An Extended Description Logic for Inconsistency-tolerant Ontological Reasoning with Sequential Information

Norihiro Kamide
Department of Information and Electronic Engineering, Faculty of Science and Engineering, Teikyo University, Toyosatodai 1-1, Utsunomiya, Tochigi, Japan

Keywords: Description Logic, Inconsistency-tolerant Reasoning, Sequential Information, Embedding Theorem, Decidability.

Abstract: Description logics are a family of logic-based knowledge representation formalisms. Inconsistency-tolerant description logics, which are extensions of standard description logics, have been studied to cope with inconsistencies that frequently occur in an open world. In this study, an extended inconsistency-tolerant description logic with a sequence modal operator is introduced. The logic proposed is intended to appropriately handle inconsistency-tolerant ontological reasoning with sequential information (i.e., information expressed as sequences, such as time, action, and event sequences). A theorem for embedding the proposed logic into a fragment of the logic is proved. The logic is shown to be decidable by using the proposed embedding theorem. These results demonstrate that using the embedding theorem enables the reuse of previously developed methods and algorithms for the standard description logic for the effective handling of inconsistent ontologies with sequential information described by the proposed logic.

1 INTRODUCTION

In this study, we introduce an extended inconsistency-tolerant description logic with a sequence modal operator that we have named *sequential inconsistency-tolerant description logic* ($\mathcal{ALCPS}$). This new logic $\mathcal{ALCPS}$ is intended to appropriately handle inconsistency-tolerant ontological reasoning with sequential information (i.e., information expressed as sequences, such as time, data, action, event, and agent-communication sequences). We then prove several theorems for embedding $\mathcal{ALCPS}$ into some fragments of $\mathcal{ALCPS}$. Using one of these embedding theorems, we show the decidability of $\mathcal{ALCPS}$.

The aim of this study is to combine and integrate an inconsistency-tolerant description logic and a sequential description logic. Therefore, we begin with a brief introduction to description logics, inconsistency-tolerant description logics, and sequential description logics. *Description logics* (Baader et al., 2003) are a family of logic-based knowledge representation formalisms that were adopted as the logical foundation of the W3C web ontology language (OWL). Many of useful description logics including the standard description logic $\mathcal{ALC}$ introduced by Schmidt-Schauss and Smolka in (Schmidt-Schauss and Smolka, 1991) have been extensively studied. *Inconsistency-tolerant description logics* (also referred to as *paraconsistent description logics*) (Ma et al., 2007; Ma et al., 2008; Meghini and Straccia, 1996; Meghini et al., 1998; Odintsov and Wansing, 2003; Odintsov and Wansing, 2008; Patel-Schneider, 1989; Straccia, 1997; Zhang and Lin, 2008; Zhang et al., 2009; Kamide, 2012; Kamide, 2013) are typical examples of such useful description logics. Inconsistency-tolerant description logics have been studied to cope with inconsistencies that frequently occur in an open world. For a brief survey of inconsistency-tolerant description logics, see the last section of this paper. *Sequential description logics*, that were obtained from $\mathcal{ALC}$ by adding a sequence modal operator, were introduced and studied by Kamide in (Kamide, 2010; Kamide, 2011), where he presented several embedding, decidability, and Craig interpolation theorems for these logics. The sequence modal operator is useful for representing sequential information (i.e., information expressed as sequences) and has also been used to obtain expressive and useful non-classical logics in several fields of computer science. For more information on such extended non-classical logics with a sequence modal operator, see the last section of this paper.

Kamide, N.
An Extended Description Logic for Inconsistency-tolerant Ontological Reasoning with Sequential Information.
DOI: 10.5220/0008876403130321
In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020) - Volume 2*, pages 313-321
ISBN: 978-989-758-395-7; ISSN: 2184-433X

The sequence modal operator $[b]$ used in $\mathcal{ALCPS}$, where b is a sequence, is useful for representing sequential information that is expressed as data sequences, action sequences, time sequences, event sequences, agent communication sequences, program-execution sequences, word (character or alphabet) sequences, DNA sequences, etc. This is regarded as plausible because a sequence structure gives a *monoid* $\langle M, ;, \emptyset \rangle$ with the following *informational interpretation* (Wansing, 1993): (1) M is a set of sequences (i.e., a set of pieces of ordered information); (2) ';' is a concatenation operator on M (i.e., a binary operator that combines two pieces of information); (3) $\emptyset$ is the empty sequence (i.e., an empty piece of information). By the informational interpretation, the intuitive meanings of the sequence modal operator can be obtained as follows: A concept of the form $[b_1 ; b_2 ; \cdots ; b_n]C$ intuitively means that "C is true based on a sequence $b_1 ; b_2 ; \cdots ; b_n$ of ordered pieces of information." Moreover, a concept of the form $[\emptyset]C$, which coincides with C, intuitively means that "C is true without any information (i.e., it is an eternal truth in the sense of classical description logic)." We remark that $[b]$ is regarded as a generalization of the temporal next-time operator X of the linear-time temporal logic LTL and the modal operator $\Box$ of the normal modal logic K. Actually, if we consider $[b]$ based on classical logic, then $[b]$ is expressive than X and $\Box$.

In this study, we develop a sequential inconsistency-tolerant description logic, $\mathcal{ALCPS}$, which is a natural combination of sequential description logic and inconsistency-tolerant description logic. To develop $\mathcal{ALCPS}$, we overcame a technical problem with the semantic interpretation for combining sequential and inconsistency-tolerant description logics; namely, some existing inconsistency-tolerant and sequential description logics have complex multiple polarities or sequence-indexed interpretation semantics. The presence of these complex interpretation semantics makes it difficult to combine these two logics. This is one reason why such a combined logic has not yet been developed. To overcome this problem, we introduce a simple single interpretation semantics that is compatible with the standard single interpretation semantics of $\mathcal{ALC}$. Using this simple interpretation semantics, we can construct $\mathcal{ALCPS}$ with the following technical merits: We can prove a theorem for embedding $\mathcal{ALCPS}$ into the $[b]$-less fragment of $\mathcal{ALCPS}$ and can simply formalize and handle the operator $[b]$.

The structure of this paper is as follows: In Section 2, we develop a basic inconsistency-tolerant description logic, $\mathcal{ALCP}$, by adding a paraconsistent negation connective $\sim$ to $\mathcal{ALC}$. This logic $\mathcal{ALCP}$ is roughly equivalent to the logic $\mathcal{SALC}$ introduced by Kamide in (Kamide, 2013), and is logically equivalent to the logic $\mathcal{PALC}$ introduced by Kamide in (Kamide, 2012). The logic $\mathcal{ALCP}$ has a simple single interpretation semantics and is shown to be embeddable into $\mathcal{ALC}$ by using the method presented in (Kamide, 2012). Using this embedding theorem, $\mathcal{ALCP}$ is also shown to be decidable. In Section 3, we develop the sequential inconsistency-tolerant description logic $\mathcal{ALCPS}$ by extending $\mathcal{ALCP}$ with the sequence modal operator $[b]$. This new logic $\mathcal{ALCPS}$ also has a simple single interpretation semantics. A translation function from $\mathcal{ALCPS}$ into $\mathcal{ALCP}$ is then defined, and a theorem for embedding $\mathcal{ALCPS}$ into $\mathcal{ALCP}$ is proved. Using this embedding theorem, we show that $\mathcal{ALCPS}$ is decidable. We also prove a theorem for embedding $\mathcal{ALCPS}$ into $\mathcal{ALC}$. In Section 4, we present our conclusions and discuss related work.

