Reasoning on Information Term Semantics with ASP for Constructive \mathcal{EL}_{\perp}

Loris Bozzato¹ and Camillo Fiorentini²

Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy
 DI, Univ. degli Studi di Milano, Via Celoria 18, 20133 Milano, Italy

Abstract. Constructive description logics represent different re-interpretations of description logics (DLs) under constructive semantics. Constructive description logics have been mostly studied for their formal properties, while limited practical approaches have been shown for their use in Knowledge Representation languages and tools (which, on the other hand, constitute the distinctive applications of description logics). To address this aspect, we recently studied the relation of constructive DLs based on Information Term semantics with Answer Set semantics in the context of the positive logic \mathcal{EL} .

In this paper we continue this study in the direction of more expressive DLs by considering the introduction of negative information, leading to a constructive interpretation for the DL \mathcal{EL}_{\perp} . We show that formal results linking the constructive semantics to answer set semantics can be extended to the case of negative information in \mathcal{EL}_{\perp} .

1 Introduction

Constructive description logics are interpretations of description logics (DLs) under different constructive semantics. The application of such non-classical semantics to description logics is motivated by the interest in applying the formal properties of constructive semantics to different aspects of knowledge representation. Starting from different constructive semantics, several constructive description logics have been proposed, see e.g. [6,9,16].

Constructive description logics have been mostly studied from a theoretical viewpoint, and they have also been applied to tackle different representation and reasoning problems (see, e.g., [4,12,13,15]); however, the interaction between formal and practical aspects of constructive DLs has been scarcely investigated. To bridge this gap, in [2] we have introduced a simple constructive DL based on \mathcal{EL} and we have disussed its relationship with Answer Set Programming (ASP). From the practical point of view, by taking advantage of such a relation, we have presented a datalog encoding managing one reasoning task over the constructive semantics (namely, the generation of valid "states" of a knowledge base) and we have developed a prototype based on the standard OWL-EL profile and "off the shelf" tools for manipulation of OWL 2 ontologies and ASP reasoning.

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Our constructive interpretation of \mathcal{EL} is based on information term semantics [5,9]: the information terms (ITs) are syntactical objects that provide a constructive justification for the classical truth of a formula. Information terms have been used to represent the state or answer of a formula (and, for extension, a notion of "snapshot" representing a valid state of a knowledge base). In particular, [2] follows the direction studied in [10], where the relation between information term and answer set semantics has been first studied over propositional theories. In fact, we remark that [2] extends for the first time the study of relations between IT semantics and ASP to the context of constructive description logics.

In this paper, we continue the investigation started in [2] by "pushing the envelope" towards more expressive DLs: as a first step we introduce falsum in \mathcal{EL} , thus providing a constructive interpretation for the DL \mathcal{EL}_{\perp} . Then, we show how the results linking IT semantics and answer set semantics can be extended in presence of negative information. Intuitively, following [10], negative information can be represented similarly to default negation in ASP: negative formulas are used as constraints and answer sets are formulated over a suitable positive reduction of the input formulas w.r.t. negative information. Our goal in this paper is to show that results connecting IT semantics and answer set semantics can be expanded to \mathcal{EL}_{\perp} in presence of negative information. These connections can be then leveraged to reason on the constructive reading of \mathcal{EL}_{\perp} using ASP tools, for example for computing the valid ITs of a knowledge base as in [2]: this task represents an essential step in order to use transformations of ITs in reasoning, e.g., to represent change of states as in [4].

In the following sections, we first present (see Section 2) the definition of the IT interpretation of \mathcal{EL}_{\perp} (that we call \mathcal{ELc}_{\perp}); then, in Section 3 we extend the definition of answer sets for formulas and answer sets to this logic and we show how to extend the results connecting the IT semantics and answer set semantics for \mathcal{ELc}_{\perp} . Finally, in Section 4 we briefly discuss how to extend to \mathcal{ELc}_{\perp} the datalog implementation from [2] for the generation of IT of knowledge bases. Proofs of the main results from Section 3 are provided in the Appendix.

2 Constructive Description Logic $\mathcal{EL}c_{\perp}$

We present a constructive semantics based on information term semantics (as in \mathcal{BCDL} [9]) for the description logic \mathcal{EL}_{\perp} [1]; we refer to this logic as \mathcal{ELc}_{\perp} .

Syntax. The language \mathcal{L} for \mathcal{ELc}_{\perp} is based on the disjoint denumerable sets NR of role names, NC of concept names and NI of individual names. In addition, we introduce a set NG of special concepts, called generators, where NG \cap NC $= \emptyset$. Generators are used in the definition of a limited form of subsumption, which facilitates the characterization of the logic in a constructive semantics. A generator G is an atomic concept associated with a finite set of individual names DOM(G) (the domain of G) which fixes the interpretation of G. In our language, we use bounded quantified formulas of the kind $\forall_G C$, meaning that every ele-

ment of DOM(G) belongs to the concept C; thus, the formula $\forall_G C$ can be read as the subsumption relation $G \subseteq C$.

In the language \mathcal{L} for $\mathcal{EL}c_{\perp}$, concepts C are inductively defined as:

$$C ::= \top \mid \bot \mid A \mid \neg C_1 \mid C_1 \sqcap C_2 \mid \exists R.C_1$$

where $A \in \mathbb{NC} \cup \mathbb{NG}$ and $R \in \mathbb{NR}$. The formulas K of \mathcal{L} are defined as:

$$K ::= R(c,d) \mid C(c) \mid \forall_G C$$

where $c, d \in NI$, $R \in NR$, $G \in NG$ and C is a concept (as defined above). We point out that $\forall_G C$ represents the inclusion (subsumption) between concepts G and C. A formula K is atomic if K has the form R(s,t), T(t), L(t), L(t), with $A \in NC \cup NG$; K is simple if it is an atomic or a negated formula (namely, $K = \neg C(t)$); K is positive if it does not contain a negation or L.

A theory \mathcal{K} (that is, a knowledge base) is a finite set of formulas which is partitioned as usual into ABox and TBox. The ABox contains formulas of the kind C(d) and R(c,d), with $R \in NR$, $c,d \in NI$ and C is a concept. The TBox contains formulas of the kind $\forall_G C$, representing subsumption axioms.

In the following we refer to languages $\mathcal{L}_{\mathcal{N}}$ restricted to finite subsets \mathcal{N} of NI. Given a finite $\mathcal{N} \subseteq \text{NI}$, let $\text{NG}_{\mathcal{N}}$ be the set of generators $G \in \text{NG}$ such that $\text{DOM}(G) \subseteq \mathcal{N}$. By $\mathcal{L}_{\mathcal{N}}$ we denote the language built on the set \mathcal{N} of individual names, the set NC of concept names, the set NR of role names and the set $\text{NG}_{\mathcal{N}}$ of generators.

