Temporal Attributes: Status and Subsumption

C. Maria Keet[1] E.A. Nasubo Ongoma[2]

[1] Department of Computer Science, University of Cape Town,
Cape Town 7701, South Africa,
Email: mkeet@cs.uct.ac.za

[2] UKZN/CSIR-Meraka Centre for Artificial Intelligence Research,
School of Mathematics, Statistics, and Computer Science,
University of KwaZulu-Natal,
Durban 4000, South Africa,
Email: 212562258@stu.ukzn.ac.za

Abstract

Representing data that changes over time in conceptual data models is required by various application domains, and requires a language that is expressive enough to fully capture the operational semantics of the time-varying information. Temporal modelling languages typically focus on representing and reasoning over temporal classes and relationships, but have scant support for temporal attributes, if at all. This prevents one to fully utilise a temporal conceptual data model, which, however, is needed to model not only evolving objects (e.g., an employee's role), but also its attributes, such as changes in salary and bonus payouts. To characterise temporal attributes precisely, we use the $\mathcal{DLR}_{\mathcal{US}}$ Description Logic language to provide its model-theoretic semantics, therewith essentially completing the temporal ER language ER_{VT}. The new notion of *status attribute* is introduced to capture the possible changes, which results in several logical implications they entail, including their interaction with temporal classes to ensure correct behaviour in subsumption hierarchies, paving the way to verify automatically whether a temporal conceptual data model is consistent.

Keywords: temporal conceptual data models, temporal attributes, dynamic data

1 Introduction

The representation of temporal information has received attention from diverse fields. In conceptual modelling, this may be informal or 'hidden', such as UML's "freeze" attribute (Object Management Group, 2012) and informal business rules about time in ORM2 (Halpin, 2008), restricted to those that fit in a non-temporal UML (McBrien, 2008), annotation-based ad hoc formal constraints without a specification of the model (Khatri *et al.*, 2014), a survey of earlier temporal ER models (Gregersen and Jensen, 1999), and spatio-temporal aspects (e.g., (Parent *et al.*, 2006)). From a viewpoint of logic-based temporal knowledge representation and reasoning at the concept-level for UML and ER or EER, there are, a.o., (Artale *et al.*, 2002; Lutz *et al.*, 2008), who use temporal logic to describe object behaviour in an object oriented specification language, Troll (Hartmann *et al.*, 1994), and extensions to the case with the presence of data (Artale *et al.*, 2013; Baader *et al.*, 2013). The linking of a temporal logic to a conceptual data modelling language is an ongoing effort, with notable achievements with the Description Logic (DL) language $\mathcal{DLR}_{\mathcal{US}}$ and the temporally extended ER, called ER_{VT}, Entity Relationship model with Valid Time, (Artale *et al.*, 2002, 2008, 2007; Keet and Artale, 2010), which covers mainly the temporal behaviour of classes and relationship. However, for a temporal conceptual data model to be fully useful in information system development that has to deal with changing information, a proper treatment of *temporal attributes* is also necessary. Such information has to be captured also at the conceptual model layer during design instead of being ignored or only encoded in an implementation. For instance, if an employee is suspended for fraud investigation, the employee should (at least temporarily) not receive a salary either; once the boolean attribute for having received a transplant is set from no to yes, it must not be changed anymore; and modelling of attribute-based access rights that may change by time of day and changing role of the user. While one can bury such data characteristics for a universe of discourse in the application, the understandability and maintainability, and, hence, software quality, will be better if such information is properly encoded in the conceptual data model first.

This paper introduces temporal attributes for temporal conceptual data models. Attributes are a type of binary relations, restricted to linking objects to data values. A "temporary attribute" was formalised in (Artale *et al.*, 2007) to the extent to say what it is, but it had no effect on modelling in ER_{VT}, due to the absence of operational semantics and its interaction with temporal classes. More recent results have been obtained for temporal ontology-based data access (Artale *et al.*, 2013), for which attributes (or: OWL data properties) are important, but it omits temporal attributes, and recent comprehensive logic-based treatments of attributes considered only the atemporal case (Artale *et al.*, 2012; Savkovic and Calvanese, 2012). To the best of our knowledge, no logic-based characterisation, including constraints on the subsumption hierarchy in the temporal setting, exists for temporal attributes (note: a temporalisation of relationships does exist (Artale *et al.*, 2008; Keet and Artale, 2010)). This hampers its potential usability for temporal knowledge representation, reasoning, and information system development. $\mathcal{DLR}_{\mathcal{US}}$ was extended recently to represent temporal attributes (Ongoma *et al.*, 2014), but this has not been integrated with the modelling of temporal attributes especially in the conceptual data modelling setting and the consequences for class subsumption are unexplored.

In order to fill this gap in understanding and representation of temporal attributes, we approach it from a *modelling expressiveness viewpoint*, instead of the a priori limitation to a computationally well-behaved logic (of low expressiveness) or no logic. We propose a comprehensive treatment of temporal attributes using the extended $\mathcal{DLR}_{\mathcal{US}}$ as a foundation to formalise precisely the notion of temporal attributes and their interaction with temporal classes in subsumption hierarchies. The comprehensive formalisation has not only the assertion of temporary attribute and 'freezing' of an attribute, but, more importantly, a full set of constraints including *status attributes*, their logical implications, and the axioms and proofs for the subsumption cases. These results can be easily transferred to temporal relations to show the interaction between status relations and status classes. With this formal characterization for temporal attributes, the ER_{VT} language—which has its logic reconstruction for ER's (temporal) classes, (temporal) relationships, and plain attributes also with $\mathcal{DLR}_{\mathcal{US}}$—now has become a fully temporalised extended ER language.

While $\mathcal{DLR}_{\mathcal{US}}$ is undecidable, it is also sufficiently expressive to enable one to obtain insight into the language features required for a full temporalization of attributes. The natural next step is the investigation on scalability trade-offs, which then can be informed also by modelling trade-offs in addition to computational trade-offs and finally thick graphical interface.

The remainder of the paper is structured as follows. We provide examples of application areas of temporal attributes in Section 2 and the extended $\mathcal{DLR}_{\mathcal{US}}$ in Section 3. Status attributes, their semantics, and logical implications are described in Section 4. We discuss in Section 5 and conclude in Section 6.

2 Application areas

Time is ingrained in every aspect of our lives, and, hence, ought to be dealt with in information systems that have to handle such data. However, traditional databases do not have the ability to show this change, as, most of the time, old attributes and attribute values are replaced by newer ones which eliminates history in the database. Many applications do need to store the history of events by capturing how to represent this changing state, and there are business rules on what can change, and how, which a database or software application is supposed to adhere to. Examples of such applications that require temporal attributes include:

- Administration: a business rule that states that when an employee is on leave or facing fraud charges, some of his attributes (e.g., Salary) are suspended, or a CEO evolves to a non-executive board member and then no longer earns a salary, but will receive a yearly bonus, i.e., one attribute (Salary), becomes 'disabled' and another ('scheduled') attribute (hasBonus) becomes active upon migration of the object.
- Medical Information Systems with patient records to monitor the attributes that change with time, for example a HIV positive patient evolves to an AIDS patient after the attribute value of CD4 count drops below 180, or once the value for the boolean attribute transplant-received is set to 'yes', it cannot be changed anymore (it is so-called 'frozen'). A related application area is in pharmacovigilance (Lora *et al.*, 2012), which aims to monitor adverse effects of drugs on patients.
- Security systems that monitor and authorise users as they log into systems. Temporal attributes here are typically time-bound passwords the users are given, and disallowing to reuse the previous passwords. For large databases that use millions of users, security administrators can use attribute-based data access (ABAC) (Priebe *et al.*, 2007) to authenticate its users. More recently, it also covers specification of different attribute-based access privileges across time in mobile applications.
- Financial institutions need to record the history for future use, e.g. before giving a credit facility. Temporal attributes are used to determine the amount a client can receive and the time in which it can be repaid. Banks need to record the history of customer transactions ((deposit) and (withdrawal)), thus the need for temporal attributes.

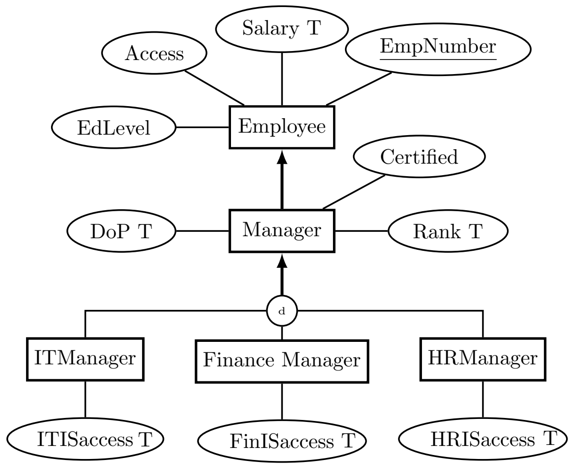

Figure 1: An ER_{VT} diagram showing temporal attributes and the interaction between status classes (rectangles) and status attributes (ovals); a "T" after the name denotes it is a temporal class or attribute.

We give a practical example in administration, for example in offices, schools, and companies, to manage information about employees, with a temporally extended ER diagram in Fig. 1, where a "T" after the name denotes temporal, as in ER_{VT} (Artale *et al.*, 2007).

Example 1. An employee in a company may have the following attributes: name, Access, Salary, EdLevel, and the identifier as EmpNumber. Employees may be promoted ('dynamically extend') to manager. A manager manages one of several departments, for example, a HR manager manages the HR department, IT manager manages the IT department and the Finance manager manages the finance department, and each departments has a different Information system, e.g. HRISaccess is the attribute assigned to the HR manager to access that system. Different managers also belong to different levels of management, from low level, mid-level to upper management. If an IT manager takes up the position of Finance manager, the ITISaccess attribute must change ('dynamically evolve') to FinanceISaccess. ♢

Practical examples with instances will be illustrated at the end of Section 4.2.