2 BASIC INCONSISTENCY-TOLERANT DESCRIPTION LOGIC

First, we introduce the inconsistency-tolerant description logic $\mathcal{ALCP}$. The $\mathcal{ALCP}$-concepts are constructed from atomic concepts, roles, $\neg$ (classical negation or complement), $\sim$ (paraconsistent negation), $\sqcap$ (intersection), $\sqcup$ (union), $\forall R$ (universal concept quantification) and $\exists R$ (existential concept quantification). We use the letter A for atomic concepts, the letter R for roles, and the letters C and D for concepts. We use an expression $C \equiv D$ to denote the syntactical equivalence between C and D. We use the symbol N_C to denote a set of atomic concepts, the symbol N'_C to denote the set $\{A' \mid A \in N_C\}$ of atomic concepts, the symbol $N^{\sim}_C$ to denote the set $\{\sim A \mid A \in N_C\}$ of negated atomic concepts, and the symbol N_R to denote a non-empty set of roles. We remark that the symbol N'_C is not used for defining $\mathcal{ALCP}$-concepts, but used for defining a translation function from the set of $\mathcal{ALCP}$-concepts into the set of $\mathcal{ALC}$-concepts, where it is used for translating the negated atomic $\mathcal{ALCP}$-concepts to the corresponding atomic $\mathcal{ALC}$-concepts in N'_C.

Definition 2.1. *Concepts C of $\mathcal{ALCP}$ are defined by the following grammar, assuming A represents atomic concepts:*

$$C ::= A \mid \neg C \mid \sim C \mid C \sqcap C \mid C \sqcup C \mid \forall R.C \mid \exists R.C$$

Definition 2.2. *A* paraconsistent interpretation *$\mathcal{PI}$ is a structure $\langle \Delta^{\mathcal{PI}}, \cdot^{\mathcal{PI}} \rangle$ such that*

1. $\Delta^{\mathcal{P}I}$ *is a non-empty set,*
2. $\cdot^{\mathcal{P}I}$ *is an interpretation function which assigns to every concept* $B \in N_C \cup N_C^{\sim}$ *a set* $B^{\mathcal{P}I} \subseteq \Delta^{\mathcal{P}I}$ *and to every role* R *a binary relation* $R^{\mathcal{P}I} \subseteq \Delta^{\mathcal{P}I} \times \Delta^{\mathcal{P}I}$.

The interpretation function is inductively extended to concepts by the following conditions:

1. $(\neg C)^{\mathcal{P}I} := \Delta^{\mathcal{P}I} \setminus C^{\mathcal{P}I}$,
2. $(C \sqcap D)^{\mathcal{P}I} := C^{\mathcal{P}I} \cap D^{\mathcal{P}I}$,
3. $(C \sqcup D)^{\mathcal{P}I} := C^{\mathcal{P}I} \cup D^{\mathcal{P}I}$,
4. $(\forall R.C)^{\mathcal{P}I} := \{a \in \Delta^{\mathcal{P}I} \mid \forall b\ [(a,b) \in R^{\mathcal{P}I} \Rightarrow b \in C^{\mathcal{P}I}]\}$,
5. $(\exists R.C)^{\mathcal{P}I} := \{a \in \Delta^{\mathcal{P}I} \mid \exists b\ [(a,b) \in R^{\mathcal{P}I} \wedge b \in C^{\mathcal{P}I}]\}$,
6. $(\sim\sim C)^{\mathcal{P}I} := C^{\mathcal{P}I}$,
7. $(\sim\neg C)^{\mathcal{P}I} := \Delta^{\mathcal{P}I} \setminus (\sim C)^{\mathcal{P}I}$,
8. $(\sim(C \sqcap D))^{\mathcal{P}I} := (\sim C)^{\mathcal{P}I} \cup (\sim D)^{\mathcal{P}I}$,
9. $(\sim(C \sqcup D))^{\mathcal{P}I} := (\sim C)^{\mathcal{P}I} \cap (\sim D)^{\mathcal{P}I}$,
10. $(\sim\forall R.C)^{\mathcal{P}I} := \{a \in \Delta^{\mathcal{P}I} \mid \exists b\ [(a,b) \in R^{\mathcal{P}I} \wedge b \in (\sim C)^{\mathcal{P}I}]\}$,
11. $(\sim\exists R.C)^{\mathcal{P}I} := \{a \in \Delta^{\mathcal{P}I} \mid \forall b\ [(a,b) \in R^{\mathcal{P}I} \Rightarrow b \in (\sim C)^{\mathcal{P}I}]\}$.

An expression $\mathcal{P}I \models C$ *is defined as* $C^{\mathcal{P}I} \neq \emptyset$. *A paraconsistent interpretation* $\mathcal{P}I := \langle \Delta^{\mathcal{P}I}, \cdot^{\mathcal{P}I} \rangle$ *is a model of a concept* C *(denoted as* $\mathcal{P}I \models C$*) if* $\mathcal{P}I \models C$. *A concept* C *is said to be* satisfiable *in* $\mathcal{ALCP}$ *if there exists a paraconsistent interpretation* $\mathcal{P}I$ *such that* $\mathcal{P}I \models C$.

Next, we introduce the logic $\mathcal{ALC}$ (Schmidt-Schauss and Smolka, 1991) as a sublogic of $\mathcal{ALCP}$. The $\mathcal{ALC}$-concepts are constructed from atomic concepts, roles, $\neg$, $\sqcap$, $\sqcup$, $\forall R$, and $\exists R$.

Definition 2.3. Concepts C *of* $\mathcal{ALC}$ *are defined by the following grammar, assuming* A *represents atomic concepts:*

$$C ::= A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall R.C \mid \exists R.C$$

Definition 2.4. *An* interpretation I *is a structure* $\langle \Delta^I, \cdot^I \rangle$ *such that*

1. Δ^I *is a non-empty set,*
2. $\cdot^I$ *is an interpretation function which assigns to every concept* $A \in N_C$ *a set* $A^I \subseteq \Delta^I$ *and to every role* R *a binary relation* $R^I \subseteq \Delta^I \times \Delta^I$.

The interpretation function is extended to concepts by the conditions 1-5 in Definition 2.2 by replacing $\cdot^{\mathcal{P}I}$ *with* $\cdot^I$.

An expression $I \models C$ *is defined as* $C^I \neq \emptyset$. *An interpretation* $I := \langle \Delta^I, \cdot^I \rangle$ *is a model of a concept* C *(denoted as* $I \models C$*) if* $I \models C$. *A concept* C *is said to be* satisfiable *in* $\mathcal{ALC}$ *if there exists an interpretation* I *such that* $I \models C$.

Remark 2.5. *We make the following remarks.*

1. *The logic* $\mathcal{CALC}^C$ *introduced in* (Odintsov and Wansing, 2003) *has the same interpretations for* A *(atomic concept),* $\sim A$ *(negated atomic concept),* $\sqcap$ *and* $\sqcup$ *as in* $\mathcal{ALCP}$. *Since* $\mathcal{CALC}^C$ *is constructive, it has no classical negation, but has constructive inclusion (constructive implication)* $\subseteq^c$.
2. $\mathcal{ALCP}$ *has the following equations with respect to* $\sim$*:*
   (a) $(\sim\sim C)^{\mathcal{P}I} = C^{\mathcal{P}I}$,
   (b) $(\sim\neg C)^{\mathcal{P}I} = (\neg\sim C)^{\mathcal{P}I}$,
   (c) $(\sim(C \sqcap D))^{\mathcal{P}I} = (\sim C \sqcup \sim D)^{\mathcal{P}I}$,
   (d) $(\sim(C \sqcup D))^{\mathcal{P}I} = (\sim C \sqcap \sim D)^{\mathcal{P}I}$,
   (e) $(\sim(\forall R.C))^{\mathcal{P}I} = (\exists R.\sim C)^{\mathcal{P}I}$,
   (f) $(\sim(\exists R.C))^{\mathcal{P}I} = (\forall R.\sim C)^{\mathcal{P}I}$.
3. $\mathcal{ALCP}$ *is regarded as a four-valued logic in the following sense. For each concept* C*, we can take one of the following cases:*
   (a) C *is verified with respect to an element* a *of* $\Delta^{\mathcal{P}I}$ *(i.e.,* $a \in C^{\mathcal{P}I}$*).*
   (b) C *is falsified with respect to an element* a *of* $\Delta^{\mathcal{P}I}$ *(i.e.,* $a \in (\sim C)^{\mathcal{P}I}$*).*
   (c) C *is both verified and falsified.*
   (d) C *is neither verified nor falsified.*
4. *A semantic consequence relation* $\models$ *is called* paraconsistent *with respect to a negation connective* $\sim$ *if there are formulas* α *and* β *such that* $\{\alpha, \sim\alpha\} \not\models \beta$. *In case of* $\mathcal{ALCP}$*, assume a paraconsistent interpretation* $\mathcal{P}I = \langle \Delta^{\mathcal{P}I}, \cdot^{\mathcal{P}I} \rangle$ *such that* $A^{\mathcal{P}I} \subseteq \Delta^{\mathcal{P}I}$, $(\sim A)^{\mathcal{P}I} \subseteq \Delta^{\mathcal{P}I}$, *and* $B^{\mathcal{P}I} \subseteq \Delta^{\mathcal{P}I}$ *for a pair of distinct atomic concepts* A *and* B. *Then,* $(A \sqcap \sim A)^{\mathcal{P}I} \not\subseteq B^{\mathcal{P}I}$*, and hence* $\mathcal{ALCP}$ *is paraconsistent with respect to* $\sim$. *Note that* $\mathcal{ALCP}$ *is not paraconsistent with respect to* $\neg$.
5. $\mathcal{ALCP}$ *and* $\mathcal{ALC}$ *can be extended to deal with an* ABox*, a* TBox*, and a knowledge base by adding a non-empty set of individual names. But, in this paper, we do not deal with these constructors, since we intend to concentrate the discussion on the essential logical reasoning.*

