Formalizing Action using Set Theory and modified Pointwise Circumscription

Eyal Amir
Department of Computer Science,
Gates Building, 2A wing
Stanford University, Stanford, CA 94305-9020, USA
eyala@cs.stanford.edu

April 15, 1997

Abstract

We present an action formalism that uses Zermelo–Frankel Set Theory and a modified version of Pointwise Circumscription, to capture the idea of *solve the Frame Problem one situation at a time*. We show that the formalization has the semantics of [LR94], and that using a set of *Allowed-Situations* lets us develop a theory for the Missionaries and Cannibals Problem (MCP), and carry out reasoning both forward and backwards in time.

1 Introduction

One of the dominating directions of research in formalizing theories of action in the past several years, is the discretization of the space of situations, and the solution of the Frame Problem one situation at a time. This idea takes some of its major variants in [LR94, LS91, KL95] and [Bak91].

In this work we present an action formalism that is built on top of a version of Situation Calculus that uses Zermelo–Frankel Set Theory. In addition we use a modification of Pointwise Circumscription (cf [Lif87]), Categorization as a tool for causality (i.e., $FrameFluents$ (cf [Lif93])), and a distinction between Qualitative Constraints and Ramification Constraints. This formalism is shown to have the semantics of [LR94], and is adequate for proving some results on the *Missionaries and Cannibals Problem*.

The use of Set Theory is motivated by the problem of the *Missionaries and Cannibals*, which is detailed in section 3. In that problem, sets are required in order to reason explicitly on the groups of people translating from one bank of a river to another. Besides the explicit use of sets in our theory, ZFC Set Theory allows for more Elaboration Tolerance (see [MA96]), *denotation* and many useful notions (such as sequences of actions, functions, relations, and sets of fluents) are easily representable in a unified theory in 1st order logic.

Set Theory was previously considered to be a key tool in formalizing Common Sense in [McC85], [McC96a], and some interesting attempts were made by [Per87], [Per88], [Bar89], and others (see [PA95] for a survey), but no attempt at formalizing theories of action with it was made. Our chosen framework of *Zermelo-Frankel Set Theory with Choice* (henceforth ZFC) has some drawbacks (e.g., see [PA95]), but, as we shall see, no obstacle is encountered in our development and application.

Previously, [Lif87] argued that Pointwise Circumscription has the power to be a tool of the kind we exercise here, namely to formalize the solution of the Frame Problem. Our work with it is similar to the treatment of [DL94], but with three major differences. We use a modified Pointwise Circumscription, adhere to the semantics of [LR94], and use *Herbrand Situations* ([DL94] use time points). We will argue that the Herbrand Situations framework, with our use of Pointwise Circumscription makes many proofs easier. We shall see an example in our treatment of the MCP. With regard to the modification in Pointwise Circumscription, we will make it look for *minimal* instead of *minimum* models (next section). In Sandewall's scenario classes taxonomy, ours fits at least in $\mathcal{K}$-$IsAd$.

The paper is organized as follows. Section 2 Introduces a modified Pointwise Circumscription, shows its semantics, then briefly describes ZFC and the semantics of the different objects needed for our Action language. It completes with the formal definition of our action theory. Section 3 demonstrates the use of the theory with a medium-sized formalization of the Missionaries and Cannibals Problem (MCP). Finally we discuss the differences among the closer approaches and ours.

For background material on Circumscription, Action, and Set Theory, the reader is referred to [Lif93], [SS94], and [Kun80]. Most of the proofs are omitted, for lack of space, and can be found in the full paper in [Ami97].

2 Action Framework

Here we present a simple action formalism, similar to the one presented in [KL95] and with many similarities to [LR94], but with the use of a variant of *Pointwise Circumscription* proposed by [Lif87], and augmented by the language and theory of sets.

2.1 Pointwise Circumscription

Pointwise Circumscription captures the ideas behind *Circumscription* (cf [McC86]) (i.e., minimize a predicate P, allowing some predicates and functions Z to change, while asserting the axiom $A(P, Z)$), while making the circumscription process more *distributed* (i.e., making it on a different subset of the theory/domain each time). Pointwise Circumscription was built with demanding the minimized predicate P to be a *minimum* rather than *minimal*[1]. In our context, the original version of Pointwise Circumscription will be sufficient if we restrict ourselves to deterministic actions, and to cases where the description of action's effects are not dependent on other action's effects. Examples where such cases are not enough are demonstrated in section 3.4. Therefore, we present a version of Pointwise Circumscription where the minimized predicate is required to be only *minimal*.

