Integrating Description Logics and Logics of Motion

Bernardo Cuenca Grau[1], Bijan Parsia[1], Evren Sirin[1], Fusun Yaman[2]

Maryland Information and Network Dynamics Lab.[1]

8400 Baltimore Av.

College Park, MD, 20740 USA

Department of Computer Science[2]

A. V. Williams Building,

University of Maryland, College Park, MD 20742

bernardo@mindswap.org, bparsia@isr.umd.edu, {evren,fusun}@cs.umd.edu

Abstract

There are numerous applications that involve the movement in space and time of complex objects, defined in terms of simpler ones. These applications heavily rely on quantitative spatial and temporal information and typically involve indeterminacy about starting times, ending times, velocities and location of the moving objects.

In this paper, we present a logic that combines Description Logics (DLs) and Logics of Motion and investigate its representational adequacy in real-world scenarios as well as its formal and practical computational behavior. DLs and logics of motion nicely complement each other in applicability and expressive power. On the one hand, DLs are adequate for the description of complex concepts (unary predicates) and their relationships, but are not well-suited for the representation of knowledge on space, time or motion. On the other hand, logics of motion are effective in describing the movement of objects on the plane, but do not provide means for describing the structure and characteristics of the moving objects.

In particular, we combine the description logic $\mathcal{SHOIQ}$, which is at the basis of the Web Ontology Language (OWL), and a prominent fragment of general motion logics called "go-theories", for which optimized decision procedures have been designed and implemented. We provide a decision procedure for the combined languages and a prototype implementation.

1 Motivation

There are numerous applications that involve objects that are moving in space and time. These applications heavily rely on quantitative spatial and temporal information and typically involve indeterminacy about starting times, ending times, velocities and location of the moving objects. In [7], Yaman et al. introduced a *logic of motion* (LOM), rooted in a mix of geometry and logic, that provides a continuous model of motion based on Newtonian Physics.

There are also many applications which require the description of complex objects in terms of simpler ones. Description Logics (DLs) are well-suited for such a task and have proved successful in a wide range of applications [1].

DLs and logic of motion nicely complement each other in applicability and expressive power. On the one hand, DLs are adequate for the description of complex concepts (unary predicates) and their relationships, but are not well-suited for the representation of knowledge on space, time or motion. On the other hand, logics of motion are effective in describing the movement of objects on the plane, but do not provide means for describing the structure and characteristics of the moving objects.

A wide range of applications would benefit from the availability of systems that combine the capabilities of both families of logics. For example:

- Air traffic control is a natural application domain for the coupling. LOM can represent the flight plans for the planes, predict the locations of the planes and answer several kinds of queries of spatio-temporal interest. However, a traffic controller needs to keep track of more than just the spatio-temporal information. For example, type of a plane (passenger or cargo), capacity of a plane, maintenance history of a plane are some of the critical information for the air traffic management. Such information can be easily represented using DLs.
- Military task planning and monitoring is another application that would benefit the coupling of DLs and LOM. One of the important notions for a military operation is the distinction between enemy and ally. Similarly differentiating between threat and safety is an important issue. We can represent such important terminology using DLs. In a military mission, the uncertainty over the movements of the vehicles is enormous and LOM is expressive enough for representing such uncertain movements.

The following are some of the queries that are interesting for these two application domains. We highlight in **bold** and *italic* the notions that are best represented using DLs and LOM respectively.

- **Q1**: What are all the **Boeing 747 passenger airplanes** with **low fuel** within a given *rectangular region centered on JFK* airport *now*?
- **Q2**: Is there any unsafe flight *in a given rectangular region now*? A flight is unsafe if either it **has not been maintained for more than 2 years** or if there is another **plane** *within 10 miles* of it.
- **Q3**: Is there now any **vehicle** likely to break the international aviation laws? By breaking the international laws we mean any **non-passenger airplane not belonging to NATO forces** that is *located now within 10 miles of a certain exclusion area R.*
- **Q4**: Are there any vulnerable ships in the gulf now? A **ship** is vulnerable if it has **little fire power** and there are less than 2 **ships with high fire power** in *1 mile radius.*
- **Q5**: Which **tanks** might be threatened by an **enemy vehicle** *o within the next 10 minutes*? By "threatened" we mean the enemy vehicle be either an **enemy tank** *within 1 mile* of *o* or an **enemy helicopter equipped with long-range missiles** *within 30 miles* of *o*.

In this paper, we combine DLs and logics of motion and investigate the representational adequacy in real-world scenarios and the formal and practical computational behavior of the combination. In particular, we couple the description logic $\mathcal{SHOIQ}$ [4], which is at the basis of the Web Ontology Language (OWL)[6], and a prominent fragment of general motion logics called "go-theories" [7] [8], for which optimized decision procedures have been designed and implemented.

2 Preliminaries

In this section, we describe the syntax and semantics of the Description Logic $\mathcal{SHOIQ}$ and recall the basics of go-theories. A more detailed discussion can be found in [4] and [7] respectively.

2.1 The Description Logic $\mathcal{SHOIQ}$

Let N_C, N_R, N_I be non-empty and pair-wise disjoint sets of *atomic concepts*, *atomic roles* and *individuals* respectively. The set of $\mathcal{SHOIQ}$ roles (roles, for short) is the set $N_R \cup \{R^- | R \in N_R\}$, where R^- denotes the inverse of the atomic role R. Concepts are inductively using the following grammar:

$$C \leftarrow A | \neg C | C_1 \sqcap C_2 | C_1 \sqcup C_2 | \exists R.C | \forall R.C | \bowtie nS.C | \{a\}$$

where $A \in N_C$, $a \in N_I$, $C_{(i)}$ a $\mathcal{SHOIQ}$ concept, R a role, S a *simple* role [1] and $\bowtie \in \{\leq, \geq\}$. We write $\top$ and $\bot$ to abbreviate $C \sqcup \neg C$ and $C \sqcap \neg C$ respectively.

A *role inclusion axiom* is an expression of the form $R_1 \sqsubseteq R_2$, where R_1, R_2 are roles. A *transitivity axiom* is an expression of the form $Trans(R)$, where $R \in V_R$. An RBox R is a finite set of role inclusion axioms and transitivity axioms. For C, D concepts, a *concept inclusion axiom* is an expression of the form $C \sqsubseteq D$. A TBox T is a finite set of concept inclusion axioms. A *knowledge base* $\mathsf{K} = (\mathsf{T}, \mathsf{R})$ consists of a TBox and an RBox.

An *interpretation* $\mathcal{I}$ is a pair $\mathcal{I} = (\mathcal{W}, .^{\mathcal{I}})$, where $\mathcal{W}$ is a non-empty set, called the *domain* of the interpretation, and $.^{\mathcal{I}}$ is the *interpretation function*. The interpretation function assigns to $A \in N_C$ a subset of $\mathcal{W}$, to each $R \in N_R$ a subset of of $\mathcal{W} \times \mathcal{W}$ and to each $a \in N_I$ an element of $\mathcal{W}$. The interpretation function is extended to complex roles and concepts as given in [4].

The satisfaction of a $\mathcal{SHOIQ}$ axiom α in an interpretation $\mathcal{I}$, denoted $\mathcal{I} \models \alpha$ is defined as follows: (1) $\mathcal{I} \models R_1 \sqsubseteq R_2$ iff $(R_1)^{\mathcal{I}} \subseteq (R_2)^{\mathcal{I}}$; (2) $\mathcal{I} \models Trans(R)$ iff for every $a, b, c \in \mathcal{W}$, if $(a, b) \in R^{\mathcal{I}}$ and $(b, c) \in R^{\mathcal{I}}$, then $(a, c) \in R^{\mathcal{I}}$; (3) $\mathcal{I} \models C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; The interpretation $\mathcal{I}$ is a model of the RBox R (respectively of the TBox T) if it satisfies all the axioms in R (respectively T). $\mathcal{I}$ is a model of $\mathsf{K} = (\mathsf{T}, \mathsf{R})$, denoted by $\mathcal{I} \models \mathsf{K}$, iff $\mathcal{I}$ is a model of T and R.