Next, we introduce a translation form $\mathcal{ALCP}$ into $\mathcal{ALC}$, and present a theorem for embedding $\mathcal{ALCP}$ into $\mathcal{ALC}$. By using this embedding theorem, we can obtain the decidability for $\mathcal{ALCP}$.

Definition 2.6. *The language* $\mathcal{L}^p$ *of* $\mathcal{ALCP}$ *is defined using* N_C, N_R, $\sim$, $\neg, \sqcap, \sqcup$, $\forall R$ *and* $\exists R$. *The language* $\mathcal{L}$ *of* $\mathcal{ALC}$ *is obtained from* $\mathcal{L}^p$ *by adding* N_C' *and deleting* $\sim$.

A mapping f *from* $\mathcal{L}^p$ *to* $\mathcal{L}$ *is defined inductively by*

1. *for any* $R \in N_R$ *and any* $f(R) := R$,

2. *for any* $A \in N_C$, $f(A) := A$ *and* $f(\sim A) := A' \in N'_C$,
3. $f(\neg C) := \neg f(C)$,
4. $f(C \sharp D) := f(C) \sharp f(D)$ *where* $\sharp \in \{\sqcap, \sqcup\}$,
5. $f(\sharp R.C) := \sharp f(R).f(C)$ *where* $\sharp \in \{\forall, \exists\}$,
6. $f(\sim\sim C) := f(C)$,
7. $f(\sim\neg C) := \neg f(\sim C)$,
8. $f(\sim(C \sqcap D)) := f(\sim C) \sqcup f(\sim D)$,
9. $f(\sim(C \sqcup D)) := f(\sim C) \sqcap f(\sim D)$,
10. $f(\sim\forall R.C) := \exists f(R).f(\sim C)$,
11. $f(\sim\exists R.C) := \forall f(R).f(\sim C)$.

Theorem 2.7 (Embedding from $\mathcal{ALCP}$ into $\mathcal{ALC}$)**.** *Let f be the mapping defined in Definition 2.6. For any concept C,*

C is satisfiable in $\mathcal{ALCP}$ iff $f(C)$ is satisfiable in $\mathcal{ALC}$.

Proof. Similar to the method presented in (Kamide, 2012) for another inconsistency-tolerant description logic $\mathcal{PALC}$ or $\mathcal{SALC}$. **Q.E.D.**

Theorem 2.8 (Decidability for $\mathcal{ALCP}$)**.** *The concept satisfiability problem for $\mathcal{ALCP}$ is decidable.*

Proof. The concept satisfiability problem for $\mathcal{ALC}$ is well known to be decidable (Baader et al., 2003; Schmidt-Schauss and Smolka, 1991). By this decidability for $\mathcal{ALC}$, for each concept C of $\mathcal{ALCP}$, it is possible to decide if $f(C)$ is satisfiable in $\mathcal{ALC}$. Then, by Theorem 2.7, the satisfiability problem for $\mathcal{ALCP}$ is decidable. **Q.E.D.**

Remark 2.9. *We make the following remarks.*

1. *A similar translation as presented in Definition 2.6 has been used by Gurevich* (Gurevich, 1977)*, Rautenberg* (Rautenberg, 1979)*, and Vorob'ev* (Vorob'ev, 1952) *to embed Nelson's constructive logic* (Almukdad and Nelson, 1984; Nelson, 1949) *into intuitionistic logic.*
2. *The satisfiability problems of a TBox, an ABox, and a knowledge base for $\mathcal{ALCP}$ are also shown to be decidable, since these problems can be reduced to those of $\mathcal{ALC}$.*
3. *The complexities of the decision problems for $\mathcal{ALCP}$ are also the same as those for $\mathcal{ALC}$, since the mapping f is a polynomial-time reduction.*

3 SEQUENTIAL INCONSISTENCY-TOLERANT DESCRIPTION LOGIC

Next, we introduce the sequential inconsistency-tolerant description logic $\mathcal{ALCPS}$. The $\mathcal{ALCPS}$-concepts are constructed from the $\mathcal{ALCP}$-concepts by adding $[b]$ (sequence modal operator) where b is a sequence. *Sequences* are constructed from countable atomic sequences, $\emptyset$ (empty sequence) and ; (composition). We use lower-case letters $b, c, \ldots$ to denote sequences, and the symbol SE to denote the set of sequences (including $\emptyset$). An expression $[\emptyset]C$ means C, and expressions $[\emptyset\ ;\ b]C$ and $[b\ ;\ \emptyset]C$ mean $[b]C$. We use the symbol $N_C^{[d]}$ ($d \in$ SE) to denote the set $\{[d]B \mid B \in N_C \cup N_C^{\sim}\}$, and the symbol N_C^d ($d \in$ SE) to denote the set $\{B^d \mid B \in N_C \cup N_C^{\sim}\}$ of atomic and negated atomic concepts where we assume $B^{\emptyset} = B$. Note that $N_C^{[\emptyset]} = N_C^{\emptyset} = N_C \cup N_C^{\sim}$. Moreover, we also take the following assumption: For any $A \in N_C$ and any $d \in$ SE,

$$(\sim A)^d = \sim(A^d) \quad \text{(commutativity of} \sim \text{and} \cdot^d\text{)}.$$

This assumption will be used for proving Lemma 3.7.

Definition 3.1. Concepts *C of $\mathcal{ALCPS}$ are defined by the following grammar, assuming A represents atomic concepts and e represents atomic sequences:*

$$C ::= A \mid \neg C \mid \sim C \mid C \sqcap C \mid C \sqcup C \mid \forall R.C \mid \exists R.C \mid [b]C$$
$$b ::= e \mid \emptyset \mid b\ ;\ b$$

The symbol ω is used to represent the set of natural numbers. An expression $\overline{[d]}$ is used to represent $[d_0][d_1]\cdots[d_i]$ with $i \in \omega$, $d_i \in$ SE and $d_0 \equiv \emptyset$. Note that $\overline{[d]}$ can be the empty sequence. We remark that $\overline{[d]}$ is not uniquely determined. For example, if $d \equiv d_1\ ;\ d_2\ ;\ d_3$ where d_1, d_2 and d_3 are atomic sequences, then $\overline{[d]}$ means $[d_1][d_2][d_3]$, $[d_1\ ;\ d_2][d_3]$, $[d_1][d_2\ ;\ d_3]$ or $[d_1\ ;\ d_2\ ;\ d_3]$. Note that $\overline{[d]}$ includes $[d]$.

