전치사 Over의 공간적 의미 표현과 추론

Representation and Reasoning of Spatial Meaning of "Over"

김 민 구
(Min-koo Kim)

Abstract This paper describes a problem of representing the visible knowledge for locative constructions, sentences including a preposition spatially. In particular, we choose the scene which is related to the trajector-landmark based on the most complicated preposition in English, "over." Actually, this work is inspired by Lakoff [6]. Lakoff proposed twenty different image schemas of the preposition "over." In this paper, we choose seven image schemas from Lakoff's and construct an ISA hierarchy. From this hierarchy, we identify several problems for nonmonotonic reasoning, and show how to handle these problems using a meta-model theory [5].

Keywords Spatial meaning, Over, Knowledge representation, Image schema, Nonmonotonic reasoning.


1. Introduction

The sense of the preposition “over” that appears to be the most central consists of elements of both “above” and “across.” Consider the following sentences: (1) The plane flew over the town; and, (2) The plane flew over the hill. Fig. 1.1 and Fig. 1.2 depict the image schemas for these two sentences. In these figures, the plane is understood as a trajector(TR) oriented relative to a landmark(LM). The arrow represents the path of the trajector. The only difference between these two image schemas is that the LM in Fig. 1.1 is horizontal, while the LM in Fig. 1.2 has a vertical orientation (we will call this convex).

![Fig 1.1 The plane flew over the town.](image-url)
Now, consider some variations of Fig 1.1 and Fig 1.2. For example, consider the following sentences: (1) The helicopter is hovering over the town; (2) The power line stretches over my yard; and, (3) The helicopter is over the hill. Fig 1.3, Fig 1.4, and Fig 1.5 depict these three sentences, respectively. The difference between Fig 1.1 and Fig 1.3 is that the TR in Fig 1.1 has a path, while the TR in Fig 1.3 has no specific path. The difference between Fig 1.1 and Fig 1.4 is that the TR in Fig 1.1 is a one-dimensional object, while the TR in Fig 1.4 is an one-dimensional object. The difference between Fig 1.2 and Fig 1.5 is that the TR in Fig 1.2 has a path, while the TR in Fig 1.5 has no specific path.

Let us consider two more different image schemas of the preposition “over.” Consider the following sentences: (1) Jack is walking over the hill; and, (2) Sam lives over the hill. Fig 1.6 and Fig 1.7 depict the above two sentences, respectively. The difference between Fig 1.2 and Fig 1.6 is that there is no contract between the TR and the LM in Fig 1.2, while there is contact between the TR and the LM in
Fig 1.6. The difference between Fig 1.6 and Fig 1.7 is that in Fig 1.6, the path is real and the TR is at the middle of the path, while in Fig 1.7, the path is potential and the TR is at the end of the path.

2. Hierarchical Representation of Image Schemas for the Preposition “Over”

In the previous section, we chose seven different image schemas for the preposition “over.” In order to represent these schemas, we first construct an ISA hierarchy for these schemas as shown in Fig 2.1.

In this hierarchy, we use intermediate image schemas, (1), (2), (3), (4), (5), (6), (7), and (12) in Fig 2.1, which were not explained in the previous section. These intermediate schemas contain some unknown properties for the TR and/or the LM. For example, in Fig 2.1-(7), the TR is an unknown object which has a path. In other words, the TR can be any object, for example, one-dimensional object or two dimensional object, which has a path.

3. A Knowledge Representation Language: UniFrame

Now, we attempt to represent the hierarchy of image schemas for the preposition “over” using a frame based language. To do this, we propose a frame-based language, called UniFrame that is based on the principle of representational uniformity[6]. This is the assumption that all concepts should be represented by the same type of data structure. In this way, it differs from well-known frame systems (e.g., FRAIL[2] and KL-ONE[1]) that use two kinds of data structures: frames (or concepts) and slots (or roles). As such, UniFrame avoids the well-know semantic difficulties associated with the use of slots (c.f., [14]). A similar approach can be found in Wilensky’s KODIAK [13]. UniFrame allows for a variable depth of processing of the meaning of a concept (c.f., [7]). This property differs from Conceptual Dependency [12], where concepts are unavoidably decomposed into their primitives.