Classical semantics. A model \mathcal{M} for $\mathcal{L}_{\mathcal{N}}$ is a pair $\langle \Delta^{\mathcal{M}}, \cdot^{\mathcal{M}} \rangle$, where the domain $\Delta^{\mathcal{M}}$ is a non-empty set and $\cdot^{\mathcal{M}}$ is a valuation map such that: for every $c \in \mathcal{N}$, $c^{\mathcal{M}} \in \Delta^{\mathcal{M}}$; for every $C \in \mathbb{NC}$, $C^{\mathcal{M}} \subseteq \Delta^{\mathcal{M}}$; for every $R \in \mathbb{NR}$, $R^{\mathcal{M}} \subseteq \Delta^{\mathcal{M}} \times \Delta^{\mathcal{M}}$; for every $G \in \mathbb{NG}_{\mathcal{N}}$, if $\mathrm{DOM}(G) = \{c_1, \ldots, c_n\}$, then $G^{\mathcal{M}} = \{c_1^{\mathcal{M}}, \ldots, c_n^{\mathcal{M}}\}$. Classical interpretation of non-atomic concepts is defined as usual:

$$\bot^{\mathcal{M}} = \emptyset \quad \top^{\mathcal{M}} = \Delta^{\mathcal{M}} \quad (C_1 \sqcap C_2)^{\mathcal{M}} = C_1^{\mathcal{M}} \cap C_2^{\mathcal{M}} \quad (\neg C)^{\mathcal{M}} = \Delta^{\mathcal{M}} \setminus C^{\mathcal{M}}$$
$$(\exists R.C)^{\mathcal{M}} = \{ c \in \Delta^{\mathcal{M}} \mid \exists d \in \Delta^{\mathcal{M}} \text{ s.t. } (c, d) \in R^{\mathcal{M}}, \ d \in C^{\mathcal{M}} \}$$

A formula K is valid in \mathcal{M} , written $\mathcal{M} \models K$, iff one of the following conditions holds:

$$\mathcal{M} \models R(c, d) \text{ iff } (c^{\mathcal{M}}, d^{\mathcal{M}}) \in R^{\mathcal{M}}$$

 $\mathcal{M} \models C(d) \text{ iff } d^{\mathcal{M}} \in C^{\mathcal{M}}$
 $\mathcal{M} \models \forall_G C \text{ iff } G^{\mathcal{M}} \subseteq C^{\mathcal{M}}$

If Γ is a set of formulas, $\mathcal{M} \models \Gamma$ means that $\mathcal{M} \models K$, for every K in Γ .

³ While non-conventional in DL languages, as noted in [9], the definition of generators with a fixed domain simplifies the following presentations of the constructive semantics. Alternatively, domain of generators can be defined by extending the language with nominals and directly including the domains declaration in the knowledge base.

⁴ Note that concept negation $\neg C_1$ does not appear in the standard syntax of $\overline{\mathcal{EL}}_{\perp}$. It can be simulated by introducing a new concept \overline{C}_1 and the axiom $C_1 \sqcap \overline{C}_1 \sqsubseteq \bot$.

Example 1. We reprise the running example presented in [2], inspired by the classical example of food and wine pairings from [8], including negative information to introduce constraints. The set NI contains names for wines, colors and regions. The set NC contains the concepts Wine, defining wines, and ExcRegion, defining regions to be excluded. We introduce the generators Food and Color, defining foods and colors respectively, and we set: $DOM(Food) = \{fish, meat\}$ and $DOM(Color) = \{red, white\}$, where fish, meat, red and white are elements of NI. The set NR contains the role names goesWith, to represent the matches between wine colors and foods, isColorOf, to associate a color with a wine, hasRegion, to map a wine to its origin region. We introduce the knowledge base \mathcal{K}_W consisting of the TBox axioms:

```
(Ax_1) \forall_{Food} \exists goes With. Color
(Ax_2) \forall_{Color} \exists is Color Of. (Wine <math> \sqcap  \neg (\exists hasRegion. ExcRegion))
```

Intuitively, axiom Ax_1 asserts that each *Food* matches an appropriate wine Color; Ax_2 states that for every Color there is at least a Wine not originating from an ExcRegion (excluded region). Note that the negated subconcept of Ax_2 behaves like a constraint. The ABox of \mathcal{K}_W consists of the following assertions:

```
 \begin{array}{ll} Wine (barolo) & is Color Of (red, barolo) \\ Wine (chardonnay) & is Color Of (white, chardonnay) \\ Wine (cabernet) & is Color Of (red, cabernet) \\ has Region (barolo, piedmont) & goes With (fish, white) \\ has Region (chardonnay, lombardy) & goes With (meat, red) \\ has Region (cabernet, california) & Exc Region (california) \end{array}
```

We can take as \mathcal{N} any finite set containing individual names occurring in the ABox. We point out that a classical model \mathcal{M} for \mathcal{K}_W must interpret the generator Food as the set $\{fish^{\mathcal{M}}, meat^{\mathcal{M}}\}$ and Color as $\{red^{\mathcal{M}}, white^{\mathcal{M}}\}$. Instead, ExcRegion can be interpreted by any superset of $\{california^{\mathcal{M}}\}$ satisfying the axioms. \diamondsuit

Information term semantics. The constructive semantics for $\mathcal{EL}_{c_{\perp}}$ is based on the notion of information term [17]. Information term semantics is related to the BHK (Brower-Heyting-Kolmogorov) interpretation of logical connectives [18]: intuitively, an information term η for a formula K is a syntactical object that constructively justifies the truth of K in a classical model \mathcal{M} . For instance, the validity of the formula $\exists R.C(d)$ in a model \mathcal{M} can be explained by an information term (e,η) providing the filler e s.t. $(d^{\mathcal{M}}, e^{\mathcal{M}}) \in R^{\mathcal{M}}$ and, inductively, an information term η justifying $e^{\mathcal{M}} \in C^{\mathcal{M}}$. A simple formula K does not need any constructive explanation, thus the associated information term is an atom, denoted by tt. Given a finite subset \mathcal{N} of NI and a formula K of $\mathcal{L}_{\mathcal{N}}$, we define

the set of information terms $IT_{\mathcal{N}}(K)$ by induction on K as follows.

```
\begin{split} \operatorname{IT}_{\mathcal{N}}(K) &= \{\, \operatorname{tt} \,\}, \text{ if } K \text{ is a simple formula} \\ \operatorname{IT}_{\mathcal{N}}((C_1 \sqcap C_2)(c)) &= \{ \ (\alpha,\beta) \mid \alpha \in \operatorname{IT}_{\mathcal{N}}(C_1(c)) \text{ and } \beta \in \operatorname{IT}_{\mathcal{N}}(C_2(c)) \ \} \\ \operatorname{IT}_{\mathcal{N}}(\exists R.C(c)) &= \{ \ (d,\alpha) \mid d \in \mathcal{N} \text{ and } \alpha \in \operatorname{IT}_{\mathcal{N}}(C(d)) \ \} \\ \operatorname{IT}_{\mathcal{N}}(\forall_G C) &= \{ \ \phi : \operatorname{DOM}(G) \to \bigcup_{d \in \operatorname{DOM}(G)} \operatorname{IT}_{\mathcal{N}}(C(d)) \mid \phi(d) \in \operatorname{IT}_{\mathcal{N}}(C(d)) \ \} \end{split}
```

Note that no constructive information is associated with negated sub-formulas, which are treated similarly to atomic formulas: negative sub-formulas can be seen as "constraints" that need to be verified by the models of the knowledge base.

