Towards Constructive Description Logics for Abstraction and Refinement

Michael Mendler and Stephan Scheele

Informatics Theory Group
University of Bamberg
{michael.mendler,stephan.scheele}@uni-bamberg.de

Abstract. This paper explores some aspects of a new and natural semantical dimension that can be accommodated within the syntax of description logics which opens up when passing from the classical truth-value interpretation to a *constructive interpretation*. We argue that such a strengthened interpretation is essential to represent applications with partial information adequately and to achieve consistency under abstraction as well as robustness under refinement. We introduce a constructive version of $\mathcal{ALC}$, called $c\mathcal{ALC}$, for which we give a sound and complete Hilbert axiomatisation and a Gentzen tableau calculus showing finite model property and decidability.

1 When Constructiveness Matters

Knowledge representation based on description logics (DL) can be used to capture the meaning of natural language statements about specific world domains (ontologies). Often, however, such knowledge is dynamic and incomplete. Entities that make up the domain may not be fixed and tangible but abstractions of real individuals whose properties are changing and defined only up to construction. Natural language concepts rarely have a static interpretation but are subject to negotiation or context and thus require a constructive approach which is robust under refinement. An application area where this aspect is particularly prominent and which motivates the work[1] reported in this paper, is auditing. The digital auditing of business mass data experiences a huge increase in importance recently. Audit executives, fraud examiners and compliance professionals are pressured on all fronts to shorten audit cycles and to increase audit efficiency and quality. In particular, the efficient verification of enterprise processes is of big interest since the audit concern is getting more critical with new regulations like SOX[2], IFRS[3] and also to respond accurately to managing risks in our competitive world.

Audit statements about the validity of accounting data, absence of fraud or conformance to financial process standards must constructively take account of many dimensions of abstraction and refinement. First, the producers of audit data usually are ongoing business processes which the audit data can only cover a limited snapshot of. E.g., a requirement such as "each delivery order must have an associated invoice" must take into account that for some delivery order the invoice is still "in the process" and only available after refinement of the audit data. Second, a role like the "legally responsible signatory" may not be fully definable once and for all but depend on the legal context. Some aspects may even deliberately be left open subject to negotiations and only refined as the auditing case progresses. Third, entities may be abstractions of physical individuals: The notion of the 'CEO of company X' in an audit statement is a virtual rather than concrete person who may be replaced perhaps while auditing in ongoing. The CEO which appears atomic at some level of abstraction really is a concept at a lower level where personal liability issues come in or where executive action needs to be taken. Forth, auditing is typically faced with vast amounts of business data. For efficiency reasons, manageable digests of the data need to be created. Such data

[1] Incipient research project funded by the DFG.
[2] Sarbanes-Oxley Act, US law of 2002 on business reporting in reaction to Enron and WorldCom scandals.
[3] International Financial Reporting Standard.

compression may ignore potentially irrelevant attributes of entities or scan only subsets of concepts. Auditing, thus, is not exact but approximated. If the quick check indicates potential irregularities then a constructive refinement of the abstracted entities and concepts must be possible to confirm or reject the case constructively.

Auditing is a prime example of a class of application domains which require the ability to express partiality and incomplete information beyond the standard open world assumption (OWA). Because the semantical meaning is context-dependent and possibly involves many levels of explication there must be a constructive notion of undefinedness which permits that concepts evolve. Classical OWA assumes that each concept is static and at the outset either includes a given entity or not. However, either option may be incorrect, if the entity or the concept is not fully defined until a later stage where lower levels of detail become available.

But if OWA is not enough, how can reasoning be both correct under abstraction and sustainable under refinement? Logic has a well-known suggestion to solve this puzzle: replace the traditional *binary* truth interpretation by a *constructive* notion of truth. Proof-theoretically, constructive logic is compatible with the idea of positive evidence and realisability [22]. It does not infer the presence of entities from the absence of others but insists on the existence of *computational witnesses*. Model-theoretically, constructive logic admits of an interpretation based on *stages of information* [23] so that truth is persistent under refinement[4].

The role of intuitionistic Kripke models for knowledge representation based on partial descriptions has been highlighted in [7]. The general benefits of the Curry-Howard Isomorphism (*proofs-as-computations*) in DL have been argued in [8,6]. In our context, more concretely, we envisage that the computational interpretation of TBox deductions as λ-terms yields verified audit tactics and that constructive ABox tableau algorithms provide engines to drive interactive games between auditee (proponent) and auditor (opponent). A third potential benefit arises from the use of DLs as a *programming type system* (see e.g., [16]) which naturally requires a constructive setting. Constructive DL concepts may not only specify the semantics of data streams in audit component interfaces but also resource requirements. This can be exploited to satisfy higher demands on robustness and efficiency in the semantic processing of mass data.

In this paper we discuss some of the model-theoretic aspects of the constructive interpretation of DLs, in contrast to [6] which is proof-theoretic and addresses the extraction of information terms.

The work of [7] presents an intuitionistic epistemic logic based on several refinement relations coding multiple *(partial) points of view*. Here we only consider one dimension of refinement reflecting a two-player scenario (e.g., auditor and auditee) but in a more general sense than [7]. Our refinement ordering $\preceq$ may have cycles and fallible descriptions. Such descriptive "oscillations" and "deadlocks" are intrinsic to real-world abstractions (see examples below).

It is important to point out that the semantic dimension along which refinement takes place is implicit in $c\mathcal{ALC}$ and not coded in the syntax. This accommodates many different notions of context generically in the standard $\mathcal{ALC}$ language. The context-dependency is built into the notion of truth rather than the terminology like in other work on special cases of context such as temporal DL [1,5,2]. $c\mathcal{ALC}$ is meant for applications where we must be robust for several implicit notions of context-dependency but do not need to reason explicitly about some specific refinement.

Our work is to be distinguished also from many-valued DL (see e.g., [17,14]) which is only finitely valued while $c\mathcal{ALC}$ is infinitely valued and from fuzzy DL (see e.g., [21,9,13]) which use a quantitative notion of approximate truth whereas $c\mathcal{ALC}$ still adheres to a crisp deductive approach. Even though the envisaged application domain of auditing may use statistical analyses, at the end of the day we must cross the t's and dot the i's and be able to name the evidence.

[4] One might say that classical DL is based on a *static* open world assumption (SOWA) while constructive DL supports an *evolving* open world assumption (EOWA).

2 Syntax and Semantics of $c\mathcal{ALC}$

Concept descriptions in $c\mathcal{ALC}$ are based on sets of *role names* N_R and *concept names* N_C and formed as follows, where $A \in N_C$ and $R \in N_R$:

$$C, D \to A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid C \sqsubseteq D \mid \exists R.C \mid \forall R.C.$$

This syntax is more general than standard $\mathcal{ALC}$ in that it includes subsumption $\sqsubseteq$ as a concept–forming operator. The TBox statement $C \sqsubseteq D$ meaning that 'D subsumes C' is expressed as the concept identity $C \sqsubseteq D = \top$. In classical $\mathcal{ALC}$ one could use the equation $\neg C \sqcup D = \top$ to do that, essentially reducing subsumption to $\neg$ and $\sqcup$. This is no longer possible in constructive logic where these operators are independent. Being a first class operator, subsumption can be nested arbitrarily as in $((D \sqsubseteq C) \sqsubseteq B) \sqsubseteq A$. The full power of such "higher–order" subsumptions will not be needed in practice but will allow us to axiomatise the full theory of $c\mathcal{ALC}$ conveniently in the form of a Hilbert calculus. Like in $\mathcal{ALC}$ the universal concept $\top$ is redundant and codable as $\neg\bot$. Also, $\bot$ and $\neg$ can represent each other, e.g. $\bot = A \sqcap \neg A$ and $\neg C = C \sqsubseteq \bot$. Otherwise, the operators are independent.

