Small notes on missionaries—try 4 [*]

John McCarthy   Eyal Amir

April 1, 1996

1 Principles

Here are some principles of this try. When this memo is debugged it will be glued into the earlier "Missionaries and Cannibals: Making it Elaboration Tolerant".

1. Using situation calculus and not asserting that situations that agree on a specific set of fluents are equal allows greater elaboration tolerance than the state based formalisms of the previous sections. In that formalism, two states were equal if the missionaries, cannibals and the boat were on the same bank. In the situation calculus formalism, new objects such as the oars can be added.

2. Groups of missionaries or cannibals will be identified with the sets of their members. I think we may lose some elaboration tolerance from this, but I don't have a definite reason for believing it.

3. When an action succeeds, the result of trying the action is equal to the result of doing the action. In missionaries and cannibals we use $tryg$, since who tries is not really identified, and we can suppose it is some god.

4. The places are lb and rb. In this try we do not take into account the fact that a river is long. There is water between lb and rb.

5. Water can be crossed only by some special method. In this case, the only method mentioned is the boat. If it were not mentioned, we could not infer that there was a solution.

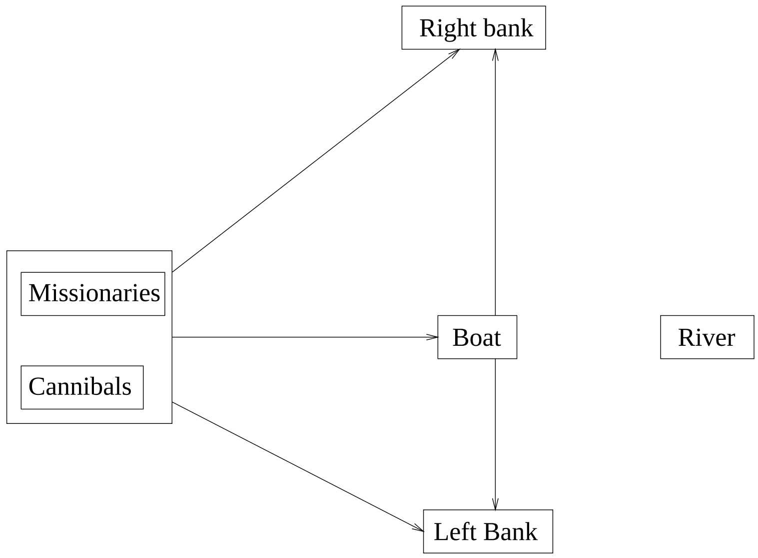
6. We use the notion of the location of a group. It is undefined if the members of a group are not in the same location. We expect to use a conventional value for "undefined", but the point hasn't come up yet.

7. Humans have the information elaborated already. We examine also the restrictions implied on the inner elaborated formalism, once the information is not elaborated but rather implicit. This is a good test of our elaboration tolerance (see also elaboration 19 in section 2).

[*] Amended and Augmented version.

2 Formulas

We formulate the relations in the following figure:



The following formulas describe these relations and objects.

2.1 Basic Formulas

1. The basic formula relating trying and succeeding is

$$can(a, s) \rightarrow result(tryg(a), s) = result(a, s) \tag{1}$$

   It doesn't say that $tryg(a, s)$ may not have the same effect as a when $can(a, s)$ is false. Therefore, it may have to be supplemented when we want to prove that a goal cannot be achieved, e.g. when there are four each missionaries and cannibals. I still hope to get by saying very little and relying on nonmonotonic reasoning to infer that the problem is not solvable, because no solution can be given.

2. For the time being we will assume that an action can be performed unless there is some reason to believe otherwise. This is expressed by

$$\neg ab1(a, s) \rightarrow can(a, s) \tag{2}$$

   I think this can't be right, but I don't yet see what trouble it will give[1].

[1] Eyal:I think we will have problems after minimizing this ab1, because we will not be able to prove that something is not possible... or it least it will force us to formalize our 'can' differently (what implies what ...)