Definition 3.2. *A* sequential paraconsistent interpretation *$\mathcal{SPI}$ is a structure $\langle \Delta^{\mathcal{SPI}}, \cdot^{\mathcal{SPI}} \rangle$ such that*

1. *$\Delta^{\mathcal{SPI}}$ is a non-empty set,*
2. *$\cdot^{\mathcal{SPI}}$ is an interpretation function which assigns to every concept $B \in N_C \cup N_C^{\sim} \cup N_C^{[d]}$ ($d \in$ SE) a set $B^{\mathcal{SPI}} \subseteq \Delta^{\mathcal{SPI}}$ and to every atomic role R a binary relation $R^{\mathcal{SPI}} \subseteq \Delta^{\mathcal{SPI}} \times \Delta^{\mathcal{SPI}}$.*
3. *For any $A \in N_C$, $(\overline{[d]}\sim A)^{\mathcal{SPI}} = (\sim\overline{[d]}A)^{\mathcal{SPI}}$.*

The interpretation function is inductively extended to concepts by the following conditions:

1. $(\overline{[d]}[b]C)^{\mathcal{SPI}} := ([d\ ;\ b]C)^{\mathcal{SPI}}$,
2. $(\overline{[d]}\neg C)^{\mathcal{SPI}} := \Delta^{\mathcal{SPI}} \setminus (\overline{[d]}C)^{\mathcal{SPI}}$,
3. $(\overline{[d]}(C \sqcap D))^{\mathcal{SPI}} := (\overline{[d]}C)^{\mathcal{SPI}} \cap (\overline{[d]}D)^{\mathcal{SPI}}$,
4. $(\overline{[d]}(C \sqcup D))^{\mathcal{SPI}} := (\overline{[d]}C)^{\mathcal{SPI}} \cup (\overline{[d]}D)^{\mathcal{SPI}}$,
5. $(\overline{[d]}\forall R.C)^{\mathcal{SPI}} := \{a \in \Delta^{\mathcal{SPI}} \mid \forall b\ [(a,b) \in R^{\mathcal{SPI}} \Rightarrow b \in (\overline{[d]}C)^{\mathcal{SPI}}]\}$,
6. $(\overline{[d]}\exists R.C)^{\mathcal{SPI}} := \{a \in \Delta^{\mathcal{SPI}} \mid \exists b\ [(a,b) \in R^{\mathcal{SPI}} \wedge b \in (\overline{[d]}C)^{\mathcal{SPI}}]\}$,

7. $(\overline{[d]}{\sim}{\sim}C)^{\mathcal{SPI}} := (\overline{[d]}C)^{\mathcal{SPI}}$,
8. $(\overline{[d]}{\sim}[b]C)^{\mathcal{SPI}} := ([d\,;b]{\sim}C)^{\mathcal{SPI}}$,
9. $(\overline{[d]}{\sim}\neg C)^{\mathcal{SPI}} := \Delta^{\mathcal{SPI}} \setminus (\overline{[d]}{\sim}C)^{\mathcal{SPI}}$,
10. $(\overline{[d]}{\sim}(C \sqcap D))^{\mathcal{SPI}} := (\overline{[d]}{\sim}C)^{\mathcal{SPI}} \cup (\overline{[d]}{\sim}D)^{\mathcal{SPI}}$,
11. $(\overline{[d]}{\sim}(C \sqcup D))^{\mathcal{SPI}} := (\overline{[d]}{\sim}C)^{\mathcal{SPI}} \cap (\overline{[d]}{\sim}D)^{\mathcal{SPI}}$,
12. $(\overline{[d]}{\sim}\forall R.C)^{\mathcal{SPI}} := \{a \in \Delta^{\mathcal{SPI}} \mid \exists b\ [(a,b) \in R^{\mathcal{SPI}} \wedge b \in (\overline{[d]}{\sim}C)^{\mathcal{SPI}}]\}$,
13. $(\overline{[d]}{\sim}\exists R.C)^{\mathcal{SPI}} := \{a \in \Delta^{\mathcal{SPI}} \mid \forall b\ [(a,b) \in R^{\mathcal{SPI}} \Rightarrow b \in (\overline{[d]}{\sim}C)^{\mathcal{SPI}}]\}$,
14. $({\sim}\overline{[d]}{\sim}C)^{\mathcal{SPI}} := (\overline{[d]}C)^{\mathcal{SPI}}$,
15. $({\sim}\overline{[d]}[b]C)^{\mathcal{SPI}} := ({\sim}[d\,;b]C)^{\mathcal{SPI}}$,
16. $({\sim}\overline{[d]}\neg C)^{\mathcal{SPI}} := \Delta^{\mathcal{SPI}} \setminus ({\sim}\overline{[d]}C)^{\mathcal{SPI}}$,
17. $({\sim}\overline{[d]}(C \sqcap D))^{\mathcal{SPI}} := ({\sim}\overline{[d]}C)^{\mathcal{SPI}} \cup ({\sim}\overline{[d]}D)^{\mathcal{SPI}}$,
18. $({\sim}\overline{[d]}(C \sqcup D))^{\mathcal{SPI}} := ({\sim}\overline{[d]}C)^{\mathcal{SPI}} \cap ({\sim}\overline{[d]}D)^{\mathcal{SPI}}$,
19. $({\sim}\overline{[d]}\forall R.C)^{\mathcal{SPI}} := \{a \in \Delta^{\mathcal{SPI}} \mid \exists b\ [(a,b) \in R^{\mathcal{SPI}} \wedge b \in ({\sim}\overline{[d]}C)^{\mathcal{SPI}}]\}$,
20. $({\sim}\overline{[d]}\exists R.C)^{\mathcal{SPI}} := \{a \in \Delta^{\mathcal{SPI}} \mid \forall b\ [(a,b) \in R^{\mathcal{SPI}} \Rightarrow b \in ({\sim}\overline{[d]}C)^{\mathcal{SPI}}]\}$.

An expression $\mathcal{SPI} \models C$ is defined as $C^{\mathcal{SPI}} \neq \emptyset$. A sequential paraconsistent interpretation $\mathcal{SPI} := \langle \Delta^{\mathcal{SPI}}, \cdot^{\mathcal{SPI}} \rangle$ is a model of a concept C (denoted as $\mathcal{SPI} \models C$) if $\mathcal{SPI} \models C$. A concept C is said to be satisfiable *in $\mathcal{ALCPS}$ if there exists a sequential paraconsistent interpretation $\mathcal{SPI}$ such that $\mathcal{SPI} \models C$.*

Proposition 3.3. *In $\mathcal{ALCPS}$, we have the following equivalence: For any concept C and any sequence d,*

$(\overline{[d]}{\sim}C)^{\mathcal{SPI}} = ({\sim}\overline{[d]}C)^{\mathcal{SPI}}$.

Proof. By induction on C. We show some cases.

• Base step:

Case $C \equiv A \in N_C$: Obvious by the definition of $\cdot^{\mathcal{SPI}}$.

• Induction step:

