Sensor Data Assimilation as Database Transactions

Marcus V. Santos *
m3santos@scs.ryerson.ca
School of Computer Science
Ryerson University
350 Victoria Street M5B 2K3 Toronto Canada

Paulo E. Santos†
p.santos@ic.ac.uk
Dep. of Electrical and Electronics Engineering
Imperial College of Science, Technology and Medicine
Exhibition Road SW7 2AZ London UK

Abstract

This paper extends a logic-based framework for robot scene interpretation by using the notion of path semantics. The main contributions of this approach are two fold. First, a logic language designed to account for the phenomenon of state changes in databases is extended with the concept of abduction and further applied on the task of robot sensor-data assimilation. Secondly, the present framework provides the theoretical foundations for symbolically interpreting long sequences of transitions of sensor data. In practice, each snapshot taken by the robot camera represents a database state. Transitions (*i.e.*, changes) between consecutive snapshots are modeled by changes in the database and interpreted by a state transition oracle, which encodes axioms about commonsense spatial reasoning.

Introduction

In this work, we assume a mobile robot whose sensor system includes a built-in camera that takes chronological sequences of snapshots of the environment. We also assume that potential changes occurred in this environment are represented by differences between image regions in consecutive camera snapshots. Our objective is to provide a logical account which allows to interpret the sensor data in terms of predicates representing common-sense concepts about space. To this aim, in this work we extend the logic-based formalism for representing knowledge about objects in space and their movement proposed in (Santos & Shanahan 2002), and show how to build up this knowledge from long sequences of snapshots noted by the robot's vision system.

The use of logic-based frameworks for image interpretation has been advocated many times in the literature. Perhaps the first such system was proposed in (Reiter & Mackworth 1990), where the task of interpreting a sketch map is defined as a model building process with respect to a first-order theory representing both, the relevant domain knowledge and a description of the map. Motivated by this work, a knowledge-based aerial image understanding system was proposed in (Matsuyama & Hwang 1990). This system interprets aerial images by generating hypotheses according to a first-order theory on a diagnostic setting (Poole, Goebel, & Aleliunas 1987); (Schroeder & Neumann 1996) extends this framework assuming a language for describing concepts. Closer to the present work is the abductive account for sensor data interpretation proposed in (Shanahan 1996). The idea there was to supply a logical account of the transition from a robot's raw sensor data to symbols denoting the existence, location and shapes of objects. These earlier works, however, assume the interpretation of static scenes described in an absolute frame of reference. On the other hand, the work presented in (Santos & Shanahan 2002) assumes the changes in a dynamic world, represented from the viewpoint of an observer, as the central element for sensor data interpretation. This work, however, falls short on interpreting paths of snapshots, rather than pairs. A solution to this issue is proposed in the present work.

Here we propose a formalism, called $\mathcal{T}$-logic, which allows the interpretation of sequences of sensor data. $\mathcal{T}$-logic is an instance of the Transaction Logic ($\mathcal{TR}$) (Bonner & Kifer 1993), a general logic of state change that accounts for the phenomenon of updating arbitrary logical theories.

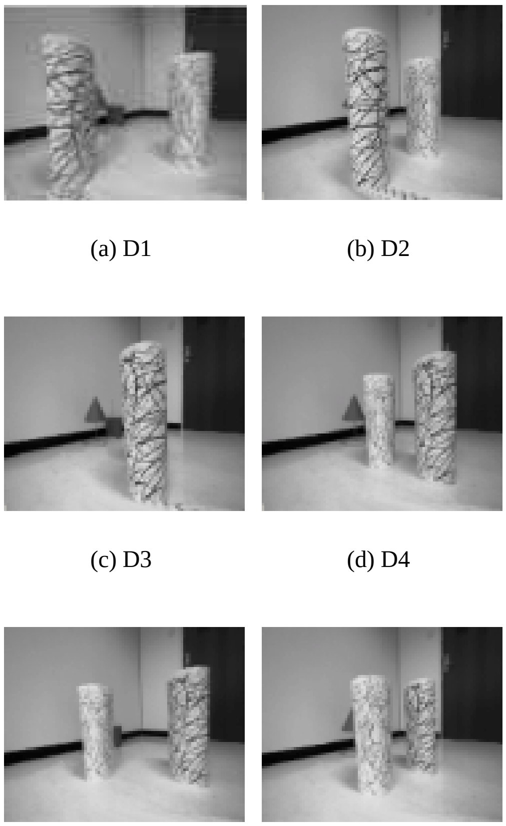
We first briefly present the essentials of $\mathcal{TR}$'s syntax and semantics. We then define $\mathcal{T}$-logic, a specialized theory for specifying dynamic transitions in object spatial relations that may occur in chronological sequences of camera snapshots. This specialized theory is based on a pair of oracles, called *state oracle* and *transition oracle*. A state, in this context, is the visual information contained in a camera snapshot. The state oracle informs us what spatial relations are true in a state. The transition oracle informs us what transitions in spatial regions explain the difference between two consecutive states. These oracles serve as the connector with the underlying language.