3. Here are the river banks and the wet river in between.

$$\begin{aligned}&lb, rb \in banks \wedge lb \neq rb \wedge opp(lb) = rb \wedge opp(rb) = lb \\ &between(River, lb, rb) \wedge watery(River, S0)\end{aligned} \tag{3}$$

4. Commonsense knowledge

$$between(a,b,c) \rightarrow between(a,c,b) \tag{4}$$

5. Here's the boat.

$$isboat(Boat) \wedge capacity(Boat) = 2. \tag{5}$$

6. Data on the river And some relations between the several predicates describing the river.

$$\begin{aligned}(\exists river) watery(river, s) \wedge between(river, bank1, bank2, s) \\ \equiv water\text{-}between(bank1, bank2, s)\end{aligned} \tag{6}$$

7. Here are some preconditions for going.

$$\begin{aligned}&can(a,s) \rightarrow ok(s) \\ &can(go(group, bank), s) \rightarrow 0 < card(group) \\ &can(go(group, bank), s) \rightarrow (\forall p_1, p_2 \in group)(location(p_1, s) = location(p_2, s))\end{aligned} \tag{7}$$

8. Here are some preconditions for using a mean of transportation (e.g., the boat)

$$\begin{aligned}&can(using(mean, go(group, bank1)), s) \wedge \\ &water\text{-}between(bank1, bank2, s) \wedge \\ &location(group, s) = bank2 \rightarrow \\ &water\text{-}crosser(mean) \wedge available(mean, group, s) \\ &can(using(mean, go(group, bank)), s) \rightarrow card(group) \leq capacity(mean) \\ &can(using(mean, go(group, bank)), s) \rightarrow can(go(group, bank), s)\end{aligned} \tag{8}$$

9. The basic axiom for the result of going is

$$can(go(x,l),s) \rightarrow location(x, result(go(x,l),s)) = l, \tag{9}$$

10. Boats are water crossers.

$$\begin{array}{l} isboat(boat) \rightarrow water\text{-}crosser(boat) \\ isboat(boat) \rightarrow accessible(boat, location(boat, s)) \end{array} \tag{10}$$

11. If a mean of transportation is in an accessible place to the group, and has the right capacity, then it is available.

$$\begin{array}{l} water\text{-}crosser(mean) \wedge available(mean, group, s) \wedge \neg ab2(group, b, s) \\ \rightarrow accessible(mean, location(group, s), s) \wedge card(group) \leq capacity(mean) \end{array} \tag{11}$$

12. If the object x uses the boat, we also have

$$\begin{array}{l} isboat(x) \wedge occurs(using(x, go(group, bank)), s) \\ \rightarrow occurs(go(x, bank), s) \end{array} \tag{12}$$

13. It is a special fact that when a boat is used to cross a river, the boat arrives at the same time as the group of passengers. If a circus cannon were used, the action of the cannon would be finished before the arrival of the passengers. Since it is a special fact, we need to be able to say it. Here we do it by equating the result of going to the other bank using the boat, the result of just going to the other bank and the result of the boat going to the other bank. I guess this commits us to linear time rather than branching time.

$$\begin{array}{rcl} \multicolumn{3}{l}{occurs(using(boat, go(group, bank)), s) \rightarrow} \\ result(using(boat, go(group, bank)), s) & = & result(go(group, bank), s) \\ & = & result(go(boat, bank), s) \end{array} \tag{13}$$

14. In the standard case of MCP, we should be able to infer

$$\begin{array}{l} location(grp, s) = bank \wedge location(boat, s) = bank \wedge ok(s) \wedge card(grp) \leq capacity(boat) \\ \rightarrow can(using(boat, go(grp, opp(bank))), s). \end{array} \tag{14}$$

This will involve some nonmonotonic reasoning.

2.2 Counting the Missionaries and Cannibals

1. We make some definitions. Their goal is to assist us in deriving a form of the problem in which only the numbers of missionaries and cannibals on

each bank need be taken into account, provided no properties of particular missionaries and cannibals are involved in the preconditions for the actions.

$$\begin{array}{l} m(s, bank) = card(\{m \in missionaries | location(m, s) = bank\}) \\ c(s, bank) = card(\{c \in cannibals | location(c, s) = bank\}) \end{array} \tag{15}$$

2. A situation is ok if the missionaries are not outnumbered on either bank of the river.

$$ok(s) \rightarrow \neg(0 < m(s, lb) < c(s, lb)) \wedge \neg(0 < m(s, rb) < c(s, rb)) \tag{16}$$

3. It is convenient to make the definitions

$$inhabitants(bank, s) = \{x \in missionaries \cup cannibals | location(x, s) = bank\}$$

$$location(group, s) = bank \equiv (\forall x)(x \in group \rightarrow location(x, s) = bank)$$

$$\begin{array}{l} m(group) \equiv card(group) = 1 \wedge group \subset missionaries \\ mm(group) \equiv card(group) = 2 \wedge group \subset missionaries \\ c(group) \equiv card(group) = 1 \wedge group \subset cannibals \\ cc(group) \equiv card(group) = 2 \wedge group \subset cannibals \\ mc(group) \equiv card(group) = 2 \wedge (\exists x)(x \in group \cap missionaries) \\ \qquad\qquad\qquad\qquad\qquad \wedge (\exists x)(x \in group \cap cannibals) \end{array} \tag{17}$$

2.3 The initial situation $S0$

1. Some missionaries, cannibals and boats are initially at the left bank lb.

$$\begin{array}{l} m0 \subset missionaries \\ c0 \subset cannibals \\ card(m0) = 3 \\ card(c0) = 3 \end{array} \tag{18}$$

$$\begin{array}{l} location(m0, S0) = lb \\ location(c0, S0) = lb \\ location(Boat, S0) = lb \end{array} \tag{19}$$

2. We will need some nonmonotonic reasoning to infer that these are all the missionaries, cannibals, boats and other *relevant* objects present at lb and rb in the initial situation $S0$.

