INTERNATIONAL STANDARD

ISO 24617-14

First edition
2023-06

Language resource management — Semantic annotation framework (SemAF) —

Part 14:
Spatial semantics

Gestion des ressources linguistiques — Cadre d'annotation sémantique (SemAF) —

Partie 14: Sémantique spatiale

Reference number
ISO 24617-14:2023(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 37, *Language and terminology*, Subcommittee SC 4, *Language resource management*.

A list of all parts in the ISO 24617 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document establishes a semantic ground for supporting ISO 24617-7 (spatial information), which specifies an abstract syntax for the annotation of spatial information in language. It also specifies a way of translating the annotation structures generated by the abstract syntax of ISO 24617-7 into well-formed semantic forms. These semantic forms are represented in a type-theoretic first-order logic and made interpretable according to a model.

This document:

— validates the abstract specification of ISO 24617-7 for the annotation of spatial information in language on semantic grounds;

— specifies an interoperable format for interpreting spatial information, both static and dynamic.

Dynamic spatial information involves spatio-temporal information as well as information about motions in space and time. This document aims at satisfying such needs. An understanding of information in natural language is necessary for many computational linguistics and artificial intelligence (AI) applications. An explicit semantics is necessary for the specification provided by ISO 24617-7, as the representations created in accord with that language will not have a significant impact on AI and automatic inference without explicit interpretation.

Language resource management — Semantic annotation framework (SemAF) —

Part 14: Spatial semantics

1 Scope

This document extends ISO 24617-7:2020, which specifies ways of annotating spatial information in natural language such as English, by establishing a formal semantics for its abstract syntax. The task of the proposed semantics is of two kinds:

a) translation of annotation structures to semantic forms;

b) model-theoretic interpretation of semantic forms.

Semantic forms are represented in a type-theoretic first-order logic. These semantic forms are then interpreted with respect to a model for part of the world to which an annotated language is referentially, or denotationally, anchored.

NOTE The basic framework and content of this document is based on Reference [1].

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 24617-7:2020, *Language resource management — Semantic annotation framework — Part 7: Spatial information*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
annotation structure
information structure created by marking up some linguistic expressions with relevant (semantic) information

Note 1 to entry: ISO 24617-7:2020, for instance, creates such annotation structures by marking up place names or motions and their spatial relations with relevant spatial information.

3.2
eigenplace
eigenspace
region or path occupied by an object

Note 1 to entry: A region may be considered as a particular finite path matching to an interval [x,x] such that its start and endpoint match or are identical. In that case, a region is considered as a point.

3.3
event-path
region of space occupied by a mover (moving object) throughout an event

3.4
first-order logic
formal language, artificially built for reasoning, with the values of its terms, particularly variables, ranging over individual objects only

Note 1 to entry: Second-order variables such as *P*, which ranges over properties of an individual, are temporarily introduced to allow the λ-operation in the process of deriving *semantic forms* (3.7), see 7.2, Note and Example 2, b) and c).

3.5
interpretation
function that maps a *semantic form* (3.7) to its denotation

Note 1 to entry: The interpretation function is represented by ⟦ ⟧ and, for each semantic form a, its denotation or the value of the interpretation, is represented by ⟦σ(a)⟧.

Note 2 to entry: In a model-theoretic semantics, the interpretation function ⟦ ⟧ is constrained by a model M and, for each semantic form a and a model M, such an interpretation is represented by $[\![\sigma(a)]\!]^M$.

3.6
model M
set-theoretical construct that represents part of the real or possible world denoted by *semantic forms* (3.7)

3.7
semantic form
logical form
representation of the semantic content of an *annotation structure* (3.1) of expressions in natural language

Note 1 to entry: The semantic form of an annotation structure a is represented by σ(a), where σ is a function that maps an annotation structure a to a semantic form that carries the semantic content of a.

Note 2 to entry: Semantic forms are often called "logical forms" because semantic forms are represented by a logical language such as *first-order logic* (3.4).