Next, using a Horn-style subset of the logic we define an inference engine based on a SLD-style proof procedure extended with the notion of abduction discussed in (Shanahan 1996). Finally, based on this inference engine, we define an abductive mechanism which, given a sequence of states and some background theory specifying properties of the spatial relations of objects, it informs us what transitions in spatial relations explain the differences in the image regions across sequences of states.

* Supported by NSERC
† Supported by CAPES.

Motivating Example: interpreting sequences of snapshots

Intuitively, the purpose of the framework presented in this paper is to interpret, by common sense concepts, sequences of snapshots of the world such as those represented in Figure 1.



(a) D1 (b) D2

(c) D3 (d) D4

(e) D5 (f) D6

Figure 1: A chronological sequence of snapshots.

For instance, assuming that the two cylindrical regions in Figure 1 are represented by the symbols a and b, the transition from $D1$ to $D2$ above can be interpreted as 'the spatial regions a and b are *approaching* each other". Similarly, the transition from $D2$ to $D3$ can be intuitively interpreted as "the spatial regions a and b are *coalescing*", from $D3$ to $D4$ as "the spatial regions a and b are *splitting*", from $D4$ to $D5$ as "a and b are *receding* from each other" and from $D4$ to $D5$ as "a and b are *approaching* each other". In a broader sense, the transitions from $D1$ to $D6$ can be intuitively interpreted as "a and b are *rotating around* each other".

As a matter of fact, interpreting such pairs of transitions was the purpose of the work described in (Santos & Shanahan 2002). As we shall see, in this paper we develop a logic language whose inference engine allows the interpretation of the sequences of snapshots with any number of transitions defining it.

Preliminaries

In this section we introduce the basic definitions of the underlying language. These definitions recall those described in (Bonner & Kifer 1993) and (Santos & Shanahan 2002).

Syntax In this paper we assume a many-sorted first-order language with sorts for spatial regions, and real numbers. The syntax includes also two infinite, enumerable sets of symbols: a set of function symbols and a set of predicate symbols. Constants, propositions, and terms are defined as usual in first-order logic. We adopt the Prolog convention that variables begin in upper case, and predicate symbols in lower case. Atomic formulae are also called *atoms*. The logic extends first-order logic with a new connective, $\otimes$, called *serial conjunction*. As a matter of fact, given two first order formulae a and b, $a \otimes b$ means intuitively that a is evaluated *before* b.

Programs: Like classical logic, the underlying language has a Horn-like fragment (called *serial-Horn* (Bonner & Kifer 1993)) with both a procedural and a declarative semantics. Serial-Horn rules are formulae of the form $p \leftarrow a_1 \otimes a_2 \otimes \cdots \otimes a_n$, where each a_i is an atomic formula ($a_i \otimes \cdots \otimes a_n$ is called a *serial goal*). Intuitively, this formula means "to compute p, it is sufficient to compute $a_1 \otimes a_2 \otimes \cdots \otimes a_n$." This procedural interpretation is very important because it provides a subroutine facility that makes logic programming possible.