2.4 The Actions

1. The first action is a group $g1$ of two cannibals crossing the river and we axiomatize trying it.

$$\begin{array}{l} cc(g1) \wedge location(g1, S0) = lb \\ S1 = result(tryg(go(g1, rb)), S0) \end{array} \tag{20}$$

2. We can generalize this action to the following

$$\begin{array}{l} available(boat, group, S0) \\ location(group, S0) = lb \\ S1 = result(tryg(using(boat, go(group, rb))), S0) \end{array} \tag{21}$$

3. And generalizing it even more yields:

$$\begin{array}{l} available(mean, group, S0) \\ S1 = result(tryg(using(mean, go(group, bank))), S0) \end{array} \tag{22}$$

3 Nonmonotonic Reasoning

We proceed informally at first. Here are some of the defaults needed in missionaries and cannibals.

1. The relevant people present are the missionaries and cannibals asserted to be present.
2. The available ways of crossing the river are those mentioned, i.e. the boat.
3. There is no way to revive dead missionaries, so that there is no use to continue investigating a case where a missionary was already eaten.
4. Everything functions normally, except as stated. The boat works and the cannibals and missionaries do what is laid out for them.
5. Many of the elaborations can be handled as additional statments with a new model determined by the old circumscription policy.

It looks like all the nonmonotonic reasoning required in all the elaborations being considered can be done before beginning the reasoning about specific actions starting from $S0$. The results of this nonmonotonic reasoning will then be monotonic theories. It looks like these theories can all be state vector theories. The elaborations will generate additional components of the state vectors and modify the actions that can be taken.

4 Elaborating the Formalism

Recall the set of elaborations mentioned in [?]. We will examine how these elaborations are formulated in the proposed formalism.

1. *There are four missionaries.* We should amend formula 18 with the following:

$$card(m0) = 4$$
$$card(c0) = 4$$

2. *One person can row the boat with one oar, but two oars are needed to carry two people. There is an oar on each bank of the river.* We add a requirement to the predicate *available* as follows:

$$available(boat, group, s) \rightarrow card(oars(boat, s)) \geq card(group)$$

adding the oars to the game:

$$\begin{aligned} & oar1, oar2 \in oars\_set \\ & location(oar1, S0) = lb \\ & location(oar2, S0) = rb \\ & oars(boat, s) = \{oar \in oars\_set \mid location(oar, s) = in(boat)\} \end{aligned}$$

and adding the proper actions:

$$\begin{aligned} & can(pickup(boat, oar), s) \rightarrow location(oar, s) = location(boat, s) \\ & can(pickup(boat, oar), s) \rightarrow oar \in oars\_set \\ & can(pickup(boat, oar), s) \\ & \quad \rightarrow location(oar, result(pickup(boat, oar), s)) = in(boat) \\ & can(putdown(boat, oar), s) \rightarrow location(oar, s) = in(boat) \\ & can(putdown(boat, oar), s) \rightarrow oar \in oars\_set \\ & can(putdown(boat, oar), s) \\ & \quad \rightarrow location(oar, result(putdown(boat, oar), s)) = location(boat, s) \end{aligned}$$

3. *There is a bridge.* Using some brute force (I call it brute force because I do not write the entire theory of bridges, but just mention a specific bridge), enables us to write

$$\begin{aligned} & available(Bridge, group, s) \\ & water\text{-}crosser(Bridge) \\ & capacity(Bridge) \geq 1 \end{aligned}$$

Problem: The definition of *location* of the water-crosser. Solution:

$$\begin{aligned} & location(Bridge, s) = between(lb, rb, s) \\ & b1, b2 \in banks \rightarrow accessible(between(b1, b2, s), b1) \end{aligned}$$

Now, *accessible* give much trouble, since the river is accessible as well, but the boat being in the river, is not always accessible to either banks. We can, however, shut the problem with saying instead,

$$accessible(Bridge, b1) \wedge accessible(Bridge, b2)$$

with no say on the location of the Bridge, which makes it undefined.