3.8
type
semantic type
kind or sort of an object denoted by a linguistic expression

4 Metamodel

This document shall be used together with ISO 24617-7:2020.

The metamodel presented in this clause outlines the basic semantic structure for the abstract syntax of ISO 24617-7 for easy reference, which specifies an annotation scheme for the markup of spatial relations, both static and dynamic, as expressed in text and other media. This specification distinguishes the following six major categories of spatially relevant elements for markup in natural language:

a) spatial entities: natural or artificial locations in the world that include places, paths and event-paths, as well as individual entities participating in spatial relations;

b) spatial relators (signals): linguistic markers that establish relations between places and spatial entities;

c) spatial measures: quantitative information associated with spatial entities;

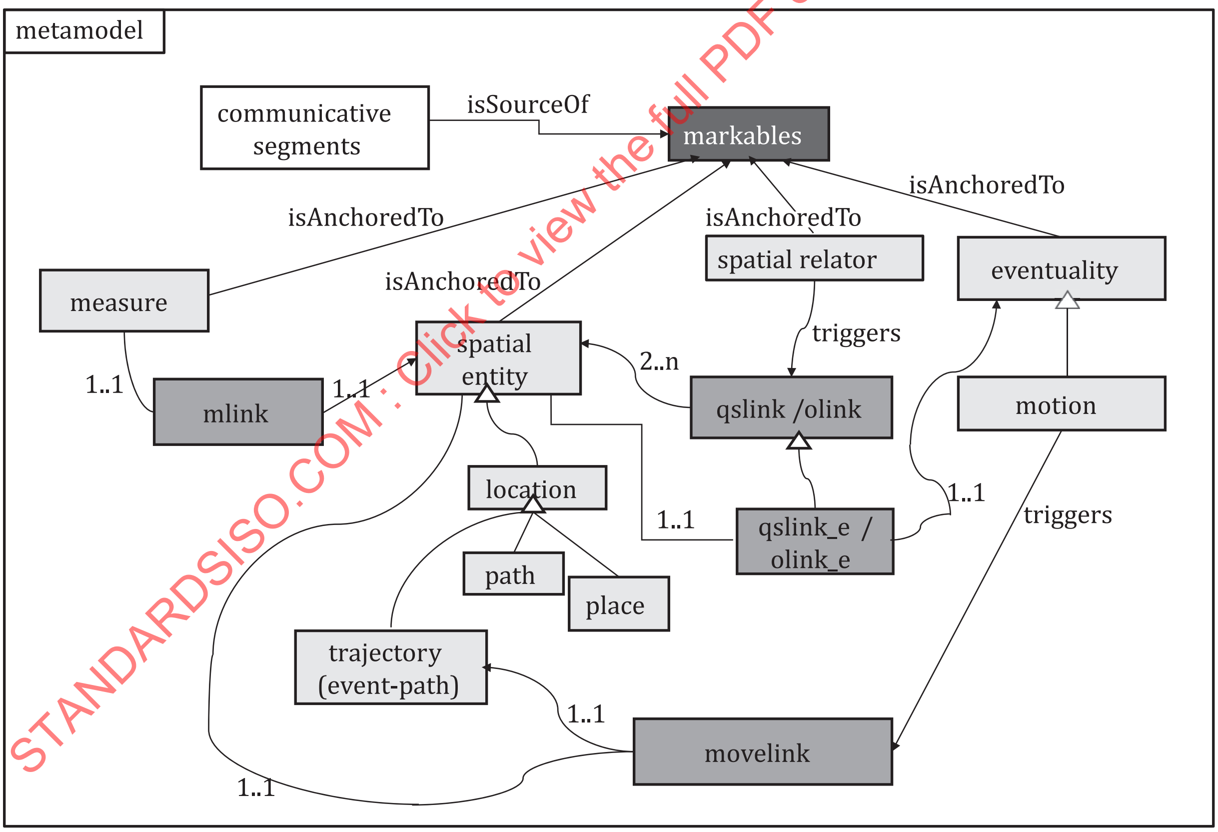
d) events and motions: eventualities either static or dynamic;

NOTE Unlike static eventualities such as referring to states, dynamic eventualities (motions) involve movement from one location to another triggering a trajectory (event-path).

e) static spatial relations: specific qualitative configurational, orientational and metric relations between objects;

f) dynamic spatial relations: movement of an object triggered by a motion from one location to another creating an event-path.

