gamma, L, O), sig(\Gamma, L, O), Mod(\Gamma, L, O))$$

Universität Bremen

# Direct Semantics of an Ontology (2)

| $O'$ | $sem(\Gamma, L, O') = \ldots$ |
|---|---|
| BASIC-ONTO $\langle\Sigma, \Delta\rangle$ | $(L, \Sigma, \{M \in Mod(\Sigma) \mid M \models \Delta\})$ |
| ONTO TRANSLATION<br>($O$ with logic $\rho$) | Let $\Sigma = sig(\Gamma, L, O)$ and $\rho = (\Phi, \alpha, \beta) : L_1 \to L_2$. Then<br>$logic(\Gamma, L, O') = L_2$, $sig(\Gamma, L, O') = \Phi(\Sigma)$, and<br>$Mod(\Gamma, L, O') = \{M \in Mod(\Phi(\Sigma)) \mid \beta(M) \subseteq Mod(\Gamma, L, O)\}$ |
| ONTO RESTRICTION | Let $\Sigma = sig(\Gamma, L, \mathtt{ONTO})$. Then $sem(L, \Sigma, \mathtt{RESTRICTION})$ determines a subsignature $\Sigma' \leq \Sigma$. Models are those $\Sigma'$-models that are reduct of some model in $Mod(\Gamma, L, \mathtt{ONTO})$ |
| $O_1$ and<br>CONS-STRENGTH? $O_2$ | $sig(\Gamma, L, O_1 \text{ and } O_2) = sig(\Gamma, L, O_1) \cup sig(\Gamma, L, O_2) =: \Sigma$<br>$Mod(\Gamma, L, O_1 \text{and } O_2) = Mod(\Gamma, L, O_1)\|^{\Sigma} \cap Mod(\Gamma, L, O_2)\|^{\Sigma}$ |
| ONTO then<br>CONS-STRENGTH?<br>BASIC-ONTO<br>($O$ then CS? $\langle\Sigma', \Delta'\rangle$) | Let $\Sigma = sig(\Gamma, L, O)$. Then $sig(\Gamma, L, O') = \Sigma \cup \Sigma'$<br>$Mod(\Gamma, L, O') =$<br>$\{M' \in Mod(\Sigma \cup \Sigma') \mid M' \models \Delta' \text{ and } M'\|_{\Sigma} \in Mod(\Gamma, L, O)\}$ |
| ONTO-REF | $(L, \Phi(\Sigma), \{M \in Mod(\Phi(\Sigma)) \mid \beta(M) \in \mathcal{M}\}$<br>where $\Gamma(\mathtt{ONTO\text{-}REF}) = (L_1, \Sigma, \mathcal{M})$ and $(\Phi, \alpha, \beta) : L_1 \to L$ is the default translation |
| logic LOGIC-REF : $O$ | $sem(\Gamma, \mathtt{LOGIC\text{-}REF}, O)$ |

Universität Bremen