The justification of formulas in classical models with respect to one of their information terms is given by the realizability relation. Let \mathcal{M} be a model for $\mathcal{L}_{\mathcal{N}}$, K a formula of $\mathcal{L}_{\mathcal{N}}$ and $\eta \in \text{IT}_{\mathcal{N}}(K)$. We define the realizability relation $\mathcal{M} \rhd \langle \eta \rangle K$ by induction on the structure of K:

$$\mathcal{M} \rhd \langle \mathsf{tt} \rangle K \text{ iff } \mathcal{M} \models K, \text{ where } K \text{ is a simple formula}$$

$$\mathcal{M} \rhd \langle (\alpha, \beta) \rangle C_1 \sqcap C_2(c) \text{ iff } \mathcal{M} \rhd \langle \alpha \rangle C_1(c) \text{ and } \mathcal{M} \rhd \langle \beta \rangle C_2(c)$$

$$\mathcal{M} \rhd \langle (d, \alpha) \rangle \exists R. C(c) \text{ iff } \mathcal{M} \models R(c, d) \text{ and } \mathcal{M} \rhd \langle \alpha \rangle C(d)$$

$$\mathcal{M} \rhd \langle \phi \rangle \forall_G C \text{ iff, for every } d \in \mathsf{DOM}(G), \, \mathcal{M} \rhd \langle \phi(d) \rangle C(d)$$

Example 2. Let us consider the knowledge base \mathcal{K}_W defined in Example 1. An element $\phi \in \text{IT}_{\mathcal{N}}(Ax_1)$ is a function mapping each food $f \in \text{DOM}(Food)$ to an information term $\phi(f) \in \text{IT}_{\mathcal{N}}(\exists goes With.Color(f))$. Thus, every $\phi(f)$ has the form (c, tt) , meaning that c is a proper color for f. An element $\psi_1 \in \text{IT}_{\mathcal{N}}(Ax_1)$ is:

[
$$fish \mapsto (white, tt), meat \mapsto (red, tt)$$
]

Similarly, we can define $\psi_2 \in IT_{\mathcal{N}}(Ax_2)$ as follows:

```
[red \mapsto (barolo, (\mathtt{tt}, \mathtt{tt})), white \mapsto (chardonnay, (\mathtt{tt}, \mathtt{tt}))]
```

where there is no constructive information associated with the negated subconcept of Ax_2 . We point out that every model \mathcal{M} of the ABox of \mathcal{K}_W satisfies both $\mathcal{M} \rhd \langle \psi_1 \rangle Ax_1$ and $\mathcal{M} \rhd \langle \psi_2 \rangle Ax_2$.

The following result, provable by induction on the structure of K, shows the relation between classical and constructive semantics (see Lemma 2 in [9]).

Proposition 1. Let \mathcal{N} be a finite subset of NI, K a formula of $\mathcal{L}_{\mathcal{N}}$ and $\eta \in IT_{\mathcal{N}}(K)$. For every model \mathcal{M} , $\mathcal{M} \rhd \langle \eta \rangle K$ implies $\mathcal{M} \models K$.

Thus, the constructive semantics preserves the classical declarative reading of DL formulas. The converse of Proposition 1 does not hold in general, unless we assume stronger conditions, such as the reachability of \mathcal{M} (i.e., every element in the domain of \mathcal{M} is denoted by a constant, see e.g. [7]).

First-order translation. We introduce a first-order translation of $\mathcal{EL}_{\mathbf{L}_{\perp}}$ formulas (see similar translations in [3]), that we exploit as an intermediate representation to define the notion of reduction of a formula. Let \mathcal{N} be a finite subset of NI. By $\mathcal{L}^1_{\mathcal{N}}$ we denote the first-order language having constants \mathcal{N} , a unary predicate symbol A, for every $A \in \mathbb{NC} \cup \mathbb{NG}$, a binary relation symbol R, for every $R \in \mathbb{NR}$, the logical constants \mathbf{t} (true) and \mathbf{f} (false), the connectives \neg and \wedge . We also introduce unary function symbols $f_{\exists R.C}$ to be used as Skolem functions to properly translate existential concepts $\exists R.C$. Firstly, we introduce the translation γ_C mapping a concept C to a first-order formula F of $\mathcal{L}^1_{\mathcal{N}}$, having x as possible free variable; $\gamma_C(t)$ denotes the formula obtained by replacing every occurrence of x in γ_C with the term t.

In translating C, we stipulate that different occurrences of the same existential subformulas $\exists R.D$ of C are associated with different Skolem function (e.g., $f^1_{\exists R.D}$, $f^2_{\exists R.D}$, and so on). Given a formula $K \in \mathcal{L}_{\mathcal{N}}$, the first-order translation $\Phi(K)$ of K is defined as follows:

$$C(d) \mapsto_{\Phi} \gamma_C(d)$$
 $R(c,d) \mapsto_{\Phi} R(c,d)$
 $\forall_G C \mapsto_{\Phi} \gamma_C(c_1) \wedge \cdots \wedge \gamma_C(c_n), \text{ where } \{c_1, \dots, c_n\} = \text{DOM}(G)$

Example 3. Let \mathcal{K}_W be the knowledge base from Example 1. The first-order translation of Ax_1 is $(C = \exists goesWith.Color)$:

$$\Phi(Ax_1) = \gamma_C(meat) \land \gamma_C(fish) \qquad \gamma_C = goesWith(x, f_C(x)) \land Color(f_C(x))$$

Similarly, Ax_2 is translated as follows:

$$D = \exists isColorOf.(Wine \sqcap \neg E) \qquad E = \exists hasRegion.ExcRegion$$

$$\Phi(Ax_2) = \gamma_D(red) \land \gamma_D(white)$$

$$\gamma_D = isColorOf(x, f_D(x)) \land \gamma_{(Wine \sqcap \neg E)}(f_D(x))$$

$$\gamma_{(Wine \sqcap \neg E)} = Wine(x) \land \neg (hasRegion(x, f_E(x)) \land ExcRegion(f_E(x)))$$

An interpretation \mathcal{I} for $\mathcal{L}^1_{\mathcal{N}}$ is a classical interpretation for the language $\mathcal{L}^1_{\mathcal{N}}$ satisfying the following condition:

for each generator
$$G \in NG_N$$
, $\mathcal{I} \models G(c)$ iff $c \in DOM(G)$ (*)

 \Diamond

Next proposition states the correspondence between first-order translation and classical semantics:

Proposition 2. Let K be a formula of $\mathcal{L}_{\mathcal{N}}$.

- (i). If $\mathcal{M} \models K$, then there is an interpretation \mathcal{I} for $\mathcal{L}^1_{\mathcal{N}}$ such that $\mathcal{I} \models \Phi(K)$.
- (ii). If $\mathcal{I} \models \Phi(K)$, with \mathcal{I} an interpretation for $\mathcal{L}^1_{\mathcal{N}}$, then there is an interpretation \mathcal{M} for $\mathcal{L}_{\mathcal{N}}$ such that $\mathcal{M} \models K$.

