TR-0584

# Some remarks on the mathemat ics of situation theory

by
Tim Fernando

August, 1990



ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

# Remarks on the mathematics of situation theory : a review of its past and a proposal for its future

Tim Fernando

September 7, 1990

Situation theory means different things to different researchers. To some, it is a theory of information (Barwise [6], Devlin [11]). This paper is concerned with the mathematical development of this view of situation theory.

Given this orientation, a *working* answer to the question "What is situation theory ?" is presented in section 1. Section 2 is a review of some past efforts to develop situation theory mathematically.[1] Addressed primarily to readers who have some familiarity with these works, it can be skipped by those who have not. Section 3 introduces the notion of a *support-map*, which the author believes to be basic to situation theory. Certain ideas concerning the investigation of this notion are then discussed in section 4.

## 1. What is situation theory?

Situation theory is a *semantic* theory of *information*. The term "semantic" refers to some notion of meaning. An example of a semantic theory is classical model theory, as initiated by Tarski. Tarski locates the *meaning* of a proposition in its relation to mathematical structures that serve as contexts (i.e., first-order models). A situation is a *partial context*. Speaking very roughly, a situation is a generalized notion of a model. If model theory is a semantic theory, what is it a semantic theory of? Is it a semantic theory of *information*? Observe that the sentences of predicate logic suppress any mention of a model, confining it to the background meta-theory:

$$1 + 0 \ = \ 1$$

is a first-order arithmetic sentence;

$$\langle N, +, \times, 0, 1\rangle \ \models \ 1 + 0 = 1$$

is *not*. It is possible to leave the context (= model) implicit only because it is assumed fixed (i.e., stable). Model theory is a semantic analysis of truth in a fixed context. But what about information? Whereas truth might be considered eternal and static, this certainly will not do with information. An essential characteristic of information is that it involves different contexts, some of which are "parts of," or otherwise related to, others. This is the thrust of

**Slogan 1.** *Information is situated.*

Also, far from being static, information is dynamic — a point related to slogan 1 by

**Slogan 2.** *Information flows between situations.*

This is why the notion of a situation is important in a semantic theory of *information*, and why the theory is called *situation* theory.

Historically, situation theory has brought slogan 1 to bear in its approach to problems in semantics. Accordingly, past efforts to develop a mathematics of situation theory are reviewed in section 2 from the point of view of slogan 1. Slogan 2 is taken up in succeeding sections.

[1] As hard as the author has worked to understand other people's research, it is quite possible that he has missed certain points. He hopes that the communication of his impressions might at the very least expose errors in his thinking.

## 2. A review of some past work

Associated with slogan 1 is the picture

$$s \models \sigma .$$

The idea is that information comes in pieces $\sigma$ called (following a suggestion by Keith Devlin) *infons*, and that situations $s$ may or may not *support* (with respect to $\models$) these infons. Roughly, an infon can be regarded as a triple

$$\langle\langle r, a; i \rangle\rangle$$

where

- $r$ is a *property* (also called a *relation*),
- $a$ is an *assignment* — i.e., a (partial) function with a set of (some of) $r$'s *argument roles* as its domain $\cdots$, and
- $i \in \{+, -\}$ is a *polarity*[2].

$\models$ is a subcollection of $Sit \times Inf$ where $Sit$ is some collection of situations and $Inf$ is some collection of infons. In the picture above, a situation occurs to the left of $\models$, *qua* context. Situation theory, however, also admits the possibility that a situation occurs *qua* object, to the right of $\models$, in the range of an assignment $a$ (inside an infon $\sigma$). This leads to the problem of circularity, such as when, for example, a situation $s$ is identified with the set $\{\sigma \mid s \models \sigma\}$ of infons $\sigma$ it supports.[3]

### 2.1. AFA and extensional reductions

Aczel's *Anti-Foundation Axiom* (AFA) provides a direct treatment of circularity in set theory, by allowing $\in$ to be non-well-founded. A useful formulation of the axiom is as a *Solution Lemma* that asserts that *systems of equations* between *indeterminates* and sets (built from indeterminates) possess unique *solutions*. Let us explain this by means of an example (referring the serious reader to Aczel [3] for a detailed presentation of the theory). The system of equations

$$\begin{aligned} x &= \{x\} \\ y &= \{\emptyset, \{z\}\} \\ z &= \{x, y\} \end{aligned}$$

over the indeterminates $x$, $y$ and $z$ has the unique solution

$$\begin{aligned} x &= \Omega \\ y &= \{\emptyset, \{a\}\} \\ z &= a \end{aligned}$$

where $\Omega$ is the unique set that has exactly itself as an element, and $a$ is defined (implicitly) as

$$a = \{\Omega, \{\emptyset, \{a\}\}\} .$$

[2] Following a practice of P. Aczel's, and a suggestion by H. Yasukawa, we write + and − instead of 0 and 1 so as not to give the impression that a polarity is a truth value (from say, a Heyting algebra), but is simply some kind of *sign*. See the discussion of forcing in section 3.3.

[3] As it turns out, more serious problems than non-well-foundedness may block the reduction of a situation to the collection of infons it supports; consider, for instance,

$$\begin{aligned} s \models \sigma \quad &\Leftrightarrow \quad s' \models \sigma \\ &\Leftrightarrow \quad \sigma = \langle\langle r, s; + \rangle\rangle \ \vee \ \sigma = \langle\langle r, s'; - \rangle\rangle \end{aligned}$$

where $r$ is some relation with one argument role, and an assignment to $r$ is indicated simply by writing the value assigned to this argument role.

(The newcomer might find the circularity in the description above of the system of equations and its solution unsettling in its simplicity. But this is actually all there is to it.)