3.1 Syntax of UniFrame

UniFrame has eight fields. There are: concept, with, isa, definition, constraint, recog, facts, and members). The first two fields, concept and with, are mandatory. The remaining fields are optional. The form of a UniFrame is shown below.

\[
\begin{array}{l}
\text{[concept: concept-name} \\
\text{with: list-of-variable-restrictions} \\
\text{isa: list-of-concept-names} \\
\text{definition: list-of-formulas} \\
\text{constrain: list-of-formulas} \\
\text{recog: list-of-formulas} \\
\text{facts: list-of-formulas} \\
\text{members: list-of-concept-names}]
\end{array}
\]

In the above, formulas are the same as sentences in first-order predicate logic. The properties of each field are explained in the following section.

3.2 Properties of Each Field

For each field of a UniFrame, we will intuitively rather than formally explain semantic properties.2)

(1) Concept
This field names the concept which we want to define. Subsequently, the concept name can be used as a predicate symbol.

(2) with
The with field is a list of typed formal variables. This indicates the number of arguments of the predicate symbol or concept name. The typing specifies the kind of object that each variable can be bound to. Each type specifier must be also be a (object-oriented) concept name. The syntax of this field is as follows.

---

1) Some of these fields are defined differently from Maida’s UniFrame[7].
2) For the formal description of semantic properties, see [5].
Fig 2.1 A hierarchy of image schema for the preposition "over"
with: \((\$v_1 \ldots \text{conceptname}_n)\)

\((\$v_n \ldots \text{conceptname}_n)\)

(3) isa

This field is used for representation ISA hierarchies. These are commonly adopted by most frame-based languages. The syntax of this field is as follows.

\text{isa:} \((\text{conceptname}_1, \ldots, \text{conceptname}_n)\)

In the above syntax, \text{conceptname}_i is a concept name, or the negation of a concept name, with a negation symbol \(\neg\). For example, consider the following UniFrame for baby elephants. It says that baby elephants are elephants, but they are not giant elephants.

\[
\begin{align*}
\text{[concept: Baby_E} \\
\text{with:} \ (\$x \text{ Elephant}) \\
\text{isa:} \ (\neg \text{Giant_E})
\end{align*}
\]

Semantically, the isa field is equivalent to the with field for object-oriented concepts (or predicates), except that isa field can specify more than one superconcepts.

(4) definition

This field describes the definitional term for the given concept. For instance, if we want to define a concept of bachelor as a conjunction of the concepts single and male, then we can have a frame for the concept of bachelor as follows.

\[
\begin{align*}
\text{[concept: Bachelor} \\
\text{with:} \ (\$x \text{ Person}) \\
\text{definition:} \ (\text{Single}($x) \land \text{Male}($x))
\end{align*}
\]

(5) constrain

The purpose of this field is to describe necessary conditions for being considered as a given concept. For example, consider what conditions must be necessary to say that something is a mammal. We might consider some condition, for instance, something is warm-blooded. The following frame describes a concept Mammal.

\[
\begin{align*}
\text{[concept: Mammal} \\
\text{with:} \ (\$x \text{ Thing}) \\
\text{constrain:} \ (\text{warm-blooded}($x))
\end{align*}
\]

The constrain field for the above frame has the following semantic property. \(\rightarrow\), we call it t-entailment, has the same meaning as the material implication \(\Rightarrow\) has.

\[
\forall x \ 	ext{Mammal}(x) \rightarrow \text{warm-blooded}(x)
\]

(6) recog

The purpose of this field is to describe minimal requirements to be recognized as a given concept. For instance, consider what conditions can be intuitively sufficient (not necessarily sufficient) to say that something is an elephant. We might consider some conditions, for instance, something is a big mammal and it has a truck. The following frame describes these conditions.