3 Answer Set Semantics for Formulas and Information Terms

Following the construction in [10], we adapt the definitions concerning logic programs with nested expressions [14] to $\mathcal{EL}_{c_{\perp}}$ formulas. The first step to define a notion of answer sets for $\mathcal{EL}_{c_{\perp}}$ formulas is thus to define a notion of interpretation akin to the one used in logic programs. In this regard, an lp-interpretation I over $\mathcal{L}_{\mathcal{N}}$ is a set of atomic formulas H of $\mathcal{L}_{\mathcal{N}}$ such that H = A(c) or H = R(c, d), where $A \in \mathbb{NC} \cup \mathbb{NG}$, $R \in \mathbb{NR}$ and $c, d \in \mathcal{N}$. Given a formula K of $\mathcal{L}_{\mathcal{N}}$, the satisfiability relation $I \models K$, and its extension to sets of formulas Γ , is defined as follows:

```
\begin{split} I &\models \top(c) \text{ for every } c \in \mathcal{N} \\ I &\models H \text{ iff } H \in I, \text{ where either } H = A(c) \text{ and } A \in \mathtt{NC} \cup \mathtt{NG}, \text{ or } H = R(c,d) \\ I &\models G(c) \text{ iff } c \in \mathtt{DOM}(G), \text{ where } G \in \mathtt{NG} \\ I &\models \neg C(c) \text{ iff } I \not\models C(c) \\ I &\models C \sqcap D(c) \text{ iff } I \models C(c) \text{ and } I \models D(c) \\ I &\models \exists R.C(c) \text{ iff there is } d \in \mathcal{N} \text{ such that } R(c,d) \in I \text{ and } I \models C(d) \\ I &\models \forall_G C \text{ iff for every } e \in \mathtt{DOM}(G), I \models C(e) \\ I &\models \Gamma \text{ iff } I \models K \text{ for every } K \in \Gamma \end{split}
```

We remark that $I \not\models \bot(c)$, for every $c \in \mathcal{N}$; moreover, I satisfies condition (*) of previous section. The definition of $J \models F$, where J is an lp-interpretation over $\mathcal{L}^1_{\mathcal{N}}$ and F a formula of $\mathcal{L}^1_{\mathcal{N}}$, is similar. We point out that J must be a set of formulas of the kind $A(t_1)$ and $R(t_1, t_2)$, where $A \in \mathbb{NC} \cup \mathbb{NG}$, $R \in \mathbb{NR}$, t_1 and t_2 are ground terms (namely, they do not contain variables).

Given an lp-interpretation I over $\mathcal{L}_{\mathcal{N}}$, the extension I^+ of I is an lp-interpretation over $\mathcal{L}^1_{\mathcal{N}}$ obtained by properly interpreting Skolem functions. More specifically, I^+ is the smallest extension of I satisfying the following property: for every formula $K = \exists R.C(c)$ of $\mathcal{L}_{\mathcal{N}}$ such that $I \models K$, let d such that $R(c, d) \in I$ and $I \models C(d)$, and let f_K be a Skolem function associated with $\exists R.C$; then, I^+ contains the formulas $R(c, f_K(c))$ and $d = f_K(c)$.

The reduct of a formula $K \in \mathcal{L}_{\mathcal{N}}$ w.r.t an lp-interpretation I, denoted by K^I , is obtained from the formula $\Phi(K)$ of $\mathcal{L}^1_{\mathcal{N}}$ by replacing every negated subformula $\neg C(c)$ with either t or f, in compliance with I^+ . Formally, the reduct of a formula $F \in \mathcal{L}^1_{\mathcal{N}}$ w.r.t. an lp-interpretation J (over $\mathcal{L}^1_{\mathcal{N}}$), denoted by F^J , is the formula of $\mathcal{L}^1_{\mathcal{N}}$ inductively defined as follows:⁵

$$\begin{split} H^J &= H, \text{ if } H \text{ is an atomic formula of } \mathcal{L}^1_{\mathcal{N}} \\ (\neg C)^J &= \begin{cases} \texttt{f} & \text{if } J \models C \\ \texttt{t} & \text{otherwise} \end{cases} & (C \land D)^J = C^J \land D^J \end{split}$$

We define the reduct on formulas K of $\mathcal{L}_{\mathcal{N}}$ and set of formulas Γ as follows:

$$K^{I} = (\Phi(K))^{I^{+}}$$
 $\Gamma^{I} = \{ K^{I} \mid K \in \Gamma \}$

⁵ Atomic formulas of $\mathcal{L}^1_{\mathcal{N}}$ are the Φ -images of atomic formulas of $\mathcal{L}_{\mathcal{N}}$.

We remark that, for positive formulas K, we have $K^{I} = \Phi(K)$.

Example 4. We show the definition of reduct of axioms in \mathcal{K}_W (see Example 1) based on the first-order translation displayed in Example 3. Let I be the lp-interpretation coinciding with the ABox of \mathcal{K}_W . The extension I^+ is obtained by adding to I the formulas $(D = \exists isColorOf.(Wine \sqcap \neg E))$ and $E = \exists hasRegion.ExcRegion)$:

```
isColorOf(red, f_D(red)), f_D(red) = barolo,

isColorOf(white, f_D(white)), f_D(white) = chardonnay,

hasRegion(f_D(red), f_E(f_D(red))), f_E(f_D(red)) = piedmont,

hasRegion(f_D(white), f_E(f_D(white))), f_E(f_D(white)) = lombardy, ...
```

The reduction \mathcal{K}_W^I is computed as follows: since Ax_1 is a positive formula, then $Ax_1^I = \Phi(Ax_1)$. Instead, Ax_2^I coincides with the formula $\Phi(Ax_2)^{I^+}$ such that:

```
\Phi(Ax_2)^{I^+} = (\gamma_D(red))^{I^+} \wedge (\gamma_D(white))^{I^+} 

(\gamma_D(red))^{I^+} = isColorOf(red, f_D(red)) \wedge Wine(f_D(red)) \wedge \mathsf{t} 

(\gamma_D(white))^{I^+} = isColorOf(white, f_D(white)) \wedge Wine(f_D(white)) \wedge \mathsf{t}
```

Note that both occurrences of the constraint $\neg E$ are reduced to t, since we have that, for $c \in \{red, white\}$, $I^+ \not\models (hasRegion(c, f_E(c)) \land ExcRegion(f_E(c))$. \diamondsuit

We introduce the notion of answer set for formulas:

Definition 1. An lp-interpretation I is an answer set for a set of positive formulas $\Gamma \subseteq \mathcal{L}_{\mathcal{N}}$ (resp. $\Gamma \subseteq \mathcal{L}_{\mathcal{N}}^1$) iff $I \models \Gamma$ and, for every $I' \subseteq I$, $I' \models \Gamma$ implies I' = I. An lp-interpretation I is an answer set for a set of formulas $\Gamma \subseteq \mathcal{L}_{\mathcal{N}}$ iff I^+ is an answer set for Γ^I .

In the following we want to extend this notion of answer set to pieces of information. We call *piece of information* over $\mathcal{L}_{\mathcal{N}}$ an expression of the kind $\langle \eta \rangle K$ with $K \in \mathcal{L}_{\mathcal{N}}$ a formula and $\eta \in \text{IT}_{\mathcal{N}}(K)$. Given a piece of information $\langle \eta \rangle K$, the following defines the sets of answers $\operatorname{ans}(\langle \eta \rangle K)$ obtainable from it:

$$\begin{split} & \operatorname{ans}(\langle\operatorname{tt}\rangle K) = \{K\}, \text{ with } K \text{ a simple formula} \\ & \operatorname{ans}(\langle(\alpha,\beta)\rangle A_1 \sqcap A_2(c)) = \operatorname{ans}(\langle\alpha\rangle A_1(c)) \cup \operatorname{ans}(\langle\beta\rangle A_2(c)) \\ & \operatorname{ans}(\langle(d,\alpha)\rangle \exists R.A(c)) = \{R(c,d)\} \cup \operatorname{ans}(\langle\alpha\rangle A(d)) \\ & \operatorname{ans}(\langle\phi\rangle \forall_G A) = \bigcup_{d \in \operatorname{DOM}(G)} \operatorname{ans}(\langle\phi(d)\rangle A(d)) \end{split}$$

We remark that $\operatorname{ans}(\langle \eta \rangle K)$ is a finite set of simple formulas. Note that negated formulas of the kind $\neg C(d)$ are considered as simple formulas and their only possible information term is tt: intuitively, this corresponds to considering these formulas as constraints, i.e. we only require that C(d) does not hold, without any constructive information about non-validity.