1. Case $C \equiv [b]D$: $(\overline{[d]}{\sim}[b]D)^{\mathcal{SPI}} = ([d\,;b]{\sim}D)^{\mathcal{SPI}} = ({\sim}[d\,;b]D)^{\mathcal{SPI}}$ (by induction hypothesis) $= ({\sim}\overline{[d]}[b]D)^{\mathcal{SPI}}$.
2. Case $C \equiv {\sim}D$: $(\overline{[d]}{\sim}{\sim}D)^{\mathcal{SPI}} = (\overline{[d]}D)^{\mathcal{SPI}} = ({\sim}\overline{[d]}{\sim}D)^{\mathcal{SPI}}$.
3. Case $C \equiv \neg D$: $(\overline{[d]}{\sim}\neg D)^{\mathcal{SPI}} = \Delta^{\mathcal{SPI}} \setminus (\overline{[d]}{\sim}D)^{\mathcal{SPI}} = \Delta^{\mathcal{SPI}} \setminus ({\sim}\overline{[d]}D)^{\mathcal{SPI}}$ (by induction hypothesis) $= ({\sim}\overline{[d]}\neg D)^{\mathcal{SPI}}$.
4. Case $C \equiv D_1 \sqcap D_2$: $(\overline{[d]}{\sim}(D_1 \sqcap D_2))^{\mathcal{SPI}} = (\overline{[d]}{\sim}D_1)^{\mathcal{SPI}} \cup (\overline{[d]}{\sim}D_2)^{\mathcal{SPI}} = ({\sim}\overline{[d]}D_1)^{\mathcal{SPI}} \cup ({\sim}\overline{[d]}D_2)^{\mathcal{SPI}}$ (by induction hypothesis) $= ({\sim}\overline{[d]}(D_1 \sqcap D_2))^{\mathcal{SPI}}$.
5. Case $\forall R.D$:
$(\overline{[d]}{\sim}\forall R.C)^{\mathcal{SPI}}$
$= \{a \in \Delta^{\mathcal{SPI}} \mid \exists b\ [(a,b) \in R^{\mathcal{SPI}} \wedge b \in (\overline{[d]}{\sim}D)^{\mathcal{SPI}}]\}$
$= \{a \in \Delta^{\mathcal{SPI}} \mid \exists b\ [(a,b) \in R^{\mathcal{SPI}} \wedge b \in ({\sim}\overline{[d]}D)^{\mathcal{SPI}}]\}$ (by induction hypothesis)
$= ({\sim}\overline{[d]}\forall R.C)^{\mathcal{SPI}}$. **Q.E.D.**

Remark 3.4. *We make the following remarks.*

1. *By using the condition $(\overline{[d]}{\sim}C)^{\mathcal{SPI}} = ({\sim}\overline{[d]}C)^{\mathcal{SPI}}$ as shown in* Proposition 3.3, *we can derive the condition $(\overline{[d]}{\sim}A)^{\mathcal{SPI}} := ({\sim}\overline{[d]}A)^{\mathcal{SPI}}$ $(A \in N_C)$ and the conditions 14-20 in* Definition 3.2. *This fact implies that we can define an alternative semantics which is obtained from the semantics defined in* Definition 3.2 *by replacing the condition $(\overline{[d]}{\sim}A)^{\mathcal{SPI}} := ({\sim}\overline{[d]}A)^{\mathcal{SPI}}$ $(A \in N_C)$ and the conditions 14-20 with the condition $(\overline{[d]}{\sim}C)^{\mathcal{SPI}} := ({\sim}\overline{[d]}C)^{\mathcal{SPI}}$.*
2. *$\mathcal{ALCPS}$ has the following equations with respect to $[b]$:*
   (a) *$([b](C \sharp D))^{\mathcal{SPI}} = (([b]C) \sharp ([b]D))^{\mathcal{SPI}}$ where $\sharp \in \{\sqcap, \sqcup\}$,*
   (b) *$([b]\sharp C)^{\mathcal{SPI}} = (\sharp[b]C)^{\mathcal{SPI}}$ where $\sharp \in \{\neg, \forall R., \exists R.\}$,*
   (c) *$(\overline{[d]}C)^{\mathcal{SPI}} = ([d]C)^{\mathcal{SPI}}$.*
3. *Similar to $\mathcal{ALCP}$, $\mathcal{ALCPS}$ is regarded as a four-valued logic and a paraconsistent logic, and can be extended to deal with* ABox, TBox, *and knowledge base.*

Definition 3.5. *The language $\mathcal{L}^{ps}$ of $\mathcal{ALCPS}$ is defined using N_C, N_R, $[b]$, $\neg$, $\sim$, $\sqcap, \sqcup$, $\forall R$, and $\exists R$. The language $\mathcal{L}^{p}$ of $\mathcal{ALCP}$ is obtained from $\mathcal{L}^{ps}$ by adding N_C^d ($d \in$ SE) and deleting $[b]$. A mapping f from $\mathcal{L}^{ps}$ to $\mathcal{L}^{p}$ is defined inductively by*

1. *for any $R \in N_R$, $f(R) := R$,*
2. *for any $A \in N_C$, $f(\overline{[d]}A) := A^d \in N_C^d$, especially, $f(A) := A$,*
3. *$f(\overline{[d]}\sharp C) := \sharp f(\overline{[d]}C)$ where $\sharp \in \{\neg, \sim\}$,*
4. *$f(\overline{[d]}(C \sharp D)) := f(\overline{[d]}C) \sharp f(\overline{[d]}D)$ where $\sharp \in \{\sqcap, \sqcup\}$,*
5. *$f(\overline{[d]}\sharp R.C) := \sharp f(R).f(\overline{[d]}C)$ where $\sharp \in \{\forall, \exists\}$,*
6. *$f(\overline{[d]}[b]C) := f([d\,;b]C)$, especially, $f(\overline{[d]}C) := f([d]C)$.*

Proposition 3.6. *Let f be the mapping defined in Definition 3.5. The following conditions hold for any concept C, and any sequences b, c, d, and k:*

1. *$f(\overline{[d]}[b][c]C) = f(\overline{[d]}[b\,;c]C)$,*
2. *$f(\overline{[d]}\overline{[k]}C) = f(\overline{[d]}[k]C)$,*
3. *$f(\overline{[d]}{\sim}C) = f({\sim}\overline{[d]}C)$.*

Proof. We show (1) and (3) below.

1. Case (1): By using the condition 6 in Definition 3.5 repeatedly, we obtain: $f(\overline{[d]}[b][c]C) = f([d\,;b][c]C) = f([d\,;b\,;c]C) = f(\overline{[d]}[b\,;c]C)$.
2. Case (3): By using the condition 3 in Definition 3.5 twice, we obtain: $f(\overline{[d]}{\sim}C) = {\sim}f(\overline{[d]}C) = f({\sim}\overline{[d]}C)$. **Q.E.D.**

Lemma 3.7. *Let f be the mapping defined in Definition 3.5. For any sequential paraconsistent interpretation $\mathcal{SPI} := \langle \Delta^{\mathcal{SPI}}, \cdot^{\mathcal{SPI}} \rangle$ of $\mathcal{ALCPS}$, we can construct a paraconsistent interpretation $\mathcal{PI} := \langle \Delta^{\mathcal{PI}}, \cdot^{\mathcal{PI}} \rangle$ of $\mathcal{ALCP}$ such that for any concept C in $\mathcal{L}^{ps}$ and any $d \in \mathrm{SE}$,*

$(\overline{[d]}C)^{\mathcal{SPI}} = f(\overline{[d]}C)^{\mathcal{PI}}$.

Proof. Suppose that $\mathcal{SPI}$ is a sequential paraconsistent interpretation $\langle \Delta^{\mathcal{SPI}}, \cdot^{\mathcal{SPI}} \rangle$ such that

1. $\Delta^{\mathcal{SPI}}$ is a non-empty set,
2. $\cdot^{\mathcal{SPI}}$ is an interpretation function which assigns to every concept $B \in \bigcup_{d\in\mathrm{SE}} N_C^{[d]}$ a set $B^{\mathcal{SPI}} \subseteq \Delta^{\mathcal{SPI}}$ and to every atomic role R a binary relation $R^{\mathcal{SPI}} \subseteq \Delta^{\mathcal{SPI}} \times \Delta^{\mathcal{SPI}}$,
3. For any $A \in N_C$, $(\overline{[d]}{\sim}A)^{\mathcal{SPI}} = ({\sim}\overline{[d]}A)^{\mathcal{SPI}}$.

We define a paraconsistent interpretation $\mathcal{PI} := \langle \Delta^{\mathcal{PI}}, \cdot^{\mathcal{PI}} \rangle$ such that

1. $\Delta^{\mathcal{PI}}$ is a non-empty set such that $\Delta^{\mathcal{PI}} = \Delta^{\mathcal{SPI}}$,
2. $\cdot^{\mathcal{PI}}$ is an interpretation function which assigns to every concept $B \in \bigcup_{d\in\mathrm{SE}} N_C^{d}$ a set $B^{\mathcal{PI}} \subseteq \Delta^{\mathcal{PI}}$ and to every atomic role R a binary relation $R^{\mathcal{PI}} \subseteq \Delta^{\mathcal{PI}} \times \Delta^{\mathcal{PI}}$.
3. for any $R \in N_R$, $R^{\mathcal{PI}} = R^{\mathcal{SPI}}$,
4. for any $B \in N_C \cup N_C^{\sim}$ and any $d \in \mathrm{SE}$, $(\overline{[d]}B)^{\mathcal{SPI}} = (B^d)^{\mathcal{PI}}$.