\[
\begin{align*}
\text{[concept: Elephant} \\
\text{with:} \ (\$x \text{ Mammal}) \\
\text{recog:} \ (\text{big}($x) \land \text{has-a-tusk}($x))
\end{align*}
\]

The recog field for the above frame has the following semantic properties.

\[
\forall x \ 	ext{Elephant}(x) \rightarrow \text{big}(x) \land \text{has-a-tusk}(x)
\]

\[
\forall x \ 	ext{big}(x) \land \text{has-a-tusk}(x) \rightarrow \text{Elephant}(x)
\]

In the above, the symbol \(\rightarrow\) intuitively means "typically implies." This entailment \(\rightarrow\) involves nonmonotonic reasoning. Actually, we use a seven-valued logic[4] to define the entailment \(\rightarrow\) called d-entailment that is a little different from the one used in one of the famous approaches to nonmonotonic reasoning: Default Logic[9], Nonmonotonic Logic[8], and Circumscription[7]. The difference will be explained in Section 5.

(7) facts

This field describes contingent facts for the given
concepts. For example, if we want to say that birds can typically fly, then we can represent it as follows.

\[
\text{[concept: Bird} \\
\quad \text{with:} \quad (\text{\$x Animal}) \\
\quad \text{.} \\
\quad \text{facts:} \quad (\neg \text{canfly(\$x))}
\]

(8) members
This field represents concepts which are considered as member concepts (a kind of subconcept) for the given concept. For example, we can categorize elephants in several ways, for instance: African elephants or Asiatic elephants, male or female, and baby elephants or adult elephants. The following frame illustrates this example.

\[
\text{[concept: Elephant} \\
\quad \text{with:} \quad (\text{\$x Mammal}) \\
\quad \text{.} \\
\quad \text{members:} \quad ((\text{African_E, Asiatic-E}), \\
\quad (\text{Male_E, Female_E}), \\
\quad (\text{Baby_E, Adult_E}))
\]

4. Representation of the Hierarchy of Image Schemas for the Preposition “Over”

Now, we represent the hierarchy of image schemas for the preposition “over” using UniFrame. In this paper, we will represent some part of the hierarchy,\(^3\) (1), (7), (12), (13), (14), and (15) in Fig 2.1, as follows. In the below, we use notational short hands: \(tr\) stands for \text{trajector}, \(lm\) stands for \text{landmark}; \(o\) stands for \text{over}, \(m\) stands for \text{move}; \(d\) stands for \text{dimensional}; and \(con\) stands for \text{convex}.

\[
\text{(1) [concept: tr_o_lm} \\
\quad \text{with:} \quad (\text{\$x Object} \\
\quad (\text{\$lm Landmark}) \\
\quad \text{recog:} \quad (\text{above(\$str, \$lm)} \land \text{across(\$str, \$lm)})
\]

\[
\text{members:} \quad ((\text{tr_o_hor_lm, tr_o_con_lm}), \\
\quad (\text{tr_m_o_lm, tr_s_o_lm})) \\
\quad \text{facts:} \quad \text{located_at_the_middle_of(\$str, \$lm),} \\
\quad \neg \text{is_touched_with(\$str, \$lm))}
\]

(7) [concept: tr_m_o_con_lm \\
\quad \text{with:} \quad (\text{\$x Object} \\
\quad \text{\$lm Landmark}) \\
\quad \text{isa:} \quad (\text{tr_m_o_lm, tr_o_con_lm}) \\
\quad \text{definition:} \quad (\text{tr_m_o_lm(\$str, \$lm)} \land \\
\quad (\text{tr_o_con_lm(\$str, \$lm})) \\
\quad \text{members:} \quad ((\text{tr_m_along_con_lm,} \\
\quad 0_d_tr_m_o_con_lm)))
\]