Constructive interpretations $\mathcal{I}$ of concept descriptions extend the classical models for $\mathcal{ALC}$ by a pre–ordering $\preceq^{\mathcal{I}}$ for expressing refinement between individuals and by a notion of fallible entities $\bot^{\mathcal{I}}$ for interpreting contradiction. Following the standard Kripke semantics of intuitionistic logic [23], entities in constructive DL are not atomic individuals but have internal structure which in general is only partially determined and thus subject to refinement. Let relation $a \preceq a'$ on entities denote that a' is *more precisely determined* than a, that a' *refines* a or that a *abstracts* a'. The relation $\preceq$ models a potential increase of information or refinement of context associated with the process of pinning down entities as real individuals. This includes the possibility that both $a \preceq b$ and $b \preceq a$, i.e., a and b have the same information content and thus are formally indistinguishable, yet still distinct $a \neq b$ because of some lower-level properties. Now, if a concept C is to be robust under refinement then $a{:}C$ and $a \preceq a'$ must imply $a'{:}C$. This is achieved by the following definition:

Definition 1. A *constructive interpretation* or *constructive model* of $c\mathcal{ALC}$ is a structure $\mathcal{I} = (\Delta^{\mathcal{I}}, \preceq^{\mathcal{I}}, \bot^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a non-empty set $\Delta^{\mathcal{I}}$ of *entities*, the universe of discourse in which each entity represents a partially defined, or abstract individual; a *refinement* pre-ordering $\preceq^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, i.e., a reflexive and transitive relation; a subset $\bot^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ of *fallible* entities closed under refinement, i.e., $x \in \bot^{\mathcal{I}}$ and $x \preceq^{\mathcal{I}} y$ implies $y \in \bot^{\mathcal{I}}$; finally an interpretation function $\cdot^{\mathcal{I}}$ mapping each role name $R \in N_R$ to a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and each atomic concept $A \in N_C$ to a set $\bot^{\mathcal{I}} \subseteq A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ which is closed under refinement, i.e., $x \in A^{\mathcal{I}}$ and $x \preceq^{\mathcal{I}} y$ implies $y \in A^{\mathcal{I}}$; Interpretation $\mathcal{I}$ is lifted from atomic $\bot$, A to arbitrary concepts, where $\Delta^{\mathcal{I}}_c =_{df} \Delta^{\mathcal{I}} \setminus \bot^{\mathcal{I}}$ is the set of *non-fallible* elements in $\mathcal{I}$:

$$\begin{aligned}
\top^{\mathcal{I}} &=_{df} \Delta^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &=_{df} \{x \mid \forall y \in \Delta^{\mathcal{I}}_c.\ x \preceq^{\mathcal{I}} y \Rightarrow y \notin C^{\mathcal{I}}\} \\
(C \sqcap D)^{\mathcal{I}} &=_{df} C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &=_{df} C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(C \sqsubseteq D)^{\mathcal{I}} &=_{df} \{x \mid \forall y \in \Delta^{\mathcal{I}}_c.\ (x \preceq^{\mathcal{I}} y \ \&\ y \in C^{\mathcal{I}}) \Rightarrow y \in D^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &=_{df} \{x \mid \forall y \in \Delta^{\mathcal{I}}_c.\ x \preceq^{\mathcal{I}} y \Rightarrow \exists z \in \Delta^{\mathcal{I}}.\ (y,z) \in R^{\mathcal{I}} \ \&\ z \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &=_{df} \{x \mid \forall y \in \Delta^{\mathcal{I}}_c.\ x \preceq^{\mathcal{I}} y \Rightarrow \forall z \in \Delta^{\mathcal{I}}.\ (y,z) \in R^{\mathcal{I}} \Rightarrow z \in C^{\mathcal{I}}\}.
\end{aligned}$$ □

Entities in $\Delta^{\mathcal{I}}$ are partial descriptions representing incomplete information about individuals. Read $a \preceq b$ as saying that a *"is an abstraction of"* b, i.e., that it has *"contains no more information"* than b. Fallible elements $b \in \bot^{\mathcal{I}}$ may be thought of as over–constrained tokens of information, self-contradictory objects of evidence or undefined computations. E.g., they may be used to model the situation in which computing a role–filler for an abstract individual a fails, i.e., $\forall b.\, R(a,b) \Rightarrow b \in \bot^{\mathcal{I}}$, yet when a is refined to a' then a non-fallible role-filler $b' \in \Delta^{\mathcal{I}}_c$ exists with $R(a', b')$ (see Ex. 3 below). Each entity implicitly subsumes all its refinements and truth is inherited. Specifically, one can show that $x \in C^{\mathcal{I}}$ and $x \preceq^{\mathcal{I}} y$

implies $y \in C^{\mathcal{I}}$ for all concepts C. Fallible entities are information–wise maximal elements and therefore included in every concept, i.e., $\bot^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ for all C.

The purpose of the present work is to show that the non-standard interpretation of Def. 1 induces a well-behaved logic, called $c\mathcal{ALC}$, which uses the same syntax but is more expressive than classical $\mathcal{ALC}$ and still admits standard TBox and ABox tableau reasoning. Before we continue expounding the theory let us look at some examples.

Example 1. Every classical interpretation $\mathcal{I}$ of $\mathcal{ALC}$ (see e.g., [3]) induces a trivial model according to Def. 1 with the *discrete* refinement relation $x \preceq^{\mathcal{I}} y$ iff $x = y$ and the empty set $\bot^{\mathcal{I}} = \emptyset$ of fallible entities. These validate the concept descriptions $C \sqcup \neg C$, $\exists R.\bot = \bot$ and $\exists R.(C \sqcup D) = \exists R.C \sqcup \exists R.D$. These three axioms essentially characterise classical models (see Sec. 4). □

Example 2. Let $a = (c, d_1)$ and $b = (c, d_2)$ be two entries in a (relational) database that share the same first attribute but are distinguished in the second. If the attributes are referenced by roles \$1 and \$2 then the situation could be specified, in ABox syntax, by $a\,\$1\,c$, $a\,\$2\,d_1$, $a\,\$1\,c$, $a\,\$2\,d_2$. Now let us abstract from the second attributes and consider the pairs as partially defined entities $a^\sharp = (c, ?)$ and $b^\sharp = (c, ?)$, respectively, say in an attempt to compress information. Ignoring d_1, d_2 means that $a^\sharp$ and $b^\sharp$ carry the same information and thus can no longer be distinguished. Since the pre-order $\preceq$ measures the information content we get $a^\sharp \preceq b^\sharp$ and $b^\sharp \preceq a^\sharp$. This cyclic refinement relationship implies an abstract equivalence $a^\sharp \cong b^\sharp$ but not an identity $a^\sharp = b^\sharp$ keeping in mind that both have incompatible realisations $a^\sharp \preceq a$ and $b^\sharp \preceq b$, respectively.