Let $A(S_1, ..., S_n)$ be a sentence, where each S_i is a predicate symbol or a function symbol (in particular, it can be a 0-ary function symbol, i.e., an object constant). We want to minimize one of the *predicate* symbols from this list, S_{i_0} (Thus S_{i_0} corresponds to P, and the other members of the list correspond to Z in the notation used above). Let us write $EQ_r(p, q)$ ("p and q are equal outside r") for

$$\forall x(\neg r(x) \implies (p(x) \equiv q(x)))$$

for p, q predicates or functions (in which case we interpret $\equiv$ as $=$), and r a predicate, all with same arity. Our version of *Pointwise Circumscription* of

[1] Here *minimum* refers to an object that is smaller (in terms of the given partial order relation) than all other objects, while *minimal* refers to objects for which there is no smaller object.

S_{i_0} in A with S_i allowed to vary on V_i, denoted $\mathcal{C}[A; S_{i_0}; S_1/V_1, ..., S_n/V_n]$, is, by definition,

$$A(S) \wedge \forall xs \neg[s_{i_0} < S_{i_0} \wedge \bigwedge_{i=1}^{n} EQ_{V_i x}(s_i, S_i) \wedge A(s)] \tag{1}$$

where S stands for $S_1, ..., S_n$, s is a list $s_1, ..., s_n$ of predicate and function variables corresponding to the predicate and function constants S, and $\lambda x, u\ V_i(x, u)$ $(i = 1, ..., n)$ is a predicate without parameters which does not contain $S_1, ..., S_n$ and whose arity is the arity of S_{i_0} plus the arity of S_i (here $V_i x = \lambda u V(x, u)$). Also, $s_{i_0} < S_{i_0}$ is $(\forall x\ s_{i_0} x \rightarrow S_{i_0} x) \wedge S_{i_0} \not\equiv s_{i_0}$ (i.e., as sets, $s_{i_0} \subsetneq S_{i_0}$).

2.2 Semantics

Take w.l.o.g., $i_0 = 1$. For a model $\mathcal{M}$ of $A(S)$, let $|\mathcal{M}|$ be the associated universe, and for every term, function, or predicate a, $a^{\mathcal{M}}$ is the realization of a in $\mathcal{M}$. Let $\mathcal{M}_1, \mathcal{M}_2$ have the same universe U, and let $\xi \in U^k$, where k is the arity of S_1. We say that $\mathcal{M}_1 \leq^{\xi} \mathcal{M}_2$ iff

1. $K^{\mathcal{M}_1} = K^{\mathcal{M}_2}$ for every function or predicate constant K which is not in S,
2. for any $i = 1, ..., n$, $S_i^{\mathcal{M}_1}$ and $S_i^{\mathcal{M}_2}$ coincide on $\{\eta \mid \neg V_i^{\mathcal{M}_1}(\xi, \eta)\}$,
3. $S_1^{\mathcal{M}_1}(\xi) \implies S_1^{\mathcal{M}_2}(\xi)$.

We define the preference relation (cf [SS94]) $\ll$ among models which prefers one model over another if they have the same universe, and for every tuple ξ in their universe, the first is preferred over the second according to $\leq^{\xi}$. More formally, $\mathcal{M}_1 \ll \mathcal{M}_2$ (a strict partial order relation) iff

1. $|\mathcal{M}_1| = |\mathcal{M}_2|$, and
2. $\forall \xi \in |\mathcal{M}_1|^k\ \mathcal{M}_1 \leq^{\xi} \mathcal{M}_2$, and
3. $\exists \xi \in |\mathcal{M}_1|^k\ \neg S_1^{\mathcal{M}_1}(\xi)$ and $S_1^{\mathcal{M}_2}(\xi)$.

Let $[\![A(S)]\!]$ be the set of models of $A(S)$. The following proposition says that a model of the circumscription formula for $A(S)$ is minimal in $[\![A(S)]\!]$ according to $\ll$, and vice versa.

Proposition 2.1 *Let* $\mathcal{M} \in [\![A(S)]\!]$.

$$\mathcal{M} \models \mathcal{C}[A; S_1; S_1/V_1, ..., S_n/V_n] \iff \\ \forall \mathcal{M}' \in [\![A(S)]\!] \neg(\mathcal{M}' \ll \mathcal{M})$$

PROOF For the forward direction, assume $\mathcal{M}' \in [\![A(S)]\!]$ $\mathcal{M}' \ll \mathcal{M}$. Then $|\mathcal{M}'| = |\mathcal{M}|$ and there is $\xi \in |\mathcal{M}|^k$ such that $\neg S_1^{\mathcal{M}'}(\xi)$ and $S_1^{\mathcal{M}}(\xi)$. Take s to be $S^{\mathcal{M}'}$, and for $x = \xi$ we get that $s_1 < S_1$ and $A(s)$ and $\bigwedge_{i=1}^{n} EQ_{V_i x}(s_i, S_i)$ (the last one is because of the 2nd condition in the definition of the relation $\mathcal{M}' \ll \mathcal{M}$). Contradiction.

