Diachronic Description Logics

** Initial Explorations **

Alex Borgida
Dept. of Computer Science
Rutgers University, USA
Email: borgida@cs.rutgers.edu

Abstract

We provide motivation and a notation for representing in description logics certain "dynamic constraints" concerning changes of state. After providing a semantics for such descriptions using pairs of ordinary interpretations, we show that one can use existing DL inference engines to reason about such extended diachronic descriptions with no additional cost compared to the standard synchronic version.

1 Motivation

Traditionally, concepts and individuals in Description Logics describe *single states* of the domain of discourse. This is reflected even in the semantic interpetation function, which maps concepts to simple subsets of the domain of individuals.

However, the real world and our knowledge of it is not static. Instead, actions (also called events or transactions) occur, changing one state to another. As part of describing an application domain, sometimes we would like to express constraints on the state changes. The following are some examples of such "dynamic integrity constraints", as these were called in the field of databases a decade ago [Lipeck&Saake 87].

1. *Unchanging properties/attributes.*
   The property expressed in the real world is unchanging. Once you know the complete value(s) of the property for some individual in the knowledge base, this cannot change. For example,

   - `#birthMother` for `#PERSON`
     /* knowledge of who gave birth to you doesn't change over time. */

2. *"Monotonic" properties.*
   The property value cannot "become smaller" as time progresses, according to some ordering of the values:

   - #`hasChildren` for #`PERSON`, as a set
     /* you can only have more children as time goes on */
   - #`ageOf` for #`PERSON` as an integer
   - #`salaryOf` for #`EMPLOYEE` as a dollar amount.

3. *Deletion dependencies.*
   When an individual A is removed from the KB, some other individual B, related to A (either by some property `r` or its inverse) must also be removed:

   - if A #`hasBrain` B, then removing A causese B to be removed
     /* when a person A dies, so does his/her brain */
   - if B #`isBrainOf` A, then removing B causes A to be removed;
     /* when the brain dies, so does the person */

4. *Changing identity conditions.*
   For certain kinds of objects, when some key part(s) change, so does the "identity" of the object. For example,

   - #`VASE` is #`madeOf` some #`MATERIAL`, so if `v.#madeOf` changes, `v` is no longer the same object it was before. (See [AFGP 96].)[1]

2 Syntax

How should one capture these kinds of constraints in a knowledge management system? One possibility is to simply check that every action provably satisfies these constraints. This is unsatisfactory because new actions may be added later to the conceptual model, and we may forget to build in the checking of these constraints. For this reason, we want to state the constraints declaratively. One available approach is to use a *temporal logic* [Artale&Franconi 01]. However, this seems to be an overkill since in many cases all we need is the ability to talk about *pairs of states* connected by the actions[2]. We therefore investigate the possibility of stating such constraints directly in a DL-like formalism.

Since we have to relate values before and after an operation is performed, it is reasonable to consider the pre/post condition approach of Floyd/Hoare, in

[1]This seems like the contrapositive of (1): the #`materialOf` for a vase cannot change.

[2]In contrast, some constraints could involve longer histories: *"The price cannot increase by more than 5% over a 12 month period."*

particular the mathematical notation of specification languages like VDM and Z: $\mathbf{x}$ (the value before an action) is compared to $\mathbf{x}'$ ("x-primed") – the value after the action. Using this simple convention, we can represent the above examples as follows[3]:

1. *Unchanging property:*

   $$\boxed{\#\text{PERSON} \ \sqsubseteq (\#\text{motherOf} \doteq \#\text{motherOf}')}$$

   #PERSON **:subclassOf**
   :slot-constraint #motherOf
   :same-value-as #motherOf′

2. *Monotonic property values:*

   $$\boxed{\#\text{PERSON} \ \sqsubseteq (\#\text{hasChildren} \subseteq \#\text{hasChildren}')}$$

   #PERSON **:subclass-of**
   :restriction #hasChildren′
   :values-superset-of #hasChildren