3 Preliminaries: The Temporal DL $\mathcal{DLR}_{\mathcal{US}}$ with Attributes

To provide a precise semantics of temporal attributes, we use a logic reconstruction for temporal conceptual data models in the temporal Description Logic $\mathcal{DLR}_{\mathcal{US}}$ extended with the operators $\mathcal{S}$ince and $\mathcal{U}$ntil, which has been used for that purpose already (Artale *et al.*, 2002, 2008, 2007; Keet and Artale, 2010), including how the temporally extended ER, ER_{VT}, maps into $\mathcal{DLR}_{\mathcal{US}}$ (Artale and Keet, 2008). $\mathcal{DLR}_{\mathcal{US}}$ (Artale *et al.*, 2002) is an expressive fragment of FOL that combines the propositional temporal logic with *Since* and *Until* operators with the (non-temporal) description logic $\mathcal{DLR}$ (Calvanese and De Giacomo, 2003) so that relationships, classes, and attributes can be temporalised. Details of $\mathcal{DLR}_{\mathcal{US}}$ can be found in (Artale *et al.*, 2002, 2007; Ongoma *et al.*, 2014); as usual, we have classes C (starting from atomic ones CN), n-ary relationships R (DL roles, with $n \geq 2$, RN), binary attributes A between a class and a datatype, DL role components (U, of which F denotes a role component in an attribute, $F \subseteq U$, and $F = \{\texttt{From}, \texttt{To}\}$). The selection expression $U_i/n : C$ denotes an n-ary relation whose i-th argument ($i \leq n$) is of type C and $[U_j]R$ denotes the j-th argument ($j \leq n$)—i.e., DL role component, which can be seen intuitively as a projection over the role—in role R (subscripts i and j are omitted if it is clear from the context). For F, which concerns the DL role components in an attribute, we thus have $F : C$, with F denoting the role component From that relates to class C, and $[F]A$ denoting the role component F of A, where if To is used, it is the DL role component that associates with the datatype of the attribute, and if From is used, it is the DL role component that associates with the class of the attribute. Thus, for each $A \in \mathcal{A}$ and denoting with Literal the top for data types (i.e., for the domain of values $\Delta^{\mathcal{I}}_D$; see below), the $\mathcal{DLR}_{\mathcal{US}}$ axiom $A \sqsubseteq \texttt{From} : \top \sqcap \texttt{To} : \texttt{Literal}$ holds. Finally, $\mathcal{U}$ntil and $\mathcal{S}$ince together with $\bot$ and $\top$ suffice to define the temporal operators: $\Diamond^+$ (some time in the future) as $\Diamond^+ C \equiv \top \,\mathcal{U}\, C$, $\oplus$ (at the next moment) as $\oplus\, C \equiv \bot \,\mathcal{U}\, C$, and likewise for their past counterparts; $\Box^+$ (always in the future) and $\Box^-$ (always in the past) are the duals of $\Diamond^+$ and $\Diamond^-$; the operators $\Diamond^*$(at some moment) and its dual $\Box^*$(at all moments) can be defined as $\Diamond^* C \equiv C \sqcup \Diamond^+ C \sqcup \Diamond^- C$ and $\Box^* C \equiv C \sqcap \Box^+ C \sqcap \Box^- C$, respectively. The syntax and semantics of the extended $\mathcal{DLR}_{\mathcal{US}}$ are included in Fig. 2. The model-theoretic semantics of $\mathcal{DLR}_{\mathcal{US}}$ assumes a flow of time $\mathcal{T} = \langle \mathcal{T}_p, < \rangle$, where $\mathcal{T}_p$ is a set of countably infinite time points also referred to as chronons and $<$ is isomorphic to the usual ordering on the integers. The language of $\mathcal{DLR}_{\mathcal{US}}$ is interpreted in temporal models over $\mathcal{T}$, which are triples in the form $\mathcal{I} = \langle \mathcal{T}, \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}(t)} \rangle$, where $\Delta^{\mathcal{I}}$ is the union of two non empty disjoint sets, the *domain of objects*, $\Delta^{\mathcal{I}}_O$, and *domain of values*, $\Delta^{\mathcal{I}}_D$, and $\cdot^{\mathcal{I}(t)}$ the interpretation function such that, for every $t \in \mathcal{T}$ ($t \in \mathcal{T}$ will be used as a shortcut for $t \in \mathcal{T}_p$), every class C, and every n-ary relation R, we have $C^{\mathcal{I}(t)} \subseteq \Delta^{\mathcal{I}}_O$ and $R^{\mathcal{I}(t)} \subseteq (\Delta^{\mathcal{I}}_O)^n$; also, $(u, v) = \{w \in \mathcal{T} \mid u < w < v\}$.

A *knowledge base* is a finite set Σ of $\mathcal{DLR}_{\mathcal{US}}$ axioms of the form $C_1 \sqsubseteq C_2$ and $R_1 \sqsubseteq R_2$, and with R_1 and R_2 being relations of the same arity. An interpretation $\mathcal{I}$ satisfies $C_1 \sqsubseteq C_2$ ($R_1 \sqsubseteq R_2$) if and only if the interpretation of C_1 (R_1) is included in the interpretation of C_2 (R_2) at all time, i.e. $C_1^{\mathcal{I}(t)} \subseteq C_2^{\mathcal{I}(t)}$ ($R_1^{\mathcal{I}(t)} \subseteq R_2^{\mathcal{I}(t)}$), for all $t \in \mathcal{T}$.

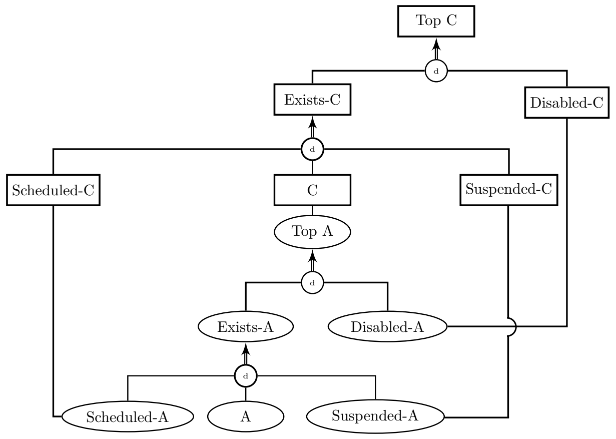

Figure 3: An EER diagram showing the interaction between status classes (rectangles) and status attributes (ovals).

4 Temporalizing Attributes

To effectively manage and present temporalised attributes, it is important to state a time-scale and give the constraints to manage the evolution of data. Timestamping (Artale *et al.*, 2007) ensures that data can be distinguished, while evolution constraints put restriction on the movement of data. These are rules that govern the valid state of the database or logic-based temporal conceptual data model. To represent temporal attributes to the level of detail required and to have them interact with temporalised classes, we extend the notion of status classes and status relations to *status attributes* to specify the operational semantics of temporal attributes.

Status models the normal behaviour in the real world: items either exist or not, also known as lifecycle (Artale *et al.*, 2007; Hartmann *et al.*, 1994), and models the evolution of data. Status classes were introduced in (Artale *et al.*, 2007) and status relations in (Artale *et al.*, 2008), but it fell short of status attributes to constrain the permissible states of affairs. Status attributes also can have four different statuses: they either exist and are scheduled, active, or suspended, or they are disabled. We describe each one of them informally and illustrate that they are relevant for conceptual modelling (and knowledge representation), and subsequently present the formalisation. Fig. 3 shows an integrated EER diagram of status classes extended with status attributes.

– *Scheduled*: an attribute is scheduled if it belongs to an active class or a scheduled class. For instance, a bonus payout to an employee occurs only after passing the probationary period successfully.
– *Active*: the status of an attribute is active if it fully instantiates the type-level attribute, thus, these are the normal attributes and they only belong to an active class and can be changed or deleted at any time in the class; e.g., an access level to certain information, date of birth, an employee's salary attribute.
– *Suspended*: These are attributes that belong to either a suspended class or an active class. They are temporarily inactive and will become active after a given period of time or until the suspended class becomes active. For instance, an employee is suspended due to an ongoing fraud case, so its attributes are suspended, too.

$$
\begin{array}{rl}
C \rightarrow & \top \mid \bot \mid CN \mid \neg C \mid C_1 \sqcap C_2 \mid \exists^{\leqslant k}[U_j]R \mid \exists[\mathbf{F}]\mathbf{A} \mid \\
& \Diamond^+ C \mid \Diamond^- C \mid \Box^+ C \mid \Box^- C \mid \oplus C \mid \ominus C \mid C_1\ \mathcal{U}\, C_2 \mid C_1\ \mathcal{S}\, C_2 \\
R \rightarrow & \top_n \mid RN \mid \neg R \mid R_1 \sqcap R_2 \mid U_i/n : C \mid \\
& \Diamond^+ R \mid \Diamond^- R \mid \Box^+ R \mid \Box^- R \mid \oplus R \mid \ominus R \mid R_1\ \mathcal{U}\, R_2 \mid R_1\ \mathcal{S}\, R_2 \\
\mathbf{A} \rightarrow & \top_{\mathbf{A}} \mid \mathbf{AN} \mid \neg \mathbf{A} \mid \mathbf{F} : \mathbf{C} \mid \\
& \Diamond^+ \mathbf{A} \mid \Diamond^- \mathbf{A} \mid \Box^+ \mathbf{A} \mid \Box^- \mathbf{A} \mid \oplus \mathbf{A} \mid \ominus \mathbf{A} \mid \mathbf{A_1}\ \mathcal{U}\, \mathbf{A_2} \mid \mathbf{A_1}\, \mathcal{S}\, \mathbf{A_2}
\end{array}
$$