The corresponding metamodel for these categories is represented in Figure 1.



NOTE Source: Reference [2] with some modifications.

Figure 1 — Metamodel

Qualitative spatial link (qslink) and orientation link (olink) each relate one spatial object to another. In contrast, qslink_e and olink_e relate an eventuality of a special type such as “live” to a location such as “Boston” with a spatial signal “in”.

These categories are constrained by semantic types. Each of the categories listed in the abstract syntax of isoSpace is shown to match one of the semantic types defined in Clause 5.

5 Semantic types

5.1 General

The semantics of isoSpace is formulated on the basis of its abstract syntax, but its interpretation rules apply to the semantic forms which are derived from annotation structures as represented by a concrete syntax. Hence, there are two levels of interpretation that shall be identified when defining a formal semantics of an annotation structure, as applied to linguistic expressions in natural language:

— language to abstract model;

— concrete model to abstract model.

This clause focuses on the first mapping. It articulates the underlying semantics of the entities represented in the metamodel in type-theoretic terms and demonstrates the composition of examples within each category. Clause 6 illustrates the second mapping, from the annotation structure (implemented as a concrete syntactic expression) into the abstract model.

5.2 Basic types

5.2.1 General

The model-theoretic semantics of ISO 24617-7:2020 is based on a theory of semantic types, which sorts out various objects denoted by linguistic expressions or their annotation structures. It is assumed that a model is characterized with the basic types in 5.2.2 and the functional types in 5.2.3, corresponding generally to the categories in Figure 1. Following Reference [3], the list of basic types is extended to eight basic types from the two basic types (*e*, the type of objects, and *t*, the type of truth values) in Montague Semantics (see Reference [4]) as given in 5.2.2.

5.2.2 Extended basic types

The basic types are as follows:

a) *t*, the type of truth values;

b) *e*, the type of objects (entities);

c) *i*, the type of time points;

d) *p*, the type of spatial points;

e) *v*, the type of events;

f) *m*, the type of measures;

g) *int*, the type of intervals;

h) *vec*, the type of vectors.

Further, following Reference [3], the group operator • (bullet) is introduced, which applies to a type to form a group type, e.g. the group of points, p•.

5.2.3 Functional types

Additional types can be constructed with conventional binary type constructors: → and ×. From these, the standard set of functional types is defined, e.g. $e \rightarrow t$, $v \rightarrow t$, $p \rightarrow t$. Further, a semi-lattice of types is defined, where ⊑ is a quasi-ordering on the set of types, such that, for types *a, b, c*: $a \sqsubseteq b$ and $b \sqsubseteq c$ implies $a \sqsubseteq c$; and $a \sqsubseteq a$. This introduces the subtyping relation between types: if $a \sqsubseteq b$, then *a* is a subtype of *b*.

The following typical functional types are derived with the binary type constructor →:

a) $e \rightarrow t$, the type of properties of an individual;

b) $p \rightarrow t$, the type r of regions;

c) $v \rightarrow t$, the type ε of eventuality descriptors;

d) $int \rightarrow p$, the type π of (static) paths;

e) $int \rightarrow vec$, the type π_v of vector-based paths.

Following the neo-Davidsonian semantics, "John walks" can be represented as [$walk(e) \wedge agent(e,j)$] such that "John" is annotated as being the agent of the event "walk". Here, the variable e refers to an eventuality of type v, while the verb is an eventuality descriptor denoting a predicate of type $v \rightarrow t$ or ε. The individual constant j referring to John is of type e. As for the type of static paths and vector-based paths, see 5.4.