(12) [concept: tr_m_along_con_lm \\
\quad \text{with:} \quad (\text{\$x Object}) \\
\quad \text{\$lm Landmark}) \\
\quad \text{isa:} \quad (\text{tr_m_o_con_lm}) \\
\quad \text{recog:} \quad (\neg \text{horizontally_m(\$tr)}) \\
\quad \text{members:} \quad ((0_d_tr_m_along_con_lm,} \\
\quad \text{tr_already_m_along_con_lm,)) \\
\quad \text{facts:} \quad \text{is_touched_with(\$str, \$lm))}
\]

(13) [concept: 0_d_tr_m_o_con_lm \\
\quad \text{with:} \quad (\text{\$x Object}) \\
\quad \text{\$lm Landmark}) \\
\quad \text{isa:} \quad (\text{tr_m_o_con_lm}) \\
\quad \text{recog:} \quad (\text{smaller_than(\$str, \$lm)})
\]

(14) [concept: 0_d_tr_m_along_con_lm \\
\quad \text{with:} \quad (\text{\$x Object}) \\
\quad \text{\$lm Landmark}) \\
\quad \text{isa:} \quad (\text{tr_m_along_con_lm}) \\
\quad \text{recog:} \quad (\text{smaller_than(\$str, \$lm)})
\]

(15) [concept: tr_already_m_along_con_lm \\
\quad \text{with:} \quad (\text{\$x Object}) \\
\quad \text{\$lm Landmark}) \\
\quad \text{isa:} \quad (\text{tr_m_along_con_lm}) \\
\quad \text{recog:} \quad (\text{smaller_than(\$str, \$lm)} \land \\
\quad \text{located_at_the_middle(\$str, \$lm))}
\]

In the above UniFrame (7), concept \text{tr_m_o_con_lm} (trajector moves over convex landmark) is defined in the

\(^3\) The entire representation of the hierarchy can be found in [4].
conjunction of two concepts \( tr\_o\_con\_lm \) and \( tr\_m\_o\_lm \) which are depicted by image schemas (3) and (4) in Fig 2.1, respectively.

5. Meta-Model Theories for Nonmonotonic reasoning

Before we state some nonmonotonic problems involved in the image schemas of the preposition "over," we introduce our meta-model theories [4]. In Section 2, we introduced two kinds of entailment, t-entailment (denoted by \( \alpha \rightarrow^t \beta \)) and d-entailment (denoted by \( \alpha \rightarrow^d \beta \)). The main difference between these entailments and the one used in Default Logic [9] (or Nonmonotonic Logic [8], or Circumscription [7]) is to use a seven-valued logic [4]. In the seven valued logic, seven truth values are used: \( \text{t}(\text{true}), \text{f}(\text{false}), \text{d}\text{t}(\text{true by default}), \text{d}\text{f}(\text{false by default}), \text{c}(\text{contradiction}), \text{d}\text{c}(\text{contradiction by default}), \text{u}(\text{unknown}) \).

These seven truth values are based on Ginsberg's bilattice [3] in Fig 5.1. Due to the properties of seven valued logic, we can assign useful semantic constraints to these entailments. Instead of describing formal semantic constraints (for the formal description, see [5]), we intuitively describe the following three constraints in the forms of inference rule. In the below, operator \( D \) (for example \( D(\alpha) \)), indicate the sentence \( \alpha \) has the truth value \( \text{d}t \) (true by default), and \( \alpha, \beta \), and \( \gamma \) are sentences.

### Constraint 1 (Implication)

1. \( \alpha \rightarrow^t \beta \)
2. \( \alpha \rightarrow^t \beta \)
3. \( \alpha \rightarrow^d \beta \)
4. \( \alpha \rightarrow^d \beta \)

### Constraint 2 (Transitivity)

1. \( \alpha \rightarrow^t \beta \)
2. \( \beta \rightarrow^t \gamma \)
3. \( \alpha \rightarrow^t \gamma \)
4. \( \alpha \rightarrow^d \beta \)
5. \( D(\alpha) \rightarrow^d \beta \)
6. \( \beta \rightarrow^t \gamma \)
7. \( D(\beta) \rightarrow^d \gamma \)
8. \( \alpha \rightarrow^d \beta \)
9. \( D(\alpha) \rightarrow^d \beta \)