Semantics Formulae are interpreted on *paths* (*i.e.*, sequence of states), as in Process Logic (Harel, Kozen, & Parikh 1982). A path represents a history of elementary changes on the world, and formulae represent what is true during periods of history. Classical connectives have their usual interpretations, except that they are interpreted on paths. For instance, $\alpha \wedge \beta$ means that α and β are both true on a path. The non-classical connectives allow a formula to relate to parts of a path. For instance, $\alpha \otimes \beta$ means that a given path can be split in two, where α is true on the prefix of the path and β is true on the suffix. Hence, it is convenient to define a *split* of a path π to be any pair of sub-paths, π_1 and π_2, such that $\pi_1 = \langle \mathbf{D}_1 \cdots \mathbf{D}_i \rangle$ and $\pi_2 = \langle \mathbf{D}_{i+1} \cdots \mathbf{D}_n \rangle$, where $\mathbf{D}_i, 1 \leq i \leq n$, is a state. In this case we shall write $\pi = \pi_1 \circ \pi_2$. Moreover, (Bonner & Kifer 1993) defines $\mathtt{state}$ as a constant that is true on any state, *i.e.*, a path of length 1. Hence,

$$^k \otimes \mathtt{state} \equiv \underbrace{\mathtt{state} \otimes \cdots \otimes \mathtt{state}}_{k}$$

denotes a formula which is true on any path of length k, *i.e.*, a path containing $k+1$ states.

Abduction In brief, abduction is an inference rule that has a form of an inverse *modus ponens*. Therefore, abductive reasoning can be intuitively understood as reasoning backwards from consequent to antecedent. Reasoning with abduction was first proposed by Peirce as *the inference that rules the first stage of scientific inquiries and of any interpretive process* (Peirce 1958). Abductive reasoning has been applied in many fields of A.I., as mentioned in (Kakas, Kowalski, & Toni 1998).

Abduction is the process of explaining a set of sentences Γ by finding a set of formulae Δ such that, given a background theory Σ, Γ is a logical consequence of $\Sigma \cup \Delta$ (Shanahan 1996). In order to avoid trivial explanations, a set of predicates is distinguished (the *abducible predicates*) such that every acceptable explanation must contain only these predicates.

Motivating examples

Next we present some examples of the underlying language syntax.

Simple Formulae: Suppose $approaching(a, b)$ means "object a is approaching object b", and $static(a, b)$ means "objects a and b are static". Then, we can write formulae of the form below.

$$approaching(a, b) \vee approaching(a, c)$$

means "object a is approaching object b or object c".

$$(\exists\, X)\ approaching(X, b)$$

means "some object, X, is approaching object b".

$$\neg(\exists\, XY)\ static(X, Y)$$

means "no object is static".

$$approaching(a, b) \otimes static(a, b)$$

means "object a is approaching object b, and then they are static."

Serial-Horn rules: An object A is avoiding another object B if A initially is approaching B, and then object A is receding from B. Formally,

$$avoiding(X, Y) \leftarrow \quad approaching(X, Y) \otimes receding(X, Y)$$

An object A is oscillating w.r.t object B if it "avoids" B once, or more times. Formally:

$$oscillating(X, Y) \leftarrow avoiding(X, Y)$$
$$oscillating(X, Y) \leftarrow avoiding(X, Y) \otimes oscillating(X, Y)$$

Defining a specialized theory

In this section we introduce an instance of the language proposed in (Bonner & Kifer 1993), which we call $\mathcal{T}$-logic. Our aim is to define a specialized theory which allows us to specify, prove, and reason about the object spatial relations based on sensor data consisting of a chronological sequence of snapshots taken by the robot camera.

$\mathcal{T}$-logic consists of a pair of oracles "plugged into" Transaction Logic, an inference system based on a SLD-style proof procedure, and an abductive mechanism to provide explanations for the object relations in the sensor data.

The oracles

The general logic proposed in (Bonner & Kifer 1993) does not commit to a particular semantics of database state. One can think of this language as a logical framework, which can be instantiated as specific logics in many ways. In Transaction Logic, a pair of oracles, called *state* and *transition oracles*, isolates elementary database operations from the logic used for combining and programming with them. The state oracle specifies a set of primitive state queries, *i.e.*, the *static* semantics of states; and the other specifying a set of primitive state *transitions*, *i.e.*, the *dynamic* semantics of states. These oracles separate the specification of elementary operations from the logic of combining them. In the present framework the state data oracle encodes the definitions used to translate the sensor data of the robot into logic predicates, while the state transition oracle comprises the set of hypotheses to assimilate the transitions in the sensor data descriptions into the robot's set of beliefs.