The situation is depicted in Fig. 1. The dashed arrows correspond to refinement and solid arrows represent the attribute roles \$1, \$2. Note that both $a^\sharp$, $b^\sharp$ have a fallible \$2 filler ($\bot$) which corresponds to a computational deadlock when selecting \$2 for $a^\sharp$ or $b^\sharp$. Formally, if $Th(x)$ denotes the set of concepts which entity x participates in, then $Th(a^\sharp) = Th(b^\sharp)$. E.g., $\exists\$1.C, \exists\$2.(D_1 \sqcup D_2) \in Th(a^\sharp)$ since every refinement of $a^\sharp$ has $c{:}C$ as filler for role \$1 and either $d_1{:}D_1$ or $d_2{:}D_2$ as a filler for \$2. The disjunction $\exists\$2.(D_1 \sqcup D_2)$ captures the choice between the two realisations of $a^\sharp$ as a concrete individual, *viz.*,(c, d_1) and (c, d_2). On the other hand, this choice cannot be resolved at the abstract level as there is no single uniform choice of the \$2-filler. This is reflected in the logic by the fact that $\exists\$2.D_i \notin Th(a^\sharp)$ $(i = 1, 2)$ which means $\exists\$2.D_1 \sqcup \exists\$2.D_2 \notin Th(a^\sharp)$.

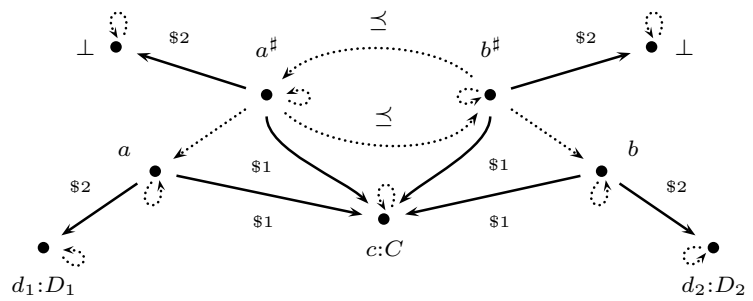

Fig. 1: A simple data model with abstraction.

Abstractions like this cannot be expressed in classical DL where existential fillers always distribute over $\sqcup$, i.e., $\exists\$2.(D_1 \sqcup D_2)$ is semantically identical to $\exists\$2.D_1 \sqcup \exists\$2.D_2$. Note that $\neg\exists\$2.D_1$ is not the same as $\forall\$2.\neg D_1$. The former says that it is inconsistent to assume that all refinements of $a^\sharp$ have \$2-filler in D_1. The latter means that no refinement has a \$2-filler in D_1. In $\mathcal{ALC}$, the concepts $\neg\exists\$2.D_1$ and $\forall\$2.\neg D_1$ are identical which is not what we want here. Also, note that the Excluded Middle $\exists\$2.D_1 \sqcup \neg\exists\$2.D_1$ is not valid for $a^\sharp$. □

Example 3. Concerning *hasCustomer* relationships between companies a, b, c, d let us assume that a has both b and c as customers, b has customer c and c has d among its customers. Further suppose that b is insolvent (concept *Insolvent*) and d solvent (concept $\neg$*Insolvent*). Regarding possible insolvency of c nothing is known. In classical OWA we have $c{:}(Insolvent \sqcup \neg Insolvent)$ regardless of c. This implies that a is an instance of the concept description $CW = \exists hasCustomer.(Insolvent \sqcap \exists hasCustomer.\neg Insolvent)$ specifying *creditworthy* companies with an insolvent customer who in turn can rely on at least one solvent

customer. In the first case c:*Insolvent* this customer of a is c, in case c:¬*Insolvent* it is b. In a *static* world the filling customer would be unknown but fixed. However, the case analysis on c is invalid if the model arises by abstraction from a concrete taxonomy where insolvency is a *context-dependent* defect.

Fig. 2 shows an example model of the situation. Each solid edge is the relation *hasCustomer* and each dotted line codes refinement. Company c may be insolvent during some specific period of time or under some specific legal understanding of the concept *Insolvent*, represented by refinement c'. It may be solvent during another period of time or other legal regulations as represented by refinement c''. Then insolvency of c is not just unknown but *undecidable* (i.e., not fixable). The required *hasCustomer*-filler for a in concept CW cannot be obtained without contradicting one of the two directions c', c'' in which c may evolve. The case c:¬*Insolvent* conflicts with refinement $c \preceq c'$ and c':*Insolvent*, if c:*Insolvent* the refinement $c \preceq c''$ obtains c'':¬*Insolvent*. Thus, neither *Insolvent* nor ¬*Insolvent* can be satisfied in c. In classical *static* OWA the case analysis is performed outside the model so that fillers may depend non-uniformly on the case analysis. In $c\mathcal{ALC}$ this choice is internalised and the filler of a role must be robust under case analysis. Thus, a:CW is invalid under *Evolving* OWA because the ∃-filler is not realisable by a single nameable entity. □

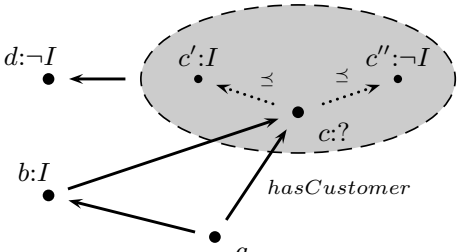

Fig. 2: Evolving OWA Model.

Example 4. Business data typically come in streams, e.g., as linearised database tables or time–series of financial market transactions. If streams are considered as abstract entities then DL concepts can act as a typing system to specify semantical properties of typical stream elements. To illustrate this let $\mathbb{D} = \mathbb{N} \uplus \mathbb{B} \uplus (\mathbb{N} \times \mathbb{B})$ be the discrete universe of booleans, naturals and their pairings. Consider the domain $\Delta^{\mathcal{I}} = \mathbb{D}^{\omega} = \mathbb{D}^{*} \cup \mathbb{D}^{\infty}$ of all finite and infinite sequences ("streams") over $\mathbb{D}$. The refinement $\preceq^{\mathcal{I}}$ is the suffix ordering, e.g., $a_0a_1a_2\cdots \preceq^{\mathcal{I}} a_2a_3\cdots$. Concepts $C^{\mathcal{I}}$ under this interpretation express *future projected behaviour* of streams. The empty stream has no future behaviour, it represents a computational deadlock, i.e., $\bot^{\mathcal{I}} = \{\epsilon\}$. For each concept there is a natural (functional) role *val* which picks the first element of the sequence considered as a singleton stream, i.e., $a_0a_1a_2\cdots\ val\, a_0$. Let NAT and BOOL be the usual programming language types considered as atomic $c\mathcal{ALC}$ concepts, i.e. $\text{NAT}^{\mathcal{I}} =_{df} \mathbb{N}^{\omega}$ and $\text{BOOL}^{\mathcal{I}} =_{df} \mathbb{B}^{\omega}$, specifying streams of naturals and streams of booleans, respectively. In a similar vein, we put $(\text{NAT} \times \text{BOOL})^{\mathcal{I}} =_{df} (\mathbb{N} \times \mathbb{B})^{\omega}$ to represent simple database tables as streams of data pairs. We then have TBox axioms $\text{NAT} \equiv \forall val.\text{NAT} \equiv \exists val.\text{NAT}$ and $\text{BOOL} \equiv \forall val.\text{BOOL} \equiv \exists val.\text{BOOL}$.