### Constraint 3 (Negation)

1. \( \alpha \rightarrow^t \neg \beta \)
2. \( D(\alpha) \rightarrow^t \neg \beta \)
3. \( \alpha \rightarrow^d \neg \beta \)
4. \( D(\alpha) \rightarrow^d \neg \beta \)

\( \neg(\alpha \rightarrow^t \beta) \quad D(\neg(\alpha \rightarrow^t \beta)) \)

\( \neg(\alpha \rightarrow^d \beta) \quad D(\neg(\alpha \rightarrow^d \beta)) \)
In Constraint 2, we can find interesting phenomenon such that rules which are generated from transitivity applied from truth rules (encoded by t-entailment) to default rules (encoded by d-entailment) are less relevant (reliable). For example, Royal elephants are elephants; and, Elephants are typically gray. From transitivity applied to these rules, we can generate a rule: Royal elephants are typically gray. Then, we can consider that this generated rule is less reliable than an asserted rule such that Royal elephants are typically white. On the other hand, we can observe that rules which are generated from transitivity applied from default rules to truth rules are at least as relevant (reliable) as originally asserted rules. For example, Elephants are typically gray, and Gray is not red. In this example, we can generate a rule Elephants are not typically red. This rule is at least as reliable as the rule Elephants are typically gray. The reason is that since every gray thing is not a red thing, we can say that elephants are not typically red with at least the same amount of reliability as the rule Elephants are typically gray has.

In the seven valued logic, we define a logical implication called relevant implication, denoted by \( \models_r \), that satisfies the above semantic constraints. Using this relevant implication, we can develop two meta-model theories. In this paper, we will intuitively describe them. In the meta-model theories, we find out relations between models (contents). The first relation represents which context is more confident. For example, consider a sentence \( \alpha \) which has the truth value \( dt \) from context C1 and also has the truth value \( t \) from context C2. Then, we can say that C2 is more reliable than C1 for \( \alpha \). Based on this relation, we define confident implication denoted by \( \models_c \) such that KB \( \models_c \alpha \) if and only if there exists a most confident context in KB from which \( \alpha \) can be relevantly implied.

The second relation describes which context is more specific. Consider sentence \( \alpha \) which has the truth value \( dt \) from context C1 and also has the truth value \( df \) from context C2. However, if context C2 has more specific evidence for \( \alpha \) than C1 has, we would decide that the truth value of \( \alpha \) is \( dt \). For example, if we know that Tweety is a bird without knowing that Tweety is a penguin, then we may conclude that Tweety can fly. However, if we know Tweety is a penguin, we would conclude Tweety cannot fly, since we have more specific information. Based on this specific relation\(^4\), we define specific implication denoted by \( \models_s \) such that KB \( \models_s \alpha \) if and only if there exists a context that is a most specific context in KB from which \( \alpha \) can be relevantly implied.

6. Some Examples Involving Nonmonotonic Reasoning

This section describes some nonmonotonic problems involved in the image schemas of the preposition "over," and applies the meta-model theories to these problems. In particular, given some descriptions of an image about the preposition "over," we attempt to find an image schema which is well-matched to the given description.

Problem 1. Suppose that Sam lives over the hill. From this information, let us assume that we find some information such as smaller than (Sam, The_hill), located at the middle (Sam, The_hill), Object(Sam), Landmark(The_hill), and tr_m.along_con_lm(Sam, The_hill). We recall that concept tr_m.along_con_lm is defined in UniFrame (12). From these formulas, we can trigger two UniFrames (14) and (15), since these formulas satisfy the conditions which are represented in the fields recog, is, and with. In other words, KB

\[
\models_r, D(0_d tr m_along_con_lm(Sam, The_hill)) \land KB
\]

\[
\models_r, D(tr_{already} m_along_con_lm(Sam, The_hill)).
\]

Since \{smaller_than(Sam, The_hill) \land located\_ at\_ the\_ middle\_ of\_ (Sam, The_hill)\} \models_r smaller_than(Sam, The_hill), KB \models_r D(tr_{already} m_along_con_lm(Sam, The_hill)). However, KB \models_r D(0_d tr m_along_con_lm(Sam, The_hill)), since the more specific information implies the opposite result. Therefore, we can pick up image schema (15) in Fig 2.1 rather than image schema (14) in Fig 2.1 for the sentence Sam lives over the hill.