$$
\begin{array}{rcl}
\top^{\mathcal{I}(t)} & = & \Delta^{\mathcal{I}}_O \\
\bot^{\mathcal{I}(t)} & = & \emptyset \\
CN^{\mathcal{I}(t)} & \subseteq & \top^{\mathcal{I}(t)} \\
(\neg C)^{\mathcal{I}(t)} & = & \top^{\mathcal{I}(t)} \setminus C^{\mathcal{I}(t)} \\
(C_1 \sqcap C_2)^{\mathcal{I}(t)} & = & C_1^{\mathcal{I}(t)} \cap C_2^{\mathcal{I}(t)} \\
(\exists^{\leqslant k}[U_j]R)^{\mathcal{I}(t)} & = & \{\ o \in \top^{\mathcal{I}(t)} \mid \sharp\{\langle o_1,\ldots,o_n\rangle \in R^{\mathcal{I}(t)} \mid o_j = o\} \leqslant k\} \\
(\exists\,[\mathbf{F}]\mathbf{A}^{\mathcal{I}(\mathbf{t})} & = & \{\ \mathbf{o} \in \top^{\mathcal{I}(\mathbf{t})} \mid \sharp\{\langle \mathbf{o},\mathbf{d}\rangle \in \mathbf{A}^{\mathcal{I}(\mathbf{t})} \geq \mathbf{1}\}\} \\
(C_1\, \mathcal{U}\, C_2)^{\mathcal{I}(t)} & = & \{\ o \in \top^{\mathcal{I}(t)} \mid \exists v > t.(o \in C_2^{\mathcal{I}(v)} \land \forall w \in (t,v).o \in C_1^{\mathcal{I}(w)})\} \\
(C_1\, \mathcal{S}\, C_2)^{\mathcal{I}(t)} & = & \{\ o \in \top^{\mathcal{I}(t)} \mid \exists v < t.(o \in C_2^{\mathcal{I}(v)} \land \forall w \in (v,t).o \in C_1^{\mathcal{I}(w)})\} \\
(\top_n)^{\mathcal{I}(t)} & = & (\Delta^{\mathcal{I}}_O)^n \\
RN^{\mathcal{I}(t)} & \subseteq & (\top_n)^{\mathcal{I}(t)} \\
(\neg R)^{\mathcal{I}(t)} & = & (\top_n)^{\mathcal{I}(t)} \setminus R^{\mathcal{I}(t)} \\
(R_1 \sqcap R_2)^{\mathcal{I}(t)} & = & R_1^{\mathcal{I}(t)} \cap R_2^{\mathcal{I}(t)} \\
(U_i/n : C)^{\mathcal{I}(t)} & = & \{\ \langle o_1,\ldots,o_n\rangle \in (\top_n)^{\mathcal{I}(t)} \mid o_i \in C^{\mathcal{I}(t)}\} \\
(R_1\, \mathcal{U}\, R_2)^{\mathcal{I}(t)} & = & \{\ \langle o_1,\ldots,o_n\rangle \in (\top_n)^{\mathcal{I}(t)} \mid \exists v > t.(\langle o_1,\ldots,o_n\rangle \in R_2^{\mathcal{I}(v)} \land \forall w \in (t,v).\ \langle o_1,\ldots,o_n\rangle \in R_1^{\mathcal{I}(w)})\} \\
(R_1\, \mathcal{S}\, R_2)^{\mathcal{I}(t)} & = & \{\ \langle o_1,\ldots,o_n\rangle \in (\top_n)^{\mathcal{I}(t)} \mid \exists v < t.(\langle o_1,\ldots,o_n\rangle \in R_2^{\mathcal{I}(v)} \land \forall w \in (v,t).\ \langle o_1,\ldots,o_n\rangle \in R_1^{\mathcal{I}(w)})\} \\
(\Diamond^+ R)^{\mathcal{I}(t)} & = & \{\langle o_1,\ldots,o_n\rangle \in (\top_n)^{\mathcal{I}(t)} \mid \exists v > t.\ \langle o_1,\ldots,o_n\rangle \in R^{\mathcal{I}(v)}\} \\
(\oplus R)^{\mathcal{I}(t)} & = & \{\langle o_1,\ldots,o_n\rangle \in (\top_n)^{\mathcal{I}(t)} \mid \langle o_1,\ldots,o_n\rangle \in R^{\mathcal{I}(t+1)}\} \\
(\Diamond^- R)^{\mathcal{I}(t)} & = & \{\langle o_1,\ldots,o_n\rangle \in (\top_n)^{\mathcal{I}(t)} \mid \exists v < t.\ \langle o_1,\ldots,o_n\rangle \in R^{\mathcal{I}(v)}\} \\
(\ominus R)^{\mathcal{I}(t)} & = & \{\langle o_1,\ldots,o_n\rangle \in (\top_n)^{\mathcal{I}(t)} \mid \langle o_1,\ldots,o_n\rangle \in R^{\mathcal{I}(t-1)} \\
(\top_{\mathbf{A}})^{\mathcal{I}(\mathbf{t})} & = & \Delta^{\mathcal{I}}_O \times \Delta^{\mathcal{I}}_D \\
\mathbf{AN}^{\mathcal{I}(\mathbf{t})} & \subseteq & (\top_{\mathbf{A}})^{\mathcal{I}(\mathbf{t})} \\
(\mathbf{F} : \mathbf{C})^{\mathcal{I}(\mathbf{t})} & = & \{\ \langle \mathbf{o},\mathbf{d}\rangle \in (\top_{\mathbf{A}})^{\mathcal{I}(\mathbf{t})} \mid \mathbf{o} \in \mathbf{C}^{\mathcal{I}(\mathbf{t})}\} \\
(\mathbf{A_1}\, \mathcal{U}\, \mathbf{A_2})^{\mathcal{I}(t)} & = & \{\ \langle \mathbf{o},\mathbf{d}\rangle \in (\top_{\mathbf{A}})^{\mathcal{I}(\mathbf{t})} \mid \exists \mathbf{v} > \mathbf{t}.(\langle \mathbf{o},\mathbf{d}\rangle \in \mathbf{A_2}^{\mathcal{I}(\mathbf{v})} \land \forall \mathbf{w} \in (\mathbf{t},\mathbf{v}).\langle \mathbf{o},\mathbf{d}\rangle \in \mathbf{A_1}^{\mathcal{I}(\mathbf{w})})\} \\
(\mathbf{A_1}\, \mathcal{S}\, \mathbf{A_2})^{\mathcal{I}(t)} & = & \{\ \langle \mathbf{o},\mathbf{d}\rangle \in (\top_{\mathbf{A}})^{\mathcal{I}(\mathbf{t})} \mid \exists \mathbf{v} < \mathbf{t}.(\langle \mathbf{o},\mathbf{d}\rangle \in \mathbf{A_2}^{\mathcal{I}(\mathbf{v})} \land \forall \mathbf{w} \in (\mathbf{v},\mathbf{t}).\langle \mathbf{o},\mathbf{d}\rangle \in \mathbf{A_1}^{\mathcal{I}(\mathbf{w})})\} \\
(\Diamond^+ \mathbf{A})^{\mathcal{I}(t)} & = & \{\langle \mathbf{o},\mathbf{d}\rangle \in (\top_{\mathbf{A}})^{\mathcal{I}(\mathbf{t})} \mid \exists \mathbf{v} > \mathbf{t}.\langle \mathbf{o},\mathbf{d}\rangle \in \mathbf{A}^{\mathcal{I}(\mathbf{v})}\} \\
(\oplus \mathbf{A})^{\mathcal{I}(t)} & = & \{\langle \mathbf{o},\mathbf{d}\rangle \in (\top_{\mathbf{A}})^{\mathcal{I}(\mathbf{t})} \mid \langle \mathbf{o},\mathbf{d}\rangle \in \mathbf{A}^{\mathcal{I}(\mathbf{t+1})}\} \\
(\Diamond^- \mathbf{A})^{\mathcal{I}(t)} & = & \{\langle \mathbf{o},\mathbf{d}\rangle \in (\top_{\mathbf{A}})^{\mathcal{I}(\mathbf{t})} \mid \exists \mathbf{v} < \mathbf{t}.\langle \mathbf{o},\mathbf{d}\rangle \in \mathbf{A}^{\mathcal{I}(\mathbf{v})}\} \\
(\ominus \mathbf{A})^{\mathcal{I}(t)} & = & \{\langle \mathbf{o},\mathbf{d}\rangle \in (\top_{\mathbf{A}})^{\mathcal{I}(\mathbf{t})} \mid \langle \mathbf{o},\mathbf{d}\rangle \in \mathbf{A}^{\mathcal{I}(\mathbf{t-1})}\}
\end{array}
$$

Figure 2: Syntax and semantics of $\mathcal{DLR_{US}}$, modified to include attributes (in bold face); o denote objects, d domain values, $v, w, t \in \mathcal{T}, F$ is a role component in an attribute.

- *Disabled*: These belong to either a disabled class or an active class. When the membership of the class has expired, its attributes are also disabled, and it can also be true for an active class that no longer requires the use of that attribute. For instance, an employee dies, so her profile is disabled, or a software company decides to change the application from for-payment to free and open source, so that the price attribute becomes disabled.

Concerning the formalization, the subsumption hierarchy and disjointness depicted in Fig. 3 are straightforward and omitted from the presentation here for brevity. We give both the model-theoretic semantics and the $\mathcal{DLR_{US}}$ axiom; a is an element in A, which can also be written as $\langle o, d\rangle$, where o represents an object in the class and d is the value domain of the attribute. We assume that the name of the attribute denotes the set of active attributes.

(Aexists1) *An Attribute either Exists or is Disabled.*
$a \in \mathit{Exists}\text{-}A^{\mathcal{I}(t)} \rightarrow \forall t'{>}t.a \in (\mathit{Exists}\text{-}A^{\mathcal{I}(t')} \lor \mathit{Disabled}\text{-}A^{\mathcal{I}(t')})$
$\texttt{Exists-A} \sqsubseteq \Box^+(\texttt{Exists-A} \sqcup \texttt{Disabled-A})$

(Aexists2) *Exist attribute involves scheduled, suspended or active attributes.*
$a \in \mathit{Exists}\text{-}A^{\mathcal{I}(t)} \rightarrow \forall t'{>}t.a \in (\mathit{Scheduled}\text{-}A^{\mathcal{I}(t')} \lor A^{\mathcal{I}(t')} \lor \mathit{Suspended}\text{-}A^{\mathcal{I}(t')})$
$\texttt{Exists-A} \sqsubseteq \Box^+(\texttt{Scheduled-A} \sqcup \texttt{A} \sqcup \texttt{Suspended-A})$

(Aexists3) *Existing attributes belong to an existing class.*
$\langle o, d\rangle \in \mathit{Exists}\text{-}A^{\mathcal{I}(t)} \rightarrow o \in \mathit{Exists}\text{-}C^{\mathcal{I}(t)}$
$\texttt{Exists-A} \sqsubseteq \texttt{From} : \texttt{Exists-C}$

(Aact1) *Active attributes belong to an active class only.*
$\langle o, d\rangle \in A^{\mathcal{I}(t)} \rightarrow o \in C^{\mathcal{I}(t)}$
$\texttt{A} \sqsubseteq \texttt{From} : \texttt{C}$

(Asch1) *Scheduled attribute will eventually become active.*
$a \in \mathit{Scheduled}\text{-}A^{\mathcal{I}(t)} \rightarrow \exists t'{>}t.a \in A^{\mathcal{I}(t')}$
$\texttt{Scheduled-A} \sqsubseteq \Diamond^+\texttt{A}$

(Asch2) *Scheduled attribute can never follow active.*
$a \in A^{\mathcal{I}(t)} \rightarrow \forall t'{>}t.a \notin \mathit{Scheduled}\text{-}A^{\mathcal{I}(t')}$
$\texttt{A} \sqsubseteq \Box^+\neg\texttt{Scheduled-A}$