The reverse direction works by the same method. If $\mathcal{M}$ is minimal, but does not satisfy (1), then there are s, x such that $s_1 < S_1 \wedge \bigwedge_{i=1}^{n} EQ_{V_i x}(s_i, S_i) \wedge A(s)$. But then we can build a model $\mathcal{M}'$ with the same universe, taking all the constants other than those in S to be the same, and the constants of S take the values of s. We got a model $\mathcal{M}'$ with $\mathcal{M}' \ll \mathcal{M}$. Contradiction. ■

2.3 Introducing Set Theory into our formalism

We use the set theoretical framework of Zermelo-Frankel Set Theory with Choice (ZFC). The 1st order language of set theory has no object constants ($\emptyset$ can be added, but is not necessary), no function constants, and a single relation constant $\in$. For lack of space we do not display the axioms of ZFC here (see [Kun80] or the full paper).

In ZFC Set Theory, every object in the model is a set. Functions, relations, and other mathematical objects are represented as sets. Of course, one can add constants to our language that will be Relation, Function or Object constants, but as we shall see, there is no need for Relation and Function constants, unless one would like to talk about the complete theory, which we will try to avoid at this stage.

For lack of space, the more ordinary mathematical tools mentioned below are displayed without a formal definition (a formula in our language), which can be found in [Kun80] or the full paper.

$\cup$,$\cap$, and others $\bigcup \mathcal{F}$,$\bigcap \mathcal{F}$, $A \cup B$, $A \cap B$, $A \subset B$, $A \setminus B$, $A \triangle B$, finite sequences $\langle x_1, ..., x_n \rangle$, and $A \times B$. ω denotes the set of natural numbers, $Seq(A)$ the set of finite sequences of elements of A, and $len(a)$, and a_i are the length of the sequence a, and the i-th element.

Relations, Functions $Relation(R, n)$, $Function(f, n)$ are the formulas stating that R, f are relations and functions (respectively) of arity n. $dom(f)$, $rank(R)$, and $range(f)$ have the intuitive interpretation.

Situations and Actions These are simple objects (i.e., sets). No special treatment is needed here. We will note, though, that we have the set of all the situations *Situations* and the set of all the actions *Actions-set*.

Fluents If we wish to use fluents as functions of s, it is straightforward[2]. If we want to use $Holds$, it demands some further treatment. In that case, there are several approaches, from which we pick the following: A *fluent* l is an object (a set) in the set $Fluents$. Then, $Holds : Fluents \times Situations \rightarrow Values$, and we interpret $Holds(\Psi(l_1, ..., l_n), s)$ as the formula $\Psi(Holds(l_1, s), ..., Holds(l_n, s))$, for a formula Ψ with the only free variables $l_1, ..., l_n$.

We will use the notation that $R(a, b)$ to say that $\langle a, b \rangle \in R$. For example, take the blocks world sentence $on(A, B)$. A, B, on are *object* constants (and thus have realization as sets). on has the property that $Relation(on, 2) \wedge rank(on) = 2$. We can furthermore add $isblock(A), isblock(B)$. At this point we can say that $on \subset isblock \times isblock$. If we would like to include the table as an object, we can say $\neg isblock(Table)$, and replace the appropriate sentence above with $on \subset isblock \times (\{Table\} \cup isblock)$. This way, every relation that we add, is actually an object in the language, on which we can reason directly.

2.4 Persistence, Qualification and Ramification

2.4.1 Language

Our language is essentially that of the *Discrete Situation Calculus* of [LR94], but adjusted to fit our set theoretic language (described in section 2.3), and our solution to the Frame and Qualification problems (described below).

Since we rely on the language and theory of ZFC, we have no sorts in our language. There are no function or relation constants on top of the language of Set Theory, but we will add object constants for functions and

[2] Here, when we want to vary $Holds$, we will actually vary all the object constants for fluents.

relation, and say that such a constant is an *object constant for a relations* (or *function*).

There are object constants for the sets of *Situations* and *Actions-set*, and an object constant for a function $Actions : Situations \rightarrow \mathbb{P}(Actions\text{-}set)$, denoting the set of actions valid for consideration in a situation. There is a unique situation constant symbol, $S0$, denoting the initial situation. There is an object constant for a function *Result* with $dom(Result) = \{\langle a, s\rangle : s \in Situations\ a \in Actions(s)\}$, $ran(Result) \subset Situations$, and two object constants for binary relations $Can \subset dom(Result)$, and $<_s \subset Situations \times Situations$. ($s <_s s'$ means that s' can be obtained from s by a sequence of executable actions). The theory includes the following axioms (in the following $s, s' \in Situations$, $a \in Actions(s), a' \in Actions(s')$).

$$
\begin{array}{l}
S0 \in Situations \\
S0 \neq Result(a, s) \\
Result(a, s) = Result(a', s') \implies (a = a' \wedge s = s') \\
<_s = \left\{ \langle s, s'\rangle \;\middle|\; \begin{array}{l} \exists q \in Seq(Can)\ \exists a\ \langle a, s\rangle = q_1 \wedge \\ \forall i < len(q)\ \exists a\ \langle a, Result(q_i)\rangle = q_{i+1} \\ \wedge\ Result(q_{len(q)}) = s' \end{array} \right\}
\end{array}
$$

Here we see the first benefit from our Set Theory. We did not write down the induction axiom because it is embodied in our use of finite sequences for the definition of $<_s$.