Example 5. Let us consider the information terms $\psi_1 \in IT_{\mathcal{N}}(Ax_1)$ and $\psi_2 \in IT_{\mathcal{N}}(Ax_2)$ defined in Example 2; we have:

```
\begin{array}{ll} H_1 &= \exists goes \textit{With.Color} & H_2 &= \exists is \textit{ColorOf.H}_3 \\ H_3 &= \textit{Wine} \sqcap \neg (\exists \textit{hasRegion.ExcRegion}) \\ &= \operatorname{ans}(\langle \psi_1 \rangle Ax_1) = \operatorname{ans}(\langle \psi_1(\textit{fish}) \rangle H_1(\textit{fish})) \ \cup \ \operatorname{ans}(\langle \psi_1(\textit{meat}) \rangle H_1(\textit{meat})) \\ &= \operatorname{ans}(\langle \operatorname{tt} \rangle \textit{Color}(\textit{white})) \ \cup \ \{\textit{goes With}(\textit{fish, white}) \} \ \cup \\ &= \operatorname{ans}(\langle \operatorname{tt} \rangle \textit{Color}(\textit{red})) \ \cup \ \{\textit{goes With}(\textit{meat, red}) \} \\ &= \{\textit{Color}(\textit{white}), \textit{goes With}(\textit{fish, white}), \\ &\quad \textit{Color}(\textit{red}), \textit{goes With}(\textit{meat, red}) \} \\ &= \operatorname{ans}(\langle \psi_2 \rangle Ax_2) = \operatorname{ans}(\langle \psi_2(\textit{red}) \rangle H_2(\textit{red})) \ \cup \ \operatorname{ans}(\langle \psi_2(\textit{white}) \rangle H_2(\textit{white})) \\ &= \{\textit{isColorOf}(\textit{red, barolo}) \} \ \cup \ \operatorname{ans}(\langle (\operatorname{tt, tt}) \rangle H_3(\textit{barolo})) \ \cup \\ &\quad \{\textit{isColorOf}(\textit{white, chardonnay}) \} \ \cup \ \operatorname{ans}(\langle (\operatorname{tt, tt}) \rangle H_3(\textit{chardonnay})) \} \\ &= \{\textit{isColorOf}(\textit{red, barolo}), \textit{Wine}(\textit{barolo}), \\ &\quad \neg (\exists \textit{hasRegion.ExcRegion})(\textit{barolo}), \\ &\quad \neg (\exists \textit{hasRegion.ExcRegion})(\textit{chardonnay}) \} \end{array}
```

We point out that $ans(\langle \eta \rangle K)$ in some sense unfolds the constructive meaning of $\langle \eta \rangle K$. Actually, by induction on the structure of K, we can prove that:

 \Diamond

Theorem 1. Let \mathcal{N} be a finite subset of NI, K a formula of $\mathcal{L}_{\mathcal{N}}$. For every model \mathcal{M} , $\mathcal{M} \triangleright \langle \eta \rangle K$ iff $\mathcal{M} \models \operatorname{ans}(\langle \eta \rangle K)$.

Accordingly, the problem of determining the realizability of a formula (w.r.t. an information term) can be reduced to the classical satisfiability of a finite set of atomic formulas. By Theorem 1, and taking into account that lp-interpretations are reachable models, we can strengthen Proposition 1 as follows:

Theorem 2. Let \mathcal{N} be a finite subset of NI, K a formula of $\mathcal{L}_{\mathcal{N}}$. For every lp-interpretation I, $I \models K$ iff there exists an information term $\eta \in IT_{\mathcal{N}}(K)$ such that $I \models ans(\langle \eta \rangle K)$.

Now we can study the relations between answer sets for $\mathcal{EL}c_{\perp}$ formulas and pieces of information.

Definition 2. Let K be a formula of $\mathcal{L}_{\mathcal{N}}$ and $K \in \text{IT}_{\mathcal{N}}(K)$. An lp-interpretation I is a minimal model of $\langle \eta \rangle K$ iff:

```
-I \models \operatorname{ans}(\langle \eta \rangle K) and,

- for every lp-interpretation I' \subseteq I, I' \models \operatorname{ans}(\langle \eta \rangle K) implies I' = I.
```

Note that, by Theorem 1, this definition implies that, for every model \mathcal{M} for $\mathcal{L}_{\mathcal{N}}$ such that $\mathcal{M} \models I$, it holds that $\mathcal{M} \triangleright \langle \eta \rangle K$.

Lemma 1. Let \mathcal{N} be a finite subset of NI, K a formula of $\mathcal{L}_{\mathcal{N}}$ and I an lp-interpretation. Then:

In one direction we can show:

Theorem 3. If I is an answer set for a formula $K \in \mathcal{L}_{\mathcal{N}}$, then there exists $\eta \in \text{IT}_{\mathcal{N}}(K)$ such that I is a minimal model of $\langle \eta \rangle K$.

In the other direction, we need to define a notion of minimality on pieces of information and introduce default reasoning as follows.

Definition 3. Let K be a positive formula of $\mathcal{L}_{\mathcal{N}}$. A piece of information $\langle \eta \rangle K$ is minimal iff there is no $\eta' \in \text{IT}_{\mathcal{N}}(K)$ such that $\text{ans}(\langle \eta' \rangle K) \subset \text{ans}(\langle \eta \rangle K)$.

Similarly to [2], in the case of (sets of) positive formulas the following property can be shown:

Theorem 4. Let K be a positive formula of $\mathcal{L}_{\mathcal{N}}$ and $\langle \eta \rangle K$ be a minimal piece of information for K. Then, $\operatorname{ans}(\langle \eta \rangle K)$ is an answer set for K.

In the general case (i.e. when negative information can occur in formulas), we need to generalize this result by including a notion of *negative* answer set: these sets represent constraints that positive answer sets need to meet, in order to be considered as valid answers for the piece of information given as input. Positive and negative answers of a piece of information $\langle \eta \rangle K$ are defined as follows:

$$\mathtt{ans}^+(\langle \eta \rangle K) = \{ \, H \in \mathtt{ans}(\langle \eta \rangle K) \mid H = A(c), \, \text{with } A \in \mathtt{NC} \cup \mathtt{NG}, \, \text{or } H = R(c,d) \, \}$$

$$\mathtt{ans}^-(\langle \eta \rangle K) = \{ \, H \in \mathtt{ans}(\langle \eta \rangle K) \mid H = \neg C(d) \, \text{ or } H = \bot(d) \, \}$$

Note that $ans^+(\langle \eta \rangle K)$ is an lp-interpretation.

Example 6. The set ans($\langle \psi_2 \rangle Ax_2$) in Example 5 can be partitioned into positive answers and constraints as follows:

```
\begin{aligned} & \operatorname{ans}^+(\langle \psi_2 \rangle Ax_2) = \{ \operatorname{isColorOf}(\operatorname{red},\operatorname{barolo}), \operatorname{Wine}(\operatorname{barolo}), \\ & \operatorname{isColorOf}(\operatorname{white},\operatorname{chardonnay}), \operatorname{Wine}(\operatorname{chardonnay}) \} \\ & \operatorname{ans}^-(\langle \psi_2 \rangle Ax_2) = \{ \neg (\exists \operatorname{hasRegion}.\operatorname{ExcRegion})(\operatorname{barolo}), \\ & \neg (\exists \operatorname{hasRegion}.\operatorname{ExcRegion})(\operatorname{chardonnay}) \} \end{aligned}
```

 \Diamond

We introduce the notion of answer set and minimality concerning pieces of information.