Formally, a *state* is a set of spatial regions as noted by the robot's sensors. To refer to states, we assume a countable collection of symbols, called *state identifiers.* We also assume a function $dist : S \times S \times D \rightarrow \Re$, where S is the set of objects, D the set of state identifiers, and $\Re$ is the set of real numbers. Function $dist/3$ defines the length of the shortest line connecting any two points in two object boundaries in a state. Thus, $dist(x, y, \mathbf{D})$ means 'the distance between objects x and y in state **D**'. Moreover, we assume the symbol δ represents a pre-defined distance value.

As seen in figure 2, the $dist/3$ function helps us identify three dyadic relations on objects: $dc(x, y)$, 'x is disconnected from y'; $ec(x, y)$, 'x is externally connected to y' ; and $co(x, y)$, 'x is coalescent with y', as defined in (Santos & Shanahan 2002). This set of relations on spatial regions are inspired by the Region Connection Calculus (Randell, Cui, & Cohn 1992).

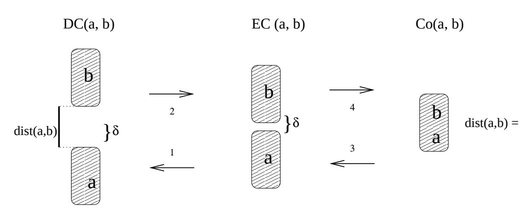


Figure 2: Relations on regions and the continuous transitions between them.

The state oracle Given a state, the objective of this oracle is to inform what dyadic relations hold for each pair of objects in a state. Therefore, the state oracle provides a mapping from states to ground instances of the above dyadic relations, as follows:

Definition 1 (State oracle) For any pair of spatial regions x and y in a snapshot of the world **D**, the state data oracle, $\mathcal{O}^d$, defines a mapping from x and y to one (and only one) dyadic relation dc, ec or co, as follows:

$$dc(x, y) \in \mathcal{O}^d(\mathbf{D}) \leftrightarrow dist(x, y, \mathbf{D}) > \delta$$
$$ec(x, y) \in \mathcal{O}^d(\mathbf{D}) \leftrightarrow dist(x, y, \mathbf{D}) \leq \delta \wedge dist(x, y, \mathbf{D}) \neq 0$$
$$co(x, y) \in \mathcal{O}^d(\mathbf{D}) \leftrightarrow dist(x, y, \mathbf{D}) = 0$$

□

In this way, the state oracle provides a built-in view to the spatial relation holding on any given pair of objects in the state.

The state transition oracle Given two consecutive snapshots taken by the robot's camera, the objective of this oracle is to inform what transitions in spatial relations explain the difference between the snapshots. To represent transitions in spatial relations we use the following *transition predicates*, as defined in (Santos & Shanahan 2002).

- $approaching(x, y)$, 'the objects x and y are approaching each other';
- $receding(x, y)$, 'the objects x and y are receding from each other';
- $coalescing(x, y)$, 'the objects x and y are coalescing';
- $splitting(x, y)$, 'the objects x and y are splitting'; and
- $static(x, y)$, 'the objects x and y are static'.

Therefore, since a snapshot represents a state, the state transition oracle provides a mapping from pair of states to grounded transition predicates, as follows:

Definition 2 (State transition oracle) Let $\mathbf{D}_1$ and $\mathbf{D}_2$ be states, and a and b be objects. Then,

- $approaching(a, b) \in Ot(\mathbf{D}_1, \mathbf{D}_2)$ `iff` $dc(a, b) \in \mathbf{D}_1 \wedge co(a, b) \notin \mathbf{D}_2 \wedge dist(a, b, \mathbf{D}_1) > dist(a, b, \mathbf{D}_2)$
- $coalescing(a, b) \in Ot(\mathbf{D}_1, \mathbf{D}_2)$ `iff` $(ec(a, b) \in \mathbf{D}_1 \vee dc(a, b) \in \mathbf{D}_1) \wedge co(a, b) \in \mathbf{D}_2$
- $splitting(a, b) \in Ot(\mathbf{D}_1, \mathbf{D}_2)$ `iff` $co(a, b) \in \mathbf{D}_1 \wedge (ec(a, b) \in \mathbf{D}_2 \vee dc(a, b) \in \mathbf{D}_2)$
- $receding(a, b) \in Ot(\mathbf{D}_1, \mathbf{D}_2)$ `iff` $(ec(a, b) \in \mathbf{D}_1 \vee dc(a, b) \in \mathbf{D}_1) \wedge dist(a, b, \mathbf{D}_1) < dist(a, b, \mathbf{D}_2)$

□

In this way, the state transition oracle provides a built-in view to the transitions between spatial relations that have occurred during two consecutive states.