The interesting inferences in $\mathcal{SHOIQ}$ are the following: The concept C is *satisfiable relative to* K iff there is a model $\mathcal{I}$ of K such that $C^{\mathcal{I}} \neq \emptyset$. We say that C *subsumes* D *relative to* K iff, for every model $\mathcal{I}$ of K, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

These reasoning services are inter-definable. Subsumption can be reduced to concept satisfiability. Through a technique called *internalization*, we can reduce concept satisfiability w.r.t. a TBox and an RBox to concept satisfiability w.r.t. an RBox only.

2.2 Logic of Motion

Let $\Re$ be the set of all real numbers and V_R be the set of variables ranging over $\Re$. Also $\mathbf{O}$ is the finite set of object names and $V_{\mathbf{O}}$ is the set of variables ranging over $\mathbf{O}$. A real-term t is any member of $\Re \cup V_{\mathbf{R}}$. Object terms are defined similarly. We now define the **LOM** atoms as follows.

- If o_1, o_2 are object terms, and d, t_1, t_2 are positive real terms, then $\mathsf{near}(o_1, o_2, d, t_1, t_2)$ is an *atom*. Intuitively, this atom says that o_1, o_2 are within distance d of each other throughout the time interval $[t_1, t_2]$.

[1]See [4] for a precise definition of simple roles

- If o_1, o_2 are object terms, and d, t_1, t_2 are positive real terms, then $\mathsf{far}(o_1, o_2, d, t_1, t_2)$ is an *atom*. This atom says that the distance between o_1 and o_2 is more than d throughout the time interval $[t_1, t_2]$.
- If o is an object term, x_1, y_1, x_2, y_2 are real terms, and t_1, t_2 are positive real terms, then $\mathsf{in}(o, x_1, y_1, x_2, y_2, t_1, t_2)$ is an *atom*. This atom says that object o is in the rectangle whose lower left (resp. upper right) corner is (x_1, y_1) (resp. (x_2, y_2)) some time in the interval $[t_1, t_2]$.
- If o is an object term, x_1, y_1, x_2, y_2 are real terms, and $t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+$ are positive real terms, then $\mathsf{go}(o, x_1, y_1, x_2, y_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ is an atom called a *go atom*. This atom states that object o leaves point (x_1, y_1) at some time in $[t_1^-, t_1^+]$ and arrives at point (x_2, y_2) during $[t_2^-, t_2^+]$, traveling on a straight line and speed is between v^- and v^+.

A **LOM** *interpretation* is a continuous function $\mathcal{I} : \mathbf{O} \times \Re^+ \rightarrow \Re \times \Re$. Intuitively, $\mathcal{I}(o, t)$ is o's location at time t. The interpretation $\mathcal{I}$ satisfies a a ground go-atom $g = \mathsf{go}(o, x_1, y_1, x_2, y_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ w.r.t. a time interval $T = [t_1, t_2]$ iff:

- $t_1 \in [t_1^-, t_1^+]$ and $\mathcal{I}(o, t_1) = (x_1, y_1)$ and $t_2 \in [t_2^-, t_2^+]$ and $\mathcal{I}(o, t_2) = (x_2, y_2)$
- $\forall t \in [t_1, t_2],\ \mathcal{I}(o, t)$ is on the line segment $[(x_1, y_1), (x_2, y_2)]$
- $\forall t, t' \in [t_1, t_2],\ t < t'$ implies $\mathsf{dist}(\mathcal{I}(o, t), (x_1, y_1)) < \mathsf{dist}(\mathcal{I}(o, t'), (x_1, y_1))$, where dist is the function that computes the Euclidean distance between two points.
- For all but finitely many times in $[t_1, t_2]$, $v = d(|\mathcal{I}(o, t)|)/dt$ is defined and $v^-(g) \leq v \leq v^+(g)$.

Satisfaction of a *ground* **LOM** atom α by a LOM-interpretation $\mathcal{I}$ is defined as follows:

1. If $\alpha = \mathsf{go}(o, x_1, y_1, x_2, y_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ then $\mathcal{I} \models \alpha$ iff there exists an interval $[t_1, t_2]$ such that $\mathcal{I}$ satisfies α over $[t_1, t_2]$.
2. If $\alpha = \mathsf{near}(o_1, o_2, d, t_1, t_2)$, then $\mathcal{I} \models \alpha$ iff $\mathsf{dist}(\mathcal{I}(o_1, t), \mathcal{I}(o_2, t)) \leq d$ for every $t_1 \leq t \leq t_2$,
3. If $\alpha = \mathsf{far}(o_1, o_2, d, t_1, t_2)$, then $\mathcal{I} \models \alpha$ iff $\mathsf{dist}(\mathcal{I}(o_1, t), \mathcal{I}(o_2, t)) > d$ for every $t_1 \leq t \leq t_2$,
4. If $\alpha = \mathsf{in}(o, x_1, y_1, x_2, y_2, t_1, t_2)$, then $\mathcal{I} \models \alpha$ iff there are reals $t \in [t_1, t_2]$, $x \in [x_1, x_2]$ and $y \in [y_1, y_2]$ such that $\mathcal{I}(o, t) = (x, y)$.

$\mathcal{I}$ *satisfies (or is a model of)* a set of ground atoms MT iff $\mathcal{I}$ satisfies every $\alpha \in \mathsf{MT}$. MT is *consistent* iff there is an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathsf{MT}$. A **LOM** atom α is a *logical consequence* of MT, denoted $\mathsf{MT} \models \alpha$, iff every model of MT is also a model of α.

3 A Combined Representation

In this section, we introduce a logic $\mathcal{SHOIQ}[\mathbf{LOM}]$, which combines and extends the expressive power of the DL $\mathcal{SHOIQ}$ and the Logic of Motion. Informally, the coupling is performed by enriching the DL with extra constructors for building complex DL concepts in terms of *queries* to the motion theory. Roughly, these queries retrieve sets of objects whose movement verifies certain specific conditions. Thus, the additional operators integrated into the concept language allow to define concepts attending to the movement of their instances.

The queries to the motion theory are treated in a way such that the LOM reasoner draws inferences separately from the rest of the system. This increases flexibility and facilitates the implementation of a combined reasoner, since the internals of the LOM checker need not be modified. The encapsulating view of the motion theory makes it possible to easily design decidable combinations of DLs and different logics of motion. The combination we present here is based on go-theories, but the same methodology could be used for integrating DLs with general motion theories. The reason for restricting ourselves to go-theories in this paper is merely the absence of a decision procedure for query answering in general motion theories. [2] In what follows, we formally define the syntax and semantics of the combination.

Let N_C and N_I be sets of atomic concepts and object names respectively and R a $\mathcal{SHOIQ}$ RBox. Let $\Re$ be the set of all real numbers and V_R a set of variables ranging over $\Re$ and disjoint with N_I and N_C. The set of $\mathcal{SHOIQ}[\mathbf{LOM}]$ concepts is given by the following grammar:

$$C \leftarrow C_1 \sqcap C_2 | C_1 \sqcup C_2 | \neg C | \{o\} | \exists R.D | \forall R.D | \bowtie nS.D | \\ \exists \mathbf{Q}_d^{t_1,t_2}.D | \forall \mathbf{Q}_d^{t_1,t_2}.D | \bowtie n\mathbf{Q}_d^{t_1,t_2}.D | \exists \mathbf{P}^{t_1,t_2}_{x_1,y_1,x_2,y_2}$$

With $d, x_1, x_2, y_1, y_2, t_1, t_2 \in \Re \cup V_R$ and $\mathbf{Q}, \mathbf{P}$ defined respectively by the following simple grammars: $\mathbf{Q} \leftarrow \mathbf{near} | \mathbf{far} | \neg \mathbf{Q}$ and $\mathbf{P} \leftarrow \mathbf{in} | \neg \mathbf{in}$.