Then, the claim is proved by induction on the complexity of C.

• Base step:

1. Case $C \equiv A \in N_C$: We obtain: $(\overline{[d]}A)^{\mathcal{SPI}} = (A^d)^{\mathcal{PI}} = f(\overline{[d]}A)^{\mathcal{PI}}$ (by the definition of f).
2. Case $C \equiv {\sim}A \in N_C^{\sim}$: We obtain: $(\overline{[d]}{\sim}A)^{\mathcal{SPI}} = (({\sim}A)^d)^{\mathcal{PI}} = {\sim}(A^d)^{\mathcal{PI}}$ (by the assumption $({\sim}A)^d = {\sim}(A^d)$) $= {\sim}f(\overline{[d]}A)^{\mathcal{PI}}$ (by the definition of f) $= f(\overline{[d]}{\sim}A)^{\mathcal{PI}}$ (by the definition of f).
3. Case $C \equiv [b]A$ where $A \in N_C$: We obtain: $(\overline{[d]}[b]A)^{\mathcal{SPI}} = (A^{d\,;\,b})^{\mathcal{PI}} = f(\overline{[d\,;b]}A)^{\mathcal{PI}}$ (by the definition of f) $= f([d\,;b]A)^{\mathcal{PI}}$ (by the definition of f) $= f(\overline{[d]}[b]A)^{\mathcal{PI}}$ (by the definition of f).
4. Case $C \equiv [b]{\sim}A$ where $A \in N_C$: We obtain: $(\overline{[d]}[b]{\sim}A)^{\mathcal{SPI}} = (({\sim}A)^{d\,;\,b})^{\mathcal{PI}} = {\sim}(A^{d\,;\,b})^{\mathcal{PI}}$ (by the commutativity of $\sim$ and $\cdot^d$) $= {\sim}f(\overline{[d\,;b]}A)^{\mathcal{PI}}$ (by the definition of f) $= {\sim}f([d\,;b]A)^{\mathcal{PI}}$ (by the definition of f) $= {\sim}f(\overline{[d]}[b]A)^{\mathcal{PI}}$ (by the definition of f) $= f(\overline{[d]}[b]{\sim}A)^{\mathcal{PI}}$ (by the definition of f).

• Induction step: We show some cases.

1. Case $C \equiv [b]D$: We obtain: $(\overline{[d]}[b]D)^{\mathcal{SPI}} = f(\overline{[d]}[b]D)^{\mathcal{PI}}$ (by induction hypothesis).
2. Case $C \equiv \neg D$: We obtain: $(\overline{[d]}\neg D)^{\mathcal{SPI}} = \Delta^{\mathcal{SPI}} \setminus (\overline{[d]}C)^{\mathcal{SPI}} = \Delta^{\mathcal{SPI}} \setminus f(\overline{[d]}C)^{\mathcal{PI}}$ (by induction hypothesis) $= \Delta^{\mathcal{PI}} \setminus f(\overline{[d]}C)^{\mathcal{PI}}$ (by the condition $\Delta^{\mathcal{SPI}} = \Delta^{\mathcal{PI}}$) $= (\neg f(\overline{[d]}D))^{\mathcal{PI}} = f(\overline{[d]}\neg D)^{\mathcal{PI}}$ (by the definition of f).
3. Case $C \equiv C_1 \sqcap C_2$: We obtain: $(\overline{[d]}(C_1 \sqcap C_2))^{\mathcal{SPI}} = (\overline{[d]}C_1)^{\mathcal{SPI}} \cap (\overline{[d]}C_2)^{\mathcal{SPI}} = f(\overline{[d]}C_1)^{\mathcal{PI}} \cap f(\overline{[d]}C_2)^{\mathcal{PI}}$ (by induction hypothesis) $= (f(\overline{[d]}C_1) \sqcap f(\overline{[d]}C_2))^{\mathcal{PI}} = f(\overline{[d]}(C_1 \sqcap C_2))^{\mathcal{PI}}$ (by the definition of f).
4. Case $C \equiv \forall R.D$: We obtain:
   $(\overline{[d]}\forall R.D)^{\mathcal{SPI}}$
   $= \{a \in \Delta^{\mathcal{SPI}} \mid \forall b\ [(a,b) \in R^{\mathcal{SPI}} \Rightarrow b \in (\overline{[d]}D)^{\mathcal{SPI}}]\}$
   $= \{a \in \Delta^{\mathcal{PI}} \mid \forall b\ [(a,b) \in R^{\mathcal{PI}} \Rightarrow b \in (\overline{[d]}D)^{\mathcal{SPI}}]\}$ (by the conditions $\Delta^{\mathcal{SPI}} = \Delta^{\mathcal{PI}}$ and $R^{\mathcal{SPI}} = R^{\mathcal{PI}}$)
   $= \{a \in \Delta^{\mathcal{PI}} \mid \forall b\ [(a,b) \in R^{\mathcal{PI}} \Rightarrow b \in f(\overline{[d]}D)^{\mathcal{PI}}]\}$ (by induction hypothesis)
   $= (\forall R.f(\overline{[d]}D))^{\mathcal{PI}}$
   $= (\forall f(R).f(\overline{[d]}D))^{\mathcal{PI}}$ (by the definition of f)
   $= f(\overline{[d]}\forall R.D)^{\mathcal{PI}}$ (by the definition of f).
5. Case $C \equiv {\sim}{\sim}D$: We obtain: $(\overline{[d]}{\sim}{\sim}D)^{\mathcal{SPI}} = (\overline{[d]}D)^{\mathcal{SPI}} = f(\overline{[d]}D)^{\mathcal{PI}}$ (by induction hypothesis) $= ({\sim}{\sim}f(\overline{[d]}D))^{\mathcal{PI}} = f(\overline{[d]}{\sim}{\sim}D)^{\mathcal{PI}}$ (by the definition of f).
6. Case $C \equiv {\sim}\neg D$: We obtain: $(\overline{[d]}{\sim}\neg D)^{\mathcal{SPI}} = \Delta^{\mathcal{SPI}} \setminus (\overline{[d]}{\sim}C)^{\mathcal{SPI}} = \Delta^{\mathcal{SPI}} \setminus f(\overline{[d]}{\sim}C)^{\mathcal{PI}}$ (by induction hypothesis) $= \Delta^{\mathcal{PI}} \setminus f(\overline{[d]}{\sim}C)^{\mathcal{PI}}$ (by the condition $\Delta^{\mathcal{SPI}} = \Delta^{\mathcal{PI}}$) $= (\neg f(\overline{[d]}{\sim}D))^{\mathcal{PI}} = (\neg{\sim}f(\overline{[d]}D))^{\mathcal{PI}} = ({\sim}\neg f(\overline{[d]}D))^{\mathcal{PI}} = f(\overline{[d]}{\sim}\neg D)^{\mathcal{PI}}$ (by the definition of f).
7. Case $C \equiv {\sim}(C_1 \sqcap C_2)$: We obtain: $(\overline{[d]}{\sim}(C_1 \sqcap C_2))^{\mathcal{SPI}} = (\overline{[d]}{\sim}C_1)^{\mathcal{SPI}} \cup (\overline{[d]}{\sim}C_2)^{\mathcal{SPI}} = f(\overline{[d]}{\sim}C_1)^{\mathcal{PI}} \cup f(\overline{[d]}{\sim}C_2)^{\mathcal{PI}}$ (by induction hypothesis) $= {\sim}f(\overline{[d]}C_1)^{\mathcal{PI}} \cup {\sim}f(\overline{[d]}C_2)$ (by the

definition of f) $= (\sim(f(\overline{[d]}C_1) \sqcap f(\overline{[d]}C_2)))^{\mathcal{PI}}$ $= \sim f(\overline{[d]}(C_1 \sqcap C_2))^{\mathcal{PI}}$ (by the definition of f) $= f(\overline{[d]}\sim(C_1 \sqcap C_2))^{\mathcal{PI}}$ (by the definition of f).