Definition 4. Let K be a formula of $\mathcal{L}_{\mathcal{N}}$ and $\langle \eta \rangle K$ a piece of information. Then, ans⁺($\langle \eta \rangle K$) is an answer set for $\langle \eta \rangle K$ iff ans⁺($\langle \eta \rangle K$) \models ans⁻($\langle \eta \rangle K$).

Definition 5. Let K be a formula of $\mathcal{L}_{\mathcal{N}}$. A piece of information $\langle \eta \rangle K$ is minimal iff, for every $\eta' \in \text{IT}_{\mathcal{N}}(K)$ such that $\text{ans}^+(\langle \eta' \rangle K) \subset \text{ans}^+(\langle \eta \rangle K)$, it holds that $\text{ans}^+(\langle \eta \rangle K) \not\models \text{ans}^-(\langle \eta' \rangle K)$.

Using these definitions, we can generalize Theorem 4 as follows:

Theorem 5. Let K be a formula of $\mathcal{L}_{\mathcal{N}}$ and $\langle \eta \rangle K$ be a minimal piece of information for K with answer set ans⁺($\langle \eta \rangle K$). Then, ans⁺($\langle \eta \rangle K$) is an answer set for K.

To complete the picture, we state the following characterization of answer sets:

Theorem 6. Let K be a formula of $\mathcal{L}_{\mathcal{N}}$. I is an answer set for K iff there exists a minimal piece of information $\langle \eta \rangle K$ such that $I = \mathtt{ans}^+(\langle \eta \rangle K)$.

4 Discussion: ASP based IT generation for $\mathcal{EL}c_{\perp}$

As discussed in [2], and as a consequence of the results of previous section, a way to solve the reasoning task of generating information terms of an input $\mathcal{EL}c_{\perp}$ knowledge base consists in computing its answer sets and then, by exploiting the recursive definition of $\mathtt{ans}(\langle \eta \rangle K)$, reconstruct the corresponding (minimal) information term η .

In [2] we have presented alternative datalog encodings for an input \mathcal{EL} knowledge base in order to generate its answer sets and build minimal information terms from them. In the case of negative information of $\mathcal{EL}c_{\perp}$, we need to adapt the datalog encoding in order to consider the constraints provided by the negative sub-formulas, so that the candidate answer sets not matching the constraints are discarded. A possible way to encode negative constraints, reflecting the definition of reduct K^I provided in previous sections, is to use default negation under the answer set semantics [11] and check that the computed answer sets do not contradict the constraints. For example (using the rules of the translation P_2 in [2]), information terms of a formula $\neg D(a)$ can be computed by the rules:

$$is_it(\mathsf{tt}, a, l_{\neg D}) \leftarrow nom(a), \mathsf{not}\ check(a, l_D).$$

 $check(a, l_D) \leftarrow is_it(x, a, l_D).$

Another option is to encode negative information as constraints rules (i.e. rules of the form $\leftarrow b_1, \ldots, b_n$.) in the rules generating the candidate answer sets, so that interpretations that verify the body of such rules are excluded from the computed answer sets.

We leave as future work the formal definition of a suitable datalog encoding for $\mathcal{EL}c_{\perp}$ for the ASP based generation of information terms and the proof of its correctness with respect to the formal results shown in this paper. As noted in [2], an issue that deserves to be investigated is how the intrinsic complexity of IT generation can be related to the computation of answer sets (and whether complexity can be limited in practical scenarios).

5 Conclusions

In this paper we present the relation between answer set semantics and information term semantics in the context of the description logic \mathcal{EL}_{\perp} . Following [2], we first provide the definition of the information term semantics for \mathcal{EL}_{\perp} . Then, we show how to extend the results presented in [2] concerning the correspondence between answer set semantics and information term semantics to \mathcal{EL}_{\perp} . By introducing a notion of reduction, similar to the one used in Answer Set Programming, it is possible to use negative information as constraints that answer sets must verify; this leads to the notion of minimal pieces of information of a KB. Finally, we briefly discuss how these results can be used to generate the information terms of an input KB over $\mathcal{EL}_{\mathsf{C}_{\perp}}$.

We note that the constructive reading of formulas provided by information term semantics can be related to the recent interest in Explainable AI (which is being discussed also in the field of symbolic Knowledge Representation and Reasoning). For example, as shown in [7], the generation of a valid "snapshot" of a knowledge base (i.e. a valid information term for its set of formulas) can be used to verify the set of constraints encoded by the KB and, in case of a violation, to constructively identify the source of inconsistencies, in order to amend the KB.

The work presented in this paper represents a first step towards the extension of this study to more expressive description logics: for example, an interesting goal is to extend the discussed results to the full language of \mathcal{ALC} , exploiting the information term semantics presented in [9]. Another issue to be investigated is the application of the presented formal results to representation and reasoning tasks, for example by extending the datalog encoding and prototype discussed in [2] for ASP based generation of information terms. We also aim at developing procedures for the manipulation of information terms (see, e.g., [4]), in order to apply IT semantics in concrete problems.

References

- 1. F. Baader. Terminological cycles in a description logic with existential restrictions. In *IJCAI-03*, pages 325–330. Morgan Kaufmann, 2003.
- L. Bozzato. ASP based generation of information terms for constructive EL. Fundam. Inform., 161(1-2):29–51, 2018.
- L. Bozzato, T. Eiter, and L. Serafini. Reasoning with Justifiable Exceptions in ££⊥ Contextualized Knowledge Repositories. In Description Logic, Theory Com-bination, and All That, volume 11560 of LNCS, pages 110–134. Springer, 2019.
- L. Bozzato and M. Ferrari. Composition of semantic web services in a constructive description logic. In RR2010, volume 6333 of LNCS, pages 223–226. Springer, 2010.
- L. Bozzato, M. Ferrari, C. Fiorentini, and G. Fiorino. A constructive semantics for ALC. In DL2007, volume 250 of CEUR-WP, pages 219–226. CEUR-WS.org, 2007.
- L. Bozzato, M. Ferrari, C. Fiorentini, and G. Fiorino. A decidable constructive description logic. In *JELIA 2010*, volume 6341 of *LNCS*, pages 51–63. Springer, 2010

