Reliable Reference for a DL Knowledge Base Under Data Update

Reference Mechanisms

Enamul Haque (Moni)

David Toman, Grant Weddell

Problem Statement

Challenge: How long can referring expressions reliably identify entities in dynamic knowledge bases?

Motivation:

- Knowledge bases undergo frequent updates
- Referring expressions may become invalid after updates
- Need guarantees on expression durability

Visual Example:

Initial KB:
`STUDENT(rob), snum(rob)=321`

↓

After Update:
`PERSON(rob)` (no longer student)

`STUDENT` $\sqcap$ $\exists$`snum.`$\{321\}$ ✓ $\rightarrow$ `STUDENT` $\sqcap$ $\exists$`snum.`$\{321\}$ ×

Key Contributions

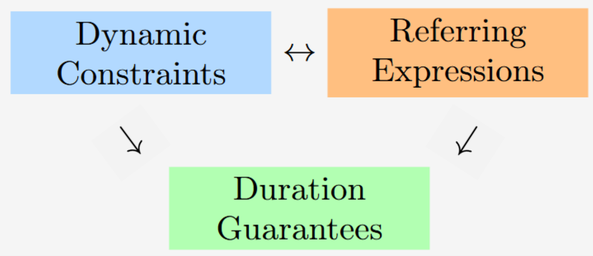
Dynamic Constraints: New constraint type (C, Pf) to preserve feature values during role transitions

ChooseRE Procedure: Algorithm to select most preferred referring expression with guaranteed duration

Durability Guarantees: Formal framework ensuring referring expressions remain valid for specified concept membership periods

Conservative Extension: Maintains logical consequence properties of original knowledge base

Constraint Relationships:



FunDL – Framework with W

A knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathcal{W}, Rt)$ consists of:

TBox $\mathcal{T}$: Subsumptions with Path Functional Dependencies

$$\text{ROOM} \sqsubseteq (\exists rnum.\top) \sqcap (\exists in.\text{BLD}) \sqcap$$
$$\text{ROOM} : rnum, in.name \rightarrow id$$
$$\text{STUDENT} \sqsubseteq \text{PERSON} \sqcap (\exists snum.\top) \sqcap$$
$$\text{STUDENT} : snum \rightarrow id$$

ABox $\mathcal{A}$: Assertions about individuals

$$\text{STUDENT}(rob),\ snum(rob) = 321,\ sin(rob) = 1234$$

WBox $\mathcal{W}$: Dynamic constraints of form (C, Pf)

$$(\text{STUDENT}, snum),\ (\text{PERSON}, sin)$$

Rt: Referring expression type defining preferences

$$\text{STUDENT} \sqcap \exists snum.\langle ? \rangle\ ;\ \text{PERSON} \sqcap \exists sin.\langle ? \rangle$$

ChooseRE – Algorithm

Goal: Compute the most preferred referring expression for individual a that can be relied upon for duration C.

Procedure: ChooseRE$(a, C, \mathcal{K})$

Requirements:

1. $\mathcal{K}$ is **durable**: $\mathcal{K} \models (\text{Con}(Rt_i), Pf)$ for all $Pf \in \text{Pfs}(Rt_i)$
2. All referring expressions in $\mathcal{L}(Rt)$ are singular w.r.t. $\mathcal{T}$

Algorithm: For $\text{Norm}(Rt) = Rt_1; \ldots; Rt_k$, return the first Rt_i such that:

1. $\mathcal{K} \models C \sqsubseteq \text{Con}(Rt_i)$ (duration compatibility)
2. $\text{ToRE}(a, Rt_i, id, \mathcal{K})$ is defined (referring expression exists)

Example Results:

$$\text{ChooseRE}(rob, \text{STUDENT}, \mathcal{K}) \rightarrow \text{STUDENT} \sqcap \exists snum.\{321\}$$
$$\text{ChooseRE}(rob, \text{PERSON}, \mathcal{K}) \rightarrow \text{PERSON} \sqcap \exists sin.\{1234\}$$

Dynamic Constraint (C, Pf) holds when:
For any structures $\mathcal{I}_1$ over $\mathcal{K}$, $\mathcal{I}_2$ over $\mathcal{K}/\mathcal{A}'$ and individuals a_1, a_2:
If $(a_1)^{\mathcal{I}_1} \in C^{\mathcal{I}_1}$, $(a_1)^{\mathcal{I}_2} \in C^{\mathcal{I}_2}$ and $Pf^{\mathcal{I}_1}((a_1)^{\mathcal{I}_1}) = (a_2)^{\mathcal{I}_1}$ then $Pf^{\mathcal{I}_2}((a_1)^{\mathcal{I}_2}) = (a_2)^{\mathcal{I}_2}$

Interpretation: If an individual maintains membership in concept C across a data update, then the path Pf must lead to the same value.