(Asusp1) *Suspended attribute was active in the past.*
$a \in \mathit{Suspended}\text{-}A^{\mathcal{I}(t)} \rightarrow \exists t'{<}t.a \in A^{\mathcal{I}(t')}$
$\texttt{Suspended-A} \sqsubseteq \Diamond^-\texttt{A}$

(Asusp2) *Suspended attributes belong to active or suspended class.*
$\langle o, d\rangle \in \mathit{Suspended}\text{-}A^{\mathcal{I}(t)} \rightarrow o \in (\mathit{Suspended}\text{-}C^{\mathcal{I}(t)} \lor C^{\mathcal{I}(t)})$
$\texttt{Suspended-A} \sqsubseteq \texttt{From} : (\texttt{Suspended-C} \sqcup \texttt{C})$

(Adisab1) *Disabled persists.*
$a \in \mathit{Disabled}\text{-}A^{\mathcal{I}(t)} \rightarrow \forall t'{>}t.a \in \mathit{Disabled}\text{-}A^{\mathcal{I}(t')}$
$\texttt{Disabled-A} \sqsubseteq \Box^+\texttt{Disabled-A}$

(Adisab2) *Disabled attribute was active in the past.*
$a \in \mathit{Disabled}\text{-}A^{\mathcal{I}(t)} \rightarrow \exists t'{<}t.a \in A^{\mathcal{I}(t')}$
$\texttt{Disabled-A} \sqsubseteq \Diamond^-\texttt{A}$

(ADISAB3) *Disabled attributes belong to a disabled or active class.*
$\langle o,d\rangle \in \mathit{Disabled\text{-}A}^{\mathcal{I}(t)} \to o \in (\mathit{Disabled\text{-}C}^{\mathcal{I}(t)} \vee C^{\mathcal{I}(t)})$
$\mathtt{Disabled\text{-}A} \sqsubseteq \mathtt{From} : (\mathtt{Disabled\text{-}C} \sqcup \mathtt{C})$

(ADISAB5) *Disabled will never become active again.*
$a \in \mathit{Disabled\text{-}A}^{\mathcal{I}(t)} \to \forall t' > t.a \notin A^{\mathcal{I}(t')}$
$\mathtt{Disabled\text{-}A} \sqsubseteq \Box^{+}\neg\mathtt{A}$

(CSUSP3) *Freezing attributes of suspended classes / Suspended class has suspended attributes only.*
$o \in \mathit{Suspended\text{-}C}^{\mathcal{I}(t)} \to \langle o,d\rangle \in \mathit{Suspended\text{-}A}^{\mathcal{I}(t)}$
$\mathtt{Suspended\text{-}C} \sqsubseteq \forall[\mathtt{From}]\mathtt{Suspended\text{-}A}$

(CACTIVEA) *An active class contains exists or disabled attributes.*
$o \in C^{\mathcal{I}(t)} \to \langle o,d\rangle \in (\mathit{Exists\text{-}A}^{\mathcal{I}(t)} \vee \mathit{Disabled\text{-}A}^{\mathcal{I}(t)})$
$\mathtt{C} \sqsubseteq \forall[\mathtt{From}](\mathtt{Exists\text{-}A} \sqcup \mathtt{Disabled\text{-}A})$

Henceforth, we denote with Σ_{sa} the above set of $\mathcal{DLR_{US}}$ axioms that formalise status attributes, and the two for status classes.

Some of these axioms are quite similar to those for relationships that were introduced in (Artale *et al.*, 2008), others are specific to attributes. Similar ones are: AACT1 corresponds to (Artale *et al.*, 2008)'s ACT, ADISAB1 to RDISAB1, ADISAB2 to RDISAB2, ASUSP1 to RSUSP1, ASUSP2 to RSUSP2, ASCH1 to RSCH1, and ASCH2 to RSCH2. The new ones specific to attributes are: AEXISTS1, AEXISTS2, AEXISTS3, ADISAB3, and CACTIVEA.

CSUSP3 requires some explanation, for we use this one instead of (Artale *et al.*, 2007)'s FREEZ axiom. Artale et al.'s FREEZ intends to capture *"Freezing attributes of suspended classes"* so as "to make unchangeable the attributes of suspended objects, the unchangeability starting at the time instant the object becomes suspended" (Artale *et al.*, 2007) whose idea originated from (Etzion *et al.*, 1998), and has the following semantics: $o \in \mathit{Suspended\text{-}C}^{\mathcal{I}(t)} \wedge \langle o,a\rangle \in A^{\mathcal{I}(t)} \to \langle o,a\rangle \in A^{\mathcal{I}(t+1)}$ and in $\mathcal{DLR_{US}}$ notation $\mathtt{Suspended\text{-}C} \sqsubseteq \neg\exists[\mathtt{From}](\mathtt{A}\sqcap\ominus\mathtt{A})$ (Artale *et al.*, 2007). However, suspended requires that it was active in the past, which applies to classes, relationships, and attributes (ASUSP1), and it is not just that the attribute may not be 'not-active', but, precisely, it has to be suspended, too. Hence, CSUSP3 captures the 'freezing' more precisely.

Observe that, as with ACT for relations (Active relations involve only active classes) (Artale *et al.*, 2008), AACT1 cannot be proved and therefore had to be added to the set of basic constraints: while by AEXISTS3, we know an attribute has to be either scheduled, active, or suspended, and one can exclude suspended thanks to CSUSP3, one cannot contradict $a \in \mathit{Scheduled\text{-}A}^{\mathcal{I}(t)}$ due to ASCH3, as scheduled attributes may belong to either scheduled or active classes.

4.1 Status Attributes: Logical Implications

Logical implications are important to be able to derive new constraints. From the $\mathcal{DLR_{US}}$ axioms above (Σ_{sa}), we can obtain the following logical implications for status attributes.

Proposition 1. (Status Attributes: Logical Implications) *Given the set of axioms Σ_{sa} and an attribute, $\mathtt{A} \sqsubseteq \mathtt{From} : \mathtt{C} \sqcap \mathtt{To} : \mathtt{D}$, with $\mathtt{D}$ a data type, the following logical implications hold:*

(CSCH) *Scheduled class has scheduled attributes only.*
$\Sigma_{sa} \models \mathtt{Scheduled\text{-}C} \sqsubseteq \forall[\mathtt{From}]\mathtt{Scheduled\text{-}A}$

(ASCH3) *Scheduled attribute belongs to an active or scheduled class.*
$\Sigma_{sa} \models \mathtt{Scheduled\text{-}A} \sqsubseteq \mathtt{From} : (\mathtt{Scheduled\text{-}C} \sqcup \mathtt{C})$

(CDISAB4) *Disabled class has only disabled attributes.*
$\Sigma_{sa} \models \mathtt{Disabled\text{-}C} \sqsubseteq \forall[\mathtt{From}]\mathtt{Disabled\text{-}A}$

(ASCH4) *Scheduled attribute persists until active.*
$\Sigma_{sa} \models \mathtt{Scheduled\text{-}A} \sqsubseteq \mathtt{Scheduled\text{-}A}\ \mathcal{U}\ \mathtt{A}$

(ASCH5) *Scheduled attribute cannot evolve directly to disabled.*
$\Sigma_{sa} \models \mathtt{Scheduled\text{-}A} \sqsubseteq \oplus\neg\mathtt{Disabled\text{-}A}$

(AACT2) *Active attribute will possibly evolve into suspended or disabled.*
$\Sigma_{sa} \models \mathtt{A} \sqsubseteq \Box^{+}(\mathtt{A} \sqcup \mathtt{Suspended\text{-}A} \sqcup \mathtt{Disabled\text{-}A})$

(ASUSP3) *Suspended attributes can never be followed by scheduled or disabled.*
$\Sigma_{sa} \models \mathtt{Suspended\text{-}A} \sqsubseteq \oplus(\neg\mathtt{Scheduled\text{-}A} \sqcup \neg\mathtt{Disabled\text{-}A})$

(ADISAB4) *Disabled attribute cannot belong to a scheduled or suspended class.*
$\Sigma_{sa} \models \mathtt{Disabled\text{-}A} \sqsubseteq \neg(\mathtt{From} : \mathtt{Scheduled\text{-}C} \sqcup \mathtt{From} : \mathtt{Suspended\text{-}C})$

Proof. See Appendix A. □

The analogues of the logical implications for attributes to those for temporal relations are that ASCH4 corresponds to RSCH3 and ASCH5 to RSCH4 in (Artale *et al.*, 2008). The new one specific to attributes are: CSCH, CDISAB4, ASCH3, AACT2, ASUSP3, and ADISAB4.

4.2 Inheritance and Temporal Attributes

Class hierarchies are relevant for conceptual modelling, and even more so for ontologies. Five subsumption (ISA) implications for temporalised class subsumption with respect to status classes were proven in (Artale *et al.*, 2007), with A and B being classes:

(ISA1) *Objects active in B must be active in A*, i.e. $\mathtt{B} \sqsubseteq \mathtt{A}$,

(ISA2) *Objects suspended in B must be either suspended or active in A*, i.e., $\mathtt{Suspended\text{-}B} \sqsubseteq (\mathtt{Suspended\text{-}A} \sqcup \mathtt{A})$,

(ISA3) *Objects disabled in B must be either disabled, suspended or active in A*, i.e., $\mathtt{Disabled\text{-}B} \sqsubseteq (\mathtt{Disabled\text{-}A} \sqcup \mathtt{A} \sqcup \mathtt{Suspended\text{-}A})$,

(ISA4) *Objects scheduled in B must exist in A*, i.e., $\mathtt{Scheduled\text{-}B} \sqsubseteq \mathtt{Exists\text{-}A}$,

(ISA5) *Objects disabled in A, and active in B in the past, must be disabled in B*, i.e., $\mathtt{Disabled\text{-}A} \sqcap \Diamond^{-}\mathtt{B} \sqsubseteq \mathtt{Disabled\text{-}B}$.

For attributes, one cannot simply replace 'object' with 'attribute of object', however, because class A may not have that attribute. Moreover, the representation of attribute hierarchies is uncommon, but attribute inheritance is important, and therewith the interaction between the permissible statuses in the temporal class subsumption with the temporal attributes. Such consequences for subsumption and their interaction with classes in a hierarchy have not been specified for status relations yet either (Artale *et al.*, 2008; Keet and Artale, 2010).

Table 1: Illustrations of logical implication of status attributes' interaction with status classes in the subsumption hierarchy, as they may be declared for some conceptual model for an application in an organisation.