In our treatment of fluents, we use the object constants $FrameFluents$ to denote the set of *basic* fluents[3], and the object constant for a function $Holds : Fluents \times Situations \rightarrow Values$, where $Values$ is the set of values each fluent can take (when a certain fluent can take only some values - we will implicitly have the right axiom for that). When using the alternative notation of $l(s)$ (instead of $Holds(l, s)$), we will actually refer to the $Holds$ notation. In the following $s \in Situations$, $a \in Actions(s)$ and $l \in Fluents$. A theory Γ in this framework is comprised of

ZFC The axioms of ZFC.

Σ_s The situation calculus axioms mentioned above.

$\mathcal{D}_{una}$ Unique names axioms.

[3]Notice that there are possibly many more fluents, which are objects in $Fluents$ (defined above), for which $Holds$ is defined.

$\mathcal{D}_{ef}$ Effect axioms of the form

$$\begin{array}{l} Can(a,s) \implies [\Psi^+ \implies Holds(l, Result(a,s))] \\ Can(a,s) \implies [\Psi^- \implies \neg Holds(l, Result(a,s))] \end{array}$$

$\mathcal{D}_{qual}$ Qualification axioms of the form

$$Can(a,s) \implies \varphi(a,s)$$

$\mathcal{D}_{dom}$ Domain constraints $Dom(s)$.

$Dom(s)$ split into two: Ramification Constraints $RC(s)$, and Qualification Constraints $QC(s)$. This split was first observed by [LR94], and we discuss it further when we deal with Qualifications. Both $QC(s)$ and $RC(s)$ are conjunctions of the right domain axioms, with s free.

2.4.2 Persistence and Ramification

Let $S_1 = Ab$, where Ab is an object constant for a relation on three objects, with the following axioms:

$$\begin{array}{l} Ab(l,a,s) \implies l \in FrameFluents \wedge Can(a,s) \\ \forall l \in FrameFluents\ \forall s \in Situations\ \forall a \in Actions(s) \\ \quad (\neg Ab(l,a,s) \implies \\ \qquad (Holds(l,s) \iff Holds(l, Result(a,s)))) \end{array} \tag{2}$$

The above formula is the persistence formula, and is denoted $\mathcal{P}$.

We now specify the circumscription policy. Let $S_2 = Holds$, $V_1(x,u)$ be the formula $\lambda x, u\ \exists l_x, a_x, s_x,\ \exists l_u, a_u, s_u\ x = \langle l_x, a_x, s_x \rangle \wedge\ u = \langle l_u, a_u, s_u \rangle\ \wedge s = s_u$, and $V_2(x,u)$ be the formula $\lambda x, u\ \exists l_x, a_x, s_x,\ \exists l_u, a_u, s_u\ x = \langle l_x, a_x, s_x \rangle \wedge u = \langle l_u, s_u \rangle \wedge Result(a_x, s_x) = s_u$.

Intuitively, we minimize Ab on one situation s at a time, letting $Holds$ change only for the result situation $Result(a,s)$. This is the application of the *Solve the frame problem one situation at a time* paradigm, mentioned above. We get that $\mathcal{C}[\Sigma; Ab; Ab/V_1, Holds/V_2]$ is

$$\begin{array}{ll} \multicolumn{2}{l}{\Sigma(Ab, Holds) \wedge \forall x, y, ab, holds} \\ \multicolumn{2}{l}{\quad \neg[\Sigma(ab, holds) \wedge ab < Ab \wedge} \\ V_1: & (s_x \neq s_y \Rightarrow ab(y) \equiv Ab(y)) \wedge \\ V_2: & (Result(a_x, s_x) \neq s_y) \Rightarrow holds(y) \equiv Holds(y)] \end{array} \tag{3}$$

where x, y are triplets of the forms $\langle l_x, a_x, s_x \rangle, \langle l_y, a_y, s_y \rangle$ for V_1, and the triplet $\langle l_x, a_x, s_x \rangle$ and tuple $\langle l_y, s_y \rangle$ for V_2. We call this formula $\mathcal{A}(\Sigma)$.

Notice that as far as nondeterministic actions are concerned, those that are expressible in our language (we do not use *Occlusion* here) are treated naturally by the formalism, and our augmentation of Pointwise Circumscription allows for several minimal models.

In the following (a minor syntactic modification of the semantics of [LR94]), we abuse notations, and write $M \models \Phi(l, a, s)$ for l, a, s objects in $|M|$ instead of using variable assignments.