5.3 Place and spatial entity

The "PLACE" tag is used for annotating geolocations, such as Germany and Boston, as well as geographic entities such as lakes and mountains. Further, administrative entities that are registered as geolocations are also tagged as PLACE, e.g. towns and counties. Hence, in Example 1, the qualitative spatial relation between the two entities is a relation between places. Both "Gothenburg" and "Sweden" are marked as PLACE, which is typed as "region". A region, r, is defined as a set of points, $p \rightarrow t$. This differs from Reference [3], where regions are defined as a subtype of p•, while • is a group operator over basic types, but either analysis can be adopted for these purposes.

NOTE 1 To differentiate tag names from common nouns, tag names are represented in upper case, e.g. PLACE is a tag for places like Gothenburg or Sweden.

The spatial words such as "location", "place", and "region" are all used equivalently. In annotation, they are all tagged PLACE. Semantically, they do not denote spatial points of type p. For example, Gothenburg refers to a location, place or region of type $p \rightarrow t$.

Further, a qualitative spatial mereo-topological relation within RCC8 (the Region Connection Calculus 8 qualitative spatial relations, see Reference [5]) is typed as a relation between regions: i.e. qslink: $r \rightarrow (r \rightarrow t)$ for qualitative spatial link, formulated in ISO 24617-7:2020.

EXAMPLE 1

a) $[\text{Gothenburg}]_{pl1}$ is $[\text{in}]_{s1}$ $[\text{Sweden}]_{pl2}$.

b) $[\![\text{Gothenburg}]\!] = G$, $< G: p \rightarrow t >$

c) $[\![\text{Sweden}]\!] = S$, $< S: p \rightarrow t >$

d) $[\![\text{in}]\!] = \lambda y \lambda x[\text{in}(x,y)]$, $< \text{in}: r \rightarrow (r \rightarrow t) >$

e) $\text{in}(G, S)$

For many spatial relations in language, however, the entities involved are not inherently typed as locations or places. For example, humans and everyday objects carry a primary type of e, which are subtyped or identified in these documents as spatialEntity. When they participate in spatial relations, a type coercion function, $\mathcal{L}$, is assumed to operate over an entity (or a collection of entities) and returns the spatial region associated with that entity (or entities), i.e. its location in space. The type for this localization operator, $\mathcal{L}$, is: $e \rightarrow (p \rightarrow t)$.

Example 2 demonstrates how this operator coerces an entity to the type required by the spatial relation, namely *r*.

EXAMPLE 2

a) $[\text{Robin}]_{se1}$ in $[\text{Sweden}]_{p}$.

b) ⟦Robin⟧ = *R*, < *R*: e >

c) ⟦Sweden⟧ = *S*, < *S*: *p*→ *t* >

d) $\mathcal{L}(R) = \lambda x[loc(x,R)]$, < $x: p, \mathcal{L}: e \rightarrow (p \rightarrow t)$>

e) ⟦in⟧ = $\lambda y \lambda x[\text{in}(x,y)]$, < in: $r \rightarrow (r \rightarrow t)$>

f) $\text{in}(\lambda x[loc(x,R)], S)$

NOTE 2 In Example 2 d), the predicate *loc* relates a spatial point of type *p* to an entity of type *e*. $\lambda x[loc(x,R)]$ is thus understood as an eigenplace, a region of type $p \rightarrow t$, where the entity R is located.

The interpretation of spatialEntity in terms of its localization will hold for how objects are situated in paths as is shown in 5.4.

5.4 Paths

The notion of a path being introduced or created by an event has its origin in several previous authors, including References [6], [7] and [8]. More recently and more in line with the present specification, the analysis of Reference [7] is adopted for the specification in ISO 24617-7:2020. Formally, paths have been analysed as sequences of spaces (see Reference [6]) and sequences of vectors (see Reference [8]). Following Reference [6], let *int* be the type of the interval $[0,1] \subset R$, and *p* be the type of a spatial point, as defined above. Then a path, π, is that function $int \rightarrow p$, which indexes locations on the path to values from the interval [0,1]. Similarly, if *vec* is the type of vectors, then a *vector-based path*, π_v, can be defined as the function $int \rightarrow vec$. That is, it indexes the vectors associated with the path to values from the interval [0,1].