Logical Implication	**Examples of the intended behaviour it can enforce**
(ISA1$_A$) Attributes of objects disabled in C, and active in D in the past, are disabled in D	When the production of a product is discontinued, the whole object and its attributes are disabled therefore the same attributes in the subclasses are also disabled
(ISA2$_A$) Attributes active in D must be either active in C or is not present in C	These are normal (temporal) attributes; e.g., salary, price of a product
(ISA3$_A$) Attributes suspended in D must be either suspended in C, or is not present in C	Going on leave, the manager's access rights can be suspended, and also as employee the access rights are suspended (but not the object itself)
(ISA4$_A$) Attributes of objects suspended in D must be either suspended in C, or is not present in C	A manager is target of a fraud case, so the object and its attributes are suspended, and as a result its attributes are suspended also as an employee
(ISA5$_A$) Attributes disabled in D must be either disabled in C, or is not present in C	When a company that produces software applications makes one of its software applications open source, the price attribute is disabled therefore the attribute price in the superclass is also disabled
(ISA6$_A$) Attributes of objects disabled in D must be either disabled in C or is not present in C	A manager leaves the company, hence, becomes a member of the Disabled-Manager class, and so will its attributes be disabled. Then also the objects' attributes for it's superclass Employee must be disabled.
(ISA7$_A$) Attributes scheduled in D must be either scheduled in C or is not present in C	Attributes can be scheduled when an employee expects to get his usual salary after probation period: attribute prob_salary is used during probation and after probation, the scheduled salary becomes active.
(ISA8$_A$) Attributes of objects scheduled in D must be either scheduled in C, or is not present in C	A manager is scheduled to start work in a few weeks, the attribute salary will be scheduled in the employee class.

To prove the logical implications for subsumption with respect to temporal attributes, we first recall the very notion of class (entity type) subsumption. Let C and D be classes, and D $\sqsubseteq$ C, which means that all instances of D are also instances of C. This is achieved in a logical theory iff D has either one of the following: 1) the same attributes as C and possibly more 2) more relationships than C, 3) more constrained attributes or relationships than C, 4) If the attribute is active in C, then it must be active in D, otherwise it would deduce C $\sqsubseteq$ D, and trivially, if D does not have some attribute A or relationship R, then neither does C. We call this the *subsumption premise*.

We obtain 8 logical implications and their proofs, which are included in Proposition 2. General, high-level examples of real world scenarios that illustrate the intuition of the logical implications are shown in Table 1, and more detailed examples are described afterward. Attributes of a class—e.g., where $a \in A^{\mathcal{I}(t)}$ and a denotes the tuple $\langle o, d\rangle \in A^{\mathcal{I}(t)}$ and $o \in C^{\mathcal{I}(t)}$—are written in shorthand notation, like $A_C^{\mathcal{I}(t)}$ or in DL notation $\mathtt{A}_\mathtt{C}$, and to prevent wieldy subscripts, we use the following abbreviations for classes: Active C, Scheduled Sch-C, Suspended Sus-C, and Disabled Dis-C; so, e.g., $a \in \mathit{Disabled}\text{-}A_{\mathit{Dis}\text{-}C}^{\mathcal{I}(t)}$ represents a disabled attribute in a disabled class C.

Proposition 2. (Status Attributes and Status Classes: ISA **Logical Implications)** *Given the set of axioms Σ_{sa}, attribute* A *and* D $\sqsubseteq$ C*, where,* D $\sqsubseteq$ [From]$\mathtt{A}_\mathtt{D}$ *and, if present,* C $\sqsubseteq$ [From]$\mathtt{A}_\mathtt{C}$*, at time t, o is the object and $a = \langle o, d\rangle$ is the attribute of the object in the class, the following logical implications hold:*

(ISA1$_A$) *Attributes of objects disabled in C, and active in D in the past, are disabled in D.*
$\mathtt{Disabled\text{-}A}_{\mathtt{Dis\text{-}C}} \sqcap \Diamond^- \mathtt{A}_\mathtt{D} \sqsubseteq \mathtt{Disabled\text{-}A}_{\mathtt{Dis\text{-}D}} \sqcup \neg\mathtt{Top\text{-}A}_{\mathtt{Top\text{-}C}}$

(ISA2$_A$) *Attributes active in D must be either active in C or is not present in C.*
$\mathtt{A}_\mathtt{D} \sqsubseteq \mathtt{A}_\mathtt{C} \sqcup \neg\mathtt{Top\text{-}A}_{\mathtt{Top\text{-}C}}$

(ISA3$_A$) *Attributes suspended in D must be either suspended in C, or is not present in C.*
$\mathtt{Suspended\text{-}A}_\mathtt{D} \sqsubseteq \mathtt{Suspended\text{-}A}_\mathtt{C} \sqcup \neg\mathtt{Top\text{-}A}_{\mathtt{Top\text{-}C}}$

(ISA4$_A$) *Attributes of objects suspended in D must be either suspended in C, or is not present in C.*
$\mathtt{Top\text{-}A}_{\mathtt{Sus\text{-}D}} \sqsubseteq \mathtt{Suspended\text{-}A}_{\mathtt{Sus\text{-}C}} \sqcup \neg\mathtt{Top\text{-}A}_{\mathtt{Top\text{-}C}}$

(ISA5$_A$) *Attributes disabled in D must be either disabled in C, or is not present in C.*
$\mathtt{Disabled\text{-}A}_\mathtt{D} \sqsubseteq \mathtt{Disabled\text{-}A}_\mathtt{C} \sqcup \neg\mathtt{Top\text{-}A}_{\mathtt{Top\text{-}C}}$

(ISA6$_A$) *Attributes of objects disabled in D must be either disabled in C or is not present in C.*
$\mathtt{Top\text{-}A}_{\mathtt{Dis\text{-}D}} \sqsubseteq \mathtt{Disabled\text{-}A}_{\mathtt{Dis\text{-}C}} \sqcup \neg\mathtt{Top\text{-}A}_{\mathtt{Top\text{-}C}}$

(ISA7$_A$) *Attributes scheduled in D must be either scheduled in C or is not present in C.*
$\mathtt{Scheduled\text{-}A}_\mathtt{D} \sqsubseteq \mathtt{Scheduled\text{-}A}_\mathtt{C} \sqcup \neg\mathtt{Top\text{-}A}_{\mathtt{Top\text{-}C}}$

(ISA8$_A$) *Attributes of objects scheduled in D must be either scheduled in C, or is not present in C.*
$\mathtt{Top\text{-}A}_{\mathtt{Sch\text{-}D}} \sqsubseteq \mathtt{Scheduled\text{-}A}_{\mathtt{Sch\text{-}C}} \sqcup \neg\mathtt{Top\text{-}A}_{\mathtt{Top\text{-}C}}$

Proof. See Appendix B. □

We illustrate the effects of ISA3$_A$ and ISA7$_A$ and ISA8$_A$ with two practical examples of a conceptual model and a corresponding database with some sample data. Note that the status classes and status attributes are orthogonal to the standard active classes and attributes, and are therefore not drawn in the diagram.

Example 2. Consider the ER diagram fragment on the left in Fig. 4 and a possible database state on the right, where t0 and t1 are represented in a separate column in the desired granularity (such as day, month, or hour—not shown). The basic $\mathcal{DLR}_{\mathcal{US}}$ axioms are shown on the left-hand side of the figure, where Name is rigid, so that the key is not allowed to change (the "$\Box^*$Name"-part) and bonuses may change (the "$\Diamond^+$Bonus"). Employee Joanne Soap is a manager earning 6 000 000 a year, and as a manager—unlike regular employees—she also receives a bonus, which is set to 50 000, and she is in the table of 'active' managers. At some point in time, she, the object as instance of the Manager entity type, is suspended due to a fraud investigation and moves to the table for suspended managers, mgrSuspended, (step 1). The suspension of the object induces suspension of its attributes (because of CSUSP3), hence Joanne's Name, Salary, and Bonus attributes are suspended, and thus cannot be modified (step 2). By the object's suspension as member of the Manager entity

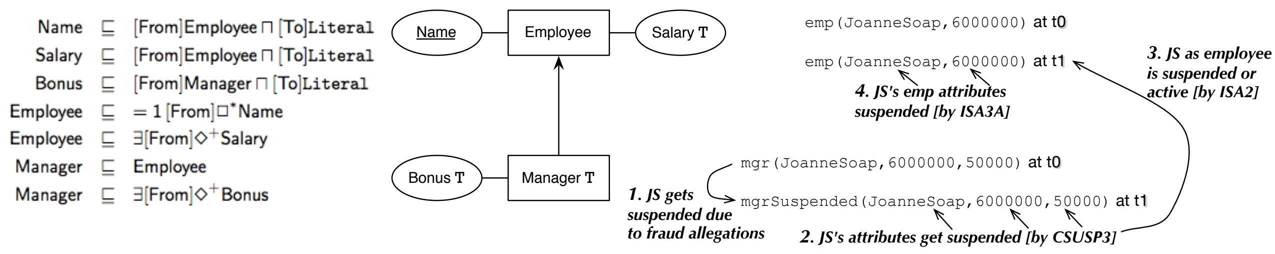
Name ⊑ [From]Employee ⊓ [To]Literal
Salary ⊑ [From]Employee ⊓ [To]Literal
Bonus ⊑ [From]Manager ⊓ [To]Literal
Employee ⊑ = 1 [From]□*Name
Employee ⊑ ∃[From]◇+Salary
Manager ⊑ Employee
Manager ⊑ ∃[From]◇+Bonus
Name
Employee
Salary T
Bonus T
Manager T
emp(JoanneSoap,6000000) at t0
emp(JoanneSoap,6000000) at t1
4. JS's emp attributes suspended [by ISA3A]
3. JS as employee is suspended or active [by ISA2]
mgr(JoanneSoap,6000000,50000) at t0
1. JS gets suspended due to fraud allegations
mgrSuspended(JoanneSoap,6000000,50000) at t1
2. JS's attributes get suspended [by CSUSP3]

Figure 4: Depiction of a scenario on attribute suspension; left: section of an extended ER_{VT} diagram and the basic $\mathcal{DLR}_{\mathcal{US}}$ axioms; right: possible database states with illustration of a change. There are several options to adorn diagrams with temporal information and to encode the statuses and time in the database and one is arbitrarily chosen here.

type, the object as an instance of its super entity type, Employee, can either remain active or be also suspended, thanks to ISA2 (step 3). Further, because of the suspension of attributes in the the subtype, the attributes that are also present in the supertype, being Name and Salary, will also be suspended, thanks to ISA3_A (step 4). (note: strictly speaking, steps 3 and 4 follow concurrently from step 2).