Now, suppose we linearise a table $t = (n_0, b_0)(n_1, b_1)(n_2, b_2)\cdots$ of (stream) type NAT × BOOL to give the flattened stream $t^{\flat} = n_0b_0n_1b_1n_2b_2\cdots$. What is the type of $t^{\flat}$? It is not the concept NAT ⊔ BOOL nor the equivalent $\exists val.\text{NAT} \sqcup \exists val.\text{BOOL}$ since this would require that all elements of $t^{\flat}$ are either NAT or all are BOOL. Really, we want set union NAT∪BOOL. In fact, this is what the concept $\exists val.(\text{NAT} \sqcup \text{BOOL})$ expresses: the first element of each suffix sequence is of value NAT or BOOL. Notice that the use of $\exists val$ here performs the decomposition of the stream so that the concept specification NAT ⊔ BOOL is applied element–wise rather than globally.

This interpretation provides a typical example for why ∃ should not distribute over ⊔ under a constructive interpretation which uses DL as a type system for programming: The difference between concepts $\exists val.(\text{NAT} \sqcup \text{BOOL})$ and $\exists val.\text{NAT} \sqcup \exists val\text{BOOL}$, or between NAT ∪ BOOL and NAT ⊔ BOOL for that matter, is the difference between local (dynamic) and global (static) choice. □

It will be convenient to introduce a semantical validity relation $\models$ as follows: Write $\mathcal{I}; x \models C$ to abbreviate $x \in C^{\mathcal{I}}$ in which case we say that entity x *satisfies* concept C in the interpretation $\mathcal{I}$. Further, $\mathcal{I}$ is a *model* of C, written $\mathcal{I} \models C$ iff $\forall x \in \Delta^{\mathcal{I}}.\mathcal{I}, x \models C$. Finally, $\models C$ means $\forall \mathcal{I}.\mathcal{I} \models C$. All notions $\mathcal{I}; x \models \Phi$, $\mathcal{I} \models \Phi$ and $\models \Phi$ are extended to sets Φ of concepts in the usual universal fashion.

In typical reasoning tasks the interpretation $\mathcal{I}$ and the entity x in a verification goal such as $\mathcal{I}; x \models C$ are not given directly but are themselves axiomatised by sets of formulas, specifically a *TBox* Θ for $\mathcal{I}$ and an *ABox* Γ for $x \in \Delta^{\mathcal{I}}$. Accordingly, we write $\Theta; \Gamma \models C$ if for all interpretations $\mathcal{I}$ which are models of all axioms in Θ it is the case that every entity x of $\mathcal{I}$ which satisfies all axioms in Γ must also satisfy concept C. Formally, $\forall \mathcal{I}. \forall x \in \Delta^{\mathcal{I}}. (\mathcal{I} \models \Theta \ \& \ \mathcal{I}; x \models \Gamma) \Rightarrow \mathcal{I}; x \models C$. Here is how standard concept reasoning is covered:

- $\Theta; \{C\} \not\models \bot$ iff concept C is *satisfiable* with respect to the TBox Θ, i.e., there exists $\mathcal{I}$ with $\mathcal{I} \models \Theta$ and non-fallible $x \in \Delta^{\mathcal{I}}_c$ such that $x \in C^{\mathcal{I}}$;
- $\Theta; \{C, D\} \not\models \bot$ iff the concepts C and D are *disjoint* with respect to Θ, i.e., $C^{\mathcal{I}}$ and $D^{\mathcal{I}}$ do not share any non-fallible entities in all models $\mathcal{I}$ of Θ;
- $\Theta; \{C\} \models D$ iff concept C is *subsumed* by concept D, i.e., for all $\mathcal{I}$ with $\mathcal{I} \models \Theta$, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; The same can be expressed by $\Theta; \emptyset \models C \sqsubseteq D$ (by reflexivity of $\preceq$);
- $\Theta; \emptyset \models (C \sqsubseteq D) \sqcap (D \sqsubseteq C)$ iff concepts C and D are *equivalent* with respect to Θ, i.e., for all $\mathcal{I}$ with $\mathcal{I} \models \Theta$ we have $C^{\mathcal{I}} = D^{\mathcal{I}}$. We define $C \equiv D$ to be the concept description $(C \sqsubseteq D) \sqcap (D \sqsubseteq C)$.

It is easy to see that $\mathcal{I} \models C \sqcap D$ iff $\mathcal{I} \models C$ and $\mathcal{I} \models D$. It follows that all the above inferences can be reduced to concept subsumption $\Theta; \{C\} \models D$ as in classical DL. Unlike classical DL, however, we cannot reduce concept inferences to the special form $\Theta; \{C\} \not\models \bot$ of satisfiability. Instead, we need to implement the generalised satisfiability check $\Theta; \{C\} \not\models D$ for *arbitrary* D. We will see in Sec. 3.2 how to build a tableau-calculus for such generalised *constructive satisfiability*. Another difference to classical DL is that whenever $\models C \sqcup D$ then $\models C$ or $\models D$. This is known as the *Disjunction Property*, a definitive feature of constructive logic. In classical DL, we have $\models C \sqcup \neg C$ for every concept C even if neither $\models C$ nor $\models \neg C$. The Disjunction Property is the key to proof extraction for $c\mathcal{ALC}$.

$c\mathcal{ALC}$ is related to the constructive modal logic CK (Constructive K) [24,4,15] as $\mathcal{ALC}$ is related to the classical modal system K [8]. In $c\mathcal{ALC}$ the classical principles of the Excluded Middle $C \sqcup \neg C = \top$, double negation $\neg\neg C = C$, the dualities $\exists R.C = \neg\forall R.\neg C$, $\forall R.C = \neg\exists R.\neg C$ and Disjunctive Distribution $\exists R.(C \sqcup D) = \exists R.C \sqcup \exists R.D$ are no longer tautologies but non-trivial TBox statements to axiomatise specialised classes of application scenarios (see Sec. 4). The fact that Excluded Middle, double negation and the dualities do not hold is a feature which $c\mathcal{ALC}$ has in common with standard intuitionistic modal logics such as [10,18,11,20]. It is well known that these principles are non-constructive and therefore need special care. In $c\mathcal{ALC}$, however, we go one step further and refute the principle of Disjunctive Distribution (and, in fact, also the nullary version $\neg\Diamond\bot$) arguing that this principle is not consistent with abstraction. Disjunctive Distribution, which corresponds to the classical $\Diamond$-dual of the normality axiom $\Box(A \wedge B) = \Box A \wedge \Box B$, is commonly accepted for intuitionistic modal logics. In other words, as a modal logic, $c\mathcal{ALC}$ is non-normal regarding $\Diamond$ and thus proofs of decidability and finite model property for standard intuitionistic modal logics (e.g., for $\mathbf{IntK}_{\Box,\Diamond}$ [12][Chap 10]) do not directly apply.

3 Constructive Proof Systems for $c\mathcal{ALC}$

In this section we show simple Hilbert and Gentzen-style deduction systems for $c\mathcal{ALC}$ which admit a direct interpretation of proofs as computations following the Curry-Howard-Isomorphism in which the refinement relation $\preceq$ is treated implicitly. The presence of the semantic refinement structure is visible in the fact that the concept operators $\sqcap$, $\sqcup$, $\sqsubseteq$ on the one hand and $\forall R$, $\exists R$ on the other are primitive and not expressible any more in terms of each other with the help of negation as in classical DLs. This makes sense since all have different computational meaning. According to the Curry-Howard-Isomorphism concept descriptions are types so that, e.g., concept conjunction $\sqcap$ corresponds to Cartesian product $\times$, disjunction $\sqcup$ to disjoint union $+$, subsumption $\sqsubseteq$ to function spaces $\rightarrow$ (see Ex. 5).