It is worth pointing out that, in practice, $co/2$ cannot be measured directly, but inferred from the case in which two externally connected regions reduce the distance separating them, i.e. two externally-connected regions approaching each other, from the robot's viewpoint.

Implementing the oracles We assume each snapshot taken by the robot camera is time-stamped and the respective sensor data is somehow stored in the computer memory. Also, we assume that image regions are properly labeled. Notice that in this setting, a state identifier actually represents a point in time, *e.g.*, $\mathbf{D}_0$ is the first snapshot taken, $\mathbf{D}_1$ is the second, and so on.

The state oracle is implemented as a procedure that determines the distance between two objects at a given state (*i.e.*, time stamp), compares it with the predefined δ threshold, and determines which of the aforementioned dyadic relations on objects hold.

Analogous to the state oracle, the transition oracle is implemented as a procedure that, given two states and two objects, finds which of the transition predicates hold for those objects in those states.

An inference system with transactions and abduction

The abductive approach to explanation introduced in this work can be realized using a mechanism which is an extension of a SLD-style resolution procedure. The inference system we use differs from the inference system introduced in (Bonner & Kifer 1993) in the sense that in (Bonner & Kifer 1993) the path is a byproduct of the refutation procedure, whereas here the path is given; the refutation procedure determines the unsatisfiability of a formula in the given path.

SLD-style resolution The inference system manipulates expressions called *sequents*, which have the form

$$\mathbf{P}, \pi_1 \circ \pi_2 \vdash (\exists)\, \phi$$

where $\mathbf{P}$ is a set of serial-Horn rules, $\pi_1 = \langle \mathbf{D}_1 \cdots \mathbf{D}_i \rangle$ and $\pi_2 = \langle \mathbf{D}_{i+1} \cdots \mathbf{D}_n \rangle$ are splits of a given path π, representing a sequence of states, and ϕ is a serial-goal. The informal meaning of such a sequent is that the formula $(\exists)\, \phi$ can be proved from state $\mathbf{D}_i$ and along π_2, *i.e.*, from the last state of the path split π_1 and along π_2.

Let the goal clause be the expression

$$\leftarrow G_0$$

where G_0 is the sequent $\mathbf{P}, \langle \mathbf{D}_1 \rangle \circ \langle \mathbf{D}_2 \cdots \mathbf{D}_n \rangle \vdash (\exists)\, \phi$ (1)

A SLD-style refutation of $\leftarrow G_0$ is a sequence of goal clauses $\leftarrow G_0 \cdots \leftarrow G_n$ where G_n is the *empty clause*, *i.e.*, the sequent $\mathbf{P}, \langle \mathbf{D}_1 \cdots \mathbf{D}_n \rangle \circ \langle\, \rangle \vdash (\,)$, where $\langle\, \rangle$ denotes the empty path, and $(\,)$ denotes the empty formula. This sequent is an axiom of the inference system, and this axiom states that the empty formula is true on any path. Each $\leftarrow G_{i+1}$ is obtained from $\leftarrow G_i$ by using the following axiom and inference rules

Axiom: $\mathbf{P}, \pi \vdash (\,)$, for any path π

Inference rules: In rules 1-3, σ is a substitution, a and b are atomic formulae, and ϕ and $rest$ are serial goals.

1. *Applying rule definitions:*
   Suppose $a \leftarrow \phi$ is a rule in $\mathbf{P}$ whose variables have been renamed so that the rule shares no variables with $b \otimes rest$. If a and b unify with mgu σ, then
   $$\frac{\mathbf{P}, \pi_1 \circ \pi_2 \vdash (\exists)\, (\phi \otimes rest)\sigma}{\mathbf{P}, \pi_1 \circ \pi_2 \vdash (\exists)\, (b \otimes rest)}$$
2. *Querying the world state:*
   If $b\sigma$ and $rest\sigma$ share no variables, and $\mathcal{O}^d(\mathbf{D}_i) \models^c (\exists) b\sigma$, then
   $$\frac{\mathbf{P}, \pi_1 \circ \pi_2 \vdash (\exists)\, rest\sigma}{\mathbf{P}, \pi_1 \circ \pi_2 \vdash (\exists)\, (b \otimes rest)}$$
   where $\pi_1 = \langle \mathbf{D}_1 \cdots \mathbf{D}_i \rangle$.