The usefulness of the solution lemma lies in taking *extensional reductions*. The principle behind extensional reductions can be understood through examples. In semantics, an *intensional* notion such as a proposition is sometimes identified with its truth-value over a fixed model (à la Tarski), or with the collection of possible worlds in which the proposition is true (the latter being the basic premise of possible worlds semantics). Going the other direction[4], a situation is, as we noted above, sometimes equated with the set of infons it supports. How the solution lemma can be applied to form such extensional reductions is shown nicely in the *final co-algebra* theorem of Aczel [3] (Theorem 7.8), on which the concept of *extensional structures* in section 4 of Fernando [14] is based.[5]

To keep a semantic framework from collapsing to utter nonsense, certain intensional notions are not extensionally reduced. In Barwise and Etchemendy [7], *Russellian propositions* and (in the Austinian case) infons are treated *as structured objects* in their own right. More specifically, a "basic" infon is identified with some object containing a property (relation), assignment and polarity. It is *not* reduced to the collection of situations supporting it. As such, it has a syntactic character of the sort studied in a constructive approach to mathematics (where the objects of study are the *presentations*).

## 2.2. Formalizing situation theory

Perhaps the primary point of Barwise and Etchemendy [7] is the separation (in the Austinian account) of two kinds of assertions. This separation is developed further in Barwise [5], where two parallel semantic systems are outlined, corresponding roughly to an *absolute* plane and a *relative* plane. The former is, in Barwise's words, "structurally determinate," consisting of *types*, each of which has an extension, and *propositions*, each of which has a truth value. By applying an object[6] $a$ to a type $t$, we obtain the proposition

$$a : t$$

which is assigned the truth value *true* exactly when $a$ is in the extension of $t$. To pass from propositions back to types, it is useful also to form parametric propositions from parameters. (We will take up later the question of whether such parametric propositions have truth values.) The intuition is that abstracting parameters $\overline{x}$ from a parametric proposition $p$ (with free parameters among $x$) yields the type $\hat{\overline{x}}.p$ whose extension consists of the objects $\overline{a}$ for which $p$ with $\overline{x}$ replaced by $\overline{a}$ has truth value *true*.

A parallel situationally-conditioned plane is set up, consisting of relations (as the relative counterparts of types), and infons (as the relative counterpart of propositions). The picture here is

| | | |
|---|---|---|
| infons | | propositions |
| application ↑ ↓ abstraction | $\xrightarrow{s,\models}$ | application ↑ ↓ abstraction |
| relations | | types |
| **relative** | | **absolute** |

Omitted from the picture above is the fact that propositions, as well as infons, are closed under $\vee, \wedge$ and $\neg$. (Quantification is not treated in Barwise [5].) In order to block paradoxes, Barwise imposes "appropriateness conditions" on application, in the form of a type *Approp-for* between types and the objects (i.e., assignments) that might be applied to them. *Approp-for* is a *primitive* type, as opposed to a *non-primitive* type which is obtained by abstraction over a parametric proposition. Although it was widely circulated in the situation theory community, Barwise [5] was never completed and therefore never published.

Instead a more careful development of the project begun in Barwise [5] was outlined at the 1989 Situation Theory conference (held in Asilomar, California). As reported in Westerstahl [26], Aczel proposed a formal development of situation theory in three stages:

[4] There is, under suitable conditions, a form of duality here (associated with Stone), which is discussed below in section 3.2.

[5] The rough idea is to start with a *syntactic* category that respects a certain theory T (containing no logical notion of equality); the solution lemma is then used to define a functor into a (possibly *final*) model of T. What system of equations ought to be solved depends on the theory T.

[6] Or more precisely, an assignment — i.e., a function.

1. a survey of (all possible) structured objects

2. a construction of an absolute theory of truth (i.e., types, propositions, and parameters)

3. an account of situations, infons, and other basic notions of situation theory.

Interpreting the proposal naively, the point of the decomposition into three stages is that investigations must first center on structured objects, then on an absolute theory of truth, and finally on "typical situation-theoretic objects" (Westerstahl [26]). We will return to this matter at the conclusion of section 2.3. For now, let us follow Aczel into stage one.

The theory of structured objects in Aczel [4] is built around the idea of collecting structured objects in a structure where they are assigned components subject to replacement. The main intuitions come from set theory and syntax, but the approach taken is rather abstract, inasmuch as the concepts are developed axiomatically, rather than coded concretely in set theory (as is customary in set-theoretic foundations of mathematics). In the pursuit of a comprehensive coverage, however, some of the ideas in Aczel [3] particularly, the solution lemma — become somewhat obscured (by, for example, relabelling maps introduced to weaken equality to isomorphism). It is not obvious to the author that the attention to abstractness is necessary for situation theory; what is clear is that an analysis of certain topics important in formalizing situation theory get postponed in Aczel [4] — namely, many sorts, *mixed fixed points*, and constructive (effective) issues.

As for stage 2, Westerstahl spoke at the Asilomar conference about treating parameters as objects of a theory, and not just as linguistic expressions (residing, as it were, in the meta-theory). Westerstahl [26] observes that parameters are useful in two ways: for anchoring (i.e., application) and abstraction. The author fails to see how these uses necessitate the introduction of parameters as objects of the theory. (Slogans concerning the value of "first-class" citizenship should not be applied blindly.) Experience in formal logic has shown over and over again that a careful separation between object- and meta-levels is crucial. In predicate logic, parameters are no more than linguistic entities; this fact is evidently not an obstacle in formalizing mathematics in first-order logic. A more dramatic case is combinatory logic, where parameters are eliminated even from the syntax.