3.1 Hilbert Calculus for $c\mathcal{ALC}$

The Hilbert calculus is given by the usual axioms for *intuitionistic propositional logic* [23],

the two *extensionality principles* $\exists K, \forall K$ for role filling as well as the rules of *Modus Ponens* MP and *Necessitation* Nec as seen in Fig. 3. Let the symbol $\vdash_H$ denote Hilbert deduction, i.e., $\Theta \vdash_H C$ if there exists a derivation $C_0, C_1, \ldots, C_n$ such that $C_n = C$ and each C_i $(i \leq n)$ is either a hypothesis $C_i \in \Theta$, or a substitution instance of an axiom scheme from Fig. 3 or arises from earlier concepts C_j $(j < i)$ through MP or Nec. The Hilbert calculus implements TBox-reasoning in the sense that it decides the semantical relationship $\Theta; \emptyset \models C$ which says that C is a *universal* concept in all models of TBox Θ.

$\forall K$: $(\forall R.\,(C \sqsubseteq D)) \sqsubseteq (\forall R.C \sqsubseteq \forall R.D)$
$\exists K$: $(\forall R.\,(C \sqsubseteq D)) \sqsubseteq (\exists R.C \sqsubseteq \exists R.D)$
IPL : All axioms of intuitionistic propositional logic
Nec : If C is a theorem, then so is $\forall R.C$.
MP : If C and $C \sqsubseteq D$ are theorems, then so is D.

Fig. 3: Hilbert Calculus for $c\mathcal{ALC}$.

Theorem 2 (Hilbert Soundness and Completeness). *$\Theta; \emptyset \models C$ iff $\Theta \vdash_H C$.* □

Example 5. We reconsider the example by Brachman et.al. (1991) as reported by [6]:

$$\Theta \vdash_H \text{FOOD} \sqsubseteq \exists \mathit{goesWith}.(\text{COLOR} \sqcap \exists \mathit{isColorOf}.\text{WINE}) \tag{1}$$

in the TBox $\Theta = \{Ax_1, Ax_2\}$ where $Ax_1 =_{df}$ FOOD $\sqsubseteq \exists \mathit{goesWith}.$COLOR and $Ax_2 =_{df}$ COLOR $\sqsubseteq \exists \mathit{isColorOf}.$WINE. The Curry-Howard-Isomorphism can be adapted to understand any Hilbert-proof of (1) as a program construction. For instance, the axiom Ax_1 can be read as a function ax_1 translating FOOD-entities f into COLOR-entities c such that $\mathit{goesWith}(f, c)$ and similarly Ax_2 is a function ax_2 from COLORs c to WINEs w so that $\mathit{isColorOf}(c, w)$. The derivation of (1) then is the construction of a uniform function from FOOD f to pairs (c, w) of COLOR c and WINE w with $\mathit{goesWith}(f, c)$ and $\mathit{isColorOf}(c, w)$. How this can be done formally has been shown by Bozzato et.al. in [6]. In the following we recall (and slightly generalise) their constructions.

With each concept C we associate a set of *realisers* or *information terms* $\text{IT}(C)$. These realisers are then taken as extra ABox parameters so that instead of $\mathcal{I}, x \models C$ we declare what it means that $\mathcal{I}, x \models \langle\alpha\rangle C$ for a particular realiser $\alpha \in \text{IT}(C)$. This so-called *realisability predicate* gives additional constructive semantics to our concepts in the sense that $\mathcal{I}, x \models \langle\alpha\rangle C$ implies $\mathcal{I}, x \models C$.

The sets $\text{IT}(C)$ and refined concepts $\langle\alpha\rangle C$ are defined by induction on C. For our example we only need the following information terms:

- $\text{IT}(A) =_{df} \{\text{tt}\}$ for atomic concepts;
- $\text{IT}(C \sqcap D) =_{df} \text{IT}(C) \times \text{IT}(D)$;
- $\text{IT}(C \sqsubseteq D) =_{df} \text{IT}(C) \to \text{IT}(D)$;
- $\text{IT}(\exists R.C) =_{df} \Delta^{\mathcal{I}} \times \text{IT}(C)$;
- $\text{IT}(\forall R.C) =_{df} \Delta^{\mathcal{I}} \to \text{IT}(C)$.

Realisability is such that

- $\mathcal{I}, x \models \langle\text{tt}\rangle A$ iff $x \in A^{\mathcal{I}}$;
- $\mathcal{I}, x \models \langle\alpha, \beta\rangle(C \sqcap D)$ iff $\mathcal{I}, x \models \langle\alpha\rangle C$ and $\mathcal{I}, x \models \langle\beta\rangle C$;
- $\mathcal{I}, x \models \langle f\rangle(C \sqsubseteq D)$ iff $\forall \alpha \in \text{IT}(C).\ \mathcal{I}, x \models \langle\alpha\rangle C \Rightarrow \mathcal{I}, x \models \langle f\,\alpha\rangle D$;
- $\mathcal{I}, x \models \langle a, \alpha\rangle(\exists R.C)$ iff $(x, a) \in R^{\mathcal{I}}$ and $\mathcal{I}, a \models \langle\alpha\rangle C$;
- $\mathcal{I}, x \models \langle\alpha\rangle(\forall R.C)$ iff $\forall a \in \Delta^{\mathcal{I}}.\,(x, a) \in R^{\mathcal{I}} \Rightarrow \mathcal{I}, a \models \langle\alpha\,a\rangle C$.

One then shows that every proof $\vdash_H C$ generates, for any interpretation $\mathcal{I}$, a function $f{:}\Delta^{\mathcal{I}} \to \text{IT}(C)$ such that $\forall u \in \Delta^{\mathcal{I}}.\,\mathcal{I}, u \models \langle f\,u\rangle C$. Specifically, every of the following Hilbert axioms IPL1:$C \sqsubseteq (D \sqsubseteq C)$, IPL2:$((C \sqsubseteq (D \sqsubseteq E)) \sqsubseteq (C \sqsubseteq D) \sqsubseteq (C \sqsubseteq E))$ and IPL3:$C \sqsubseteq (D \sqsubseteq (C \sqcap D))$ is realised by a λ-term: For instance, IPL1 $=_{df}$ $\lambda u.\lambda x.\lambda y.x$, IPL2 $=_{df}$ $\lambda u.\lambda x.\lambda y.\lambda z.\ (xz)(yz)$ and IPL3 $=_{df}$ $\lambda u.\lambda x.\lambda y.\,(x, y)$. Axiom $\exists K$ is the function $\exists K =_{df}$ $\lambda u.\lambda x.\lambda y.(\pi_1 y, x(\pi_1 y)(\pi_2 y))$. The rule of MP and Nec are refined to

If $\langle\alpha\rangle C$ and $\langle\beta\rangle(C \sqsubseteq D)$ then $\langle\lambda u.(\beta\,u)(\alpha\,u)\rangle D$
If $\langle\alpha\rangle C$ then $\langle\lambda u.\lambda x.\,\alpha\,x\rangle(\forall R.C)$.