Definition 2.2 ([LR94]) *A model M of a theory W is* "minimal" *iff there is no other model M' of W, and an object $s \in Situations^M$ such that*

1. $|M| = |M'|$.
2. *M and M' differ only in their interpretations of $Holds, Ab$.*
3. *For any fluent $l \in Fluents$, $M \models Holds(l, s) \iff M' \models Holds(l, s)$*
4. *For any fluent $l \in Fluents$, and $a \in Actions(s)$, $M \models Can(a, s) \wedge \neg Ab(l, a, s)$ implies $M' \models \neg Ab(l, a, s)$*
5. *There is $l \in FrameFluents$ and $a \in Actions(s)$, such that $M, \models Can(a, s) \wedge Ab(l, a, s)$ but, $M' \models \neg Ab(l, a, s)$*

Let Γ, be all the theory developed above, including $\mathcal{P}$.

Theorem 2.3 *$\mathcal{M} \models \mathcal{A}(\Gamma)$ iff $\mathcal{M}$ is a* minimal *model of Γ.*

This theorem holds for both deterministic and nondeterministic actions (as far as our language allows us to specify those).

2.4.3 Qualification

To determine the preconditions of actions, we add the axioms

$$\begin{array}{l} Abq \subset dom(Result) \\ \forall s \in Situations\ a \in Actions(s) \\ \qquad\qquad \neg Abq(a, s) \implies Can(a, s) \end{array} \tag{4}$$

Every precondition $\varphi(a, s)$ for an action a is expressed using an axiom of the form.

$$Can(a, s) \implies \varphi(a, s)$$

We would like to conclude that an action is applicable iff its preconditions are met, and it does not lead us to violations of our QC. We add the object constant *Allowed-Situations* $\subset$ *Situations* and the object constant for a relation *Applicable*, and say

$$\begin{array}{l}
\forall s \in Situations\ RC(s) \\
\forall s \in Allowed\text{-}Situations\ QC(s) \\
\forall s \in Situations \setminus Allowed\text{-}Situations\ \forall a \in Actions(s) \\
\qquad\qquad Result(a, s) \notin Allowed\text{-}Situations \\
S0 \in Allowed\text{-}Situations \\
Applicable(a, s) \iff \\
\qquad Can(a, s) \wedge Result(a, s) \in Allowed\text{-}Situations
\end{array}$$

These axioms make sure that if the situation we are in is *Allowed* then every situation on the way is also *Allowed*.[4] Now, we simply aggregate the preconditions for $Can(a, s)$ using

$$\begin{array}{l}
\mathcal{C}[\Gamma; Abq; Abq/V_1, Can/V_1, Ab/V_1, Holds/V_2, \\
\quad Allowed\text{-}Situations/True, Applicable/True] \\
\\
Can(a, s) \iff \varphi_1(a, s) \wedge ... \wedge \varphi_n(a, s)
\end{array} \tag{5}$$

Since the solution to the Persistence Problem depends on Can alone, we succeed in producing the changes for $Result(a, s)$. We are left to demand that the end/goal situation is an *Allowed-Situation* (somewhat similar to Observations in Sandewall's PCMF (Filter Pref. Entailment)).

As mentioned above, the split of domain constraints to RC and QC was first observed in [LR94]. They do not give means for deciding where to put a domain constraint, though. The above framework gives us a proposed definition. $l \in QC \implies$ violating l makes the situation invalid. $l \in RC \implies$ violating l changes some relations/functions (e.g., $Holds/Can$)[5]. This definition does not give us a syntactic solution, but rather turns to the knowledge

[4] See the Persistence Problem treatment for the MCP in section 3.3.

[5] by $l \in QC$ ($l \in RC$) we mean that the compound fluent (cf [Lif93]) l is in the conjunction of QC (RC).

engineer to make the distinction, according to her perception of the knowledge. (e.g., putting a block on top of an unclear block, causes the block initially on the top to fall) are in RC. Formulas that describe external restrictions on the situation (e.g., a block can have only one thing on top) are in QC.

Recently [Thi97] showed that causality can be accounted for using *Causal Relationships* (instead of *Causal Rules* ([MT95]), or Categorization (e.g., [Lif93])). His analysis does not include a split of Domain Constraints to Ramification and Qualification constraints, but he rather uses Causal Rules instead of our Ramification constraints, where Qualification constraints are given no special treatment. His approach therefore strengthens our analysis of the distinction, as Ramifications are those that are causal, while Qualifications are not.

The next section demonstrated the way this solution for the Qualification and Persistence Problems work.

3 The Missionaries and Cannibals Problem

The missionaries and cannibals problem (abbreviated MCP) is stated as follows:

> **Three missionaries and three cannibals come to a river and find a boat that holds two. If the cannibals ever outnumber the missionaries on either bank, the missionaries will be eaten.**
>
> **How shall they cross?**

In this section we develop a formalism to solve this problem, and show that our formalism allows us to prove that both the *3 missionaries, 3 cannibals* problem is solvable ($MCP(3,3)$), and that the *4 missionaries, 4 cannibals* version, is not ($\neg MCP(4,4)$). It can be argued that Set Theory is not needed for the formalization of the MCP, and that the rather simple Amarel formalization can be used. We chose to use Set Theory here because it allows us to express many more elaborations and common sense facts (see [McC96b],[MA96]).