This brings us to work on situation theory over a combinatory, type-free framework. The approach of Plotkin [23], [24] is to formalize situation theory in an extension of the theory of *Frege structures* (Aczel [2]). The theory of Frege structures is essentially a theory of propositions built on top of the untyped lambda calculus. Let us call these propositions *Fregean-propositions* to distinguish them from the propositions of situation theory. Plotkin equates infons (previously called "states of affairs," or soa's) with Fregean propositions. Just as Fregean propositions can be true, so too can infons (according to Plotkin). To deal with circularity, the $Y$-combinator (i.e., the recursion theorem) is used, from which a number of negative results (more or less familiar from recursion theory) also follow. As a piece of research in *situation* theory, it is slightly curious that Plotkin [24] completely leaves out situations, investigating semantic notions that are *un-situated*. In Plotkin [23], situations are introduced as certain sets of infons, the idea then being that a "true" infon (or *fact*) is one that is an element of some (actual) situation. This approach, however, severely constrains the notion of an infon (or more precisely, $\models$), since a consequence of the approach is that no two (actual) situations can support contradictory infons.[7] Evidently, what Plotkin has done is to ignore the situation $s$ and predicate $\models$ in

$$s \models \sigma$$

with the idea of extracting $s$ and $\models$ from $\sigma$. This completely blurs the distinction between infons and propositions; in the author's opinion, the distinction must first be made clear, before (in certain special cases) it is muddied. From this point of view, Plotkin's work concerns a theory of infons (and relations) for a fixed situation and a fixed predicate $\models$. But then if $s$ and $\models$ are fixed, why bother introducing the notions at all? And why develop a theory named *situation* theory?

[7] As will become clear in section 3, this point is related to the notion of *(strong) persistence*, as well as the assumption that information flow is directed.

The author does, in fact, believe in the idea of a *situation* theory, and regrets that in the outline proposed by Aczel, the development of situation theory proper is put off to the last stage. Should basic situation-theoretic principles be taken seriously from the very beginning of work on formalizing situation theory? Of course — but what if one does not know what these principles are? Afterall, is not the discovery of these principles the very reason for formalizing situation theory? Yes, but perhaps the project is not so hopeless, as long as

- there is some (minimal) understanding of what the theory is about, and
- sufficient attention is paid to developing this understanding.

Peripheral topics should not distract us away from central problems. The subject of structured objects can be very interesting from a purely mathematical point of view, and surely it ought to be pursued for reasons not necessarily related to the immediate concerns of situation theory. Nevertheless, some critical reflection is perhaps also in order. Consider, for example, the work on obtaining non-well-founded sets as ideal completions of well-founded sets. What is the point of this exercise? Lindstrom [21] shows that Aczel's construction can be carried out in a constructive setting (Martin-Löf type theory). As for the feeling that well-founded sets are in some sense simpler than non-well-founded ones, who will deny that $\Omega = \{\Omega\}$ is a perfectly finite object that is not as problematic as $\omega = \{0, 1, \ldots\}$ is finitistically, or $Pow(\omega)$ is predicatively? The mathematics involved might be very pretty (as the works of Mislove, Moss, Oles [22] and Abramsky [1] demonstrate), but work on situation theory should not lose sight of situation theory.

## 2.3. A set-theoretic formalization

At the Asilomar conference, the author reported that he had carried out all three of Aczel's stages mentioned above. He did not (and does not now) claim that work on the project can stop. The point of this section is to describe what is accomplished in Fernando [14].

The primary contribution of the paper is the proposal of a notion of a *structure* for situation theory, and a definition of truth over such a structure involving a rich collection of situation-theoretic constructs. (In other words, what is supplied is the equivalent in situation theory to Tarski's truth definition in first-order logic.) The term "rich" is used above to indicate, for example, that

- in addition to logical connectives such as $\wedge$ and $\neg$, quantification on infons and propositions is allowed (in contrast to Barwise [5]),
- a general system of abstraction and application between propositions and types is constructed, as is one between infons and relations, and
- parameters are subject to arbitrary type restrictions.

The basic idea is to build a "type theory" over ZFC/AFA, wherein certain types are *primitive* and others are *non-primitive*, the latter being obtained from abstraction over parametric propositions. Although Barwise [5] is used as the starting point, Fernando [14] differs in certain significant ways from it. For instance, primitive types in Fernando [14] come in three kinds:

- *free* types with extensions given at the outset by the notion of a structure,
- *generated* types (for example, infons and propositions) obtained from *substitutional recursive definitions* (Fernando [13]), and
- a universal type *obj*.

Furthermore, whereas Barwise [5] assumes that there is a primitive type *Approp-for* to block paradoxes, the corresponding type in Fernando [14] is non-primitive. Instead, a positive inductive definition of truth reminiscent of Kripke [20] is applied, resulting in a 3-valued logic (from which fragments respecting 2-valued logic can then be extracted).

Fernando [14] might be better understood by breaking down the construction according to Aczel's three stages.

- In stage one, a concrete theory of structured objects based on AFA is used to generate the necessary objects that populate a structure. Parameters and parametric objects are among the objects so generated.
- In stage two, however, parametric objects with *free* occurrences of parameters are weeded out in the definitions of structure and truth. In other words, following logical tradition, free parameters are confined to the meta-level. The proposal of a notion of structure provides a clean distinction between the object- and meta-levels. It is the contention of the author that the need to anchor (= apply) and abstract parameters is met by passing between propositions (or infons) and types (or relations). An entity can be a parameter from the point of view of the meta-theory (in which case it is subject to the system of abstraction and application in a structure), or from the point of view of the object-theory (in which case it may be formally denoted by a constant); but it can*not* be a parameter in both senses.
- As for stage three, Fernando [14] essentially carries out this work in stages one and two. (Indeed, stages one and two are directed solely towards developing situation theory.)

Fernando [14] goes on to define notions of syntax, and *extensional structures*, leading to a "logic" over a set-theoretic foundation (ZFC/AFA).