In this way, the derivation of (1), up to reductions in the λ-calculus, corresponds to the term $prf = \lambda u.\lambda x.(\pi_1(ax_1\,x), (\pi_2(ax_1\,x), (\pi_1(ax_2(\pi_2(ax_1\,x))), \pi_2(ax_2(\pi_2(ax_1\,x))))))$ which is an information term so that $\forall u.\mathcal{I}, u \models \langle prf\ u\rangle(\text{FOOD} \sqsubseteq \exists goesWith.(\text{COLOR} \sqcap \exists isColorOf.\text{WINE}))$ assuming that $\forall u.\mathcal{I}, u \models \langle ax_1\,u\rangle Ax_1$ and $\forall u.\mathcal{I}, u \models \langle ax_2\,u\rangle Ax_2$. Such realisers ax_1, ax_2 can be obtained from a concrete ABox [6]. □

3.2 Gentzen Tableau Calculus for $c\mathcal{ALC}$

Refutation or tableau calculi play an important role in automated reasoning. These combine both goal-directed proof-search and counter-model construction. In this section we will present such a tableau system for $c\mathcal{ALC}$ based on Gentzen-style sequents. In contrast to tableau systems for classical DL it is consistent with the Curry-Howard Isomorphism and thus permits proof-extraction. In contrast to natural deduction systems such as [6], Gentzen-systems not only support the constructive interpretation of proofs as λ-terms but also formalise tableau-style refutation procedures.

The tableau calculus manipulates Gentzen-style sequents $\Theta; \Sigma; \Gamma \vdash \Phi; \Psi$, where Θ, Γ, Φ are sets of concepts, not necessarily finite, and Σ, Ψ are partial functions mapping role names $R \in N_R$ to sets of concepts $\Sigma(R)$, $\Psi(R)$ which may be infinite, too. The domains of the latter functions are assumed to be finite and identical. We call $dom = dom(\Sigma) = dom(\Psi) \subseteq N_R$ the *domain* of the sequent. A sequent $\Theta; \Sigma; \Gamma \vdash \Phi; \Psi$ formalises and refines the semantic validity relationship $\Theta; \Gamma \models \Phi$ (see page 6) by extra constraints Σ, Ψ as follows: Θ is the TBox which are model assumptions. The ABox is given by the sets Σ, Γ, Φ, Ψ of the sequent. These encode information about individual entities relative to Θ. The first, Σ, Γ specify what we want an entity to satisfy and the latter Φ, Ψ what we do *not* want them to satisfy. The fact that we sandwich entities between explicit positive and negative constraints is the novel constructive aspect of the following Definition 3:

Definition 3 (Constructive Satisfiability). Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \preceq^{\mathcal{I}}, \bot^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation and $a \in \Delta^{\mathcal{I}}$ an entity in $\mathcal{I}$. We say that the pair $(\mathcal{I}, a)$ *satisfies* a sequent $\Theta; \Sigma; \Gamma \vdash \Phi; \Psi$ if $\mathcal{I}$ is a model of Θ, $\mathcal{I} \models \Theta$, and for all $R \in dom$, $L \in \Sigma(R)$, $M \in \Gamma$, $N \in \Phi$, $K \in \Psi(R)$:

- $\forall a'.\forall b.\,(a \preceq a'\ \ \&\ \ a'\,R\,b) \Rightarrow \mathcal{I}, b \models L$, i.e., all R-fillers of a and of its refinements a' are part of all concepts of $\Sigma(R)$;
- $\mathcal{I}, a \models M$, i.e., a and hence all its refinements are part of all concepts of Γ;
- $\mathcal{I}, a \not\models N$, i.e., a is contained in none of the concepts in Φ;
- $\forall b.\,a\,R\,b \Rightarrow \mathcal{I}, b \not\models K$, i.e., none of the R-fillers b of a is contained in any concept of $\Psi(R)$.

A sequent $\Theta; \Sigma; \Gamma \vdash \Phi; \Psi$ is *(constructively) satisfiable*, written $\Theta; \Sigma; \Gamma \not\models \Phi; \Psi$, iff there exists an interpretation $\mathcal{I}$ and entity $a \in \Delta^{\mathcal{I}}$ such that $(\mathcal{I}, a)$ satisfies the sequent. □

The purpose of a *tableau* or *refutation* proof is to establish that an entity specification presented as a sequent is not satisfiable. On the other hand, if no closed tableau can be found and the calculus is complete then the failed proof search implies the existence of a satisfying entity. Our tableau calculus for $c\mathcal{ALC}$ is given by the rules seen in Fig. 4.

Definition 4 (Tableau and Constructive Consistency). A *tableau for* $\Theta; \Sigma; \Gamma \vdash \Phi; \Psi$ is a finite and closed derivation tree $\mathcal{T}$ built using instances of the rules in Fig. 4 which has $\Theta; \Sigma; \Gamma \vdash \Phi; \Psi$ as its root. The sequent is *(constructively) consistent*, written $\Theta; \Sigma; \Gamma \not\vdash \Phi; \Psi$, if no tableau exists for it. □

Our calculus is formulated in the spirit of Gentzen with left introduction rules $\sqcap L$, $\sqcup L$, $\sqsubseteq L$, $\forall L$, $\exists L$ and right introduction rules $\sqcap R$, $\sqcup R$, $\sqsubseteq R$, $\forall R$, $\exists R$ for each logical connective. These rules can be interpreted not only as tableau-style refutation steps but also have computational meaning.

Proposition 5. *The Hilbert and Tableau calculi are equivalent. For any TBox Θ and set of concepts Φ we have $\Theta \vdash_H \Phi$ iff the sequent $\Theta; \emptyset; \emptyset \vdash \Phi; \emptyset$ has a tableau derivation (i.e., is inconsistent).* □

$$\frac{}{\Theta\ ;\ \Sigma\ ;\ \Gamma, C \vdash \Phi, C\ ;\ \Psi}\ Ax \qquad \frac{|\Phi \cup \Psi| \geq 1}{\Theta\ ;\ \Sigma\ ;\ \Gamma, \bot \vdash \Phi\ ;\ \Psi}\ \bot L$$

$$\frac{\Theta\ ;\ \Sigma\ ;\ \Gamma, C, D \vdash \Phi\ ;\ \Psi}{\Theta\ ;\ \Sigma\ ;\ \Gamma, C \sqcap D \vdash \Phi\ ;\ \Psi}\ \sqcap L \qquad \frac{\Theta\ ;\ \Sigma\ ;\ \Gamma \vdash \Phi, C\ ;\ \Psi \qquad \Theta\ ;\ \Sigma\ ;\ \Gamma \vdash \Phi, D\ ;\ \Psi}{\Theta\ ;\ \Sigma\ ;\ \Gamma \vdash \Phi, C \sqcap D\ ;\ \Psi}\ \sqcap R$$

$$\frac{\Theta\ ;\ \Sigma\ ;\ \Gamma \vdash \Phi, C, D\ ;\ \Psi}{\Theta\ ;\ \Sigma\ ;\ \Gamma \vdash \Phi, C \sqcup D\ ;\ \Psi}\ \sqcup R \qquad \frac{\Theta\ ;\ \Sigma\ ;\ \Gamma, C \vdash \Phi\ ;\ \Psi \qquad \Theta\ ;\ \Sigma\ ;\ \Gamma, D \vdash \Phi\ ;\ \Psi}{\Theta\ ;\ \Sigma\ ;\ \Gamma, C \sqcup D \vdash \Phi\ ;\ \Psi}\ \sqcup L$$

$$\frac{\Theta\ ;\ \Sigma\ ;\ \Gamma \vdash \Phi, C\ ;\ \Psi \qquad \Theta\ ;\ \Sigma\ ;\ \Gamma, D \vdash \Phi\ ;\ \Psi}{\Theta\ ;\ \Sigma\ ;\ \Gamma, C \sqsubseteq D \vdash \Phi\ ;\ \Psi}\ \sqsubseteq L \qquad \frac{\Theta\ ;\ \Sigma\ ;\ \Gamma, C \vdash D\ ;\ \emptyset}{\Theta\ ;\ \Sigma\ ;\ \Gamma \vdash \Phi, C \sqsubseteq D\ ;\ \Psi}\ \sqsubseteq R$$