It may look 'odd' that Joanne remains a member of the active emp, whereas as manager she is suspended. The reason for this is, that it cannot be proven that if an attribute is suspended, then therefore the object must be suspended, and likewise for relationships. It depends on the business rule what should happen. $\diamondsuit$

Example 3 illustrates the effects of ISA7_A and ISA8_A with a practical example, also intended only to illustrate in an intuitive manner the effect of the axioms.

Example 3. Using the motivating example in Section 2, a manager, Joe Soap, is recruited and is scheduled to start work sometime in the future, say t_1, so all his attributes are scheduled at t_0 (where $t_0 < t_1$), thanks to CSCH, until he becomes active in the database, i.e., starts working for the company. The employee details can be added into the database at time t_0, with the manager details recorded as scheduled, being an employee ID JS123 and salary of 6 000 000 (step 1 in Fig. 5). Thanks to ISA8_A, both the object and the attribute Salary in the employee class (database table in the implementation) will automatically become scheduled (step 2) until the object, an instance of Manager, becomes active (step 3), ensuring that the new employee will not be paid before commencing to work. Put differently: it would result in an inconsistency to have Salary active in the employee class and scheduled in the manager class, thanks to ISA7_A. At time t_1, when the object is activated (step 3), ISA1 forces the employee instance to become active as well (step 4), and by ISA2_A, we see that the attribute Salary is active in the employee class, after the manager class is active (step 5). $\diamondsuit$

These examples show the need for temporal attributes and their status in databases, which now can be modelled at the conceptual modelling layer. Also, additional business rules can be specified and manually added, yet adhering to the logical implications.

This completes the logical implications with respect to subsumption in the context of status classes and status attributes. The interaction between subsumption for status classes that participate in status relations is expected to yield similar implications.

5 Discussion

The main contribution—and focus—of this paper is the comprehensive temporalisation of attributes with a rigorously defined semantics, which has shown to be feasible within the well-researched framework with $\mathcal{DLR}_{\mathcal{US}}$ and ER_{VT}. The full treatment of temporal attributes was the remaining gap towards obtaining a logic-based fully temporalised modelling language, which is an essential component for designing temporal databases and knowledge bases. An advantage of using $\mathcal{DLR}_{\mathcal{US}}$ over another logic, is that the temporal attributes integrate well with the temporal entity types and relationships already characterised with $\mathcal{DLR}_{\mathcal{US}}$ axioms, compared to having to start from scratch with a new logic. The major benefit will be reaped for the modelling of temporal information in such a way that it is consistent, therewith preventing bugs in the information system, thanks to the logic-based reconstruction of the temporally extended conceptual data modelling language in $\mathcal{DLR}_{\mathcal{US}}$, the status classes, relationships, and attributes, and their logical implications.

With the new insights presented in this paper, one can successfully build a *complete* temporal ER model that would be translated from the temporally extended EER to $\mathcal{DLR}_{\mathcal{US}}$ and therewith enable the option to check the consistency of a conceptual schema, hence, improve or guarantee its quality. For instance, it will disallow the undesirable state where all employees and their attributes are suspended but managers still receive a salary, or forcing a customer through a payment procedure for a payment of 0.0 Rand to download free software. Similarly for Example 3, where the employee is recorded in the database but has not yet commenced working: our temporal model can spot inconsistencies by having the scheduled class which would then cause some attributes in the employee class to be scheduled until the manager is active at work. Finally, recall Example 1 from Section 2 regarding a manager's access rights: now if a system administrator mistakenly gave a manager a higher access than he is allowed to have, our results can spot the consistency and recommend the correct solutions to be addressed. For instance, suppose an HRManager's HRISaccess is not active but his manager access profile is active, we see by (ISA2_A), it would automatically show this inconsistency and alert the system administrator.

The results presented here are the final theoretical step toward creation of a tool that would check automatically the consistency of temporal data models and thus would be able to spot such issues. Although the notation chosen here is in EER, the results can be transferred easily to other conceptual models, such as UML Class Diagrams and ORM2, thanks to the unifying metamodel from (Keet and Fillottrani, 2013) and similar formal and conceptual unification attempts.

$\mathcal{DLR}_{\mathcal{US}}$ is undecidable, however, with the known consequences for a potential automated reasoner for

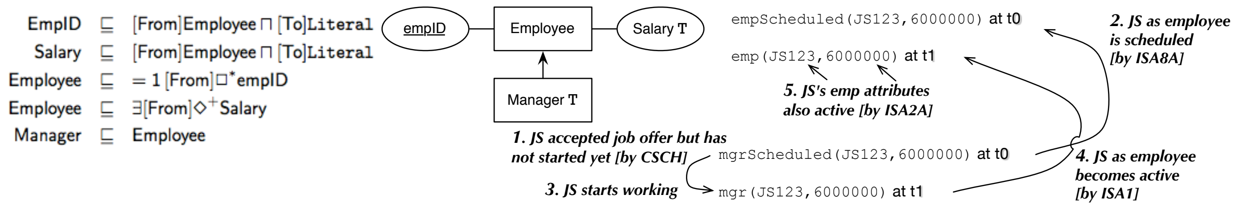
EmpID ⊑ [From]Employee ⊓ [To]Literal
Salary ⊑ [From]Employee ⊓ [To]Literal
Employee ⊑ = 1 [From]□*empID
Employee ⊑ ∃[From]◇+Salary
Manager ⊑ Employee
empID
Employee
Salary T
Manager T
empScheduled(JS123,6000000) at t0
emp(JS123,6000000) at t1
2. JS as employee is scheduled [by ISA8A]
5. JS's emp attributes also active [by ISA2A]
1. JS accepted job offer but has not started yet [by CSCH]
mgrScheduled(JS123,6000000) at t0
3. JS starts working
mgr(JS123,6000000) at t1
4. JS as employee becomes active [by ISA1]

Figure 5: Center: Partial extended ER_{VT} diagram for a scenario for scheduled attributes in a scheduled subclass and Manager and Salary declared temporal; Left: a $\mathcal{DLR}_{\mathcal{US}}$ notation; Right: some sample data informally demonstrating the effect of the axioms.

it. While time-consuming computation is an acceptable trade-off for a modeller focussing on expressiveness, a slightly less expressive temporal language and better performance with the reasoner may be preferred by others. At the other end of this spectrum are results obtained with TDL-Lite and temporal Ontology-Based Data Access (OBDA) (Artale *et al.*, 2013). It would be useful to conduct a detailed investigation as to what would be the best trade-off between a subset of temporal constructs that is most desired from a viewpoint of conceptual data modelling and the complexity 'costs' and what can be implemented in temporal OBDA. Once a temporal conceptual data modelling tool is available, one can obtain quantitative results as to how often a construct is actually used, which further can inform the notion of 'preferred constructs' and the further development of decidable temporal logics.

Finally, $\mathcal{DLR}_{\mathcal{US}}$ being in the DL family of languages, the advances described here can be relatively easily transferred to an ontology engineering setting.

6 Conclusion

A refined description logic language $\mathcal{DLR}_{\mathcal{US}}$ was used to properly include temporal constructors for attributes in its syntax and semantics. These constructors enabled the specification of a logic-based semantics for fully temporalised attributes, aided by the notion of status common in temporal conceptual data modelling. In addition, the logical implications for subsumption have been proven, in particular the interaction between status classes and status attributes in a subsumption hierarchy, thereby providing rules for effectively modelling and managing temporal data.

With these results obtained, the next natural step is to extend the ER_{VT} graphical component of the temporal conceptual data modelling language and develop a modelling tool to make the features easily available to modellers. A further step is to check automatically the properties of the temporal schema for satisfiability and subsumption.

References

Artale, A. and Keet, C. M. (2008). Essential, mandatory, and shared parts in conceptual data models. In T. Halpin, H. Proper, and J. Krogstie, editors, *Innovations in Information Systems modeling: Methods and Best Practices*, Advances in Database Research Series, pages 17–52. IGI Global.

Artale, A., Franconi, E., Wolter, F., and Zakharyaschev, M. (2002). A temporal description logic for reasoning about conceptual schemas and queries. In S. Flesca, S. Greco, N. Leone, and G. Ianni, editors, *Proc. of the 8th JELIA-02*, volume 2424 of *LNAI*, pages 98–110. Springer Verlag.

Artale, A., Parent, C., and Spaccapietra, S. (2007). Evolving objects in temporal information systems. *Ann. Math. Artif. Intell.*, **50**(1-2), 5–38.

Artale, A., Guarino, N., and Keet, C. M. (2008). Formalising temporal constraints on part-whole relations. In G. Brewka and J. Lang, editors, *Proc. of the 11th KR-08*, pages 673–683. AAAI Press. Sydney, Australia, September 16-19, 2008.

Artale, A., Ryzhikov, V., and Kontchakov, R. (2012). DL-Lite with attributes and datatypes. In *Proc. of ECAI-12*, pages 61–66. IOS Press.

Artale, A., Kontchakov, R., Wolter, F., and Zakharyaschev, M. (2013). Temporal description logic for ontology-based data access. In *Proc. of IJCAI'13*, pages 711–717. AAAI Press.

Baader, F., Borgwardt, S., and Lippmann, M. (2013). Temporalizing ontology-based data access. In M. Bonacina, editor, *Automated Deduction – CADE-24*, volume 7898 of *LNCS*, pages 330–344. Springer.

Calvanese, D. and De Giacomo, G. (2003). *The DL Handbook: Theory, Implementation and Applications*, chapter Expressive description logics, pages 178–218. Cambridge University Press.

Etzion, O., Gal, A., and Segev, A. (1998). *Temporal Databases – Research and Practice*, chapter Extended update functionality in temporal databases, pages 56–95. LNCS. Springer-Verlag.

Gregersen, H. and Jensen, C. S. (1999). Temporal entity-relationship models - a survey. *IEEE Trans. Knowl. Data Eng.*, **11**(3), 464–497.

Halpin, T. (2008). Temporal modeling and ORM. In *OTM 2008 Workshops*, volume 5333 of *LNCS*, pages 688–698. Springer.

Hartmann, T., Saake, G., Jungclaus, R., Hartel, P., and Kusch, J. (1994). Revised version of the modelling language troll.

Keet, C. M. and Artale, A. (2010). A basic characterization of relation migration. In *OTM Workshops*, volume 5231 of *LNCS*, pages 484–493. Springer.

Keet, C. M. and Fillottrani, P. R. (2013). Toward an ontology-driven unifying metamodel for UML class diagrams, EER, and ORM2. In *Proc. of the 32nd ER-2013*, pages 313–326.