3.1 Basic Formulas (mainly action related)

The following formulas describe the relations and objects in the basic story.

3.1.1 Language

Our language will include (besides the constants of the above framework) the following object constants: M, C, $Boat$, $Banks$, which are the sets of Missionaries and Cannibals, the Boat, and the set of Banks, respectively.

For simplification, we restrict ourselves to actions of the form

$$A(group, bank) = using(Boat, go(group, bank))$$

Notice that the Boat is fixed, and we therefore implicitly assume that the boat is needed, that there is exactly one, and that there are no other actions possible. We further restrict the set of actions to be

$$Actions(s) = \left\{ \begin{array}{l|l} A(group, & bank \in Banks \wedge \\ bank) & group \subset inhab(\\ & \quad opp(bank), s) \end{array} \right\} \tag{6}$$

All of the notions used in the above definitions are defined below.

3.1.2 Qualification

Let A be a macro for $A(group, bank)$.

$$\begin{array}{l} \neg Abq(A, s) \rightarrow Can(A, s) \\ Can(A, s) \rightarrow A \in Actions(s) \\ Can(A, s) \rightarrow 0 < card(group) \\ Can(A, s) \rightarrow (\forall p \in group) \\ \qquad (location(p, s) = opp(bank)) \\ Can(A, s) \rightarrow (\forall p \in group) \\ \qquad (location(p, Result(A, s)) = bank) \\ Can(A, s) \rightarrow location(Boat, Result(A, s)) = bank \\ Can(A, s) \rightarrow card(group) \leq capacity(Boat) \\ Can(A, s) \rightarrow water\text{-}crosser(Boat) \\ Can(A, s) \rightarrow available(Boat, group, s) \end{array} \tag{7}$$

3.1.3 Counting the Missionaries and Cannibals

The only formulas we need for our previous definitions/formulas are

$$\begin{array}{l} inhab(bank, s) = \{x \in M \cup C \mid location(x, s) = bank\} \\ m(s, bank) = card(\{m \in M | location(m, s) = bank\}) \\ c(s, bank) = card(\{c \in C | location(c, s) = bank\}) \\ ok(s) \equiv \forall b \in Banks\ \neg(0 < m(s, b) < c(s, b)) \wedge \\ \qquad\qquad s \in Situations \\ QC(s) \equiv ok(s) \end{array} \tag{8}$$

The last sentence says that the only Qualification Constraint is *ok*. *RC* is the rest of the domain constraints mentioned, i.e., the law about *available* below.

3.1.4 Commonsense Axioms

The above axiom set relied on the following definitions and axioms.

$$\begin{array}{l} \{Lb, Rb\} = Banks \wedge Lb \neq Rb \wedge opp(Lb) = Rb \wedge \\ opp(Rb) = Lb \wedge isboat(Boat) \wedge capacity(Boat) = 2. \\ isboat(boat) \rightarrow water\text{-}crosser(boat) \\ available(means, group, s) \iff isboat(means) \wedge \\ \quad group \subset inhab(location(means, s), s) \end{array} \tag{9}$$

3.2 Closing our domain

We would like to conclude that if an action of going, or using the boat to go, is feasible, we maintain the restrictions over the relative quantities of missionaries and cannibals. By using fixed sets of Missionaries and Cannibals (M,C), and fixed sets of all the other objects under consideration, we solve many possible frame problems, such as humans appearing or disappearing suddenly (here we restrict many axioms in order to make our theory simpler). The following axioms deal with these restrictions and the *initial situation*.

$$\begin{array}{l} m(S0, Lb) = M \\ c(S0, Lb) = C \\ location(Boat, S0) = Lb \end{array} \tag{10}$$

3.3 The Nonmonotonic Framework

We use the framework developed above to solve the Qualification and Persistence Problems. Let $FrameFluents = \{location(obj) \mid obj \in M \cup C \cup \{Boat\}\}$.

Lemma 3.1 *The Qualification solution (5) entails*

$$\begin{array}{l} Can(using(Boat, go(group, bank)), s) \iff \\ \quad bank \in Banks \wedge isboat(Boat) \wedge \\ \quad 0 < card(group) \leq capacity(Boat) \wedge \\ \quad location(Boat, s) = opp(bank) \wedge \\ \quad (\forall p \in group)(p \in M \cup C \wedge \\ \qquad location(p, s) = opp(bank)) \end{array} \tag{11}$$

The reason for not including $Can(a, s)$ in the antecedent of our inertia law (2) will now become apparent. If we add $Can(a, s)$ to the antecedent, we know nothing about results of situations in which $Can(a, s)$ is not true. This way, people can "jump" from one bank to the other, still maintaining *Allowed-Situations*-ness. The following lemma summarizes the results of our inference.