The set of $\mathcal{SHOIQ}[\mathbf{LOM}]$ formula is an expression of one of the following forms: (1) A **LOM** ground go-atom $go(o, x_1, y_1, x_2, y_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$, with $o \in N_I$ and $x_1, y_1, x_2, y_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+ \in \Re$; (2) A GCI $C \sqsubseteq D$, with C, D $\mathcal{SHOIQ}[\mathbf{LOM}]$ concepts; (3) any axiom in the RBox R.

A knowledge base $(\mathsf{K}, \mathsf{MT})$ is a finite set of $\mathcal{SHOIQ}[\mathbf{LOM}]$ formulas, where MT is a finite set of ground go-atoms and K a finite set of GCIs and RBox axioms. Let MT be a consistent go-theory. An interpretation $\mathcal{I} = (\mathcal{W}, .^{\mathcal{I}})$ is a $\mathcal{SHOIQ}$ interpretation relative to MT if the interpretation function $.^{\mathcal{I}}$ extends to the new concepts as follows:

[2] Even the decidability of the consistency problem for general motion theories remains an open question.

- $(\exists \mathbf{Q}_d^{t_1,t_2}.D)^\mathcal{I} = \{x \in \mathcal{W} | \exists y \in \mathcal{W} \text{ s.t. } y \neq x, \mathsf{MT} \models Q(x, y, d, t_1, t_2) \text{ for } [t_1, t_2] \text{ and } y \in D^\mathcal{I}\}$
- $(\forall \mathbf{Q}_d^{t_1,t_2}.D)^\mathcal{I} = \{x \in \mathcal{W} | \forall y \in \mathcal{W}, \text{ if } \mathsf{MT} \models Q(x, y, d, t_1, t_2) \text{ for } [t_1, t_2] \text{ and } y \neq x, \text{ then } y \in D^\mathcal{I}\}$
- $(\geq n\mathbf{Q}_d^{t_1,t_2}.D)^\mathcal{I} = \{x \in \mathcal{W} | ||\{y \in \mathcal{W} | \mathsf{MT} \models Q(x, y, d, t_1, t_2) \text{ for } [t_1, t_2], y \neq x \text{ and } y \in D^\mathcal{I}\}|| \geq n\}$
- $(\leq n\mathbf{Q}_d^{t_1,t_2}.D)^\mathcal{I} = \{x \in \mathcal{W} | ||\{y \in \mathcal{W} | \mathsf{MT} \models Q(x, y, d, t_1, t_2) \text{ for } [t_1, t_2], y \neq x \text{ and } y \in D^\mathcal{I}\}|| \leq n\}$
- $(\exists \mathbf{P}_{x_1,y_1,x_2,y_2}^{t_1,t_2})^\mathcal{I} = \{o \in \mathcal{W} | \mathsf{MT} \models Q'(o, x_1, y_1, x_2, y_2, t_1, t_2) \text{ for } [t_1, t_2]\}$

A $\mathcal{SHOIQ}$ interpretation $\mathcal{I} = (\mathcal{W}, .^\mathcal{I})$ relative to MT satisfies RBox axioms and GCIs as usual and it satisfies a knowledge base $(\mathsf{K}, \mathsf{MT})$ if MT is consistent and $\mathcal{I}$ satisfies every axiom in K. Note that the queries embedded in the new concept constructors coincide with the query language for go-theories. Although near(), in() and far() atoms, as well as their negation, are not allowed in the language of go-theories, it is possible to use them in the *query language* and suitable algorithms for answering these queries have been designed and implemented [7][8]. As an example, consider the concepts:

$$Airplane \sqcap \exists \mathbf{near}_{\mathbf{10}}^{\mathbf{1,2}}.Airplane; \qquad Tank \sqcap \exists \mathbf{in}_{1,2,10,20}^{1,2}$$

which represent, respectively, the airplanes that have been closer than 10 miles to another airplane during the *whole* period between 1am and 2am, and the set of tanks that have entered the rectangular region defined by the points $(1, 2)$ and $(10, 20)$ at some time point between 1am and 2am in the night.

4 Decidability of $\mathcal{SHOIQ}$[LOM]

In this section, we provide a decision procedure for concept satisfiability in $\mathcal{SHOIQ}$[**LOM**]. As it is usual in DLs, we will be able to reduce other interesting problems, such as subsumption and query answering, to concept satisfiability. The decision procedure in this section extends the one for $\mathcal{SHOIQ}$ concept satisfiability, recently presented in [4]. As mentioned in Section 3, the LOM reasoner can be treated as an oracle and hence LOM algorithms do not need to be modified. However, standard DL tableau algorithms need to be extended to cope with the new constructs.

4.1 A Tableau for $\mathcal{SHOIQ}$[LOM]

Before describing the algorithm in detail, we need some preliminary notions. We say that a concept is in *Negation Normal Form* (NNF) if negation occurs only in front of atomic concepts, nominals or restrictions of the form

$\exists \mathbf{P}^{t_1,t_2}_{x_1,y_1,x_2,y_2}$. NNF can be computed using the usual rewrite rules (see [4]) plus the following ones:

$$\neg\exists \mathbf{Q}^{t_1,t_2}_d.D \equiv \forall \mathbf{Q}^{t_1,t_2}_d.\neg D \qquad \neg\forall \mathbf{Q}^{t_1,t_2}_d.D \equiv \exists \mathbf{Q}^{t_1,t_2}_d.\neg D$$
$$\neg \leq n\mathbf{Q}^{t_1,t_2}_d.C \equiv \geq (n+1)\mathbf{Q}^{t_1,t_2}_d.C \qquad \neg \geq (n+1)\mathbf{Q}^{t_1,t_2}_d.C \equiv \leq n\mathbf{Q}^{t_1,t_2}_d.C$$
$$\neg(\geq 0\mathbf{Q}^{t_1,t_2}_d.C) \equiv \bot$$

We will assume that the operator $\sim C$ maps every concept C to the NNF of $\neg C$. For a concept D and an RBox R, the set of "relevant" sub-concepts $cl(D, \mathsf{R})$ is defined as in [4]. For establishing the soundness and completeness of a tableau algorithm for expressive DLs, it is a common practice to define a more convenient representation of a model, called a *tableau*.

Definition 1 *(*Tableau*)*
Let MT *be a Go-Theory,* R *a* $\mathcal{SHOIQ}$ *RBox and* D *a* $\mathcal{SHOIQ}[\mathbf{LOM}]$ *concept in NNF. A tableau* $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ *for* D *w.r.t.* R *and* MT *is a triple such that:* $\mathbf{S}$ *is a set of individuals,* $\mathcal{L} : \mathbf{S} \rightarrow 2^{cl(D,\mathsf{R})}$ *maps each individual to a set of concepts which is a subset of* $cl(D, \mathsf{R})$*,* $\mathcal{E} : R_D \rightarrow 2^{\mathbf{S}\times\mathbf{S}}$ *maps each role in* R_D *to a set of pairs of individuals and there is some individual* $s \in \mathbf{S}$ *s.t.* $D \in \mathcal{L}(s)$*. Then,* T *verifies the same conditions as a* $\mathcal{SHOIQ}$ *tableau (see [4]), plus the following ones:*