3. *Deletion dependencies:*
   To describe these, we must first model deletion/creation. One approach is to assume that the instances of every primitive concept are the *currently existing* individuals, and "deleted" instances remain, but only as instances of a new special class `DeletedThing`.
   Aside: We then need general laws stating that deleted things cannot participate in any relations or be instances of ordinary primitive concepts:

   $$(\forall x, y) DeletedThing(x) \ \Rightarrow \ \neg C(x) \wedge \neg r(x, y)$$

   and

   $$(\forall x, y) DeletedThing'(x) \ \Rightarrow \ \neg C'(x) \wedge \neg r'(x, y)$$

   for every role identifier r, and and every concept identifier C different from `DeletedThing` and $\top$. Such laws can be captured by requiring that every object (i.e., every instance of $\top$) satisfy descriptions like $\neg$`(DeletedThing` $\sqcap$ `C)` and $\neg$`(DeletedThing` $\sqcap$ $\exists$. `r)`

   We can then state the deletion constraint (3) as the following FOPC formula:

[3]We give first, in a box, a DL notation for our description, then one resembling OIL [OIL] — a language for describing web page semantics

$$(\forall x)[Person(x) \;\Rightarrow (\forall y).[hasBrain(x,y) \wedge DeletedThing'(x) \;\Rightarrow\; DeletedThing'(y)]]$$
$$\Longleftrightarrow$$
$$(\forall x).\; \not Person(x) \vee \not DeletedThing'(x) \vee (\forall y).[hasBrain(x,y) \;\Rightarrow\; DeletedThing'(y)]$$

This universal constraint can be translated into the following subsumption constraint:

$$\#\top \;\sqsubseteq\; \neg\#\mathtt{PERSON} \sqcup \neg\mathtt{DeletedThing}' \sqcup (\forall\#\mathtt{hasBrain} : \#\mathtt{DeletedThing}')$$

4. *Identity conditions* .
   These can be expressed by FOPC formulae like the following:
   $\forall v_1, v_2, m_2, m_1[\#isMadeOf(v_1, m_1) \wedge \#isMadeOf'(v_2, m_2) \wedge m_1 \neq m_2] \;\Rightarrow\; (v_1 \neq v_2)$
   and from [Borgida 96] we know that any such formula can be expressed as a concept in a “universal” DL (whose reasoning is of course undecidable).

Note that the identifier x' is intimately related to the identifier x, so that it is best to view x' as a shorthand for a term $primed(x)$, which acts as a composite name. This leads us to the formal definition

Definition 1 *A* **diachronic description over concept and role identifier set** $\mathcal{N}$ *is an ordinary DL description, whose primitive identifiers come from the set* $\mathcal{N} \cup \{primed(E) \mid E \in \mathcal{N}\}$*.*

In fact entire composite descriptions can be primed by just associating a name with them. An obvious extension is to handle this directly, by making $primed()$ a full-fledged concept/role constructor, so that `primed(#PERSON` $\sqcap$ $\forall$ `#age : #INTEGER)` is equivalent to `primed(#PERSON)` $\sqcap$ $\forall$ `primed(#age) : #INTEGER`.

An alternative syntactic approach would be to introduce special constructors/restrictions to express each kind of condition we encounter. For example,

$\boxed{\text{(:unchanging \#motherOf)}}$

:restriction #motherOf
:unchanging

$\boxed{\text{(:non-decreasing \#childrenOf)}}$

:restriction #childrenOf
:numeric-non-decreasing

The advantage of this approach would be that it does not use the full power of constructors like **subset-of**, which are known to lead to undecidability in the general case. The disadvantage is having to build a new reasoner whenever we add a new kind of a constraint, and proving it correct. (This approach to extensible reasoning is explored in [Borgida 99].)