Next, it is only natural to ask whether the author (in his arrogance) believes that his formalization is definitive. Can more interesting accounts be given? But interesting in what sense? We ought to be clear about what problems we wish to address. Fernando [14] was carried out at a time when the logical consistency of situation theory (as applied in natural language semantic analysis) was in grave doubt. Barwise [5] and Plotkin [23] were intended to provide some assurance that situation theory was coherent. After working on the problem as well, the author is no longer troubled by the question of consistency, and is convinced that paradoxes can be avoided by exercising sufficient care. Having made (or at least claimed to make) a preliminary pass through Aczel's three stages, he thinks that the stages are *not* independent nor properly developed simply in a serial manner. (From stage three, it may be appropriate to return to stage two or to stage one, and then perhaps back to ...) The point of a formalization is to shed light on basic principles of situation theory. In particular, he believes that an important next step in the "mathematics of situation theory" is to study the picture

$$\begin{array}{ccc} \text{infons} & & \text{propositions} \\ \uparrow\ \downarrow & \xrightarrow{s,\models} & |\ \ | \\ \text{relations} & & \text{types} \\ \textbf{Information} & & \textbf{Truth} \end{array}$$

adapted from that in section 2.2, page 3. Some notion of computation must be brought into this picture, particularly on the side of information. After briefly attempting to work out a theory of effective fixed points for the substitutional recursive definitions (Fernando [13]) that underly Fernando [14], the author feels that a formalization of situation theory in *explicit mathematics*[8] (Feferman [12]) may be more natural. Indeed, having emphasized that Fernando [14] provides a set-theoretic formalization of situation theory, it should also be stressed that the notions constructed inductively for the theory (e.g., type membership ":") are highly intensional. In any case, before undertaking another formalization of situation theory, it is crucial to have a clearer sense of what the theory is about.

## 3. Introducing support-maps

Given a collection *Sit* of situations, and a collection *Inf* of (pieces of information called) infons, a *support-predicate* $\models$ is a certain kind of binary relation between *Sit* and *Inf*. The underlying idea here is slogan 1: "Information is situated." Missing in $\models$, however, is a treatment of slogan 2: "Information flows between situations."

The notion of a *support-map* is an extension of the notion of a support-predicate that is intended to analyze information flow. Roughly, a support-map $\Delta$ is a support-predicate $\models$ plus an intepretation of a

[8] Explicit mathematics is not unrelated to the Frege structures (Aczel [2]) on which Plotkin [23], [24] are based.

pre-order (indicating information flow) between situations in terms of arrows (to be explained in section 3.2) between the (sets of) infons supported by the situations.

### 3.1. Support-maps and support-predicates

Let us first understand the relation between support-maps and support-predicates. Fix a collection *Sit* of situations, and a collection *Inf* of infons. For now, we will not ask what situations or infons are. We will simply assume that collections of these are given. To every support-predicate $\models$ between *Sit* and *Inf*, associate the function $\Delta_{\models} = \Delta$ defined on situations $s$ as follows:

$$\Delta s \quad = \quad \{\sigma \mid s \models \sigma\} \ .$$

That is, going the opposite direction,

$$s \models \sigma \quad \Leftrightarrow \quad \sigma \in \Delta s \ .$$

Next, rather than considering the full collection $Pow(Inf)$ of subsets of *Inf*, it will turn out to be useful to isolate the notion of an *infon-packet*, by postulating a collection $IP$ of these infon-packets. An infon-packet is a set of infons; i.e., $IP \subseteq Pow(Inf)$. We will not require every set of infons to be an infon-packet; i.e., we do not assume $IP = Pow(Inf)$. But we will require that sets of the form $\Delta s$ be infon-packets. In other words, a support-map $\Delta$ is (in part) a function from a collection *Sit* of situations to a collection $IP$ of infon-packets

$$\Delta : Sit \rightarrow IP \ .$$

$\Delta$ need not be surjective (i.e., onto); however, it is often natural to assume an infon-packet is a set of infons that satisfies certain *closure* (as well as *coherence*) *conditions*. Writing $\Sigma$ for an infon-packet, a closure condition can generally be put in the form

$$\frac{\Sigma_0 \subseteq \Sigma \qquad P(\Sigma_0, \tau)}{\tau \in \Sigma}$$

where $P$ is some predicate (or, in the terminology of section 2.2, type). A concrete example is given by the infon algebras $\langle Sit, Inf, \Rightarrow, \models \rangle$ in Barwise and Etchemendy [8], where the infons (collected together in *Inf*) are partially ordered by some notion of entailment $\Rightarrow$. The requirement there that an infon-algebra satisfy the condition

$$\frac{s \models \sigma \qquad \sigma \Rightarrow \tau}{s \models \tau}$$

can be restated (without mentioning situations) as the requirement that a support-map have range $IP$, where the infon-packets $\Sigma \in IP$ satisfy

$$\frac{\sigma \in \Sigma \qquad \sigma \Rightarrow \tau}{\tau \in \Sigma} \ .$$

But now, what is the point of introducing the notion of an infon-packet, and reformulating Barwise and Etchemendy's conditions as conditions on infon-packets? The answer lies in our category-theoretic formulation of information flow, to which we now turn.

### 3.2. Support-maps and information flow

To give a mathematically precise account of information flow, it is convenient to adopt some terminology from category theory. A *category* is roughly

- a collection of *objects*, and
- a collection of *arrows* (or *morphisms*) between objects (where an arrow is a generalization of the notion of function).

Associated with every arrow are two (possibly identical) objects, its *domain* and *codomain*. When the codomain of an arrow $f$ is identical to the domain of an arrow $g$, it is possible to *compose* the two arrows

$$f \; ; \; g$$

to get back an arrow, with the domain of $f$, and the codomain of $g$. Composition of arrows is also required to be associative, and have identities. Without precisely spelling this out, let us simply remark that the essential structure in a category lies in the arrows.

The notion of a support-predicate concerns only the object component (as opposed to the crucial arrow component) of categories. What is interesting about information is that it flows; similarly, what is interesting about a support-map is its interpretation of the arrows ($\approx$ flow).

We will form a category out of the collection $IP$ of infon-packets by adding a collection $Arr(IP)$ of arrows between infon-packets. We do this gradually, starting with a collection $Arr(Inf)$ of arrows between infons, which in turn is constructed from a monoid $\langle Links, \cdot, e\rangle$ of "links." The plan is that $Arr(Inf)$ will be a certain subcollection of $Inf \times Links \times Inf$, and $Arr(IP)$ will be a certain subcollection of $IP \times Links \times IP$. For $\sigma, \sigma' \in Inf$, $\Sigma, \Sigma' \in IP$, and $\alpha \in Links$, let us write

$$\sigma \xrightarrow{\alpha} \sigma'$$

for $(\sigma, \alpha, \sigma') \in Arr(Inf)$ (where $domain(\sigma, \alpha, \sigma') = \sigma$ and $codomain(\sigma, \alpha, \sigma') = \sigma'$), and

$$\Sigma \xrightarrow{\alpha} \Sigma'$$

for $(\Sigma, \alpha, \Sigma') \in Arr(IP)$ (where $domain(\Sigma, \alpha, \Sigma') = \Sigma$ and $codomain(\Sigma, \alpha, \Sigma') = \Sigma'$).