- ***P0****: If* $\exists \mathbf{P}^{t_1,t_2}_{x_1,y_1,x_2,y_2} \in \mathcal{L}(s)$*, then there exists an* $a \in N_I$*, with* $\{a\} \in \mathcal{L}(s)$ *and* $\mathsf{MT} \models in(a, x_1, y_2, x_2, y_2, t_1, t_2)$*.*
- ***P1****: If* $\exists \mathbf{Q}^{t_1,t_2}_d.C \in \mathcal{L}(s)$*, then there exists a* $t \in \mathbf{S}$ *with* $t \neq s$ *and object names* $a, b \in N_I$ *s.t.* $\{a\} \in \mathcal{L}(s)$*,* $\{b, C\} \subseteq \mathcal{L}(t)$ *and* $\mathsf{MT} \models Q(a, b, d, t_1, t_2)$*.*
- ***P2****: If* $\forall \mathbf{Q}^{t_1,t_2}_d.C \in \mathcal{L}(s)$ *and* $\mathsf{MT} \models Q(a, b, d, t_1, t_2)$ *with* $\{a\} \in \mathcal{L}(s)$*,* $\{b\} \in \mathcal{L}(t)$ *and* $t \neq s$*, then* $C \in \mathcal{L}(t)$
- ***P3****: If* $\geq n\mathbf{Q}^{t_1,t_2}_d.C \in \mathcal{L}(s)$*, then there exists an* $a \in N_I$ *with* $\{a\} \in \mathcal{L}(s)$ *and* n *objects* $u_1, ..., u_n \in \mathbf{S}$ *with* $u_i \neq s$ *s.t.* $\{\{b_i\}, C\} \subseteq \mathcal{L}(u_i)$*, for* $b_i \in N_I$*, and* $\mathsf{MT} \models Q(a, b_i, d, t_1, t_2)$*, for every* $1 \leq i \leq n$*.*
- ***P4****: If* $\leq n\mathbf{Q}^{t_1,t_2}_d.C \in \mathcal{L}(s)$ *and there exists an* $a \in N_I$ *with* $\{a\} \in \mathcal{L}(s)$*, then there are at most* n *objects* $u_1, ..., u_n \in \mathbf{S}$ *with* $u_i \neq s$ *such that* $\{\{b_i\}, C\} \subseteq \mathcal{L}(u_i)$*, for* $b_i \in N_I$*, and* $\mathsf{MT} \models Q(a, b_i, d, t_1, t_2)$*, for every* $1 \leq i \leq n$*.*
- ***P5****:If* $(\leq n\mathbf{Q}^{t_1,t_2}_d.C) \in \mathcal{L}(s)$ *and* $\mathsf{MT} \models Q(a, b, d, t_1, t_2)$ *with* $\{a\} \in \mathcal{L}(s)$*,* $\{b\} \in \mathcal{L}(t)$ *and* $s \neq t$ *then* $\{C, \neg C\} \cap \mathcal{L}(t) \neq \emptyset$

Lemma 1 *A* $\mathcal{SHOIQ}[\mathbf{LOM}]$ *concept* D *in NNF is satisfiable w.r.t. a role hierarchy* R *and a motion theory* MT *iff* D *has a tableau w.r.t.* R *and* MT*.*

4.2 A Tableau Algorithm for $\mathcal{SHOIQ}$[LOM]

From the lemma above, an algorithm that constructs a tableau for a $\mathcal{SHOIQ}$[**LOM**] concept D can be used as a decision procedure for the satisfiability of D with respect to a RBox R and a go-theory MT.

The tableau algorithm for $\mathcal{SHOIQ}$[**LOM**] concepts with respect to an RBox and a go-theory is an extension of the known tableau algorithm for $\mathcal{SHOIQ}$ concepts w.r.t. RBoxes. DL tableaux-based algorithms decide consistency by trying to construct an abstraction of a model, called a *completion graph* by applying a set of *expansion rules*. Termination is ensured by using a special cycle detection technique called *blocking* and inconsistencies are reported when a *clash* (a contradiction) is detected in the completion graph. Completion graphs as well as blocking are defined identically to the case of $\mathcal{SHOIQ}$. The differences enforced by the coupling occur in the initialization of the algorithm, in the expansion rules and in the definition of a clash.

As is $\mathcal{SHOIQ}$, a completion graph for D w.r.t. R and MT is a labeled directed graph $\mathcal{G} = (V, E, \mathbf{L}, \dot{\neq})$, where V is a set of nodes, E a set of edges, $\mathbf{L}$ a labeling function for nodes and edges and $\dot{\neq}$ a binary relation for indicating which nodes cannot be merged by the application of an expansion rule. Let $o_1, ..., o_l$ be all the object names occurring in MT and $a_1, ...a_p$ all the nominals occurring in the input $\mathcal{SHOIQ}$[**LOM**] concept D and not in MT, then the tableau algorithm starts with the completion graph $\mathcal{G} = (\{v_0, ..., v_l, w_1, ..., w_p\}, \emptyset, \mathbf{L}, \dot{\neq})$ with $\mathbf{L}(v_0) = \{D\}$, $\mathbf{L}(v_i) = \{\{o_i\}\}$ with $v_i \dot{\neq} v_j$ for $1 \leq i < j \leq l$ and $\mathbf{L}(w_i) = \{\{a_i\}\}$ for $1 \leq i \leq p$. Then, $\mathcal{G}$ is expanded by repeatedly applying the $\mathcal{SHOIQ}$ expansion rules [4] plus the additional rules in Figure 1, stopping if a clash occurs. As an example, suppose that we are checking the consistency of the concept:

$$Airplane \sqcap \exists \mathbf{near^{1,2}_{10}}.Airplane$$

with respect to some go-theory. In the tableaux expansion, we would generate a root node x that will contain, after applying the conjunction rule, the concept $\exists \mathbf{near^{1,2}_{10}}.Airplane$ in its label. Then, the combined reasoner would call the LOM oracle to retrieve the pairs of objects that belong to the answer set of the query $near(?z, ?w, 10, 1, 2)$, non-deterministically select one of the retrieved pairs, say (a, b), add the nominal $\{a\}$ to the label of x and add the concept *Airplane* to the label of the nominal node corresponding to the individual b in the completion graph. In case the answer (a, b) led to a clash, the algorithm would backtrack, non-deterministically select a different answer to the query and repeat the process. In case no answer had been found for the query in the first place, a clash would have been immediately reported. Note that the semantics of the combination combination requires that all the individuals that occur in the go-theory *must* exist as nominal nodes in the completion graph and that for those individuals the Unique Name Assumption applies.

Interestingly, in order to handle correctly cardinality restrictions on LOM queries, the $\rightarrow choose_Q$ rule is required. The reason is similar to why an analogous rule is required for ensuring soundness in the presence of DL cardinality restrictions. Without the $\rightarrow choose_Q$ the unsatisfiability of the following concept would not be detected:

$$\leq 1\mathbf{near^{1,2}_{10}}.Airplane \sqcap \leq 1\mathbf{near^{1,2}_{10}}.\neg Airplane \sqcap \geq 3\mathbf{near^{1,2}_{10}}.\top$$

Also note that the additional expansion rules require the ability to answer in(), near() and far() (possibly negated) queries over a go-theory. In [7] [8] suitable algorithms were devised for such a purpose. Basically, in $\mathcal{SHOIQ}$[**LOM**], the reasoner over the motion theory can be treated as an "oracle", that, whenever required, will execute queries and provide the answer to the tableau reasoner. Thus, the algorithm presented in this section can be implemented by modifying the DL reasoner only.

Finally, in [4], the expansion rules are applied with different priorities in order to guarantee termination. The strategy basically consists on applying the shrinking rules before any other rules and to apply them to "lower level" nominal nodes before applying them to "higher level" nodes. Note that the additional rules we have introduced can *only* result, except for the $\rightarrow\leq_Q$ rule, in the addition of concepts to the label of a node in the expansion graph and hence should be applied with the lowest priority. The $\rightarrow\leq_Q$ rule must be applied with the same priority as the "ordinary" $\rightarrow\leq$ rule.