3 Formal Semantics

Traditionally, the semantics of Description Logics is presented as an interpretation which describes the *current state* of the world. Such an interpretation, provides (i) a domain of values Δ for the interpretation; and (ii) a (post-fix) *interpretation function* $^{\mathcal{I}}$ which maps every concept C to a subset $C^{\mathcal{I}}$ of Δ, and every property/slot R to a subset $R^{\mathcal{I}}$ of $\Delta \times \Delta$. The semantics of various concept constructors (e.g. conjunction $\sqcap$) is then presented as restrictions on $^{\mathcal{I}}$, e.g., $(A \sqcap B)^{\mathcal{I}} = A^{\mathcal{I}} \cap B^{\mathcal{I}}$ An alternative, which works in the absence of recursive definitions, is to have $\mathcal{I}$ provide interpretations for atomic identifiers, and then extend it over complex descriptions using the above equalities.

To extend this to diachronic descriptions, we will use a pair of ordinary interpretations, describing the pre- and post-activity state. Formally,

Definition 2 *A* **d-interpretation** $\mathcal{H}$ **over atomic identifiers** $\mathcal{N}$ *consists of a domain* Δ *of individuals, and a pair* $(^{\mathcal{H}_{pre}}, ^{\mathcal{H}_{post}})$*, such that* $^{\mathcal{H}_{pre}}$ *and* $^{\mathcal{H}_{post}}$ *are ordinary interpretation functions over* $\mathcal{N}$*.*

Let D be a diachronic description over atomic symbols $\mathcal{N}$*. The* **d-denotation of D under** $\mathcal{H}$*, written as* $\mathcal{H}\langle D\rangle$*, is defined as*

- $D^{\mathcal{H}_{pre}}$ *if D is in* $\mathcal{N}$*;*
- $D^{\mathcal{H}_{post}}$ *if D=prime(E), for E in* $\mathcal{N}$*;*
- *otherwise satisfying the standard equations for composite descriptions.*

Note that under this definition, the d-denotation of a constraint concept such as ($\texttt{hasChildren} \subseteq \texttt{hasChildren}'$) is $\{\ y \mid \forall v.\mathrm{child}^{\mathcal{H}_{pre}}(y,v) \Rightarrow \mathrm{child}^{\mathcal{H}_{post}}(y,v)\ \}$, representing individuals y such that if v is related to y by `hasChildren` in the "stard-state" $\mathcal{H}_{pre}$, then v is also related to y in the "end state", represented by $\mathcal{H}_{post}$; this is the desired intuitive interpetation.

Description logics are traditionally interested in relationships such as *subsumption* (written here as $\sqsubseteq$). Formally, an ordinary interpretation $\mathcal{I}$ is said to satisfy A $\sqsubseteq$ B iff $A^{\mathcal{I}} \subseteq B^{\mathcal{I}}$. More generally, given a knowledge base KB, consisting of a set of subsumption constraints $\{\ A_i \sqsubseteq B_i,...\ \}$, KB is said to entail the subsumption C $\sqsubseteq$ D (written as KB $\models$C $\sqsubseteq$ D), if every interpretation $\mathcal{I}$ that satisfies each $A_i \sqsubseteq B_i$ also satisfies $C \sqsubseteq D$. We generalize this to diachronic descriptions the obvious way:

Definition 3 *Given diachronic descriptions C and D, d-interpretation* $\mathcal{H}$ **d-satisfies** $C \sqsubseteq D$ *if* $\mathcal{H}\langle C\rangle \subseteq \mathcal{H}\langle D\rangle$*.*

A knowledge base KB is said to **d-entail** *the subsumption* $C \sqsubseteq D$ *(written as* $\{A \sqsubseteq B, ...\} \models_\delta C \sqsubseteq D$ *) iff for all* $\mathcal{H}$ *d-satisfying every constraint* $A \sqsubseteq B$ *in KB,* $\mathcal{H}$ *also d-satisfies* $C \sqsubseteq D$*. (As a special case, D* **h-subsumes** *C iff all d-interpretations* $\mathcal{H}$ *d-satisfy* $C \sqsubseteq D$*.).*