The intuition behind $\langle Links, \cdot, e\rangle$ is that an $\alpha \in Links$ is a "link" between situations, as illustrated in the following examples.

- In dynamic logic, a situation can be viewed as an environment for a program (e.g., the variable bindings), and a link as some program such as

  $$x \;:=\; x + 1$$

  that transforms the environment.

- In model theory, a situation can be taken to be a first-order model, and a link a function between non-empty sets (regarded as universes of models).

Next, the idea is that

$$\sigma \xrightarrow{\alpha} \sigma'$$

means *roughly* that for every $s, s' \in Sit$ "linked" by $\alpha$,

$$s \text{ supports } \sigma \quad \text{implies} \quad s' \text{ supports } \sigma' \; .$$

(What is rough here is the notion of "supports," which is used informally above.) Returning to the examples above, observe that

- in dynamic logic, the Hoare triple

  $$\{P\} \quad \alpha \quad \{Q\}$$

  asserting that (the postcondition) $Q$ always holds following the execution of the program $\alpha$ at a state in which (the precondition) $P$ is true corresponds to

  $$P \xrightarrow{\alpha} Q \; .$$

- Continuing the example of model theory above, an infon can be identified with a pair $(\varphi(\overline{x}), f)$ of a first-order formula $\varphi(\overline{x})$ and an environment $f$ for it (i.e., a function with $\overline{x}$ in its domain). Then we might take

$$(\varphi(\overline{x}), f) \quad \xrightarrow{\alpha} \quad (\psi(\overline{y}), g)$$

to mean that $\varphi(\overline{x}) = \psi(\overline{y})$, and $g = \alpha \circ f$.

Having described $Arr(Inf)$, how do we pass to $Arr(IP)$? As an infon-packet is interpreted "conjunctively" (i.e., $\Sigma \approx \bigwedge \Sigma$), typically[9], the idea is to apply the *Hoare pre-order*

$$\Sigma \xrightarrow{\alpha} \Sigma' \quad \leftrightarrow \quad \forall \sigma \in \Sigma \; \exists \sigma' \in \Sigma' \; \sigma \xrightarrow{\alpha} \sigma' \; .$$

For example, if *Inf* is made into a *discrete* category (i.e.,

$$\begin{aligned} Links &= \{e\} \\ Arr(Inf) &= \{(\sigma, e, \sigma) \mid \sigma \in Inf\} \; ) \end{aligned}$$

then (suppressing the link $\alpha$ since it can only mean $e$), the Hoare recipe yields

$$\Sigma \rightarrow \Sigma' \quad \text{iff} \quad \Sigma \subseteq \Sigma' \; .$$

More interesting examples are provided in model theory, where a situation is a first-order model $\mathcal{M}$. If

$$\Delta \mathcal{M} \quad = \quad \{(\varphi(\overline{x}), f) \mid \varphi(\overline{x}) \text{ is atomic, } \mathcal{M} \models \varphi[f]\}$$

then

$$\Delta \mathcal{M} \xrightarrow{\alpha} \Delta \mathcal{N} \quad \text{iff} \quad \alpha \text{ restricted to } \mathcal{M} \text{ is a homomorphism ;}$$

if

$$\Delta \mathcal{M} \quad = \quad \{(\varphi(\overline{x}), f) \mid \varphi(\overline{x}) \text{ is atomic or negated atomic, } \mathcal{M} \models \varphi[f]\}$$

then

$$\Delta \mathcal{M} \xrightarrow{\alpha} \Delta \mathcal{N} \quad \text{iff} \quad \alpha \text{ restricted to } \mathcal{M} \text{ is a monomorphism ;}$$

and if

$$\Delta \mathcal{M} \quad = \quad \{(\varphi(\overline{x}), f) \mid \varphi(\overline{x}) \text{ is elementary (arbitrary), } \mathcal{M} \models \varphi[f]\}$$

then

$$\Delta \mathcal{M} \xrightarrow{\alpha} \Delta \mathcal{N} \quad \text{iff} \quad \alpha \text{ restricted to } \mathcal{M} \text{ is an elementary map .}$$

Now, a ***support-map*** can be defined[10] to be a *functor* $\Delta : Sit \rightarrow IP$, where *Sit* is pre-ordered by $\leq$ (indicating information flow). Spelled out, this means that for every $s, s' \in Sit$ with $s \leq s'$, $\Delta$ associates an $f_{s,s'} \in Arr(IP)$ with domain $\Delta s$ and codomain $\Delta s'$ such that

$$\begin{aligned} f_{s,s} \quad &= \quad identity_{\Delta s, \Delta s} \\ s' \leq s'' \quad \text{implies} \quad & f_{s,s''} = f_{s,s'} ; f_{s',s''} \; . \end{aligned}$$

Notice that in case

$$\Sigma \rightarrow \Sigma' \quad \text{iff} \quad \Sigma \subseteq \Sigma' \; ,$$

[9] A more general recipe from arrows between infons to arrows between infon packets is currently being studied by the author.

[10] A slightly more elaborate definition is given in Fernando [15].

the functoriality of $\Delta$ amounts simply to

$$s \leq s' \quad \text{implies} \quad \Delta s \subseteq \Delta s' ,$$

a property often referred to as *(strong) persistence*.

The reader familiar with the notion of a Kripke model might ask why not call support-maps Kripke models? Yes, the idea is quite old; and, in fact, examples of it (such as elementary chains and ultraproducts) predate Kripke. The author proposes to carry out a fundamental re-examination of this notion, and to investigate whether an analysis of the notion in a setting more general than that considered in the past can be enlightening (or otherwise interesting). At the very least, such a study ought to clarify where certain simplifying assumptions are smuggled in (which, however mathematically fruitful, may for certain applications be problematic).