We now establish the specific conditions for a clash (i.e. a contradiction) to occur. A completion graph $\mathcal{G}$ is said to contain a *clash* if, for some node x of $\mathcal{G}$ either of the following conditions hold: *1)* $\{A, \neg A\} \subseteq \mathbf{L}(x)$ with $A \in N_C$; *2)* $\leq nS.C \in \mathcal{L}(x)$ and there are $n+1$ S-neighbors $y_0, ..., y_n$ of x with $C \in \mathbf{L}(y_i)$ for each $0 \leq i \leq n$ and $y_i \dot{\neq} y_j$ for each $0 \leq i < j \leq n$; *3)* for some $o \in N_I$ there is a node y with $x \dot{\neq} y$ s.t. $o \in \mathbf{L}(x) \cap \mathbf{L}(y)$; *4)* $\exists \mathbf{P}^{t_1,t_2}_{x_1,y_1,x_2,y_2} \in \mathbf{L}(x)$ and there is no $a \in N_I$ s.t. $\mathsf{MT} \models in(a, x_1, y_1, x_2, y_2)$; *5)* $\exists \mathbf{Q}^{t_1,t_2}_d.D \in \mathbf{L}(x)$ and there are no $a, b \in N_I$ s.t. $\mathsf{MT} \models Q(a, b, d, t_1, t_2)$; *6)* $\geq n\mathbf{Q}^{t_1,t_2}_d.D \in \mathbf{L}(x)$ and there are no n pairs (a, b) with $a, b \in N_I$ s.t. $\mathsf{MT} \models Q(a, b, d, t_1, t_2)$; *7)* $\{\{a\}, \leq n\mathbf{Q}^{t_1,t_2}_d.D\} \subseteq \mathbf{L}(x)$, $\mathsf{MT} \models Q(x, a, b_i, t_1, t_2)$ for $0 \leq i \leq n$ and there are n+1 nominal nodes $y_0, ..., y_n$ in $\mathcal{G}$ s.t. , $\{D, \{b_i\}\} \subseteq \mathbf{L}(y_i)$ and $y_i \dot{\neq} y_j$ for each $0 \leq i < j \leq n$.

Finally, as in the case of $\mathcal{SHOIQ}$, we say that a completion graph is *complete* either if it contains a clash, or when none of the rules is applicable. If the expansion rules can be applied to D, R, MT in such a way that they yield

- $\rightarrow in$ **rule**: Let $\exists \mathbf{P}^{t_1,t_2}_{x_1,y_1,x_2,y_2} \in \mathbf{L}(x)$, x not blocked and $\{a_1,...,a_n\}$ with $a_i \in V_I$ for $1 \leq i \leq n$ be the set of objects s.t. $\mathsf{MT} \models in(a_i, x_1, y_1, x_2, y_2, t_1, t_2)$ for every a_i. Let $\{\{a_1\},...,\{a_n\}\} \cap \mathbf{L}(x) = \emptyset$,
  Then $\mathbf{L}(x) \leftarrow \mathbf{L}(x) \cup \{a_i\}$ for some $\{a_i\} \in \{\{a_1\},...,\{a_n\}\}$

- $\rightarrow \exists_Q$ **rule** Let $\exists \mathbf{Q}^{t_1,t_2}_d.D \in \mathbf{L}(x)$, x not blocked and $\{(a_1,b_1)...,(a_n,b_n)\}$ with $a_i, b_i \in V_I$ for $1 \leq i \leq n$ be the set of pairs of objects s.t. $\mathsf{MT} \models Q(a_i, b_i, d, t_1, t_2)$ for every (a_i, b_i). If there is no pair (a_i, b_i) and nominal node y s.t. $\{a_i\} \in \mathbf{L}(x)$, $D \in \mathbf{L}(y)$ and $\{b_i\} \in \mathbf{L}(y)$,
  Then for some pair (a_i, b_i), do
  $\mathbf{L}(x) \leftarrow \mathbf{L}(x) \cup \{a_i\}$, if $\{a_i\} \notin \mathbf{L}(x)$ and
  $\mathbf{L}(y) \leftarrow \mathbf{L}(y) \cup \{D\}$, if $D \notin \mathbf{L}(y)$, where y is a nominal node with $\{b_i\} \in \mathbf{L}(y)$

- $\rightarrow \forall_Q$ **rule**: Let $\forall \mathbf{Q}^{t_1,t_2}_d.D \in \mathbf{L}(x)$, x not indirectly blocked, $\{a\} \in \mathbf{L}(x)$, $\{b\} \in \mathbf{L}(y)$ and $\mathsf{MT} \models Q(a, b, d, t_1, t_2)$
  Then $\mathbf{L}(y) \leftarrow \mathbf{L}(y) \cup \{D\}$

- $\rightarrow \geq_Q$ **rule**: Let $\geq n\mathbf{Q}^{t_1,t_2}_d.D \in \mathbf{L}(x)$, x not blocked and $\{(a_1,b_1)...,(a_n,b_n)\}$ with $a_i, b_i \in V_I$ for $1 \leq i \leq n$ be the set of pairs of objects s.t. $\mathsf{MT} \models Q(a_i, b_i, d, t_1, t_2)$ for every (a_i, b_i). If there are no n pairs (a_i, b_i) and distinct nominal nodes y_i s.t. $\{a_i\} \in \mathbf{L}(x)$ $\{b_i\} \in \mathbf{L}(y_i)$ and $D \in \mathbf{L}(y)$
  Then $\mathbf{L}(x) \leftarrow \mathbf{L}(x) \cup \{a_i\}$, if $\{a_i\} \notin \mathbf{L}(x)$ and
  $\mathbf{L}(y_i) \leftarrow \mathbf{L}(y_i) \cup \{D\}$, if $D \notin \mathbf{L}(y_i)$, where y_i is a nominal node with $\{b_i\} \in \mathbf{L}(y_i)$
  for some n pairs (a_i, b_i)

- $\rightarrow choose_Q$ rule: if $(\leq n\mathbf{Q}^{t_1,t_2}_d.C) \in \mathbf{L}(x)$, x is not indirectly blocked, $\{a\} \in \mathbf{L}(x)$ and there is a nominal node y with $\mathsf{MT} \models Q(a, b, d, t_1, t_2)$, $\{b\} \in \mathbf{L}(y)$ and $\{C, \sim C\} \cap \mathbf{L}(y) = \emptyset$, then:
  $\mathbf{L}(y) \leftarrow \mathbf{L}(y) \cup \{E\}$ for some $E \in \{C, \sim C\}$

- $\rightarrow \leq_Q$ rule: if $(\leq n\mathbf{Q}^{t_1,t_2}_d.C) \in \mathbf{L}(z)$, z is not indirectly blocked, $\{b\} \in \mathbf{L}(z)$ and there are $n+1$ nominal nodes $y_0, ..., y_n$ with $\mathsf{MT} \models Q(b, a_i, d, t_1, t_2)$ and $\{C, a_i\} \in \mathbf{L}(y_i)$ for $0 \leq i \leq n$ and there are two of them y_j, y_k s.t. not $y_j \dot{\neq} y_k$, then $Merge(y_j, y_k)$

Figure 1: Additional Expansion Rules

to a complete, clash-free completion graph, then the algorithm decides that D is satisfiable w.r.t. R and unsatisfiable otherwise. Termination, soundness and completeness are given by the following lemma:

Lemma 2 *Let D be a $\mathcal{SHOIQ}[\mathbf{LOM}]$ concept in NNF, R an RBox and MT a go-theory. Then: **(1)** When started with D, R and MT, the tableau algorithm terminates; **(2)** D has a tableau w.r.t. R if and only if the expansion rules can be applied to D, R and MT such that they yield a complete, clash-free completion graph.*

As a direct consequence of the Lemma, the algorithm we have presented is a decision procedure for satisfiability and subsumption of $\mathcal{SHOIQ}[\mathbf{LOM}]$ concepts w.r.t. TBoxes, RBoxes and go-theories.