3. *Verifying a state transition:*
If $b\sigma$ and $rest\sigma$ share no variables, and $\mathcal{O}^t(\mathbf{D}_i, \mathbf{D}_{i+1}) \models^c (\exists)b\sigma$ or $b = \mathtt{state}$, then

$$\frac{\mathbf{P}, \pi'_1 \circ \pi'_2 \vdash (\exists)\, rest\sigma}{\mathbf{P}, \pi_1 \circ \pi_2 \vdash (\exists)\, (b \otimes rest)}$$

where $\pi_1 = \langle \mathbf{D}_0 \cdots \mathbf{D}_i \rangle$, $\pi_2 = \langle \mathbf{D}_{i+1}\mathbf{D}_{i+2} \cdots \mathbf{D}_n \rangle$,$\pi'_1 = \langle \mathbf{D}_0 \cdots \mathbf{D}_i\mathbf{D}_{i+1} \rangle$, and $\pi'_2 = \langle \mathbf{D}_{i+2} \cdots \mathbf{D}_n \rangle$

Each inference rule consists of two sequents, and has the following interpretation: if the upper sequent (G_{i+1}) can be inferred, then the lower sequent (G_i) can also be inferred.

The inference rules capture the roles of serial-Horn rules, the state oracle and the transition oracle, as follows:

Rule 1: It deals with serial-Horn rule definitions. Intuitively, this rule replaces an instance of the rule 'head' by an instance of its body. Notice that the path split does not change.

Rule 2: It deals with state tests. It says that a condition b satisfied in $\mathbf{D}_i$ can be added to the front of the formula $rest$. Notice that the path split does not change.

Rule 3: It deals with tests on a pair of states (*i.e.*, the last state of π_1 and the first state of π_2). Intuitively, b is attached to the front of $rest$, so that the first state of π_2 is removed from the path split and added to the end of π_1, *i.e.*, the formula can now be proved from $\mathbf{D}_i$ and in the path split $\langle \mathbf{D}_{i+1} \cdots \mathbf{D}_n \rangle$.

Based on the above, notice that, in a sequent, the leftmost atom of the serial-goal is always selected.

Now suppose we have the goal clause, $\leftarrow G_0$, where G_0 is the sequent[1]

$$\mathbf{P}, \langle \mathbf{D}_1 \rangle \circ \langle \mathbf{D}_2 \cdots \mathbf{D}_n \rangle \vdash \ ^{n-1}\!\otimes \mathtt{state} \qquad (2)$$

If we are searching for a serial-goal Δ, which is true in the same path π as $^{n-1}\!\otimes \mathtt{state}$, *i.e.*,

$$\mathbf{P}, \pi \models \ ^{n-1}\!\otimes \mathtt{state} \wedge \Delta,$$

then clearly Δ can be represented as a serial conjunction $a_1 \otimes \cdots \otimes a_{n-1}$, where each a_i is a transition predicate, *i.e.*, $a_i \in \mathcal{O}^t\mathbf{D}_k, \mathbf{D}_{k+1})$, $(1 \leq k < n)$. Moreover, Δ can be seen as an explanation for the state transitions in π. For instance, suppose $\pi = \langle \mathbf{D}_1\mathbf{D}_2\mathbf{D}_3 \rangle$, where

$$\begin{aligned}
&\mathcal{O}^d(\mathbf{D}_1) = dc(a,b)\\
&\mathcal{O}^d(\mathbf{D}_2) = dc(a,b)\\
&\mathcal{O}^d(\mathbf{D}_3) = co(a,b)\\
&\mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2) = approaching(a,b)\\
&\mathcal{O}^t(\mathbf{D}_2, \mathbf{D}_3) = coalescing(a,b)\\
&\Delta = approaching(a,b) \otimes coalescing(a,b)
\end{aligned}$$

Notice that Δ means: "initially a was approaching b, and then a was coalescing with b". That is, Δ explains the transitions in π. The mechanism for obtaining Δ consists of the following extension to the SLD-style resolution.