4 Reasoning

We are interested in reasoning with diachronic descriptions. Recall that given a diachronic description D over $\mathcal{N}$, we viewed x' as shorthand for a term $primed(x)$, for $x \in \mathcal{N}$. We will use $\check{y}$ as an *indecomposable* atomic identifier, unrelated to (and different from) all the other ones, including y. Let us call the alphabet of identifiers for this case $\mathcal{N}\check{\mathcal{N}}$, so $\mathcal{N}\check{\mathcal{N}}$= {E, Ě | E $\in\mathcal{N}$}. Extend the notation so that Ď, for a complex diachronic description D, results from substituting Ě for E$'$, for every E in $\mathcal{N}$.. We then get the following useful result:

Theorem 1 *Given diachronic descriptions A,B,C,D,...*
$\{A \sqsubseteq B,...\} \models_\delta C \sqsubseteq D$ *if and only if* $\{\check{A} \sqsubseteq \check{B},...\} \models \check{C} \sqsubseteq \check{D}$

The proof of the theorem rests on relating the interepretation $\mathcal{I}$ over $\mathcal{N}\check{\mathcal{N}}$and the d-interpretation $(\mathcal{H}_{pre}, \mathcal{H}_{post})$ over $\mathcal{N}$as follows: $\mathrm{E}^{\mathcal{I}}=\mathrm{E}^{\mathcal{H}_{pre}}$ and $\check{\mathrm{E}}^{\mathcal{I}}=\mathrm{E}^{\mathcal{H}_{post}}$, for E in $\mathcal{N}$.

This theorem shows that we can use the existing theorem provers for description logics (such as FaCT) to reason about diachronic descriptions by simply treating V$'$as a new identifier, for every ordinary identifier V.

5 Conclusions

Traditionally, description logics have dealt with issues involving synchronic snapshots of the world. We have provided motivation for extending description logics to allow them to model certain aspects of the world dynamics. The syntactic form of our extension is simple, and more limited in expressive power than temporal DLs. We have provided the semantics of these new kinds of diachronic descriptions in terms of pairs of states, which are supposed to model the beginning and ending states of actions. Finally, we showed the rather desirable feature of our extension that reasoning with it reduces to reasoning in standard description logics.

Some observations/extensions: For the consequences of the theorem to be truly useful, we need to extend the current DLs to represent and reason with descriptions like the ones used in our examples (e.g., "objects whose value(s) for role p are the same-as/numerically-greater-than/subsets-of the values of role q"). Some of these extensions (involving numbers for example) can be added as suggested in [Baader&Hanschke 91]. The others will need to be built in directly, but should not pose the traditional computational difficulties because they deal with single roles rather than chains.

A second observation is that if we want to mix diachronic and synchronic constraints in the same knowledge base, we must be careful to extend the semantics so that subsumption conditions not involving primed identifiers are required to hold *both* in the initial and the final state of actions.

References

[Artale&Franconi 01] A. Artale and E. Franconi. "A Survey of Temporal Extensions of Description Logics.", *Annals of Mathematics and Artificial Intelligence (AMAI), Vol. 30 No. 1-4*, 2001, Kluwer Academic Publishers.

[AFGP 96] Alessandro Artale, Enrico Franconi, Nicola Guarino, Luca Pazzi, *Part-Whole Relations in Object-Centered Systems: An Overview, DKE 20(3)*, pp.347-383, (1996)

[Baader&Hanschke 91] Baader, F., Hanschke, P., 'A Scheme for Integrating Concrete Domains into Concept Languages', *IJCAI-91*, pp. 452-457, 1991.

[Borgida 96] A. Borgida, "On the Relative Expressiveness of Description Logics and Predicate Logics", *Artificial Intelligence Journal*, **82** (1996): 353–367.

[Borgida 99] A. Borgida, "Extensible Knowledge Representation: the Case of Description Reasoners",*Journal of Artificial Intelligence Research*, Volume 10, pages 399-434, 1999.

[Lipeck&Saake 87] Udo W. Lipeck, Gunter Saake: "Monitoring dynamic integrity constraints based on temporal logic", *Information Systems 12(3)*: pp.255-269 (1987)

[OIL] Dieter Fensel, Ian Horrocks, Frank van Harmelen, Stefan Decker, Michael Erdmann, Michel C. A. Klein. "OIL in a Nutshell", *EKAW 2000*, pp.1-16