$$\frac{\Theta\ ;\ \Sigma\ ;\ \Gamma \vdash \emptyset\ ;\ [R \mapsto C]}{\Theta\ ;\ \Sigma\ ;\ \Gamma \vdash \Phi, \exists R.C\ ;\ \Psi}\ \exists R \qquad \frac{\Theta\ ;\ \emptyset\ ;\ \Sigma(R), C \vdash \Psi(R)\ ;\ \emptyset}{\Theta\ ;\ \Sigma\ ;\ \Gamma, \exists R.C \vdash \Phi\ ;\ \Psi}\ \exists L$$

$$\frac{\Theta\ ;\ \Sigma \cup [R \mapsto C]\ ;\ \Gamma \vdash \Phi\ ;\ \Psi}{\Theta\ ;\ \Sigma\ ;\ \Gamma, \forall R.C \vdash \Phi\ ;\ \Psi}\ \forall L \qquad \frac{\Theta\ ;\ \emptyset\ ;\ \Sigma(R) \vdash C\ ;\ \emptyset}{\Theta\ ;\ \Sigma\ ;\ \Gamma \vdash \Phi, \forall R.C\ ;\ \Psi}\ \forall R$$

$$\frac{\Theta\ ;\ \Sigma \cup [R \mapsto C]\ ;\ \Gamma \vdash \Phi\ ;\ \Psi \qquad R \in dom}{\Theta, C\ ;\ \Sigma\ ;\ \Gamma \vdash \Phi\ ;\ \Psi}\ Hyp_1 \qquad \frac{\Theta\ ;\ \Sigma\ ;\ \Gamma, C \vdash \Phi\ ;\ \Psi}{\Theta, C\ ;\ \Sigma\ ;\ \Gamma \vdash \Phi\ ;\ \Psi}\ Hyp_2$$

In all rules, the hypotheses Θ, $\Sigma(R)$, Γ and conclusions Φ, $\Psi(R)$ are treated as sets rather than lists. For instance, $\Gamma, C \sqsubseteq D$ in rule $\sqsubseteq L$ is $\Gamma \cup \{C \sqsubseteq D\}$. Hence, if $C \sqsubseteq D \in \Gamma$ then Γ in the premise of $\sqsubseteq L$ is identical to $\Gamma, C \sqsubseteq D$ in the conclusion of the rule. $\emptyset$ is used both as the empty set and the constant function $\emptyset(R) = \emptyset$. $[R \mapsto C]$ is the finite function with domain $\{R\}$ mapping R to the singleton set $\{C\}$ and $\Sigma \cup [R \mapsto C]$ is the union of functions with domain $dom(\Sigma) \cup \{R\}$ such that $(\Sigma \cup [R \mapsto C])(S) = \Sigma(S)$ for $S \neq R$ and $(\Sigma \cup [R \mapsto C])(R) = \Sigma(R) \cup \{C\}$, otherwise.

Fig. 4. Gentzen-tableau rules for $c\mathcal{ALC}$.

Theorem 6 (Strong Soundness and Completeness). *A sequent is satisfiable iff it is consistent, i.e.,* $\Theta; \Sigma; \Gamma \not\models \Phi; \Psi \Leftrightarrow \Theta; \Sigma; \Gamma \not\vdash \Phi; \Psi$. □

A sequent $\Theta; \Sigma; \Gamma \vdash \Phi; \Psi$ is *finite* if it has a finite domain and for all $R \in dom$ the sets $\Sigma(R), \Psi(R)$ as well as Θ, Γ, Φ are finite as well. The tableau rules in Fig. 4 induce a decidable deduction system for finite sequents because all rules have the *sub-formula* property: The premises of each rule only contain (not necessarily proper) sub-formulas of formulas in the conclusion. In fact, the proof of Thm. 6 shows that finite counter-models can be obtained essentially by unfolding unprovable finite end–sequents.

Theorem 7 (Finite Model Property). *A finite sequent is satisfiable iff it is satisfiable in a finite interpretation.* □

As a corollary of Thms. 6 and 7 it follows that consistency of finite sequents is decidable. This is not surprising since $c\mathcal{ALC}$ can be embedded into $\mathcal{ALC}$ with transitive roles, $\mathcal{ALC}_{R^+}$. Therefore, the PSPACE-complexity of $\mathcal{ALC}_{R^+}$ [19] forms an upper bound for satisfiability of $c\mathcal{ALC}$-concepts. On the other hand it is easy to show that concepts in negation normal form (NNF) coincide in $\mathcal{ALC}$ and $c\mathcal{ALC}$. Since all $\mathcal{ALC}$-concepts can be transformed into NNF (in linear time) and satisfiability of $\mathcal{ALC}$-concepts is PSPACE, satisfiability in $c\mathcal{ALC}$ is PSPACE-complete.

4 Some Specialisations between $c\mathcal{ALC}$ and $\mathcal{ALC}$

There are at least three natural dimensions in which $c\mathcal{ALC}$ is a constructive weakening of $\mathcal{ALC}$ corresponding to the axiom schemes of *Non-contradictory Fillers* $\neg\exists R.\bot$, *Disjunctive Distribution* $\exists R.(C \sqcup D) \sqsubseteq (\exists R.C \sqcup \exists R.D)$ and the *Excluded Middle* $C \sqcup \neg C$. Each of them is associated with a specific semantical restriction of interpretations which can be captured by a simple modification (strengthening) of the $c\mathcal{ALC}$ tableau calculus.

In this way, depending on the application at hand, a combination of non-classical DLs may be generated between $c\mathcal{ALC}$ and $\mathcal{ALC}$:

- *Non-contradictory Fillers.* Interpretations without fallible elements, i.e., $\bot^{\mathcal{I}} = \emptyset$, can be axiomatised by the scheme $\neg\exists R.\bot$ which says that any entity can always be refined so it becomes fully defined for role R, i.e., all its R-fillers (if they exist) are non-fallible. In fact, the invalidity of $\neg\exists R.\bot$ is the only effect of fallibility. It indicates the existence of entities all of whose refinements have fallible R-fillers. In the tableau system we exclude fallibility and implement the scheme $\neg\exists R.\bot$ by rule $\bot L^+$ in Fig. 5 which is $\bot L$ without the side-condition $|\Phi \cup \Psi| \geq 1$.
- *Disjunctive Distribution.* If we add the axiom $\exists R.(C \sqcup D) \sqsubseteq (\exists R.C \sqcup \exists R.D)$ then $\exists R.$ distributes over $\sqcup$ and we are essentially saying that role filling via R is *confluent* with refinement, i.e., whenever $xR^{\mathcal{I}}y$ and $x \preceq x'$ then there exists y' such that $y \preceq y'$ and $x'R^{\mathcal{I}}y'$. In the tableau system this can be accommodated by strengthening $\exists R$ to $\exists R^+$ as in Fig. 5, thereby making it perfectly dual to rule $\forall L$.
- *Excluded Middle.* We can implement the scheme $C \sqcup \neg C$ by replacing the intuitionistic rule for implication $\sqsubseteq R$ by the classical $\sqsubseteq R^+$ seen in Fig. 5. The difference is that in applying $\sqsubseteq R^+$ backwards we do not lose the contexts Φ, Ψ like in $\sqsubseteq R$. This is the standard extension which turns the intuitionistic into the classical sequent calculus.

$$\frac{}{\Theta\ ;\ \Sigma\ ;\ \Gamma, \bot \vdash \Phi\ ;\ \Psi}\ \bot L^+$$

$$\frac{\Theta; \Sigma; \Gamma \vdash \Phi; \Psi \cup [R \mapsto C]}{\Theta; \Sigma; \Gamma \vdash \Phi, \exists R.C; \Psi}\ \exists R^+$$

$$\frac{\Theta; \Sigma; \Gamma, C \vdash \Phi, D; \Psi}{\Theta; \Sigma; \Gamma \vdash \Phi, C \sqsubseteq D; \Psi}\ \sqsubseteq R^+$$

Fig. 5. Variations of Tableau Rules.