- L. Bozzato, M. Ferrari, and P. Villa. Actions over a constructive semantics for description logics. Fundam. Inform., 96(3):253–269, 2009.
- 8. R.J. Brachman, D.L. McGuinness, P.F. Patel-Schneider, L.A. Resnick, and A. Borgida. Living with CLASSIC: When and how to use a KL-ONE-like language. In *Principles of Semantic Networks*, pages 401–456. Morgan Kaufmann, 1991.
- 9. M. Ferrari, C. Fiorentini, and G. Fiorino. \mathcal{BCDL} : basic constructive description logic. J. of Automated Reasoning, 44(4):371–399, 2010.
- C. Fiorentini and M. Ornaghi. Answer set semantics vs. information term semantics. In ASP2007: Answer Set Programming, Advances in Theory and Implementation, 2007.
- M. Gelfond and V. Lifschitz. Classical Negation in Logic Programs and Disjunctive Databases. New Generation Computing, 9:365–385, 1991.
- 12. E. Hermann Haeusler, V. de Paiva, and A. Rademaker. Intuitionistic description logic and legal reasoning. In *DEXA 2011 Workshops*, pages 345–349. IEEE Computer Society, 2011.
- M. Hilia, A. Chibani, K. Djouani, and Y. Amirat. Semantic service composition framework for multidomain ubiquitous computing applications. In *ICSOC* 2012, volume 7636 of *LNCS*, pages 450–467. Springer, 2012.
- 14. V. Lifschitz, L. R. Tang, and H. Turner. Nested expressions in logic programs. *Ann. Math. Artif. Intell.*, 25(3-4):369–389, 1999.
- 15. M. Mendler and S. Scheele. Towards a type system for semantic streams. In *SR2009 Stream Reasoning Workshop (ESWC 2009)*, volume 466 of *CEUR-WP*. CEUR-WS.org, 2009.
- M. Mendler and S. Scheele. Towards Constructive DL for Abstraction and Refinement. J. Autom. Reasoning, 44(3):207–243, 2010.
- P. Miglioli, U. Moscato, M. Ornaghi, and G. Usberti. A constructivism based on classical truth. Notre Dame Journal of Formal Logic, 30(1):67–90, 1989.
- 18. A. S. Troelstra. From constructivism to computer science. *Theoretical Computer Science*, 211(1-2):233–252, 1999.

A Appendix: Proofs

Lemma 1. Let \mathcal{N} be a finite subset of NI, K a formula of $\mathcal{L}_{\mathcal{N}}$ and I an lp-interpretation. Then:

Proof.

- (i). We first note that, by the definition of I^+ , if $f_D(d) = e$ in I^+ , then $f_D(d)$ behaves like e in I^+ , namely:
- (‡) for every concept C, $I \models C(e)$ iff $I^+ \models C(f_D(d))$.

We can show the claim by induction of the form of the formula K.

- Let K be atomic. If K = A(c) or R(c,d), then $\Phi(K) = K$ and (i) trivially holds. If K = T(c), then $\Phi(K) = \mathbf{t}$ and $K = \bot(c)$, then $\Phi(K) = \mathbf{f}$; in both cases (i) holds.
- Let $K = \neg C(d)$. Note that $\Phi(\neg C(d)) = \neg \Phi(C(d))$. If $I \models \neg C(d)$, then by definition $I \not\models C(d)$. By induction hypothesis, $I^+ \not\models \Phi(C(d))$, which implies $I^+ \models \neg \Phi(C(d))$, namely $I^+ \models \Phi(\neg C(d))$. Similarly, if $I^+ \models \Phi(\neg C(d))$, we conclude $I \models \neg C(d)$.
- Let $K = C_1 \sqcap C_2(d)$. Note that $\Phi(C_1 \sqcap C_2(d)) = \Phi(C_1(d)) \land \Phi(C_2(d))$. If $I \models K$, then $I \models C_1(d)$ and $I \models C_2(d)$. By induction hypothesis, $I^+ \models \Phi(C_1(d))$ and $I^+ \models \Phi(C_2(d))$. Conversely, if $I^+ \models \Phi(K)$, we get $I \models K$.
- Let $K = \exists R.C(d)$. Note that $\Phi(K) = R(d, f_{\exists R.C}(d)) \land \gamma_C(f_{\exists R.C}(d))$. Let us assume $I \models \exists R.C(d)$. By hypothesis, there exists $e \in \mathcal{N}$ such that $R(d, e) \in I$ and $I \models C(e)$. By induction hypothesis and the definition of the extension $I^+, I^+ \models R(d, e)$ and $I^+ \models \Phi(C(e))$. By (\ddagger) , we get $I^+ \models R(d, f_{\exists R.C}(d))$ and $I^+ \models \Phi(C(f_{\exists R.C}(d)))$, namely $I^+ \models \gamma_C(f_{\exists R.C}(d))$; we conclude $I^+ \models \Phi(K)$. Conversely, let us assume $\mathcal{I}^+ \models \Phi(K)$. Then, $I^+ \models R(d, f_{\exists R.C}(d))$ and $I^+ \models \gamma_C(f_{\exists R.C}(d))$. By the definition of I^+, I contains a formula of the kind R(d, e) and, by (\ddagger) , we get $I^+ \models \gamma_C(e)$. By induction hypothesis we get $I \models C(e)$, thus $I \models \exists R.C(d)$.
- Let $K = \forall_G C$. We have $\Phi(K) = \Phi(C(c_1)) \wedge \cdots \wedge \Phi(C(c_n))$, thus we can proceed as in the case concerning \square .
- (ii). One can easily prove that, for every formula F of $\mathcal{L}^1_{\mathcal{N}}$, $I \models F$ iff $I \models F^I$. Thus, $I \models K$ iff $I^+ \models \Phi(K)$ (by point (i)), iff $I^+ \models (\Phi(K))^{I^+}$ (by the previous remark), iff $I^+ \models K^I$ (by definition of K^I).
- (iii). Considering the definition of ans and reduction K^{I} , we show the claim by induction on the structure of K (and definition of ans):
- Let K be atomic. Then we have that $K^I = K$ and $(ans(\langle tt \rangle K))^I = \{K\}$ thus the claim immediately follows.

- Let $K = \neg C(d)$. Since K is a simple formula, we have that $(\operatorname{ans}(\langle \operatorname{tt} \rangle K))^I = \{K\}^I$. Thus, it immediately follows that $I^+ \models K^I$ iff $I^+ \models (\operatorname{ans}(\langle \operatorname{tt} \rangle K))^I$.
- Let $K = C_1 \sqcap C_2(d)$. Then we have $K^I = (\gamma_{C_1}(d))^I \wedge (\gamma_{C_2}(d))^I$ and $(\operatorname{ans}(\langle (\alpha,\beta)\rangle K))^I = \operatorname{ans}(\langle \alpha\rangle C_1(d))^I \cup \operatorname{ans}(\langle \beta\rangle C_2(d))^I$. By induction hypothesis, $I^+ \models (\gamma_{C_1}(d))^I$ iff $I^+ \models \operatorname{ans}(\langle \alpha\rangle C_1(d))^I$ and $I^+ \models (\gamma_{C_2}(d))^I$ iff $I^+ \models \operatorname{ans}(\langle \alpha\rangle C_2(d))^I$. Thus $I^+ \models K^I$ iff $I^+ \models (\operatorname{ans}(\langle (\alpha,\beta)\rangle K))^I$.

 Let $K = \exists R.C(d)$. Then $K^I = R(d,f_{\exists R.C}(d)) \wedge (\gamma_C(f_{\exists R.C}(d)))^I$ and
- Let $K = \exists R.C(d)$. Then $K^I = R(d, f_{\exists R.C}(d)) \land (\gamma_C(f_{\exists R.C}(d)))^I$ and $(\operatorname{ans}(\langle (e, \alpha) \rangle K))^I = \{R(d, e)\} \cup (\operatorname{ans}(\langle \alpha \rangle A(e)))^I$. Let us suppose that $I^+ \models K^I$: then, we have $I^+ \models R(d, f_{\exists R.C}(d))$ and $I^+ \models (\gamma_C(f_{\exists R.C}(d)))^I$. Thus, by (\ddagger) , $I^+ \models R(d, e)$ and $I^+ \models (\gamma_C(e))^I$. By induction hypothesis we have that $I^+ \models \operatorname{ans}(\langle \alpha \rangle C(e))^I$. Thus, we have that $I^+ \models (\operatorname{ans}(\langle (e, \alpha) \rangle K))^I$. Conversely, suppose that $I^+ \models (\operatorname{ans}(\langle (e, \alpha) \rangle K))^I$. Then we have $I^+ \models R(d, e)$ and $I^+ \models \operatorname{ans}(\langle \alpha \rangle C(e))^I$. By definition of I^+ and induction hypothesis, we obtain that $I^+ \models K^I$.
- Induction hypothesis, we obtain that I = I and I =

Theorem 3. If I is an answer set for a formula $K \in \mathcal{L}_N$, then there exists $\eta \in \text{IT}_N(K)$ such that I is a minimal model of $\langle \eta \rangle K$.