Take, for instance, the duality often asserted between situations and infons. Under the classic Stone duality, the situations are organized in a certain kind of topological space, and the infons in a certain kind of distributive lattice. Infons supported by the same situations are identified, as are situations with the same infon-packet. This becomes problematic when situations are allowed to occur *qua* object, inside an infon. (See footnote 3.) Furthermore, the arrows in *IP* and *Sit* are not touched by the duality. Indeed, the disparity in structural complexity between a pre-order on situations and infon-packet arrows formed from a possibly complicated monoid $\langle Links, \cdot, \epsilon \rangle$ makes an adjunction between *Sit* and *IP* hopeless (in general). Why not then allow *Sit* to have richer categorical structure? After all, the intuition behind links can be explained (as done above) by thinking of these links as existing between situations. Another intuition, however, is that the arrows between situations are *undifferentiated* and require analysis; that is what support-maps and infons are for. If situations and infons were inter-derivable, then why bother introducing both notions? To this one may respond that a semantic analysis is often taken to mean an extensional reduction, and to carry out such a reduction, we need (intensional) notions that, at the outset, are separate (before showing that, in fact, they are not). There is, however, a consideration that is less perverse, namely, the principle of simplicity; and simplicity comes down heavily on the side of *Sit* as a pre-order. To study different pre-orders between situations, the idea then would be to study different support-maps (with possibly different domains). For surely, if the notion of a support-map is basic to situation theory, then we had better be prepared to study the different instances of the notion.

## 3.3. Some examples (classical, intuitionistic and everyday reasoning)

The author believes that support-maps ought to be treated as fundamental objects of situation theory, and, as such, constructions on them ought to be studied.[11] Three kinds of constructions, in particular, seem to occur with notable frequency.

**Logical extensions.** Given a support-map $\Delta$, another support-map $\Delta'$ is constructed with the same domain as $\Delta$, where for every situation $s$ in the domain,

$$\Delta s \quad \subset \quad \Delta' s .$$

The infons in $\Delta' s - \Delta s$ are in a natural sense (logically) *compound*, and are typically added to $\Delta' s$ intrasituationally (i.e., according only to $\Delta s$, and not to some $\Delta s'$ where $s' \neq s$). An example is Tarski's inductive truth definition (for first-order logic) of

$$\mathcal{M} \quad \models \quad \varphi[f]$$

from atomic $\varphi$ to (arbitrary) elementary $\varphi$.

**Persistent extensions.** This kind of construction again yields a support-map with the same domain, but differs in that it places emphasis on the pre-order $\leq$ on situations. Expressed in terms of support-predicates, the idea is that given a support-predicate $\models$, to construct a support-predicate $\models^*$ which is related to $\models$ in that

$$s \models^* \sigma \quad \text{implies} \quad \exists s' \geq s \; s' \models \sigma ,$$

[11] This then suggests the problem of forming categories of support-maps, and viewing these constructions as functors (Fernando [15]).

and which, moreover, is strongly persistent (i.e.,

$$s \models^* \sigma\ ,\quad s \leq s' \quad \text{implies} \quad s' \models^* \sigma\ ).$$

Such predicates are discussed at some length in Fernando [15]. For now, we remark only that the largest such support-predicate is $\models^\diamond$ defined by

$$s \models^\diamond \sigma \quad \Leftrightarrow \quad \forall s' \geq s\ \exists s'' \geq s'\ s'' \models \sigma\ ,$$

which, when $\models$ is strongly persistent and $\leq$ is linear, amounts to taking the union of a chain.

**Restrictions on situations.** A given support-map can have too many situations in its domain in which case it is useful to weed out some situations. More precisely, given a support-map $\Delta : Sit \to IP$, we might restrict $\Delta$ to a certain subcollection $S \subseteq Sit$.

With this general description, let us turn to perhaps the fundamental construction in first-order model theory (which is related to the Lowenheim-Skolem, Gödel completeness and compactness theorems)

A consistent first-order $L$-theory T has a model of cardinality $\leq max(|L|, \omega)$.

From the point of view of the preceding paragraph, the construction can be broken into two steps, where

- $IP$ is the collection of all sets $\Sigma$ of $L'$-sentences consistent with $\mathrm{T} \cup \mathsf{Henkin}_{L'}$, where $L' \supseteq L$ contains Henkin witnesses as specified by a certain fixed theory $\mathsf{Henkin}_{L'}$;
- $IP$ is partially ordered by the subset relation;
- $Sit = IP$, and $\Delta$ is the identity functor.

In step one, a chain $C \subseteq Sit$ is constructed that is maximal in the sense that no $s \in Sit$ is strictly larger than every $s' \in C$. This corresponds to a restriction of situations:

$$\Delta \quad \leadsto \quad \Delta \text{ restricted to } C\ .$$

In step two, the union of the chain $C$ — i.e., a persistent extension of $\Delta$ restricted to $C$ — is taken (from which the required model is then extracted by taking equivalence classes of constants).

Now, one criticism that might be raised against the preceding "situation-theoretic" account is that it misses the cleverness in the construction (e.g., the idea of Henkin witnesses, and the conversion of a complete Henkin theory to a model). But surely it is too much to expect situation theory to make thinking unnecessary. The more serious objection to the account above is that it introduces notions that contribute absolutely nothing to the construction. There is simply no denying that the notions of situation (as distinct from infon-packet) and support-map are completely irrelevant here. And, in fact, how could we ever dream otherwise, given that first-order logic is (as described in section 1) a theory of truth in a fixed context (i.e., situation), free from the "anomalies" that plague, say, everyday reasoning? The very point of the completeness theorem (a corollary of the construction above) is that in first-order logic, situations (i.e., first-order models) can (in a precise sense) be eliminated because the syntax is complete.