Lemma 3.2 *(1) Nothing Changes but what ought to.*

$$\begin{array}{l} \forall s \in Situations\ \forall A \in Actions(s)\ (Can(A, s) \implies \\ \quad (\forall p \in group)(location(p, Result(A, s)) = bank) \wedge \\ \quad location(Boat, Result(A, s)) = bank \wedge \\ \quad (\forall p \in (M \cup C \setminus group)) \\ \qquad (location(p, Result(A, s)) = location(p, s))) \end{array}$$

(2) If something changed, then a possible *action occurred.*

$$\begin{array}{l} \forall s \in Situations \forall A \in Actions(s) \exists p \in M \cup C \cup \{Boat\} \\ (location(p, Result(A, s)) \neq location(p, s) \implies \\ \qquad Can(A, s)) \end{array}$$

Let $MCP(m, c)$ be the formula $\lambda m, c\ card(M) = m \wedge card(C) = c \wedge \exists s' \in$ *Allowed-Situations* $S0 \leq_s s'$.

Conclusion 3.3 *Let* MCP(m,c) *be the above theory with* $card(M) = m \wedge card(C) = c$.

$$\begin{array}{l} \mathcal{A}(\Gamma \wedge \text{MCP(3,3)}) \models MCP(3, 3) \\ \mathcal{A}(\Gamma \wedge \text{MCP(4,4)}) \models \neg MCP(4, 4). \end{array}$$

3.4 Elaborations on MCP

To articulate the power of using the proposed Pointwise Circumscription, consider the following examples:

- The action of crossing the river picks people according to their ethnic group, but is nondeterministic with regard to the actual individuals

  In this case we get a nondeterministic action, and using the original Pointwise Circumscription would lead to inconsistency (there is no minimum Ab (see formula (2))).

- Let us add the status of the cannibals being $Hungry$ (as a group). In $S0$ they are not hungry, but there is an action that makes them hungry, and there is exactly one such action (same action for all situations): $\neg Hungry(S0)$, and $\exists! a \forall s \in Situations\ a \in Actions(s) \wedge Hungry(Result(a, s))$.

  This is a different kind of nondeterminism, where the nondeterminism is not in the action, but rather in our knowledge. In this case the original Pointwise Circumscription given us inconsistency again (there is no minimum Ab).

- We will use set theory more vigorously in the following case. Let us expand the MCP to multinational participants, and let higher-level restrictions be in place for pacts between groups of people. More precisely, we say that there should not be on either bank a majority of pacts that are hostile. In this case we should represent pacts between sets of people, and reason about sets of pacts. We can even expand that to pacts between blocks of groups, and so on to any degree[6].

4 Discussion

Some comparisons are essential. First and foremost, our formalization is the only one we know to use Set Theory. Moreover, our formalization is done with the formal tool of Pointwise Circumscription, while [KL95] use Nested Abnormality Theories, and do not refer to the Qualification Problem, and [LR94] use a two-phase algorithm. The use of Pointwise Circumscription

[6] Although it seems that at some point this tower is becoming artificial, it does make our point though.

style allows us to blend the two stages of the solution, and treat it as one (although we did not bother to write the policy down in this paper).

Our use of Pointwise Circumscription style also allowed us to argue that *Qualification Constraints* are not *Real World* constraints (as the *Ramification Constraints* are), but superficial constraints imposed by us to either (1) shortcut some physical phenomenon (e.g., with the MCP, the cannibals will eat the missionaries), or (2) block some situations from materializing (e.g., "the king said that only one block can be colored yellow at one time" (an example taken from [LR94])). As such, the "label" of such a situation as *Allowed* makes more sense (see also our argument at the end of section 2.4.3).

In [LR94] Ghost-situations (situations that are results of impossible actions) are treated as situations we know nothing about. For inductive reasoning (e.g., proving that the MCP with 4 missionaries and 4 cannibals is unsolvable) this is insufficient[7]. This is the reason for us omitting the *Can* from the antecedent of the persistence axiom (2), which is also demonstrated in lemma 3.2. Lemma 3.2 also demonstrates the easiness of proving frame sentences with our framework. Whereas in [DL94] change in one time point may influence many later time points, here we are restricted to exactly *two* situations with which we work at a time (Here we mostly refer to the use of Pointwise Circumscription style, and the isolation of reasoning to two situations at a time). We found the proof to be quite straightforward (see [Ami97] for the proofs omitted here).

5 Conclusions

We present a simple action framework in which Set Theoretic Situation Calculus is simply and naturally demonstrated. The use of *Augmented Pointwise Circumscription* allows us to deal simply with nondeterministic actions (we did not experiment with *Occlusion*), and the reduction of reasoning to two situations at a time simplifies many proofs.

On the down side, the use of Categorization as a means for dealing with

[7]To see that, notice that $\neg MCP(4,4)$ means that there is no situation s' with the required missionaries and cannibals on the right bank of the river, such that $S0 \leq_s s'$. But then, since we know nothing about ghost-situations (more precisely - the frame axiom does not apply for impossible actions), such a situation and a sequence of actions can actually occur!

causality limits its application as observed by [Thi97]. As far as the MCP is concerned, Categorization is enough for the examples we examined.