8. Case $C \equiv \sim\forall R.D$: We obtain:

   $(\overline{[d]}\sim\forall R.D)^{\mathcal{SPI}}$
   $= \{a \in \Delta^{\mathcal{SPI}} \mid \exists b\ [(a,b) \in R^{\mathcal{SPI}} \wedge b \in (\overline{[d]}\sim D)^{\mathcal{SPI}}]\}$
   $= \{a \in \Delta^{\mathcal{PI}} \mid \exists b\ [(a,b) \in R^{\mathcal{PI}} \wedge b \in (\overline{[d]}\sim D)^{\mathcal{SPI}}]\}$ (by the conditions $\Delta^{\mathcal{SPI}} = \Delta^{\mathcal{PI}}$ and $R^{\mathcal{SPI}} = R^{\mathcal{PI}}$)
   $= \{a \in \Delta^{\mathcal{PI}} \mid \exists b\ [(a,b) \in f(R)^{\mathcal{PI}} \wedge b \in f(\overline{[d]}\sim D)^{\mathcal{PI}}]\}$ (by the condition $f(R) = R$ and induction hypothesis)
   $= (\sim\forall f(R).f(\overline{[d]}D))^{\mathcal{PI}}$
   $= (\sim f(\overline{[d]}\forall R.D))^{\mathcal{PI}}$ (by the definition of f)
   $= f(\overline{[d]}\sim\forall R.D)^{\mathcal{PI}}$ (by the definition of f).

Q.E.D.

Lemma 3.8. *Let f be the mapping defined in Definition 3.5. For any sequential paraconsistent interpretation $\mathcal{SPI} := \langle\Delta^{\mathcal{SPI}}, \{\cdot^{\mathcal{SPI}}\rangle$ of $\mathcal{ALCPS}$, we can construct a paraconsistent interpretation $\mathcal{PI} := \langle\Delta^{\mathcal{PI}}, \cdot^{\mathcal{PI}}\rangle$ of $\mathcal{ALCP}$ such that for any concept C in $\mathcal{L}^{ps}$ and any $d \in$ SE,*

$\mathcal{SPI} \models \overline{[d]}C$ iff $\mathcal{PI} \models f(\overline{[d]}C)$.

Proof. We obtain: $\mathcal{SPI} \models \overline{[d]}C$ iff $(\overline{[d]}C)^{\mathcal{SPI}} \neq \emptyset$ iff $f(\overline{[d]}C)^{\mathcal{PI}} \neq \emptyset$ (by Lemma 3.7) iff $\mathcal{PI} \models f(\overline{[d]}C)$. **Q.E.D.**

Lemma 3.9. *Let f be the mapping defined in Definition 3.5. For any paraconsistent interpretation $\mathcal{PI} := \langle\Delta^{\mathcal{PI}}, \cdot^{\mathcal{PI}}\rangle$ of $\mathcal{ALCP}$, we can construct a sequential paraconsistent interpretation $\mathcal{SPI} := \langle\Delta^{\mathcal{SPI}}, \cdot^{\mathcal{SPI}}\rangle$ of $\mathcal{ALCPS}$ such that for any concept C in $\mathcal{L}^{ps}$ and any $d \in$ SE,*

$\mathcal{PI} \models f(\overline{[d]}C)$ iff $\mathcal{SPI} \models \overline{[d]}C$.

Proof. Similar to the proof of Lemma 3.8. **Q.E.D.**

Theorem 3.10 (Embedding from $\mathcal{ALCPS}$ into $\mathcal{ALCP}$)**.** *Let f be the mapping defined in Definition 3.5. For any concept C,*

C is satisfiable in $\mathcal{ALCPS}$ iff $f(C)$ is satisfiable in $\mathcal{ALCP}$.

Proof. By Lemmas 3.8 and 3.9. **Q.E.D.**

Theorem 3.11 (Decidability for $\mathcal{ALCPS}$)**.** *The concept satisfiability problem for $\mathcal{ALCPS}$ is decidable.*

Proof. By Theorems 2.8 and 3.10. **Q.E.D.**

We can also obtain the following results.

Theorem 3.12 (Embedding from $\mathcal{ALCPS}$ into $\mathcal{ALC}$)**.** *Let f be the composition of the mappings defined in Definitions 3.5 and 2.6. For any concept C,*

C is satisfiable in $\mathcal{ALCPS}$ iff $f(C)$ is satisfiable in $\mathcal{ALC}$.

Proof. By combining Theorems 2.7 and 3.10. **Q.E.D.**

Remark 3.13. *We make the following remarks.*

1. *The satisfiability problems of a TBox, an ABox, and a knowledge base for $\mathcal{ALCPS}$ are also shown to be decidable, since these problems can be reduced to those of $\mathcal{ALC}$.*
2. *The complexities of the decision problems for $\mathcal{ALCPS}$ are also the same as those for $\mathcal{ALC}$, since the mapping (composition) used in Theorem 3.12 is a polynomial-time reduction.*

4 CONCLUSIONS AND RELATED WORKS

In this study, we introduced the sequential inconsistency-tolerant description logic $\mathcal{ALCPS}$ that can appropriately handle inconsistency-tolerant ontological reasoning with sequential information. We proved the theorems for embedding $\mathcal{ALCPS}$ into the fragments $\mathcal{ALCP}$ and $\mathcal{ALC}$ of $\mathcal{ALCPS}$. Using one of these embedding theorems, we proved the decidability of the satisfiability problem for $\mathcal{ALCPS}$. These results demonstrate that the existing framework for the standard description logic $\mathcal{ALC}$ can be extended to handle the useful constructors $\sim$ (paraconsistent negation) and $[b]$ (sequence modal operator). Namely, these results demonstrate that using the embedding theorem enables the reuse of previously developed methods and algorithms for $\mathcal{ALC}$ for the effective handling of inconsistency-tolerant ontologies with sequential information described by $\mathcal{ALCPS}$. In the following paragraphs, we discuss related work in the literature.

Inconsistency-tolerant description logics obtained from standard description logics by adding $\sim$ have been studied. An *inconsistency-tolerant four-valued terminological logic*, which is regarded as the original inconsistency-tolerant description logic, was introduced by Patel-Schneider in (Patel-Schneider, 1989). A sequent calculus for reasoning in four-valued description logics was introduced by Straccia in (Straccia, 1997). An application of four-valued description logic to information retrieval was studied by Meghini et al. in (Meghini and Straccia, 1996; Meghini et al., 1998). Three *inconsistency-tolerant con-*

structive description logics, which are based on constructive logic, were studied by Odintsov and Wansing in (Odintsov and Wansing, 2003; Odintsov and Wansing, 2008). Kaneiwa (Kaneiwa, 2007) studied $\mathcal{ALC}^n_{\sim}$, which is an *extended description logics with contraries, contradictories, and subcontraries* that does not strictly qualify as an inconsistency-tolerant description logic. Some *paraconsistent four-valued description logics*, including $\mathcal{ALC}4$, were studied by Ma et al. in (Ma et al., 2007; Ma et al., 2008). Some *quasi-classical description logics* were studied by Zhang et al. in (Zhang and Lin, 2008; Zhang et al., 2009). An inconsistency-tolerant description logic, $\mathcal{PALC}$, was studied by Kamide in (Kamide, 2012). In almost all these logics, except the logics of Odintsov and Wansing, some dual interpretation semantics were used. An inconsistency-tolerant description logic, $\mathcal{SALC}$, that is logically equivalent to $\mathcal{PALC}$ was introduced by Kamide in (Kamide, 2013) by using a simple single interpretation semantics that is similar to those found in the logics of Odintsov and Wansing. The logic $\mathcal{ALCP}$ that is introduced in this paper is a slight modification of $\mathcal{SALC}$.