Khatri, V., Ram, S., Snodgrass, R. T., and Terenziani, P. (2014). Capturing telic/atelic temporal data semantics: Generalizing conventional conceptual models. *Transactions on Knowledge and Data Engineering*, **26**(3), 528–548.

Lora, R., Sabaini, A., Combi, C., and Moretti, U. (2012). Designing the reconciled schema for a pharmacovigilance data warehouse through a temporally-enhanced ER model. In C. C. Yang, H. Chen, H. D. Wactlar, C. Combi, and X. Tang, editors, *Proc. of SHB-12*, pages 17–24. ACM.

Lutz, C., Wolter, F., and Zakharyaschev, M. (2008). Temporal description logics: A survey. In *Proc. of the 15th TIME-08*. IEEE Computer Society Press.

McBrien, P. (2008). Temporal constraints in non-temporal data modelling languages. In *Proc. of ER-08*, volume 5231 of *LNCS*, pages 412–425. Springer.

Object Management Group (2012). Superstructure specification. Standard 2.4.1, Object Management Group. http://www.omg.org/spec/UML/2.4.1/.

Ongoma, E. A. N., Keet, C. M., and Meyer, T. (2014). Transition constraints for temporal attributes. In *Proc. of the 27th DL-14*, volume 1193 of *CEUR-WS*, pages 684–695. 17-20 July 2014, Vienna, Austria.

Parent, C., Spaccapietra, S., and Zimányi, E. (2006). *Conceptual modeling for traditional and spatio-temporal applications—the MADS approach*. Berlin Heidelberg: Springer Verlag.

Priebe, T., Dobmeier, W., Schläger, C., and Kamprath, N. (2007). Supporting attribute-based access control in authorization and authentication infrastructures with ontologies. *Journal of software: JSW*, **2**(1), 27–38.

Savkovic, O. and Calvanese, D. (2012). Introducing datatypes in DL-Lite. In *Proc. of ECAI-12*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 720–725. IOS Press.

Appendix A

Proof of Proposition 1: 'Status Attributes: Logical Implications'.

Proof. The proof for Asch4 is similar to the proof of sch3 in (Artale *et al.*, 2007), and Asch5 is similar to sch4, so we prove here only the other ones.

(Csch). Let $o \in \textit{Scheduled-}C^{\mathcal{I}(t)}$, then if $\langle o,d\rangle \in A^{\mathcal{I}(t)}$, then $o \in C^{\mathcal{I}(t)}$ by Aact1, which contradicts the premise; if $\langle o,d\rangle \in \textit{Suspended-}A^{\mathcal{I}(t)}$, the $o \in C^{\mathcal{I}(t)}$ or $o \in \textit{Suspended-}C^{\mathcal{I}(t)}$ (by Asusp2), which also contradicts; if $\langle o,d\rangle \in \textit{Disabled-}A^{\mathcal{I}(t)}$, it contradicts likewise due to Adisab3; therefore $\langle o,d\rangle \in \textit{Scheduled-}A^{\mathcal{I}(t)}$ (which does not contradict Asch3).

(Asch3). Let $\langle o,d\rangle \in \textit{Scheduled-}A^{\mathcal{I}(t)}$, then $o \notin \textit{Disabled-}C^{\mathcal{I}(t)}$ because by Aexists3, $o \in \textit{Exists-}C^{\mathcal{I}(t)}$, and $\textit{Exists-}C^{\mathcal{I}(t)}$ is disjoint from $\textit{Disabled-}C^{\mathcal{I}(t)}$; if $o \in \textit{Suspended-}C^{\mathcal{I}(t)}$ it contradicts Csusp3, for it would force $\langle o,d\rangle \in \textit{Suspended-}A^{\mathcal{I}(t)}$, which contradicts the premise; hence $\langle o,d\rangle \in \textit{Scheduled-}A^{\mathcal{I}(t)} \rightarrow o \in (\textit{Scheduled-}C^{\mathcal{I}(t)} \vee C^{\mathcal{I}(t)})$.

(Cdisab4). Let $o \in \textit{Disabled-}C^{\mathcal{I}(t)}$, then if $\langle o,d\rangle \in A^{\mathcal{I}(t)}$, o's status contradicts because of Aact1; if $\langle o,d\rangle \in \textit{Suspended-}A^{\mathcal{I}(t)}$, it leads to a contradiction because of Asusp2; if $\langle o,d\rangle \in \textit{Scheduled-}A^{\mathcal{I}(t)}$, likewise a contradiction because of Asch3; therefore $\langle o,d\rangle \in \textit{Disabled-}A^{\mathcal{I}(t)}$ (which does not contradict Adisab3).

(Asusp3). Let $a \in \textit{Suspended-}A^{\mathcal{I}(t)}$, then by Asusp1, $\exists t'{<}t.a \in A^{\mathcal{I}(t')}$, so a was active in the past; by Asch1 $\exists t'{>}t''.a \in A^{\mathcal{I}(t'')}$ (with $a \in \textit{Scheduled-}A^{\mathcal{I}(t'')}$), but there cannot be a t''' such that $t''{<}t'''{<}t'$ due to Asch4, i.e., a must first become active before any possible migration to another status, and by Adisab1 disabled persists, so $a \notin \textit{Disabled-}A^{\mathcal{I}(t+1)}$.

(Aact2). An attribute remaining active, becoming suspended, or disabled is the same as saying it cannot become scheduled anymore. By Asch2, if $a \in A^{\mathcal{I}(t)}$ then $\forall t'{>}t.a \notin \textit{Scheduled-}A^{\mathcal{I}(t')}$, therefore $\forall t'{>}t.a \in (A^{\mathcal{I}(t')} \vee \textit{Suspended-}A^{\mathcal{I}(t')} \vee \textit{Disabled-}A^{\mathcal{I}(t')})$.

(Adisab4). Let $\langle o,d\rangle \in \textit{Disabled-}A^{\mathcal{I}(t)}$, then if $o \in \textit{Scheduled-}C^{\mathcal{I}(t)}$, it contradicts Adisab3 as it says that either $o \in \textit{Disabled-}C^{\mathcal{I}(t)}$ or $o \in C^{\mathcal{I}(t)}$, thus also if $o \in \textit{Suspended-}C^{\mathcal{I}(t)}$, it contradicts Adisab3 too, therefore, $a \notin \textit{Scheduled-}A^{\mathcal{I}(t)}$ and $a \notin \textit{Suspended-}A^{\mathcal{I}(t)}$. □

Appendix B

Proof of Proposition 2: 'Status Attributes and Status Classes: isa Logical Implications'.

Proof. (isa1_A) Attributes of objects disabled in C, and active in D in the past, are disabled in D.

Given: $o \in \textit{Disabled-}C^{\mathcal{I}(t_0)}$, by Cdisab4, we have $a \in \textit{Disabled-}A^{\mathcal{I}(t_0)}_{\textit{Dis-C}}$ and $a \in A^{\mathcal{I}(t_1)}_D$, where $t_1 < t_0$. By Subsumption premise, A is also an attribute of D and by Aact1, for any $a \in A^{\mathcal{I}(t_1)}$, then $o \in D^{\mathcal{I}(t_1)}$, i.e. $\Diamond^{-}\texttt{D}$. By isa5, $\texttt{Disabled-C} \sqcap \Diamond^{-}\texttt{D} \sqsubseteq \texttt{Disabled-D}$; thus, because $o \in \textit{Disabled-}C^{\mathcal{I}(t_0)}$, then also $o \in \textit{Disabled-}D^{\mathcal{I}(t_0)}$. Therefore by Cdisab4, $a \in \textit{Disabled-}A^{\mathcal{I}(t_0)}_{\textit{Dis-D}}$.

(isa2_A) Attributes active in D must be either active in C or is not present in C.

Given: $o \in D^{\mathcal{I}(t)}$, and its active attribute, $a \in A^{\mathcal{I}(t)}_D$.

part 1. From isa1, we get $\texttt{D} \sqsubseteq \texttt{C}$, and thus by Cactive, $\texttt{C} \sqsubseteq \texttt{From} : (\texttt{Exists-A} \sqcup \texttt{Disabled-A})$, and thus $(\texttt{Scheduled-A} \sqcup \texttt{A} \sqcup \texttt{Suspended-A})$ or $\texttt{Disabled-A}$ (Aexists2).

part 2. We prove by contradiction that if $a \in A^{\mathcal{I}(t)}_D$, we have $a \in A^{\mathcal{I}(t)}_C$. If attributes in C are:

- $a \in \textit{Disabled-}A^{\mathcal{I}(t)}_C$, it contradicts: by Adisab5, $a \notin A^{\mathcal{I}(t')}$ where $t' > t$, together with isa1_A, if $a \in \textit{Disabled-}A^{\mathcal{I}(t)}_C$, forces $a \in \textit{Disabled-}A^{\mathcal{I}(t)}_D$, therefore $a \notin \textit{Disabled-}A^{\mathcal{I}(t)}_D$.
- $a \in \textit{Scheduled-}A^{\mathcal{I}(t)}_C$, it contradicts: by Asch1, $a \in A^{\mathcal{I}(t')}$, where $t' > t$, but we have $a \in A^{\mathcal{I}(t)}_D$, which means that it must have been active in C, but by Asch2, $a \notin A^{\mathcal{I}(t')}$, therefore $a \notin \textit{Scheduled-}A^{\mathcal{I}(t)}_C$.
- $a \in \textit{Suspended-}A^{\mathcal{I}(t)}_C$, contradicts: $a \in A^{\mathcal{I}(t')}_C$, where $t' > t$, due to Asusp1, and the subsumption premise would deduce $\texttt{A}_\texttt{C} \sqsubseteq \texttt{A}_\texttt{D}$.
- $a \in A^{\mathcal{I}(t)}_C$ does not contradict, by Aact1, $\texttt{A} \sqsubseteq \texttt{From} : \texttt{C}$, since $\texttt{D} \sqsubseteq \texttt{C}$ by isa1.

Therefore, if $a \in A^{\mathcal{I}(t)}_D$, then $a \in A^{\mathcal{I}(t)}_C$, or it is not present in C.

(ISA3_A) Attributes suspended in D must be either suspended in C, or is not present in C.