EXAMPLE 1

a) $[\text{Prague}]_{pl1}$ is on $[\text{the Vltava River}]_{p1}$.

b) $[\text{Boston}]_{pl2}$ is at the end of $[\text{the Massachusetts Turnpike}]_{p2}$.

In Example 1, the qualitative spatial relation introduced by the predication identifies a place as situated within (or on) a path. Hence, the preposition “on” which governs the path-PP, [pp on [NP the Vltava River]], carries a more specific type than a general qslink relation, namely: $\pi_v \rightarrow (r \rightarrow t)$. The type derivation for Example 1 a) is illustrated in Example 2.

EXAMPLE 2

a) Prague_{pl1} is on $[\text{the Vltava River}]_{p1}$.

b) ⟦Prague⟧ = *P*, < *P*: *p*→ *t* >

c) ⟦the Vltava River⟧ = *M*, < *M*: π_v >

d) ⟦on⟧ = $\lambda y \lambda x[\text{onPath}(x,y)]$, < onPath: $\pi_v \rightarrow (r \rightarrow t)$>

e) onPath(*P*, *M*)

NOTE Rivers and roads both form paths. Unlike roads, rivers flow and the objects on them also flow, thus forming dynamic paths.

As Example 1 b) illustrates, the endpoints of paths can be explicitly mentioned in text. Examples 3 a) and 3 b), which are annotated in accordance with ISO 24617-7:2020, demonstrate reference to both endpoints and mid-points.

EXAMPLE 3

a) ... the railroad$_{pl}$ between Boston$_{pl1}$ and [New York]$_{pl2}$ ...

Annotation structure: path(p1, beginID = pl1, endID = pl2, form = NOM)

b) John took the road$_{p2}$ through Boston$_{pl1}$.

Annotation structures: path(p2, midIDs = pl1, form = NOM)

Formally, the expressions introducing end- and mid-point locations are acting as functions from paths to path positions: $\pi_v \rightarrow int$; e.g. given a path < 3,4,5,2,1,8> , $end(\pi_v) = 8$.

EXAMPLE 4

a) Boston$_{pl1}$ is at the end of [the Massachussets Turnpike]$_{p1}$

b) $[\![\text{Boston}]\!] = B$, $< B: p \rightarrow t >$

c) $[\![\text{the Massachusetts Turnpike}]\!] = MT$, $< MT: \pi_v >$

d) $[\![\text{end}]\!] = \lambda x[\text{endOf}(x)]$, $< x: \pi_v, \text{endOf}: \pi_v \rightarrow int >$

e) $[\![\text{on}]\!] = \lambda y \lambda x[\text{onPath}(x,y)]$, $< \text{onPath}: \pi_v \rightarrow (r \rightarrow t)>$

f) $\text{onPath}(B,MT) \wedge \text{endOf}(MT) = B$

As mentioned above, the localized spatialEntity can be situated on a path by coercion: namely, $\mathcal{L}$ coerces an entity referred to by the name "John" to his localized place or path, called "eigenplace", and then the spatial relation predication situates this region onto the path, π_v.

EXAMPLE 5

a) [John]$_{se1}$ is on [the road]$_{p1}$.

b) $\mathcal{L}(J) = \lambda x[loc(x, J)]$, $< x: p, \mathcal{L}: e \rightarrow (p \rightarrow t)>$

6 Events and paths generated from events

6.1 General

The term "event" as it is used in ISO 24617-7:2020 is borrowed directly from ISO 24617-1:2012 and is used as a cover term for situations that happen, occur, hold or take place. Following References [9] and [10], the event can be represented as an individual predicated of an event class (the verb), where the arguments are then related by semantic role relations. It has further been proposed that there is an internal structure to events which structurally differentiates the Aktionsarten (action types) of Vendler's classes. This has come to be known as "event structure"[11][12]. On this theory, the subevent structure of the event is explicitly represented in the lexical semantics and subsequent compositional interpretations, giving rise to three basic event structures: state, process and transition. The event tag captures TimeML events, as specified in ISO 24617-1, that are related to another spatial entity by way of a link tag (e.g. a spatial anchoring such as "sleeping in the courtyard").