Sequential description logics that were obtained from $\mathcal{ALC}$ by adding $[b]$ were introduced by Kamide in (Kamide, 2010; Kamide, 2011), where he presented embedding and interpolation theorems for these logics. However, these logics were constructed based on complex multiple interpretation semantics. The $\sim$-free fragment of $\mathcal{ALCPS}$ introduced in this paper is not logically equivalent to the sequential description logic developed in (Kamide, 2010; Kamide, 2011). The condition of the interpretation function with respect to the classical negation connective $\neg$ is different. Although no other sequential description logic equipped with $[b]$, some extended non-classical logics with $[b]$ have been studied in some applications. An *extended sequential paraconsistent computation tree logic*, SPCTL, was introduced by Kamide in (Kamide, 2015), and this logic was used for the verification of clinical reasoning. A *sequence-indexed linear-time temporal logic*, SLTL, was introduced by Kaneiwa and Kamide in (Kaneiwa and Kamide, 2010), and this logic was used for describing security issues with agent communication. An extended linear logic with $[b]$, called a *sequence-indexed linear logic*, was developed by Kamide and Kaneiwa in (Kamide and Kaneiwa, 2013), and this logic was used for formalizing resource-sensitive reasoning with agent communication and event sequences. An extended full computation tree logic with $[b]$ was developed by Kaneiwa and Kamide in (Kaneiwa and Kamide, 2011), and this logic was used for conceptual modeling in domain ontologies. Some temporal logics with $[b]$ have recently been introduced by Kamide and Yano in (Kamide and Yano, 2017; Kamide, 2018), and these logics were used for the logical foundation of hierarchical model checking. In these non-classical logics, which are not description logics, $[b]$ was used for expressing hierarchies, data sequences, event sequences, action sequences, and agent communication sequences.

In addition to the aforementioned studies on inconsistency-tolerant description logics for inconsistency-tolerant ontological reasoning (without handling sequential information), there is another direction of promising studies on inconsistency-tolerant ontological reasoning based on description logics. Such studies do not introduce a new inconsistency-tolerant description logic, but, several inconsistency-tolerant semantics, which are not a semantics for a description logic itself, are introduced and investigated for query answering in description logic knowledge bases (Lembo et al., 2010). For more information on recent developments of inconsistency-tolerant semantics for query answering in description logic knowledge bases, see e.g., (Lembo et al., 2010; Bienvenu et al., 2014; Lukasiewicz et al., 2019) and the references therein.

ACKNOWLEDGEMENTS

We would like to thank the anonymous referees for their valuable comments and suggestions. This research was supported by JSPS KAKENHI Grant Numbers JP18K11171, JP16KK0007, and JSPS Core-to-Core Program (A. Advanced Research Networks).

REFERENCES

Almukdad, A. and Nelson, D. (1984). Constructible falsity and inexact predicates. *Journal of Symbolic Logic*, 49:231–233.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. (2003). *The description logic handbook: Theory, implementation and applications*. Cambridge University Press.

Bienvenu, M., Bourgaux, C., and Goasdoué, F. (2014). Querying inconsistent description logic knowledge bases under preferred repair semantics. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI 2014)*, pages 996–1002.

Gurevich, Y. (1977). Intuitionistic logic with strong negation. *Studia Logica*, 36:49–59.

Kamide, N. (2010). Sequential description logic. *Far East Journal of Applied Mathematics*, 47 (1):205–214.

Kamide, N. (2011). Interpolation theorems for some extended description logics. In *Proceedings of the 15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES 2011), Lecture Notes in Artificial Intelligence*, volume 6882, pages 246–255.

Kamide, N. (2012). Embedding-based approaches to paraconsistent and temporal description logics. *Journal of Logic and Computation*, 22 (5):1097–1124.

Kamide, N. (2013). A comparison of paraconsistent description logics. *International Journal of Intelligence Science*, 3 (2):99–109.

Kamide, N. (2015). Inconsistency-tolerant temporal reasoning with hierarchical information. *Information Sciences*, 320:140–155.

Kamide, N. (2018). Logical foundations of hierarchical model checking. *Data Technology and Applications*, 52 (4):539–563.

Kamide, N. and Kaneiwa, K. (2013). Reasoning about resources and information: A linear logic approach. *Fundamenta Informaticae*, 125 (1):51–70.

Kamide, N. and Yano, R. (2017). Logics and translations for hierarchical model checking. In *Proceedings of the 21st International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Procedia Computer Science*, volume 112, pages 31–40.

Kaneiwa, K. (2007). Description logics with contraries, contradictories, and subcontraries. *New Generation Computing*, 25 (4):443–468.

Kaneiwa, K. and Kamide, N. (2010). Sequence-indexed linear-time temporal logic: Proof system and application. *Applied Artificial Intelligence*, 24 (10):896–913.

Kaneiwa, K. and Kamide, N. (2011). Conceptual modeling in full computation-tree logic with sequence modal operator. *International Journal of Intelligent Systems*, 26 (7):636–651.

Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., and Savo, D. F. (2010). Inconsistency-tolerant semantics for description logics. In *Proceedings of the International Conference on Web Reasoning and Rule Systems (RR 2010), Lecture Notes in Computer Science*, volume 6333, pages 103–117.

Lukasiewicz, T., Malizia, E., and Vaicenavičlus, A. (2019). Complexity of inconsistency-tolerant query answering in datalog+/- under cardinality-based repairs. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, pages 2962–2969.

Ma, Y., Hitzler, P., and Lin, Z. (2007). Algorithms for paraconsistent reasoning with owl. In *Proceedings of the 4th European Semantic Web Conference (ESWC 2007), Lecture Notes in Computer Science*, volume 4519, pages 399–413.

Ma, Y., Hitzler, P., and Lin, Z. (2008). Paraconsistent reasoning for expressive and tractable description logics. In *Proceedings of the 21st International Workshop on Description Logic (DL 2008), Sun SITE Central Europe (CEUR) Electronic Workshop Proceedings*, volume 553.

Meghini, C., Sebastiani, F., and Straccia, U. (1998). Mirlog: A logic for multimedia information retrieval. In *Uncertainty and Logics: Advanced Models for the Representation and Retrieval of Information*, pages 151–185.

Meghini, C. and Straccia, U. (1996). A relevance terminological logic for information retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 197–205.

Nelson, D. (1949). Constructible falsity. *Journal of Symbolic Logic*, 14:16–26.

Odintsov, S. and Wansing, H. (2003). Inconsistency-tolerant description logic: Motivation and basic systems. In *V.F. Hendricks and J. Malinowski, Editors, Trends in Logic: 50 Years of Studia Logica*, pages 301–335.

Odintsov, S. and Wansing, H. (2008). Inconsistency-tolerant description logic. Part II: Tableau algorithms. *Journal of Applied Logic*, 6:343–360.

Patel-Schneider, P. F. (1989). A four-valued semantics for terminological logics. *Artificial Intelligence*, 38:319–351.

Rautenberg, W. (1979). *Klassische und nicht-klassische Aussagenlogik*. Vieweg, Braunschweig.

Schmidt-Schauss, M. and Smolka, G. (1991). Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26.

Straccia, U. (1997). A sequent calculus for reasoning in four-valued description logics. In *Proceedings of International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 1997), Lecture Notes in Computer Science*, volume 1227, pages 343–357.

Vorob'ev, N. (1952). A constructive propositional calculus with strong negation (in Russian). *Doklady Akademii Nauk SSR*, 85:465–468.

Wansing, H. (1993). The logic of information structures. In *Lecture Notes in Computer Science*, volume 681, pages 1–163.

Zhang, X. and Lin, Z. (2008). Paraconsistent reasoning with quasi-classical semantics in *ALC*. In *Proceedings of the 2nd International Conference on Web Reasoning and Rule Systems (RR 2008)*, volume 5341, pages 222–229.

Zhang, X., Qi, G., Ma, Y., and Lin, Z. (2009). Quasi-classical semantics for expressive description logics. In *Proceedings of the 22nd International Workshop on Description Logic (DL 2009), Sun SITE Central Europe (CEUR) Electronic Workshop Proceedings*, volume 477.