Given: $o \in D^{\mathcal{I}(t)}$ with an attribute that is suspended: $a \in \textit{Suspended-}A_D^{\mathcal{I}(t)}$, then, if the attribute is present also in C, it is existing or disabled (refer to ISA2_A, part 1). We prove that if $a \in \textit{Suspended-}A_D^{\mathcal{I}(t)}$, we have $a \in \textit{Suspended-}A_C^{\mathcal{I}(t)}$. Suppose the attribute in C is:

- $a \in \textit{Disabled-}A_C^{\mathcal{I}(t)}$. By ADISAB1, $a \in \textit{Disabled-}A^{\mathcal{I}(t')}$, where $t' > t$, then from ISA1_A, if $a \in \textit{Disabled-}A_C^{\mathcal{I}(t)}$, it forces $a \in \textit{Disabled-}A_D^{\mathcal{I}(t)}$, but by premise, $a \in \textit{Suspended-}A_D^{\mathcal{I}(t)}$, therefore $a \notin \textit{Disabled-}A_C^{\mathcal{I}(t)}$.
- $a \in \textit{Scheduled-}A_C^{\mathcal{I}(t)}$. By ASCH1, $a \in A^{\mathcal{I}(t')}$, where $t' > t$, but we have $a \in A_D^{\mathcal{I}(t)}$, which means that it must have been active in C. But by ASCH2, $a \notin A^{\mathcal{I}(t')}$ therefore $a \notin \textit{Scheduled-}A_C^{\mathcal{I}(t)}$.
- $a \in A^{\mathcal{I}(t)}$. By ASUSP1, $\exists t' < t.a \in A^{\mathcal{I}(t')}$, but by subsumption premise if $a \in \textit{Suspended-}A_D^{\mathcal{I}(t)}$ and $a \in A_C^{\mathcal{I}(t)}$, then C would have more constrained attributes than D, forcing the deduction $\texttt{C} \sqsubseteq \texttt{D}$, which contradicts the premise. Therefore $a \notin A^{\mathcal{I}(t)}$.
- $a \in \textit{Suspended-}A^{\mathcal{I}(t)}$ does not lead to a contradiction: by ASUSP2, $\texttt{Suspended-A} \sqsubseteq \texttt{From} : (\texttt{Suspended-C} \sqcup \texttt{C})$.

Therefore, if $a \in \textit{Suspended-}A_D^{\mathcal{I}(t)}$, it must be $a \in \textit{Suspended-}A_C^{\mathcal{I}(t)}$ or it does not exist in C, or: if the attribute exists in C, then $\texttt{Sus-A}_\texttt{D} \sqsubseteq \texttt{Sus-A}_\texttt{C}$.

(ISA4_A) Attributes of objects suspended in D must be either suspended in C, or is not present in C.

Given: $o \in \textit{Suspended-}D^{\mathcal{I}(t)}$, then from CSUSP3 we get $a \in \textit{Suspended-}A_{\textit{Susp-D}}^{\mathcal{I}(t)}$ and from ISA2, $o \in \textit{Suspended-}C^{\mathcal{I}(t)}$ or $o \in C^{\mathcal{I}(t)}$. We prove by contradiction that if $a \in \textit{Suspended-}A_{\textit{Susp-D}}^{\mathcal{I}(t)}$, we have $a \in \textit{Suspended-}A_{\textit{Susp-C}}^{\mathcal{I}(t)}$.

- CSUSP3, i.e., $\texttt{Suspended-C} \sqsubseteq \texttt{From} : \texttt{Suspended-A}$, forces $a \in \textit{Suspended-}A_{\textit{Susp-C}}^{\mathcal{I}(t)}$.
- If $o \in C^{\mathcal{I}(t)}$, then by ISA3_A, $a \in \textit{Suspended-}A^{\mathcal{I}(t)}$ if the attribute is exists in D.

Therefore attributes in C can only be suspended else it is not present in C.

(ISA5_A) Attributes disabled in D must be either disabled in C, or is not present in C.

Given: $o \in D^{\mathcal{I}(t)}$ and its disabled attribute $a \in \textit{Disabled-}A_D^{\mathcal{I}(t)}$, then C has existing or disabled attributes (refer to ISA2_A, part 1). We prove by contradiction that if $a \in \textit{Disabled-}A_D^{\mathcal{I}(t)}$, then $a \in \textit{Disabled-}A_C^{\mathcal{I}(t)}$. Now suppose the attribute in C is:

- $a \in \textit{Scheduled-}A^{\mathcal{I}(t)}$. ASCH1 implies $a \in A^{\mathcal{I}(t')}$, where $t < t'$. But if $a \in \textit{Disabled-}A_D^{\mathcal{I}(t)}$ then $a \in A_D^{\mathcal{I}(t_0)}$ with $t_0 < t$ (from ADISAB2), i.e., it must have been active before, and therefore $a \in A_C^{\mathcal{I}(t_0)}$ (thanks to ISA2_A), but by ASCH2 this cannot happen, therefore $a \notin \textit{Scheduled-}A_C^{\mathcal{I}(t)}$.
- $a \in \textit{Suspended-}A^{\mathcal{I}(t)}$ or $a \in A^{\mathcal{I}(t)}$, contradicts because of the subsumption premise (if D does not have some attribute A or relationship R, then neither does C).

Therefore, if $a \in \textit{Disabled-}A_D^{\mathcal{I}(t)}$, then $a \in \textit{Disabled-}A_C^{\mathcal{I}(t)}$, or it is not present in C.

(ISA6_A) Attributes of objects disabled in D must be either disabled in C or is not present in C.

Given: $o \in \textit{Disabled-}D^{\mathcal{I}(t)}$ and its attributes are disabled by CDISAB4, $a \in \textit{Disabled-}A_{\textit{Dis-D}}^{\mathcal{I}(t)}$, From ISA3, we have $(\texttt{Disabled-C} \sqcup \texttt{C} \sqcup \texttt{Suspended-C})$. We prove by contradiction that if $a \in \textit{Disabled-}A_{\textit{Dis-D}}^{\mathcal{I}(t)}$, the attribute can only be $a \in \textit{Disabled-}A_{\textit{Dis-C}}^{\mathcal{I}(t)}$

- We start with $o \in \textit{Suspended-}C^{\mathcal{I}(t)}$, by CSUSP3, it must be that $a \in \textit{Suspended-}A^{\mathcal{I}(t)}$, but this contradicts ISA5_A, because if $a \in \textit{Disabled-}A_D^{\mathcal{I}(t)}$, then $a \in \textit{Disabled-}A_C^{\mathcal{I}(t)}$.
- Next, $o \in C^{\mathcal{I}(t)}$. By CACTIVE, $\texttt{C} \sqsubseteq \texttt{From} : (\texttt{Exists-A} \sqcup \texttt{Disabled-A})$, then if $a \in \textit{Disabled-}A_{\textit{Dis-D}}^{\mathcal{I}(t)}$, the attribute in D can only be disabled by ISA5_A, because if $\texttt{Exists-A}$ then it violates the subsumption premise.
- Last, $o \in \textit{Disabled-}C^{\mathcal{I}(t)}$. CDISAB4 forces $a \in \textit{Disabled-}A^{\mathcal{I}(t)}$, which holds.

Therefore if $a \in \textit{Disabled-}A_{\textit{Dis-D}}^{\mathcal{I}(t)}$, then it must be $a \in \textit{Disabled-}A_{\textit{Dis-C}}^{\mathcal{I}(t)}$ else, it does not exist.

(ISA7_A) Attributes scheduled in D must be either scheduled in C or is not present in C.

Given: $o \in D^{\mathcal{I}(t)}$ and its attribute scheduled, $a \in \textit{Scheduled-}A_D^{\mathcal{I}(t)}$. C has existing or disabled attributes (see ISA2_A, part 1). We prove by contradiction that if $a \in \textit{Scheduled-}A_D$, we must have $a \in \textit{Scheduled-}A_C$. Now suppose:

- $a \in A_C^{\mathcal{I}(t)}$, then, by the subsumption premise, $a \in A_D^{\mathcal{I}(t)}$, but by ASCH2, $a \in A^{\mathcal{I}(t)} \to \forall t'{>}t.a \notin \textit{Scheduled-}A^{\mathcal{I}(t)}$, hence, a contradiction;
- $a \in \textit{Suspended-}A_C^{\mathcal{I}(t)}$, then because of ASUSP1, $\exists t'{<}t.a \in A^{\mathcal{I}(t')}$, hence, the same argument as in the previous item applies.
- $a \in \textit{Disabled-}A_C^{\mathcal{I}(t)}$, which means there must be a time $t' < t$, s.t. $a \in A^{\mathcal{I}(t')}$ (from ADISAB2) that contradicts (see first item);

Therefore, either $a \in \textit{Scheduled-}A_C^{\mathcal{I}(t)}$ or the attribute is not present in C.

(ISA8_A) Attributes of objects scheduled in D must be either scheduled in C, or is not present in C.

Given: $o \in \textit{Scheduled-}D^{\mathcal{I}(t)}$, therefore also $a \in \textit{Scheduled-}A_{\textit{Sched-D}}^{\mathcal{I}(t)}$ (by CSCH) and $\texttt{Scheduled-D} \sqsubseteq \texttt{Exists-C}$, i.e., $\texttt{Scheduled-D} \sqsubseteq (\texttt{Scheduled-C} \sqcup \texttt{C} \sqcup \texttt{Suspended-C})$ from ISA4. We prove that if $a \in \textit{Scheduled-}A_{\textit{Sched-D}}^{\mathcal{I}(t)}$, then $a \in \textit{Scheduled-}A_{\textit{Sched-C}}^{\mathcal{I}(t)}$ must hold.

- If $o \in \textit{Suspended-}C^{\mathcal{I}(t)}$, then by CSUSP3, $a \in \textit{Suspended-}A^{\mathcal{I}(t)}$, which contradicts: by ASUSP1, $\exists t'{<}t.a \in A^{\mathcal{I}(t')}$ but by ASCH2, $a \in A^{\mathcal{I}(t)} \to \forall t'{>}t.a \notin \textit{Scheduled-}A^{\mathcal{I}(t)}$.
- If $o \in C^{\mathcal{I}(t)}$, then by ISA7_A, $a \in \textit{Scheduled-}A^{\mathcal{I}(t)}$ if the attribute exists in C.
- If $o \in \textit{Scheduled-}C^{\mathcal{I}(t)}$, by CSCH $\texttt{Scheduled-C} \sqsubseteq [\texttt{From}]\texttt{Scheduled-A}$, we have $a \in \textit{Scheduled-}A^{\mathcal{I}(t)}$.

Therefore if $a \in \textit{Scheduled-}A_{\textit{Sched-D}}^{\mathcal{I}(t)}$, then $a \in \textit{Scheduled-}A_{\textit{Sched-C}}^{\mathcal{I}(t)}$ else it is not present in C. $\square$