Of course, this does not mean that there is no place for model-theoretic arguments; and the notions of situation and support-map do, in fact, arise in more significant ways in various first-order model-theoretic constructions such as (elementary) chains and ultraproducts. A particularly instructive example is the very general formulation of forcing presented in Keisler [19]. The basic approach can be described as a kind of intuitionistic analog to the fundamental result for classical first-order logic quoted above. And here the three kinds of constructions on support-maps mentioned above lie very much at the heart of the method, which proceeds as follows. Fix a countable first-order language $L$, and a countable set $C$ of fresh constants. A *forcing property* is a triple $(P, \leq, f)$ where

- $(P, \leq)$ is a partial order with a least element 0,

- $f : P \to Pow(\{\text{atomic } L \cup C\text{-sentences}\})$ maps an element of $P$ to a set of atomic $L \cup C$-sentences such that for all $p, q \in P$,

$$p \leq q \quad \text{implies} \quad f(p) \subseteq f(q) \ ,$$

and certain other technical conditions are satisfied.

For our purposes, the point to be made is that a forcing property is a particular kind of support-map. Next, from a forcing property $(P, \leq, f)$, we pass to a notion of *(strong) forcing* $\Vdash$ (between $p \in P$ and arbitrary elementary $L \cup C$-sentences $\varphi$) according to

$$\begin{aligned} p \Vdash \varphi \quad &\Leftrightarrow \quad \varphi \in f(p) \quad \text{for atomic } \varphi \\ p \Vdash \bigvee \Phi \quad &\Leftrightarrow \quad \exists \varphi \in \Phi \ p \Vdash \varphi \\ p \Vdash \neg\varphi \quad &\Leftrightarrow \quad \forall q \geq p \ \text{ not } q \Vdash \varphi \\ p \Vdash \exists x \varphi(x) \quad &\Leftrightarrow \quad \exists c \in C \ p \Vdash \varphi(c) \ . \end{aligned}$$

$\Vdash$ determines another support-map, which is a logical extension of the forcing property. It is convenient to isolate a third support-map $\Vdash^w$ (*weak forcing*) obtained by a persistent extension of $\Vdash$

$$\begin{aligned} p \Vdash^w \varphi \quad &\Leftrightarrow \quad p \Vdash \neg\neg\varphi \\ &\text{iff} \quad \forall q \geq p \ \exists r \geq q \ r \Vdash \varphi \ . \end{aligned}$$

Finally, a model-theoretic analysis of $\Vdash$ and $\Vdash^w$ leads to the notion of a generic set. Given a forcing property $(P, \leq, f)$, a *generic set* is a subset $G \subseteq P$ such that

$$\begin{aligned} p \in G \ , \ q \leq p \quad &\text{implies} \quad q \in G \\ p, q \in G \quad &\text{implies} \quad \exists r \in G \ \ p, q \leq r \\ \text{for every sentence } \varphi \text{ in } L \cup C \quad &\exists p \in G \quad p \Vdash \varphi \ \text{ or } \ p \Vdash \neg\varphi. \end{aligned}$$

It turns out that every $p \in P$ is contained in some generic set, and that for every generic set $G$, the theory

$$\mathsf{T}_G \quad = \quad \{\varphi \mid \exists p \in G \ p \Vdash \varphi\}$$

has a model (in fact, a *generic model*). The step from $P$ to $G$ is evidently a restriction on situations, while the formation of $\mathsf{T}_G$ is another persistent extension.

Insofar as talk of the many truth values in a Boolean algebra is replaced by talk of many situations, Keisler's formulation above can be considered a situation-theoretic approach to Boolean-valued models. (In connection with the notion of polarity described above in section 2, it is clear in this context that a polarity is *not* a truth value. This is hardly surprising, given the assertion in section 2.2 that infons are not true or false; propositions are.)

In the preceding example, strong persistence holds (for forcing properties, strong forcing and weak forcing)[12]; that is, the arrows between infon-packets are rather boring. In many cases (such as dynamic logic), strong persistence fails, and the nature of the arrows between infon-packets requires investigation. It is in these cases where situation theory should be especially relevant, because the shifts in situations cannot be ignored. As stressed by Barwise, common sense reasoning is such an example. In particular, non-monotonic reasoning can be viewed as a consequence of the failure of certain inference operators on support-maps to preserve strong persistence. Put more simply, non-monotonic reasoning is the natural outcome of the application of non-monotonic operators (see, for example, Jäger [18]).

[12] Related, under certain conditions, to forcing constructions are priority arguments from recursion theory. As pointed out to the author by Feferman, the notion of injury in priority arguments suggests that the natural logic of these arguments violates strong persistence.

## 4. Further work

Concerning the study of support-maps, two points, perhaps obvious, are, nonetheless, worth making. One, support-maps must be studied side by side (rather than in isolation). And two, the computational character of support-maps deserves investigation.

In view of the first point (— not to mention the definition of a support-map in categorical terms —), a category-theoretic approach to support-maps would appear reasonable. This is pursued in Fernando [17], where the relevance of certain developments in categorical logic is examined. As for the second point, recall that section 2 closed with the suggestion of a formalization of situation theory in explicit mathematics. The idea of a formalization in an untyped axiomatic setting (of which explicit mathematics is an example) goes back to Plotkin [23]. The author, however, was unable to appreciate the motivations behind such an approach until he observed certain similarities between explicit mathematics and situation theory described in Fernando [16].[13] While that paper spells out some guiding intuitions concerning operations, types, propositions and situations, it fails to forge a precise connection between situation theory and explicit mathematics. (A more precise — but decidedly partial — formulation of the connection is attempted in Fernando [17].)