[1]Recall that $^{k}\!\otimes \mathtt{state} \equiv \underbrace{\mathtt{state} \otimes \cdots \otimes \mathtt{state}}_{k}$

An abductive mechanism for transactions The transition predicates defined by the transition oracle, $\mathcal{O}^t$, are designated as abducibles. Given a path π, to find a serial-goal Δ_n such that $\mathbf{P}, \pi \models \ ^{n-1}\!\otimes \mathtt{state} \wedge \Delta_n$ and Δ_n mentions only abducibles, a refutation of the form $\leftarrow G_0, \Delta_0 \cdots \leftarrow G_n, \Delta_n$ is constructed, where each $\leftarrow G_i$ is a goal clause, each Δ_i is a serial-goal mentioning only abducibles, $\leftarrow G_n$ is the empty clause, Δ_0 is the empty formula, and each $\leftarrow G_{i+1}, \Delta_{i+1}$ is obtained from $\leftarrow G_i, \Delta_i$ as follows: Assume $G_i = \mathbf{P}\langle \mathbf{D}_1 \cdots \mathbf{D}_j \rangle \circ \langle \mathbf{D}_{j+1} \cdots \mathbf{D}_n \rangle \vdash \ ^{m}\!\otimes \mathtt{state}$, $m \geq 1$. If there is a predicate d such that $d \in \mathcal{O}^t(\mathbf{D}_j, \mathbf{D}_{j+1})$, then $\Delta_{i+1} = \Delta_i \otimes d$, and using inference rule 3 on G_i we get

$$G_{i+1} = \mathbf{P}\langle \mathbf{D}_1 \cdots \mathbf{D}_j\mathbf{D}_{j+1} \rangle \circ \langle \mathbf{D}_{j+2} \cdots \mathbf{D}_n \rangle \vdash \ ^{m-1}\!\otimes \mathtt{state}$$

Example 1 (Sequential composition) Recalling the example presented in the introduction, assume $\mathbf{P}$ consists of the rule

$$\begin{aligned}
rotating(a,b) \leftarrow \quad & approaching(a,b) \otimes coalescing(a,b) \otimes \\
& splitting(a,b) \otimes receding(a,b) \otimes \\
& approaching(a,b)
\end{aligned}$$

Assume also $\Sigma = \mathbf{P} \cup \mathcal{O}^d \cup \mathcal{O}^t$. In the expressions below, '$\dashv$' denotes *abductive inference*, whereas '$\vdash$' denotes *deductive inference*. The snapshot sequence represented in Figure 1 is interpreted as follows.

- $\Sigma, \langle \mathbf{D}_1 \rangle \circ \langle \mathbf{D}_2 \rangle \dashv approaching(a,b)$, where $\mathcal{O}^d(\mathbf{D}_1) = dc(a,b)$, $\mathcal{O}^d(\mathbf{D}_2) = dc(a,b)$, and $\mathcal{O}^t(\mathbf{D}_1, \mathbf{D}_2) = approaching(a,b)$;
- $\Sigma, \langle \mathbf{D}_1\mathbf{D}_2 \rangle \circ \langle \mathbf{D}_3 \rangle \dashv coalescing(a,b)$, where $\mathcal{O}^d(\mathbf{D}_2) = dc(a,b)$, $\mathcal{O}^d(\mathbf{D}_3) = co(a,b)$ and $\mathcal{O}^t(\mathbf{D}_2, \mathbf{D}_3) = coalescing(a,b)$;
- $\Sigma, \langle \mathbf{D}_1\mathbf{D}_2\mathbf{D}_3 \rangle \circ \langle \mathbf{D}_4 \rangle \dashv splitting(a,b)$, where $\mathcal{O}^d(\mathbf{D}_3) = co(a,b)$, $\mathcal{O}^d(\mathbf{D}_4) = dc(a,b)$ and $\mathcal{O}^t(\mathbf{D}_3, \mathbf{D}_4) = splitting(a,b)$;
- $\Sigma, \langle \mathbf{D}_1\mathbf{D}_2\mathbf{D}_3\mathbf{D}_4 \rangle \circ \langle \mathbf{D}_5 \rangle \dashv receding(a,b)$, where $\mathcal{O}^d(\mathbf{D}_4) = dc(a,b)$, $\mathcal{O}^d(\mathbf{D}_5) = dc(a,b)$ and $\mathcal{O}^t(\mathbf{D}_4, \mathbf{D}_5) = receding(a,b)$;
- $\Sigma, \langle \mathbf{D}_1\mathbf{D}_2\mathbf{D}_3\mathbf{D}_4\mathbf{D}_5 \rangle \circ \langle \mathbf{D}_6 \rangle \dashv approaching(a,b)$, where $\mathcal{O}^d(\mathbf{D}_4) = dc(a,b)$, $\mathcal{O}^d(\mathbf{D}_5) = dc(a,b)$ and $\mathcal{O}^t(\mathbf{D}_5, \mathbf{D}_6) = approaching(a,b)$;
- $\Sigma, \langle \mathbf{D}_1 \rangle \circ \langle \mathbf{D}_2\mathbf{D}_3\mathbf{D}_4\mathbf{D}_5\mathbf{D}_6 \rangle \dashv approaching(a,b) \otimes coalescing(a,b) \otimes splitting(a,b) \otimes receding(a,b) \otimes approaching(a,b)$.
- $\Sigma, \langle \mathbf{D}_1 \rangle \circ \langle \mathbf{D}_2\mathbf{D}_3\mathbf{D}_4\mathbf{D}_5\mathbf{D}_6 \rangle \vdash rotating(a,b)$