Same as concept subsumption, query answering can also be reduced to concept satisfiability. Let N_V be a set of variable names, disjoint with the set of individual names N_I. A *boolean conjunctive query* is a conjunction of terms of the form $C(a), R(a,b)$ and terms of the form $near(a, b, d, t_1, t_2)$, $far(a, b, d, t_1, t_2)$, $in(a, x_1, y_1, x_2, y_2, t_1, t_2)$ or their negation where C is a $\mathcal{SHOIQ}[\mathbf{LOM}]$-concept, R a role, $a, b \in N_I \cup N_V$ and $x_1, y_1, x_2, y_2, d, t_1, t_2$

are real numbers. A boolean query is ground if it contains no variables. The boolean query answering problem is the problem of determining whether a ground conjunctive query q is a logical consequence of the $\mathcal{SHOIQ}[\mathbf{LOM}]$ KB.

Obviously, ground queries of the form $near(a, b, d, t_1, t_2)$, $far(a, b, d, t_1, t_2)$, $in(a, x_1, y_1, x_2, y_2, t_1, t_2)$ hold w.r.t. to a $\mathcal{SHOIQ}[\mathbf{LOM}]$ KB $\Sigma = (\mathsf{K}, \mathsf{MT})$ iff the concepts $\{a\} \sqcap \exists \mathbf{near}_d^{t_1,t_2}.\{b\}$, $\{a\} \sqcap \exists \mathbf{far}_d^{t_1,t_2}.\{b\}$ and $\{a\} \sqcap \exists \mathbf{in}_{x_1,y_1,x_2,y_2}^{t_1,t_2}.\{b\}$ respectively are unsatisfiable w.r.t. Σ. Here is how we can represent the queries presented in the introduction using conjunctive queries:

- **Q1**: $Boeing747(?p) \wedge LowFuel(?p) \wedge in(?p, \mathsf{x_1}, \mathsf{y_1}, \mathsf{x_2}, \mathsf{y_2}, \mathsf{t_1}, \mathsf{t_2})$
- **Q2**: $(Unmaintained \sqcup \exists \mathbf{near}_{10}^{\mathsf{t_{now}},\mathsf{t_{now}}}.Airplane)(?p)$
- **Q3**: $\neg PassengerPlane(?p) \wedge country(?p, ?c) \wedge \neg NATOMember(?c) \wedge in(?p, \mathsf{x_1} - 10, \mathsf{y_1} - 10, \mathsf{x_2} + 10, \mathsf{y_2} + 10)$
- **Q4**: $(Ship \sqcap LowFirePower \sqcap \leq 1 near_1^{\mathsf{t_{now}},\mathsf{t_{now}}}.HighFirePower)(?p)$

It is not hard to see that the problem of answering the queries **Q1** and **Q3** is equivalent, using the standard rolling-up technique, to retrieving the instances of the following concepts:

$$Boeing747 \sqcap LowFuel \sqcap \exists \mathbf{in}_{x_1,y_1,x_2,y_2}^{t_1,t_2}$$
$$\neg PassengerPlane \sqcap \exists country.\neg NATOMember \sqcap \exists \mathbf{in}_{\mathsf{x_1}-10,\mathsf{y_1}-10,\mathsf{x_2}+10,\mathsf{y_2}+10}^{t_1,t_2}$$

5 Implementation

We have built a prototype system that integrates the Description Logic reasoner Pellet and the LOM checker presented in [7]. As a first step, we have considered the problem of answering mixed conjunctive queries [3] over an "ordinary" $\mathcal{SHOIQ}$ ontology and a LOM KB. In the absence of LOM-based constructs in the $\mathcal{SHOIQ}$ KB, it is possible to decompose a mixed query Q into a query Q_1 to the DL KB and a query Q_2 to the LOM theory. The queries Q_1 and Q_2 are handled independently by the DL reasoner and the LOM checker respectively. The combined reasoner makes sure that the answers for Q adequately combine the results for the independent queries Q_1 and Q_2. Note that, although none of the LOM-based constructors is used in the DL KB, there is still a coupling [4] between the DL and the LOM theories, since both share individuals and thus talk about the same objects.

[3] This is, queries containing both LOM and DL query atoms

[4] Of course, looser

We have explored three strategies for query processing: 1) Run the DL query first and then filter the results using the results from LOM query [5] 2) Run the LOM query first and filter the results using the results from the DL query [6] 3) Execute both queries completely independently and then combine the results.

Pellet implements an optimization technique called *binary-instance-retrieval* [3], which makes it much more efficient to retrieve all the answers from the DL reasoner first and then filter the results using the output of the LOM checker. Thus, the first approach proved to be more efficient than 2) and 3) In the near future, we plan to extend the DL reasoner Pellet to handle the full expressive power provided by the combined formalism and to perform a thorough empirical evaluation.

6 Conclusion and Related Work

In this paper, we have presented a decidable combination of the Description Logic $\mathcal{SHOIQ}$ with a prominent fragment of general motion theories, called *go-theories*. This combination allows to describe the movement of complex objects as well as complex objects in terms of the characteristics of their motion. The expressive power provided by our combined formalism can be useful to many real-world scenarios, ranging from traffic control to military applications. We have provided a practical decision procedure and prototype implementation using the open-source DL reasoner Pellet.

In the last few years, researchers in AI and Philosophy have intensively studied *spatio-temporal* logics. Most of them (see for example [2]) involve qualitative logics, in contrast to the Logic of Motion, which is heavily focused on quantitative information and is rooted in a mix of geometry and logic, rather than logic alone.

Recently, it has been shown that DL knowledge bases can be coupled to spatial (both qualitative, e.g. $\mathcal{RCC}$-8, and quantitative, e.g.) and/or temporal KBs (e.g. **PTL**) using the $\mathcal{E}$-Connections technique [5]. However, to the best of our knowledge, there is no $\mathcal{E}$-Connected logic that combines DLs with both quantitative spatial and quantitative temporal formalisms, nor there is a decision procedure for $\mathcal{E}$-Connections involving DLs and either quantitative spatial logics or qualitative temporal logics. Besides, these formalisms do not allow uncertainty about starting times, ending times and velocities.

[5]LOM queries become ground since we already have possible answers generated

[6]Now DL queries are ground

References

[1] F. Baader and W. Nutt. Basic description logics. In Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 43–95. Cambridge University Press, 2003.

[2] D. Gabelaia, R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyaschev. Combining spatial and temporal logics: expressiveness vs. complexity. *Journal of Artificial Intelligence Research (JAIR)*, 23:167–243, 2005.

[3] V. Haarslev and R. Möller. Optimization techniques for retrieving resources described in OWL/RDF documents: First results. In *Proc. of the 9th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 163–173, 2004.

[4] Ian Horrocks and Ulrike Sattler. A tableaux decision procedure for SHOIQ. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*. Morgan Kaufman, 2005.

[5] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. $\mathcal{E}$-Connections of Abstract Description Systems. *Artificial Intelligence 156(1):1-73*, 2004.

[6] P.F. Patel-Schneider, P. Hayes, and I.Horrocks. Web ontology language OWL Abstract Syntax and Semantics. *W3C Recommendation*, 2004.

[7] F. Yaman, D. Nau, and V.S. Subrahmanian. A logic of motion. In *Proc. 2004 International Conference on the Principles of Knowledge Representation and Reasoning (KR-2004), Whistler, Canada, June 2-5*, 2004.

[8] F. Yaman, D. Nau, and V.S. Subrahmanian. Going far, logically. In *Proc. of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-2005), EDINBURGH, SCOTLAND*, 2005.