Example: Constraint $(\text{STUDENT}, snum)$ ensures that if rob remains a student after an update, his student number must remain 321.

Data Update: Replacing ABox $\mathcal{A}$ with $\mathcal{A}'$ while:

- Maintaining consistency of $\mathcal{K}/\mathcal{A}'$
- Satisfying all dynamic constraints in $\mathcal{W}$

Theoretical Results

Theorem 1 (Singularity Test): All referring expressions in $\mathcal{L}(Rt)$ are singular iff for every preference-free component Rt' of $\text{Norm}(Rt)$:

$$\mathcal{T} \models \text{Con}(Rt') \sqsubseteq \text{Con}(Rt') : \text{Pfs}(Rt') \rightarrow id$$

Theorem 2 (Conservative Extension): Dynamic constraints are conservative w.r.t. logical consequence:

$$\mathcal{K} \models \mathcal{C} \text{ iff } (\mathcal{T}, \mathcal{A}, \{\}, Rt) \models \mathcal{C}$$

Theorem 3 (Durability Guarantee): If $\mathcal{K}$ is durable, all referring expressions in $\mathcal{L}(Rt)$ are singular, and $\text{ChooseRE}(a, C_1, \mathcal{K})$ returns C_2, then:

$$C_2^{\mathcal{I}_1} = C_2^{\mathcal{I}_2} = \{(a)^{\mathcal{I}_1}\}$$

for any structure $\mathcal{I}_1$ over $\mathcal{K}$, data update $\mathcal{A}'$, and structure $\mathcal{I}_2$ over $\mathcal{K}/\mathcal{A}'$.

Inference Rules for Dynamic Constraints:

$$\frac{\mathcal{K} \models (C_1, Pf),\ \mathcal{K} \models C_2 \sqsubseteq C_1}{\mathcal{K} \models (C_2, Pf)}$$

$$\frac{\mathcal{K} \models (C_1, Pf_1),\ \mathcal{K} \models C_1 \sqsubseteq \exists Pf_1.C_2,\ \mathcal{K} \models (C_2, Pf_2)}{\mathcal{K} \models (C_1, Pf_1.Pf_2)}$$

Example

Knowledge Base Components:
TBox: Rooms have unique room numbers and building names; students have unique student numbers; persons have unique social insurance numbers.
ABox:

$$\text{ROOM}(room1),\ rnum(room1) = 5678,$$
$$in(room1) = bld1,\ name(bld1) = \text{“Davis Center”}$$
$$\text{PERSON}(rob),\ sin(rob) = 1234,$$
$$\text{STUDENT}(rob),\ snum(rob) = 321$$

Dynamic Constraints:

$$(\text{ROOM}, rnum),\ (\text{STUDENT}, snum),\ (\text{PERSON}, sin)$$

Referring Expression Type:

$$Rt = \text{ROOM} \sqcap \exists rnum.\langle ? \rangle \sqcap \exists in.name.\langle ? \rangle\ ;$$
$$\text{STUDENT} \sqcap \exists snum.\langle ? \rangle\ ;$$
$$\text{PERSON} \sqcap \exists sin.\langle ? \rangle$$

Query Results:

- For student duration: use student number
- For person duration: use social insurance number
- Room references are permanently reliable

Applications:

- Multi-level storage systems with caching
- Query answering with reliable references
- Ontology-based data access
- Document data source understanding

Key Benefits:

- Transparent query answers
- Guaranteed reliability durations
- Efficient reference computation
- Conservative extension property

Future Directions:

- Extended dynamic constraint types
- Incremental update algorithms
- Integration with stream processing
- Practical implementation studies

Contact: {enamul.haque, david, gweddell}@uwaterloo.ca **Workshop:** DL 2025, September 3–6, Opole, Poland

WATERLOO DSg Data Systems Group

UNIVERSITY OF WATERLOO