Proof. Since I is an answer set for K, we have that $I^+ \models K^I$ and thus, by Lemma 1, $I \models K$. By Theorem 2, then there exists $\eta \in \text{IT}_{\mathcal{N}}(K)$ s.t. $I \models \text{ans}(\langle \eta \rangle K)$.

We can prove that I is minimal: suppose that $J \subseteq I$ with $J \models \operatorname{ans}(\langle \eta \rangle K)$. Then, by Theorem 2, $J \models K$. Let us show that $J^+ \models (\operatorname{ans}(\langle \eta \rangle K))^I$. Consider $H \in \operatorname{ans}(\langle \eta \rangle K)$. If H is atomic (and H is not $\bot(d)$), then $H^I = H$ and thus it holds $J^+ \models H^I$. Otherwise, if $H = \neg C(d)$, since $I^+ \models H^I$ then we have $H^I = \mathbf{t}$ and thus it holds $J^+ \models H^I$. This proves that $J^+ \models (\operatorname{ans}(\langle \eta \rangle K))^I$: thus, by Lemma 1, we obtain that $J^+ \models K^I$. Since I is an answer set for K, then by definition we obtain J = I.

Theorem 4. Let K be a positive formula of $\mathcal{L}_{\mathcal{N}}$ and $\langle \eta \rangle K$ be a minimal piece of information for K. Then, $\operatorname{ans}(\langle \eta \rangle K)$ is an answer set for K.

Proof. By Theorem 2, for every lp-interpretation I s.t. $I \models \mathtt{ans}(\langle \eta \rangle K)$ we have $I \models K$. Hence, considering $I = \mathtt{ans}(\langle \eta \rangle K)$, also $\mathtt{ans}(\langle \eta \rangle K) \models K$.

Moreover, let us consider $I' \subseteq \mathtt{ans}(\langle \eta \rangle K)$ with $I' \models K$. Then, by Theorem 2, there exists a $\beta \in \mathsf{IT}_{\mathcal{N}}(K)$ s.t. $I' \models \mathtt{ans}(\langle \beta \rangle K)$. Thus, $\mathtt{ans}(\langle \beta \rangle K) \subseteq I'$ and, for the minimality of $\langle \eta \rangle K$, this implies that $\mathtt{ans}(\langle \beta \rangle K) = I' = \mathtt{ans}(\langle \eta \rangle K)$.

Theorem 5. Let K be a formula of $\mathcal{L}_{\mathcal{N}}$ and $\langle \eta \rangle K$ be a minimal piece of information for K with answer set $\operatorname{ans}^+(\langle \eta \rangle K)$. Then, $\operatorname{ans}^+(\langle \eta \rangle K)$ is an answer set for K.

Proof. Assuming that $I = \mathtt{ans}^+(\langle \eta \rangle K)$ is an answer set for the minimal piece of information $\langle \eta \rangle K$, then by definition $I \models \mathtt{ans}^-(\langle \eta \rangle K)$. Let us show that I is an answer set for K.

By the condition above (and the definition of positive an negative answers), it holds that $I \models \operatorname{ans}(\langle \eta \rangle K)$. Thus, by Theorem 2 we have $I \models K$ and by Lemma 1, it holds that $I^+ \models K^I$.

To show that I is minimal, let us consider $J \subset I$ s.t. $J^+ \models K^I$. Then, by Theorem 2 and Lemma 1, there exists $\beta \in \text{IT}_{\mathcal{N}}(K)$ s.t. $J^+ \models (\text{ans}(\langle \beta \rangle K))^I$. This implies that:

```
(a). J^+ \models (\operatorname{ans}^+(\langle \beta \rangle K))^I
(b). J^+ \models (\operatorname{ans}^-(\langle \beta \rangle K))^I
```

Considering (a), we have that $\mathtt{ans}^+(\langle \beta \rangle K)^I = \mathtt{ans}^+(\langle \beta \rangle K)$ (i.e. the reduction does not change the positive contents of the set). Thus, $\mathtt{ans}^+(\langle \beta \rangle K) \subseteq J \subseteq I$.

Moreover, considering (b) (and the fact that J is consistent), the reduct $(ans^-(\langle \beta \rangle K))^I = \{t\}$. This implies that $I \models ans^-(\langle \beta \rangle K)$: this is an absurd, as it would contradict the minimality condition of $\langle \eta \rangle K$.

Theorem 6. Let K be a formula of $\mathcal{L}_{\mathcal{N}}$. I is an answer set for K iff there exists a minimal piece of information $\langle \eta \rangle K$ such that $I = \mathtt{ans}^+(\langle \eta \rangle K)$.

Proof. By Theorem 4 we directly have the "if" direction: if $\langle \eta \rangle K$ is a minimal piece of information, $I = \mathtt{ans}^+(\langle \eta \rangle K)$ is an answer set for K.

Let us now prove the other direction and consider an answer set I for K: by definition, then $I^+ \models K^I$. By Lemma 1, we have that there exists $\eta \in \text{IT}_{\mathcal{N}}(K)$ such that:

(a).
$$I^+ \models (\operatorname{ans}(\langle \eta \rangle K))^I$$
.

Moreover, given that I is an answer set:

(b). for every $J \subset I$, $J^+ \not\models K^I$, that is for any $\beta \in IT_{\mathcal{N}}(K)$, $J^+ \not\models (\mathtt{ans}(\langle \beta \rangle K))^I$.

From (a) and the definition of the reduct, we have that $(\mathsf{ans}^-(\langle \eta \rangle K))^I = \{\mathsf{t}\}$, and so $I \models \mathsf{ans}^-(\langle \eta \rangle K)$.

From (b), for every $J \subset I$, $J^+ \not\models (\mathtt{ans}^+(\langle \eta \rangle K))^I$, thus we have that $I = \mathtt{ans}^+(\langle \eta \rangle K)$.

We need to show that $\langle \eta \rangle K$ is minimal. It holds that I is an answer set for $\langle \eta \rangle K$ since $I \models \mathtt{ans}^-(\langle \eta \rangle K)$. Let us consider a piece of information $\langle \beta \rangle K$ s.t. $\mathtt{ans}^+(\langle \beta \rangle K) \subset I$. We have by (b) that, if $J = \mathtt{ans}^+(\langle \beta \rangle K)$, $J^+ \not\models (\mathtt{ans}(\langle \beta \rangle K))^I$: thus, in particular $J^+ \not\models (\mathtt{ans}^-(\langle \beta \rangle K))^I$. This means that $\mathbf{f} \in (\mathtt{ans}^-(\langle \beta \rangle K))^I$ and thus $I \not\models \mathtt{ans}^-(\langle \beta \rangle K)$. By definition of minimality, this implies that $\langle \eta \rangle K$ is minimal.