Note that $c\mathcal{ALC}$ + Excluded Middle is properly more expressive than $\mathcal{ALC}$. Therefore, $c\mathcal{ALC}$ is not the intuitionistic analog of $\mathcal{ALC}$ in the sense of Simpson [20] but a *constructive* or *sub-intuitionistic* analog. In fact, $c\mathcal{ALC}$ + Non-contradictory Fillers yields the multimodal version of Wijesekera's constructive modal logic [24]. However, if we further add Disjunctive Distribution and the axiom scheme $(\exists R.C \to \forall R.D) \to \forall R.(C \to D)$ (not discussed above) then we obtain the multi-modal version $i\mathcal{ALC}$ of the standard intuitionistic logic of Fischer-Servi [11] known as IK [20] or **FS** [12]. See [8] for a deeper discussion of the difference between $c\mathcal{ALC}$ and $i\mathcal{ALC}$. Here it suffices to point out that $i\mathcal{ALC}$ is a special theory of $c\mathcal{ALC}$ which enforces additional relationships between role filling and refinement which may or may not be adequate for a given application.

5 Conclusion and Future Work

We presented a new constructive interpretation of DL which refines the classical one and generates a family of theories that admit computational interpretations of proofs in line with the Curry-Howard Isomorphism. This new interpretation is consistent with the idea of concepts comprising abstract entities with hidden fine–structure. It supports intensional ABox and TBox theories with semantic slack in the sense that not all aspects of the low-level structure of entities can necessarily be captured by the concept language. This gives rise to the notion of *constructive satisfiability* and a stronger form of OWA, which we tentatively call the *Evolving Open World Assumption*.

In this paper we applied this interpretation to $\mathcal{ALC}$ as the core DL obtaining $c\mathcal{ALC}$ together with sound and complete Hilbert and Tableau deduction systems. The semantics is general enough that it should be applicable to other DLs, too. It is conservative in that all constructions of $c\mathcal{ALC}$ are sound in $\mathcal{ALC}$. The point is that $c\mathcal{ALC}$ does not permit constructions which are incompatible with refinement. We have given examples where $\mathcal{ALC}$ would not be adequate. $c\mathcal{ALC}$ enjoys semantical robustness and admits decidable tableau with proof extraction and counter-model construction. Where the application supports it we can specialise $c\mathcal{ALC}$ back towards $\mathcal{ALC}$ by adding axioms or strengthen some tableau rules suitably as discussed.

We aim to extend $c\mathcal{ALC}$ for the domain of mass data business auditing by designing specialised example ontologies. We plan to extract and automate auditing processes from proof terms by using the calculus as an interactive design and type specification system of data streams and audit component interfaces. Towards this end we will give full separation between ABox and TBox reasoning, specifically explicit representation of ABoxes in sequents.

As in standard DL tableaux each node would then describe information about a full ABox rather than a single entity, which yields a more global construction.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. We also acknowledge the support by the German Research Council (DFG).

References

1. A. Artale and E. Franconi. A survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence*, 30(1–4), 2001.
2. A. Artale, C. Lutz, and D. Toman. A description logic of change. In *Int'l Workshop on Description Logics (DL 2006)*, pages 97–108, 2006.
3. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, 2003.
4. G. Bellin, V. de Paiva, and E. Ritter. Extended Curry-Howard correspondence for a basic constructive modal logic. In *Methods for Modalities II*, November 2001.
5. A. Borgida. Diachronic description logics. In *Int'l Workshop on Description Logics (DL 2001)*, pages 106–112, 2001.
6. L. Botazzo, M. Ferrari, C. Fiorentini, and G. Fiorino. A constructive semantics for ALC. In *Int'l Workshop on Description Logics (DL 2007)*, pages 219–226, 2007.
7. O. Brunet. A logic for partial system description. *Journal of Logic and Computation*, 14(4):507–528, 2004.
8. V. de Paiva. Constructive description logics: what, why and how. In *Context Representation and Reasoning*, Riva del Garda, August 2006.
9. M. Dürig and Th. Studer. Probabilistic ABox reasoning: Preliminary results. In *Int'l Workshop on Description Logics (DL 2005)*, 2005.
10. W. B. Ewald. Intuitionistic tense and modal logic. *Journal of Symbolic Logic*, 51, 1986.
11. G. Fischer-Servi. Semantics for a class of intuitionistic modal calculi. In M. L. Dalla Chiara, editor, *Italian Studies in the Philosophy of Science*, pages 59–72. Reidel, 1980.
12. D. M. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-dimensional modal logics*. Elsevier, 2003.
13. S. Hölldobler, Nguyen Hoang Nga, and Tran Dinh Khang. The fuzzy description logic ALCFLH. In *Int'l Workshop on Description Logics (DL 2005)*, 2005.
14. Yue Ma, P. Hitzler, and Zuoquan Lin. Paraconsistent resolution for four-valued description logics. In *Int'l Workshop on Description Logics (DL 2007)*, 2007.
15. M. Mendler and V. de Paiva. Constructive CK for contexts. In L. Serafini and P. Bouquet, editors, *Context Representation and Reasoning (CRR-2005)*, volume 13 of *CEUR Proceedings*, July 2005. Also presented at the Association for Symbolic Logic Annual Meeting, Stanford University, USA, 22nd March 2005.
16. A. Paschke. Typed hybrid description logic programs with order-sorted semantic web type systems on OWL and RDFS. Technical report, TU Munich, December 2005.
17. P. F. Patel-Schneider. A four-valued semantics for terminological logics. *Artificial Intelligence*, 38:319–351, 1989.
18. G. Plotkin and C. Stirling. A framework for intuitionistic modal logics. In *Theoretical aspects of reasoning about knowledge*, Monterey, 1986.
19. Ulrike Sattler. A concept extended with different kinds of transitive roles. In S. Hölldobler G. Görtz, editor, *German Annual Conference on Artificial Intelligence (KI96)*, volume 1137 of *Lecture Notes in Computer Science*, pages 333–345. Springer, 1996.
20. A.K. Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD thesis, University of Edinburgh, 1994.
21. U. Straccia. Fuzzy ALC with fuzzy concrete domains. In *Int'l Workshop on Description Logics (DL 2005)*, 2005.
22. A. S. Troelstra. Realizability. In S. R. Buss, editor, *Handbook of Proof Theory*, chapter VI, pages 407–474. Elsevier, 1998.
23. D. van Dalen. Intuitionistic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume III, chapter 4, pages 225–339. Reidel, 1986.
24. D. Wijesekera. Constructive modal logic I. *Annals of Pure and Applied Logic*, 50:271–301, 1990.