\(^4\) For the formal definition of specific context, see [5]
Problem 2. In the previous example, since \( KB \models tr\_m\_along\_con\_lm(Sam, The\_hill) \), following isa field of UniFrame (12), \( KB \models tr\_m\_o\_con\_lm(Sam, The\_hill) \) (represented in UniFrame (7)). Therefore, it is possible to trigger UniFrame (13), since conditions in the recog field of UniFrame (13) are satisfied. That is \( KB \models D(0\_d\_tr\_m\_o\_con\_lm(Sam, The\_hill)) \). We recall that UniFrame (7) for concept \( tr\_m\_o\_con\_lm \) has a field members which represents disjoint subconcepts \( 0\_d\_tr\_m\_o\_con\_lm \) (represented in UniFrame (13)) and \( tr\_m\_along\_con\_lm \) (represented in UniFrame (12)). The concept \( tr\_m\_along\_con\_lm \) has also two disjoint subconcepts \( 0\_d\_tr\_m\_o\_con\_lm \) (represented in UniFrame (14)) and \( tr\_already\_m\_along\_con\_lm \) (represented UniFrame (15)). This implies that formula \( 0\_d\_tr\_m\_o\_lm(Sam, The\_hill) \) conflicts with formula \( tr\_already\_m\_along\_con\_lm(Sam, The\_hill) \). However, since \( \{ smaller\_than(Sam, The\_hill) \land \neg located\_at\_the\_middle\_of(Sam, The\_hill) \} \models smaller\_than(Sam, The\_hill), KB \models D(0\_d\_tr\_m\_o\_con\_lm(Sam, The\_hill)). \) Even, KB \( \models D(0\_d\_tr\_m\_o\_con\_lm(Sam, The\_hill)) \). Therefore, we can pick up image schema (15) in Fig 2.1 rather than image schema (13) in Fig 2.1 for the sentence Sam lives over the hill.

Problem 3. In this hierarchy of image schema for the preposition over, we can find that properties of subconcepts override properties of superconcepts. Suppose that \( KB \models D(\neg located\_at\_the\_middle\_of(Sam, The\_hill)) \) from the properties of the recog field in UniFrame (15). Following the isa field in UniFrame (15), we can get up to UniFrame (1). That is \( KB \models D(tr\_o\_lm(Sam, The\_hill)) \). Therefore, formulas in the facts field of UniFrame (1) can be fired. This implies that \( KB \models D(\neg(\neg located\_at\_the\_middle\_of(Sam, The\_hill))) \). So, we can have a conflict. Since \( KB \models tr\_already\_m\_along\_con\_lm(Sam, The\_hill) \)

\[ \rightarrow tr\_o\_lm(Sam, The\_hill), KB \models D(located\_at\_the\_middle\_of(Sam, The\_hill)) \) and \( KB \models D(located\_at\_the\_middle\_of(Sam, The\_hill)) \).

Therefore, we can conclude that Sam is not located at the middle of the hill.

7. Conclusions

We constructed a hierarchy of image schemas for the preposition “over” using UniFrame, and identified several nonmonotonic reasoning problems. In particular, given some sentence including the preposition “over,” we attempted to find an image schema which is well-matched to the given sentence. The members fields of UniFrame was efficiently used in encoding this kind of knowledge, since this kind of knowledge contained many disjoint concepts. For example, consider following concepts: white, gray, green, black, and so on. If we want to say that they are distinct (disjoint) concepts, then imagine how many formulas are needed to represent the fact that the are distinct. In applying our meta-model theory, we could not find any implausible result. In particular, the qualification problem[8] did not occur, and unintended conflicts were not produced.

References