□

Discussion and Open Issues

In this paper we presented the initial definitions of a logic language for interpreting sequences of snapshots of the world noted by a mobile robot vision system. This language brings together the notion of path semantics (Bonner & Kifer 1993) and the spatial reasoning theory proposed in (Santos & Shanahan 2002). The main contributions of this work are two fold. First, the inference engine based on transactions (as defined in (Bonner & Kifer 1993)) was extended with the notion of abduction proposed in (Shanahan 1996). On the other hand, the present framework extends the work described in (Santos & Shanahan 2002) by allowing long sequences of transitions, rather than pairs, to be interpreted. The predicates interpreting such long sequences of transitions can be understood as second-order hypotheses since they are defined over interpretations of pairs of snapshots.

The further investigation of how deduction and abduction interplay in the process of sensor data interpretation and action remains as the main open issue of the above framework. Another important open problem is the connection of hypotheses on images (such as the predicates defining the state transition oracle) to hypotheses on physical bodies. A discussion of this issue was started in (Santos & Shanahan 2002); however, this problem still warrants further investigation.

References

Bonner, A., and Kifer, M. 1993. Transaction logic programming. In *Proceedings of the Tenth International Conference on Logic Programming (ICLP)*, 257–279. MIT Press.

Harel, D.; Kozen, D.; and Parikh, R. 1982. Process logic: Expressiveness, decidability, completeness. *Journal of Computer and Systems Sciences* 2(25):144–170.

Kakas, A.; Kowalski, R.; and Toni, F. 1998. The role of abduction in logic programming. In D.M.Gabbay, C. J. H., and Robinson, J., eds., *Handbook of Logic in Artificial Intelligence and Logic Programming*. Oxford University Press. 235–324.

Matsuyama, T., and Hwang, V. S. 1990. *SIGMA: A Knowledge-Based Image Understanding System*. Plenum Press.

Peirce, C. S. 1958. *Collected papers*. Harvard University Press.

Poole, D.; Goebel, R.; and Aleliunas, R. 1987. Theorist: A logical system for defaults and diagnosis. In Cercone, N., and McCalla, G., eds., *The Knowledge Frontier*. 331–352.

Randell, D.; Cui, Z.; and Cohn, A. 1992. A spatial logic based on regions and connection. In *Proc. of the KR*, 165–176.

Reiter, R., and Mackworth, A. 1990. A logical framework for depiction and image interpretation. *Artificial Intelligence* 41:125–155.

Santos, P., and Shanahan, M. 2002. Hypothesising object relations from image transitions. In van Harmelen, F., ed., *Proc. of ECAI-02*, 292–296.

Schroeder, C., and Neumann, B. 1996. On the logics of image interpretation: model construction in a formal knowledge representation framework. In *International Conference on Image Processing*, volume 2, 785–788.

Shanahan, M. 1996. Robotics and the common sense informatic situation. In *Proc. of ECAI-96*, 684–688.