The motion tag, on the other hand, identifies those events involving movement of an object through space. All motion tags participate in a movelink relation, as specified in ISO 24617-7:2020.

6.2 Two types of verb constructions

There are two basic strategies that languages typically exploit to convey the movement of an object through space[13]: path verb constructions and manner verb constructions.

EXAMPLE 1

a) Path-involving motion: “John arrived at home”.

b) Manner-involving motion: “John walked”.

In terms of their event structure, path-verbs are transitions while manner-verbs are processes. In addition, path verbs are those predicates that presuppose a specific path for the moving object (the figure), along with a possible distinguished point or region on this path (the ground), which the figure is moving toward or away from. Manner verbs can be seen as creating a path as the motion event unfolds. This is illustrated formally in Example 2.

EXAMPLE 2

a) Path-presupposing verb (with temporal anchor):

$\lambda y\, \lambda x\, \lambda i\, \lambda e\, \exists\{e1, e2, p\}[@i\ \mathrm{arrive}(e) \wedge \mathrm{arrive\_act}(e1, x, p) \wedge DC(e1, x, y) \wedge \mathrm{arrive\_result}(e2, x, p) \wedge EC(e2, x, y) \wedge \mathrm{end}(y, p) \wedge e1 = e1 \circ e2 \wedge e1 \leq e2 \wedge e1 \preccurlyeq e \wedge e2 \preccurlyeq e\,]$

b) Path-introducing verb (with temporal anchor):

$\lambda x\, \lambda p\, \lambda i\, \lambda e[@i\ \mathrm{walk}(e) \wedge \mathrm{walk\_act}(e, x, p)]$

Example 2 a) is a case of an eventuality of type transition, whereas Example 2 b) is a case of an eventuality of type process or activity. Each part of Example 2 a) is explained as follows:

a) @*i* is a temporal anchoring function meaning that some event *e* occurs at time *i*: @*i* arrive(*e*) thus means that the whole event *e* of arriving occurs at time *i*;

b) arrive_act(e_1, *x*, *p*) means that e_1 refers to the activity or process of arriving such that *x* is the object that moves and *p* is the path that x traverses (through);

c) $DC(e_1, x, y)$ and end(*y*, *p*) together mean that the mover *x* of e_1 or at the stage e_1 is disconnected (*DC*) from the end *y* of the event-path *p*, thus meaning that the mover has not arrived at the destination;

d) [arrive_result(e_2, *x*, *p*) ∧ $EC(e_2, x, y)$ ∧ end(*y*, *p*)] means that e_2 is the result of the whole process or arriving such that *x* is the mover and *p* is the event-path, the mover *x* is now externally connected (*EC*) to the end point *y*;

e) $[e = e_1 \circ e_2 \wedge e_1 \leq e_2 \wedge e_1 \preccurlyeq e \wedge e_2 \preccurlyeq e]$ means that *e* consists of e_1 and e_2, e_1 precedes e_2, e_1 and e_2 are parts of *e*.

NOTE *DC* and *EC* in Example 1 a) are two of the RCC relations, representing two-dimensional qualitative spatial relations (see Reference [14]).

Path predicates make the change of location explicit in the subevent representation (see Reference [15]). This states that the figure *x* moves along a path *p*, represented by the event *e*. This entails a transition from not being at the ground e_1 to finally being at the ground e_2. It further gives the necessary temporal constraints along with the constraint that the ground must be the termination of the path.