In addition to these lines of investigation (which are of a more or less mathematical nature), there is the question of providing an account of the kind of "real world" examples of information flow in page 17 of Barwise and Perry [9], and page 44 of Barwise [6]. The present work considers examples of information flow that are of an abstract character of the sort considered in page 14 of Barwise [6], where the inventor's paradox is discussed in relation to getting "the flow of information to work out properly" to prove that

$$1 + 3 + \cdots + (2k+1) \quad = \quad (k+1)^2 \ .$$

The author regards information as being something abstract, and is not yet prepared to ascribe to infons the kind of physical existence ordinarily (and perhaps naively) ascribed to leptons. A treatment of "real world" examples would appear to require a theory of the "real world," and this is something that the author has tried to avoid (insofar as it is possible) by working on the *mathematics* of situation theory. Nevertheless, he does hope that such a theory can be superimposed on the framework described in sections 2 and 3 that would provide an account of more concrete cases of information flow. Ideas, for instance, about inquiry, inference (Burke [10]), and perspective (Seligman [25]) might be put in terms of movement between support-maps. The hope is that such exercises (grounding, as they do, a mathematical framework in a conception of "reality") may lead to further insights and intuitions about what situations and infons "really" are.

## 5. Conclusion

The reader may have noticed that the situation-theoretic approach sketched above goes against a certain trend in attempts at formalizing or mechanizing thought. Since Gödel's completeness theorem, researchers engaged in such attempts have often suppressed the notion of a situation, preferring syntactic formalisms that slight the significance of the notion. Implicit in this is perhaps a suspicion that the notion is computationally intractable. It is not clear to the author how to evaluate the validity of such concerns. It would appear that even without an explicit notion of situation, logic can be computationally hopeless (in any number of senses). The question then is whether serious semantic theories can be developed without the notion (recognized as distinct from other concepts such as infon and infon-packet). The author believes the answer to be "no," and trusts that the matter can be studied mathematically.

## Acknowledgments

This paper incorporates various notational suggestions by J. Barwise, under whom the author is a graduate student research assistant at CSLI. The author has also enjoyed the patient supervision of S. Feferman, whose interest in these matters the author deeply appreciates.

[13] Furthermore, the author recognized the fact that untyped theories provide some handle on computational character only after some exposure to the explicit mathematics literature.

This paper is a revision based on a number of comments by Feferman, T. Burke and Barwise of a technical report written during a visit to the Third Laboratory of ICOT. The author is very grateful to H. Yasukawa, K. Mukai and the other members of the laboratory for many interesting discussions, and also for their uncommon generosity, kindness, and support. He should like also to include S. Tutiya in this list, and thank S. Hayashi, D. Israel and G. O'Hair for some stimulating discussions.

## References

[1] S. Abramsky. In preparation.

[2] P. Aczel. Frege structures and the notion of proposition, truth and set, in *The Kleene Symposium*, ed. J. Barwise et al. North Holland, Amsterdam, 1980.

[3] P. Aczel. *Non-well-founded sets*. CSLI Lecture Notes Number 14, Stanford, 1988.

[4] P. Aczel. Replacement systems and the axiomatization of situation theory, in *Situation theory and its applications*, ed. R. Cooper et al. CSLI Lecture Notes Number 22, Stanford, 1990.

[5] J. Barwise. Notes on a model of a theory of situations. Unpublished notes, 1987.

[6] J. Barwise. *The situation in logic*. CSLI Lecture Notes Number 17, Stanford, 1989.

[7] J. Barwise and J. Etchemendy. *The liar: an essay on truth and circularity*. Oxford University Press, Oxford, 1987.

[8] J. Barwise and J. Etchemendy. Information, infons, and inference, in *Situation theory and its applications*, ed. R. Cooper et al. CSLI Lecture Notes Number 22, Stanford, 1990.

[9] J. Barwise and J. Perry. *Situations and attitudes*. MIT Press, Cambridge, MA, 1983.

[10] T. Burke. Dewey on defeasibility, in *Situation theory and its applications*, ed. R. Cooper et al. CSLI Lecture Notes Number 22, Stanford, 1990.

[11] K. Devlin. *Logic and information*. Cambridge University Press, Cambridge, to appear.

[12] S. Feferman. A language and axioms for explicit mathematics, in *Algebra and logic*, ed. J. Crossley. Springer-Verlag, Berlin, 1975.

[13] T. Fernando. On substitutional recursion over non-well-founded sets, in *Fourth annual symposium on Logic in Computer Science*. IEEE, Computer Science Press, Washington, D.C., 1989.

[14] T. Fernando. On the logic of situation theory, in *Situation theory and its applications*, ed. R. Cooper et al. CSLI Lecture Notes Number 22, Stanford, 1990.

[15] T. Fernando. Information and truth in situation theory. Unpublished manuscript, 1990.

[16] T. Fernando. Operations, types, propositions and situations. Unpublished manuscript, 1990.

[17] T. Fernando. Applying categorical logic to situation theory. In preparation.

[18] G. Jäger. Non-monotonic reasoning by axiomatic extensions, in *Logic, methodology and philosophy of science VIII*, ed. J. Fenstad et al. North Holland, Amsterdam, 1989.

[19] J. Keisler. Forcing and the omitting types theorem, in *Studies in model theory*, ed. M. Morley. The Mathematical Association of America, 1973.

[20] S. Kripke. Outline of a theory of truth. *The Journal of Philosophy*, 72, 1975.

[21] I. Lindstrom. A construction of non-well-founded sets within Martin Löf's type theory. *The Journal of Symbolic Logic*, 54, 1989.

[22] M. Mislove, L. Moss, F. Oles. Non-well-founded sets obtained from ideal fixed points, in *Fourth annual symposium on Logic in Computer Science*. IEEE, Computer Science Press, Washington, D.C., 1989.

[23] G. Plotkin. Notes on a formal theory and model for situation theory. Unpublished notes, 1987.

[24] G. Plotkin. An illative theory of relations, in *Situation theory and its applications*, ed. R. Cooper et al. CSLI Lecture Notes Number 22, Stanford, 1990.

[25] J. Seligman. Perspectives in situation theory, in *Situation theory and its applications*, ed. R. Cooper et al. CSLI Lecture Notes Number 22, Stanford, 1990.

[26] D. Westerstahl. Parametric types and propositions in first-order situation theory, in *Situation theory and its applications*, ed. R. Cooper et al. CSLI Lecture Notes Number 22, Stanford, 1990.