A Appendix: Proofs

A.1 Proof for Lemma 1

(⇒)

Let $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ be a tableau for D w.r.t. R and $\mathbf{MT}$. A model $\mathcal{I}$ can be constructed as in [4]. As in [4], we need to prove the following statement:

$$\text{if } C \in \mathcal{L}(s) \Rightarrow s \in C^{\mathcal{I}}$$

The proof works by induction of the structure of concepts. We only consider the additional constructs we provide. The rest is identical to [4]. We need to consider the following additional cases:

- Let $\exists \mathbf{P}^{t_1,t_2}_{x_1,y_1,x_2,y_2} \in \mathcal{L}(s)$, then there exists an $a \in N_I$ s.t. $\{a\} \in \mathcal{L}(s)$ and $\mathbf{MT} \models in(a, x_1, y_1, x_2, y_2, t_1, t_2)$. By induction, $s \in (\{a\})^{\mathcal{I}}$, i.e. $s = a^{\mathcal{I}} = a$. Hence, $\mathbf{MT} \models in(s, x_1, y_1, x_2, y_2, t_1, t_2)$ and $s \in \mathcal{W}$, which implies that $s \in (\exists \mathbf{P}^{t_1,t_2}_{x_1,y_1,x_2,y_2})^{\mathcal{I}}$
- Let $\exists \mathbf{Q}^{t_1,t_2}_d.D \in \mathcal{L}(s)$, then there exists a $t \in \mathbf{S}$ and object names $a, b \in N_I$ s.t. $\{a\} \in \mathcal{L}(s), \{b, D\} \subseteq \mathcal{L}(t)$ and $\mathbf{MT} \models Q(a, b, d, t_1, t_2)$. By induction, $s \in (\{a\})^{\mathcal{I}}$, i.e. $s = a^{\mathcal{I}} = a$. Also, $t \in (\{b\})^{\mathcal{I}}$, i.e. $t = b^{\mathcal{I}} = b$ and $t \in D^{\mathcal{I}}$. Hence, we have that $\mathbf{MT} \models Q(s, t, d, t_1, t_2)$ and by definition of the semantics $s \in (\exists \mathbf{Q}^{t_1,t_2}_d.D)^{\mathcal{I}}$.
- The proof for the case $\geq n\mathbf{Q}^{t_1,t_2}_d.D \in \mathcal{L}(s)$ is completely analogous to the above case.
- Let $\leq n\mathbf{Q}^{t_1,t_2}_d.D \in \mathcal{L}(s)$ and $\{a\} \in \mathcal{L}(s)$, then:
  - There are at most n elements $u_1, ..., u_n \in \mathbf{S}$ s.t. $u_i \neq s$, $\{D, b_i\} \subseteq \mathcal{L}(u_i)$ and $\mathbf{MT} \models Q(a, b_i, d, t_1, t_2)$ for $1 \leq i \leq n$
  - For $w \in \mathbf{S}$ with $\{b\} \in \mathcal{L}(w)$ and $\mathbf{MT} \models Q(a, b, d, t_1, t_2)$ it follows that $\{D, \sim D\} \cap \mathcal{L}(b) \neq \emptyset$

  Using ♣, it is easy to verify that $s \in (\leq n\mathbf{Q}^{t_1,t_2}_d.D)^{\mathcal{I}}$
- Let $\forall \mathbf{Q}^{t_1,t_2}_d.D \in \mathcal{L}(s)$, $\{a\} \in \mathcal{L}(s)$, $b \in \mathcal{L}(t)$ and $\mathbf{MT} \models Q(a, b, d, t_1, t_2)$. Then $D \in \mathcal{L}(t)$. By induction, $s = a$ and $t = b$ and $t \in D^{\mathcal{I}}$. By the semantics, $s \in (\forall \mathbf{Q}^{t_1,t_2}_d.D)^{\mathcal{I}}$.

(⇐)

For the converse, suppose that $\mathcal{I} = (\mathcal{W}, .^{\mathcal{I}})$ is a model for D w.r.t. R and $\mathbf{MT}$. A tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ can be easily constructed as in [4]:

- $\mathbf{S} = \mathcal{W}$
- $\mathcal{E}(R) = R^{\mathcal{I}}$
- $\mathcal{L}(s) = \{C \in clos(D) | s \in C^{\mathcal{I}}\}$

We denote the above definition as ♣.

We prove that the tableau thus defined verifies the new conditions we have introduced in the definition of a tableau:

- Let $\exists \mathbf{P}^{t_1,t_2}_{x_1,y_1,x_2,y_2} \in \mathcal{L}(s)$. By ♣, $s \in (\exists \mathbf{P}^{t_1,t_2}_{x_1,y_1,x_2,y_2})^{\mathcal{I}}$. By the semantics, $\mathbf{MT} \models in(s, x_1, y_1, x_2, y_2, t_1, t_2)$ and s must be a named individual. This implies that $\{s\} \in \mathcal{L}(s)$ and hence the condition holds.
- Let $\exists \mathbf{Q}^{t_1,t_2}_{d}.D \in \mathcal{L}(s)$. By ♣, $s \in (\exists \mathbf{Q}^{t_1,t_2}_{d}.D)^{\mathcal{I}}$. By the semantics, there exists a $t \in \mathcal{W}$ s.t. $t \in D^{\mathcal{I}}$ and $\mathbf{MT} \models Q(s, t, d, t_1, t_2)$, where s, t must be named individuals. Using ♣ again it is easy to see that the condition holds. Similar considerations apply in case $\geq n\mathbf{Q}^{t_1,t_2}_{d}.D \in \mathcal{L}(s)$, $\leq n\mathbf{Q}^{t_1,t_2}_{d}.D \in \mathcal{L}(s)$ and $\forall \mathbf{Q}^{t_1,t_2}_{d}.D \in \mathcal{L}(s)$.

A.2 Proof for Lemma 2

Proof

The proof consists of three parts: termination, soundness and completeness.

TERMINATION: Termination is a consequence of the termination of both the $\mathcal{SHOIQ}$ tableau algorithm and the query answering algorithms for go-theories. More specifically, termination is a consequence of the following facts; let $m = cl(D)$, k the number of roles and their inverses in D and R, $(n_{\geq})$ the maximal number of at-least number restrictions, $(n_{\leq})$ the maximal number of at-most number restrictions, $o_1, ..., o_l$ all the nominals occurring in D and object names in $\mathbf{MT}$ and $\lambda = 2^{2m+k}$.

- All but the shrinking rules strictly extend the completion graph by adding new nodes (and edges) or extending node labels. The non-shrinking rules never remove nodes (or edges) and never remove elements from node labels.
- New nodes are only added by the generating rules and each of those rules can be triggered at most once for a given concept in the label of a given node or in its heirs. Note that none of the additional rules we have introduced is generating.

- As for the $\mathcal{SHOIQ}$ case, a generating rule being applied to the label of a node x can generate at most $(n_{\geq})$ blockable successors. As there are at most m concepts in $\mathbf{L}(x)$, a node can have at most $m \times (n_{\geq})$ blockable successors.

- As for $\mathcal{SHOIQ}$, the blocking condition ensures that the length of a path consisting entirely of blockable nodes is bounded by λ.

- As for $\mathcal{SHOIQ}$, the number of nominal nodes in the completion graph is bounded by $\mathcal{O}()l(m(n_{\leq}))^{\lambda})$. This is a direct consequence from the fact that the additional rules we are introducing do not generate new nominal nodes.

- The algorithms in [7][8] for answering queries over go-theories terminate.

- The additional rules will be applied at most once. Each application of the $\rightarrow in$, $\rightarrow \exists_Q$ and $\rightarrow \geq_Q$ rules generates a single query and hence, since there are at most m of those concepts in the label of a node x, then we can fire at most m queries to the go-theory. Moreover the number of objects (or pairs of objects) returned by those queries is finite.

SOUNDNESS: We show that we can obtain a tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ from a complete and clash-free completion graph $\mathcal{G}$.