The use of Set Theory makes the theory more Elaboration Tolerant (cf [MA96]) and the theories are more clear and concise. *Isabelle* is a theorem prover that among other things deals with Set Theory in an efficient way. Our future plan includes running some experiments with variants of the Missionaries and Cannibals on it. So far, our exercises with the *PVS* Theorem Prover did not come successful (we were able to prove MCP(3,3), but not before sweating considerably), as PVS's library for Set Theory and strategies were not powerful enough for our purposes.

6 Acknowledgments

John McCarthy, with the help of Tom Costello and Sasa Buvac, came up with the first formalization of the Missionaries and Cannibals Problem, without which this work was not possible. We owe many thanks to Vladimir Lifschitz and Tom Costello for reading and commenting on drafts of this paper. Tom Costello also pointed out some works we were not aware of, notes for which we are indebted. We also wish to thank the anonymous referees for their many important notes and insights. This research was supported by an ARPA (ONR) grant N00014-94-1-0775.

References

[Ami97] Eyal Amir. Formalizing Action using Set Theory and Pointwise Circumscription - extended notes. http://www-formal.stanford.edu/eyal/nmr/foundations.ps, 1997.

[Bak91] Andrew B. Baker. Nonmonotonic reasoning in the framework of the situation calculus. *Artificial Intelligence*, 49:5–23, 1991.

[Bar89] Jon Barwise. Afa and the unification of information. In Jon Barwise, editor, *Situation in Logic*, pages 277–283. Center for the Study of Language and Information, Stanford, CA, 1989.

[DL94] Patrick Doherty and Witold Lukaszewicz. Circumscribing features and fluents. In D. Gabbay and H.J. Ohlback, editors, *Proceedings of the 1st int'l conf. on Temporal Logic*, pages 82–100, 1994.

[KL95] G. N. Kartha and V. Lifschitz. A simple formalization of actions using circumscription. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1995.

[Kun80] K. Kunen. *Set Theory, an introduction to independence proofs*. North-Holland, New York, 1980.

[Lif87] Vladimir Lifschitz. Pointwise circumscription. In Matthew Ginsberg, editor, *Readings in nonmonotonic reasoning*, pages 179–193. Morgan Kaufmann, San Mateo, CA, 1987.

[Lif93] Vladimir Lifschitz. Circumscription. In J.A.Robinson D.M. Gabbay, C.J.Hogger, editor, *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning*. Oxford University Press, 1993.

[LR94] Fangzhen Lin and Raymond Reiter. State constraints revisited. *Journal of Logic and Computation, Special Issue on Actions and Processes*, 1994.

[LS91] Fangzhen Lin and Yoav Shoham. Provably correct theories of action: Preliminary report. In *Proceedings of the Ninth National Conference on Artificial intelligence (AAAI-91)*, 1991.

[MA96] John McCarthy and Eyal Amir. Missionaries and Cannibals: Making it Elaboration Tolerant. http://www-formal.stanford.edu/eyal/nmr/m4.ps, 1996.

[McC85] John McCarthy. Acceptance address, 1985. IJCAI Award for Research Exellence.

[McC86] John McCarthy. Applications of Circumscription to Formalizing Common Sense Knowledge. *Artificial Intelligence*, 28:89–116, 1986. Reprinted in [McC90].

[McC90] John McCarthy. *Formalization of common sense, papers by John McCarthy edited by V. Lifschitz*. Ablex, 1990.

[McC96a] John McCarthy. From Here to Human-Level AI. In *KR96 Proceedings*, 1996. Available as http://www-formal.stanford.edu/jmc/human.html.

[McC96b] John McCarthy. Missionaries and Cannibals: Making it Elaboration Tolerant. In *McCarthy's web page*, 1996. Available as http://www-formal.stanford.edu/jmc/cs323/missionaries.ps.

[MT95] N. McCain and H. Turner. A causal theory of ramifications and qualifications. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1978–1984, 1995.

[PA95] M. Pakkan and V. Akman. Issues in Commonsense Set Theory. *Artificial Intelligence Review*, 8:279–308, 1995.

[Per87] Donald Perlis. Circumscribing with sets. *Artificial Intelligence*, 31:201–211, 1987.

[Per88] Donald Perlis. Commonsense set theory. In P. Maes and D. Nardi, editors, *Meta-Level Architectures and Reflection*, pages 87–98. Elsevier, Amsterdam, 1988.

[SS94] Erik Sandewall and Yoav Shoham. Nonmonotonic Temporal Reasoning. In D.M. Gabbay, C.J.Hogger, and J.A.Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 4: Epistemic and Temporal Reasoning*. Oxford University Press, 1994.

[Thi97] Michael Thielscher. Ramification and causality. *Artificial Intelligence*, 89(1-2):317–364, 1997.