The type of the path variable *p* introduced above is no different than that used in the examples in Clause 5, namely π_v or *int* → *vec*. The difference, however, is that there is no lexical offset (markable) in the sentences in Example 2, which can be associated with this path.

The concept of “path” is distinguished from the motion-dependent notion of an “event-path”. A path is a component part of the domain of space (or a vector space), whereas an event-path is that region of space occupied by a mover throughout an event.

6.3 Typing event-paths

An event-path can be typed as that path which is associated with an object over time. Assuming that the moving object *x* can be represented spatially as its eigenplace $\mathcal{L}(x)$, the trace of the path created by *x* is typed as an event-path. Event-paths are treated as a functional type π_v or *int* → *vec*, the function from intervals to vectors, thus being dynamic paths so that objects on them move as time moves on. This is a function from time-intervals to paths.

NOTE 1 Reference [16] following Reference [17] argues that the metamodel and its associated semantics become much simpler if the concept of event-path is reintroduced as a basic element in the model. The notion of event-path with a distinct tag is thus incorporated into ISO 24617-7:2020.

NOTE 2 Paths can be seen minimally as a vector of two points, (x, y).

7 Semantic interpretation of annotation structures

7.1 Overview

Clause 7 demonstrates how the concrete syntax of ISO 24617-7:2020, as deployed over a natural language example, receives an intermediate semantic interpretation, which can then be subsequently interpreted in a model.

7.2 Semantic forms

Given an annotation structure a, its semantic form σ(a) is represented in a type-theoretic first-order logic. The semantic form generator σ is a function from annotation structures, specified by the abstract syntax of an annotation scheme, independent of how these input annotation structures are represented. In practice, it applies to a concrete representation of a, for instance, in the generalized extensible markup language (XML). Instead of XML, a predicate-logic-like format, called "pFormat", is adopted for representing annotation structures.

NOTE Semantic forms are represented in a first-order logic but use second-order variables such as P to allow the lambda operation over it, e.g. $\lambda P P(x)(\sigma(x1))$, which turns into a first-order formula of type t, as illustrated in Example 2.

EXAMPLE 1

a) Data: John arrived in Gothenburg.

b) Word-segmented: John_w1 arrived_w2 in_w3 Gothenburg_w4.

c) Annotation structures represented:

- Annotation (is1, language="en", aScheme="ISO 24617-7:2020")
- *Entity structures*
- entity(x1, w1, type="person", form="nam")
- motion(m1, w2, type="transition", tense="past")
- sRelation(sr1, w3, type="endPt-defining")
- place(pl1, w4, type="ppl", form="nam", ctv="city", country="sw")
- eventPath(ep1, endpoint="pl1")
- *Link structure*
- movelink(mvL1, reltype="traverses", figure="x1", ground="ep1", trigger="m1")

Based on Example 1 c), semantic representations can be obtained, as in Example 2.

EXAMPLE 2

a) Semantic representation of the entity structures

$\sigma(\mathrm{x1}) := \lambda x_1[named(x_1,John) \wedge person(x_1)]$

$\sigma(\mathrm{m1}) := \lambda e_1[arrive(e_1) \wedge past(e_1)]$

$\sigma(\mathrm{pl1}) := \lambda l_1[named(l_1,Gothenburg) \wedge city(l_1) \wedge in(l_1,sw)]$

$\sigma(\mathrm{ep1}) := \lambda p_1[route(p) \wedge starts(p_1, < l_0,i_0 >) \wedge ends(p_1, < l_1,i_1 >)]$

b) Semantic representation of the link structure

$\sigma(\mathrm{mvL1}) := [[mover(x,e) \wedge \lambda PP(x)(\sigma(x1))] \wedge [\lambda PP(p)(\sigma(\mathrm{ep1})) \wedge traverses(x,p)]]$