A *path* is a sequence of pairs of blockable nodes of $\mathcal{G}$ of the form $\tilde{p} = (\frac{x_0}{x'_0}, ..., \frac{x_n}{x'_n})$. For such a path we define $Tail(p) = x_n$ and $Tail'(\tilde{p}) = x'_n$. With $(\tilde{p}|\frac{x_{n+1}}{x'_{n+1}})$ we denote the path $\tilde{p} = (\frac{x_0}{x'_0}, ..., \frac{x_n}{x'_n}, \frac{x_{n+1}}{x'_{n+1}})$. The set $Paths(\mathcal{G})$ is inductively defined as follows:

- For each blockable node x of $\mathcal{G}$ that is a successor of a nominal node or a root node, $(\frac{x}{x}) \in Paths(\mathcal{G})$, and

- For a path $\tilde{p} \in Paths(\mathcal{G})$ and a blockable node y in $\mathcal{G}$:
  - If y is a successor of $Tail(\tilde{p})$ and y is not blocked, then $(p|\frac{y}{y}) \in Paths(\mathcal{G})$ and
  - If y is a successor of $Tail(\tilde{p})$ and y is blocked by y', then $(p|\frac{y'}{y}) \in Paths(\mathcal{G})$

Due to the construction of $Paths(\mathcal{G})$, all nodes occurring in a path are blockable and for $\tilde{p} \in Paths(\mathcal{G})$ with $\tilde{p} = (\tilde{p}'|\frac{x}{x'})$, x is not blocked, x' is blocked iff $x \neq x'$ and x' is never indirectly blocked. Furthermore the blocking condition implies $\mathbf{L}(x) = \mathbf{L}(x')$. We denote by $Nom(\mathcal{G})$ the set of nominal nodes in $\mathcal{G}$ and define a tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ from $\mathcal{G}$ as follows (denoted by $\diamondsuit$):

- $\mathbf{S} = Nom(\mathcal{G}) \cup Paths(\mathcal{G})$
- $\mathcal{L}(\tilde{p}) = \mathbf{L}(Tail(\tilde{p})$, if $\tilde{p} \in Paths(\mathcal{G})$ and $\mathbf{L}(p)$ if $p \in Nom(\mathcal{G})$
- $\mathcal{E}(R) = \{(\tilde{p}, \tilde{q}) \in Paths(\mathcal{G} \times \mathcal{G})|$
  $\tilde{q} = (p|\frac{x}{x'})$ and x' is an R-successor of $Tail(\tilde{p})$ or
  $\tilde{p} = (q|\frac{x}{x'})$ and x' is an inv(R)-successor of $Tail(\tilde{q})\} \cup$
  $\{(\tilde{p}, a) \in Paths(\mathcal{G}) \times Nom(\mathcal{G})|$ a is an R-neighbor of $Tail(\tilde{p})\} \cup$
  $\{(a, \tilde{p}) \in Nom(\mathcal{G}) \times Paths(\mathcal{G})|$ $\tilde{p}$ is an R-neighbor of $a\} \cup$
  $\{(a, b) \in Nom(\mathcal{G}) \times Nom(\mathcal{G})|$ b is an R-neighbor of $a\}$

In order to prove that T is a tableau for D w.r.t. **R** and **MT**, we only need to show that T verifies the additional conditions we have introduced in Definition 1. First, note that if $\mathcal{G}$ is clash-free and complete, then the MT-related concepts can only appear in the label of *nominal* nodes in $\mathcal{G}$.

- **Property P0**: Let $\exists \mathbf{P}^{t_1,t_2}_{x_1,y_1,x_2,y_2} \in \mathcal{L}(s)$ with $s \in Nom(\mathcal{G})$. Then, by $\diamondsuit$, $\exists \mathbf{P}^{t_1,t_2}_{x_1,y_1,x_2,y_2} \in \mathbf{L}(s)$. Again, there is an $\{a_i\}$ s.t. $\{a_i\} \in \mathcal{L}(s)$ and $\mathbf{MT} \models in(a_i, x_1, y_1, x_2, y_2, t_1, t_2)$. Since $\mathcal{G}$ is clash-free, such an a_i must exist and hence the condition holds.
- **Property P1**: Let $\exists \mathbf{Q}^{t_1,t_2}_{d}.D \in \mathcal{L}(s)$ with $s \in Nom(\mathcal{G})$; then, by $\diamondsuit$, $\exists \mathbf{Q}^{t_1,t_2}_{d} \in \mathbf{L}(s)$. The $\rightarrow \exists_Q$ rule and clash-freeness ensure that there exists a pair (a, b) of object names s.t. $\{a\} \in \mathbf{L}(s)$, $\{D, \{b\}\} \subseteq \mathbf{L}(y)$, with y a nominal node, and $\mathbf{MT} \models Q(a, b, d, t_1, t_2)$. Hence, using $\diamondsuit$ again, $\{a\} \in \mathcal{L}(s)$ and $\{D, \{b\}\} \subseteq \mathcal{L}(b)$.
- **Property P2**: The proof is straightforward by completeness of the $\rightarrow \forall_Q$ rule.
- **Property P4**: Let $(\leq n\mathbf{Q}^{t_1,t_2}_{d}.D) \in \mathcal{L}(s)$, with $s \in Nom(\mathcal{G})$ and $\{a\} \in \mathcal{L}(s)$. Then, $\{(\leq n\mathbf{Q}^{t_1,t_2}_{d}.D), \{a\}\} \subseteq \mathbf{L}(s)$ The $\rightarrow \leq_Q$ rule and clash-freeness ensure that there are at most n nominal nodes $y_1, ..., y_n$ s.t. $\{D, \{b_i\}\} \subseteq \mathbf{L}(y_i)$ and $\mathbf{MT} \models Q(a, b_i, d, t_1, t_2)$ for $1 \leq i \leq n$. By $\diamondsuit$, there are at most n elements $w_1, ..., w_n \in \mathbf{S}$ s.t. $\{D, \{b_i\}\} \subseteq \mathcal{L}(w_i)$ and $\mathbf{MT} \models Q(a, b_i, d, t_1, t_2)$ for $1 \leq i \leq n$.

- Similar arguments apply for proving **P3**

- **Property P5**: Let $(\leq n\mathbf{Q}_d^{t_1,t_2}.D) \in \mathcal{L}(s)$, $\{a\} \in \mathcal{L}(s)$, $\mathbf{MT} \models Q(a,b,d,t_1,t_2)$ and $\{b\} \in \mathcal{L}(t)$. By $\diamondsuit$, s,t are nominal nodes in $\mathcal{G}$, $(\leq n\mathbf{Q}_d^{t_1,t_2}.D) \in \mathbf{L}(s)$, $\{a\} \in \mathbf{L}(s)$, $\mathbf{MT} \models Q(a,b,d,t_1,t_2)$ and $\{b\} \in \mathbf{L}(t)$. The $\rightarrow$ *choose*$_Q$ rule ensures that $\{D, \sim D\} \cap \mathbf{L}(t) \neq \emptyset$. By $\diamondsuit$, $\{D, \sim D\} \cap \mathcal{L}(t) \neq \emptyset$.

COMPLETENESS: We show that, given a tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ for D w.r.t. R and **MT**, it is possible to apply the non-deterministic rules, i.e. the $\rightarrow \sqcup$, $\rightarrow$ *choose*, $\rightarrow\leq$, $\rightarrow NN$, $\rightarrow in$, $\rightarrow \exists_Q$, $\rightarrow\geq_Q$ and $\rightarrow\leq_Q$ rules in such a way that we obtain a complete and clash-free graph. Fur such a purpose, we define inductively with the generation of new nodes a mapping π from nodes in the completion graph to individuals in $\mathbf{S}$ of the tableau in such a way that:

- $\mathbf{L}(x) \subseteq \mathcal{L}(\pi(x))$, for each node x

- If y is an R-successor of x, then $(\pi(x), \pi(y)) \in \mathcal{E}(R)$

- $x \neq y$ implies $\pi(x) \neq \pi(y)$

The proof is analogous to the proof in [4] with the additional observation that the application of the extra rules does not lead to a clash due to the additional properties that T needs to satisfy, that force the result of the queries to the motion theory to be non-empty.

P.E.D