$:= [mover(x,e) \wedge [named(x,John) \wedge person(x)] \wedge [route(p) \wedge starts(p, < l_0,i_0 >) \wedge$
$ends(p, < l_1,i_1 >)] \wedge traverses(x,p)]$

c) Semantic representation of the whole annotation structure

$\sigma(\mathrm{is1}) := [\sigma(\mathrm{movelink}) \oplus [\sigma(\mathrm{x1}) \oplus \sigma(\mathrm{m1}) \oplus \sigma(\mathrm{pl1}) \oplus \sigma(\mathrm{ep1})]]$

$:= [[[mover(x,e) \wedge \lambda PP(x)(\sigma(\mathrm{x1}))] \wedge [\lambda PP(p)(\sigma(\mathrm{ep1})) \wedge traverses(x,p)]] \wedge$

$:= [mover(x,e) \wedge [named(x,John) \wedge person(x)]$

$[arrive(e) \wedge past(e)] \wedge$

$[named(l_1,Gothenburg) \wedge city(l_1) \wedge in(l_1,sw)] \wedge$

$[route(p) \wedge starts(p, < l_0,i_0 >) \wedge ends(p, < l_1,i_1 >)]]$

By existentially binding the free occurrences of the variables:

$:= \exists\{x, e, p, l_0, l_1, i_0, i_1\}$

$[[mover(x,e) \wedge named(x,John) \wedge person(x)] \wedge$

$[arrive(e) \wedge past(e)] \wedge$

$[named(l_1,Gothenburg) \wedge city(l_1) \wedge in(l_1,sw)] \wedge$

$[route(p) \wedge starts(p, < l_0,i_0 >) \wedge ends(p, < l_1,i_1 >)] \wedge$

$traverses(x,p)]$

Table 1 — Semantics based on the abstract syntax

John arrived in Gothenburg.				
Syntax			**Semantics**	
ID	**TARGET**	**ANNOTATION**	**semTYPE**	**semFORM**
x1	John	`type="person"`	*x: e* (entity)	*named*(*x,John*) *person*(*x*)
m1	arrived	`tense="past"`	*e: event* (event)	*arrive*(*e*) *past*(*e*)
s1	in			
pl1	Gothenburg	`ctv="city"`	*l: r* (region)	$named(l_1,Gothenburg)$ $city(l_2)$
ep1	Ø	`start="unknown"` `end="pl1"`	*p*: π_v (path)	$starts(p,<l_1,i_1>)$ $ends(p,<l_2,i_2>)$
	.	.	*t* (truth-value)	***Φ***

mvL1		`figure="x1"`
		`ground="ep1"`
		`relType="traverses"`
				mover(*x,e*)
				route(*p*)
				traverses(*x,p*)

The semantic representations a) and b) in Example 2, as in Table 1, show how each of the annotation structures is translated through the interpretation function σ into a first-order expression. Being a complex structure with IDREFs for the values of its attributes, the link structure has extra λP-expressions each of which allows a required variable adjustment. The free occurrences of the variables are bound by the existential quantifier, ∃.

The semantic form of Example 2 is that of the entire annotation structure (is1), compositionally obtained from the list of the semantic forms of the entity and link structures, which are given as a) and b). It is then interpreted with respect to a model that represents part of the world to which the annotated linguistic expressions are denotationally anchored.

7.3 Model theory

7.3.1 General

A model theory specifies ways of interpreting the semantic forms σ(a) of annotation structures a with respect to a model M. It also specifies a model M as a set-theoretic construct that represents part of the real or possible world to which a list of linguistic expressions or their semantic forms are referentially or denotatively anchored. Subclause 7.3.2 presents several examples that involve spatial expressions and motion predicates to illustrate the semantics outlined in 7.2. In particular, attention is paid to:

— static spatial relations between entities and locations: in, under, on;

— predicates involving translocation (motion) of the theme argument that have a directional preposition phrase as adjunct: walk to the corner, drive to the store, swim across the lake;

— predicates involving translocation (motion) of the theme argument with no spatial prepositional or stative prepositional phrases as adjunct: leave Boston, arrive in Gothenburg.

The model consists of the entities given in 5.2.2 and 5.2.3, sorted by type.