4. *The boat is defective and must be repaired.* This is simple, since we have $ab2$ as a precondition for availability.

$$\begin{aligned}&defective(Boat, s)\\ &defective(boat, s) \rightarrow ab2(boat, group, s)\end{aligned}$$

We do, however, have to give our god the chance to repair the boat.

$$\begin{aligned}&can(repair(boat), s) \rightarrow defective(boat, s)\\ &can(repair(boat), s) \rightarrow \neg defective(boat, result(repair(boat), s))\end{aligned}$$

5. *The missionaries can't row.* We can make it in the easier way (brute force, i.e., say that groups of missionaries alone can not operate the boat (the boat is not available to these)), or in the more decent way of explaining what rowing is, and then saying that missionaries can not row. We take the latter. First we explain the meaning of rowing

$$\begin{aligned}&can(using(boat, go(group, bank)), s)\\ &\quad \rightarrow (\exists p \in group)(can(row(p, boat), s))\end{aligned} \tag{23}$$

Note: We can say the same thing using occurs, which gives somewhat different meaning.

$$\begin{aligned}&occurs(using(boat, go(group, bank)), s)\\ &\quad \rightarrow (\exists p \in group)(occurs(row(p, boat), s))\end{aligned}$$

For restricting the rowing only to the cannibals we then write

$$m \in missionaries \rightarrow \neg row(m, boat)$$

Note: We should be carefull and define what kind of nonmonotonic inference do we intend to use, since this will influence the kind of completion we will get for the above formula.

6. *Only one missionary and one cannibal can row.* Using formula 23 above, we add

$$(\exists! m \in m0\ \exists! c \in c0)(row(m, boat) \wedge row(c, boat))$$

7. *If the biggest cannibal is isolated with the smallest missionary, the latter will be eaten.* We add the following general properties

$$\begin{aligned}&(\exists! m \in m0) smallest(m)\\ &(\exists! c \in c0) biggest(c)\end{aligned} \tag{24}$$

and the current restriction

$$\begin{aligned}&ok(s) \rightarrow smallest(m) \wedge biggest(c)\\ &\quad \rightarrow (\forall bank)(inhabitants(bank, s) \neq \{m, c\})\end{aligned}$$

8. *The biggest cannibal cannot fit in the boat with another person.* We use the above formula 24 to formalize

$$available(boat, group, s) \wedge card(group) \geq 2$$
$$\rightarrow p \in group \cap cannibals \rightarrow \neg biggest(p)$$

9. *There is an island.* We should check everything is intact after we add the new bank, but, if everything else is as it should be, we just add the new bank

$$newbank \in banks$$

Note: We amended the definition of banks to be nonmonotonic (instead of declaring what the set 'banks' is equal to.

10. *The boat is a motor boat but cannot carry two across and back unless fuel has been stored on the far side of the river. It takes a back and forth trip with one person in the boat to take enough fuel for a return with two people.* First, let us describe the fuel tank of the boat, the loss of fuel in the action of using the boat for crossing the river, and the refueling action:

$$available(boat, group, s) \rightarrow fuel(boat, s) >= card(group))$$
$$fuel(Boat, s0) = 2$$
$$isboat(boat) \rightarrow fuel\_capacity(boat) = 2 \wedge fuel\_tank(boat)$$

$$can(using(boat, go(group, bank)), s)$$
$$\rightarrow fuel(boat, result(using(boat, go(group, bank)), s)) +$$
$$+card(group) = fuel(boat, s)$$

$$can(refuel(boat, tank), s)$$
$$\rightarrow final = fuel(boat, result(refuel(boat, tank), s)) =$$
$$= min(fuel\_capacity(boat), fuel(boat, s) + fuel(tank, s)) \wedge$$
$$fuel(tank, result(refuel(boat, tank), s)) =$$
$$= fuel(tank, s) - (final - fuel(boat, s)) \wedge$$
$$fuel\_tank(tank) \wedge fuel\_tank(boat)$$

The fuel tank that is carried from one bank to the other requires some sophistication, and the solution to the problem is highly dependent on the realism required. Before formalizing it, note that nowhere in the previous formalization did we require the members of the group to be missionaries or cannibals. It followed only from the CWA. In order to add fuel tanks to groups, we need to make sure that there is someone that can row the boat, and we do that by using formula (23)

$$fuel\_tank(tank) \rightarrow \neg can(row(tank, boat), s)$$
$$fuel\_tank(Tank1) \wedge fuel\_tank(Tank2)$$
$$location(Tank1) = location(Tank2) = lb$$
$$fuel(Tank1, S0) = 1000 \wedge fuel(Tank2, S0) = 0$$
$$fuel\_capacity(Tank2) = 8$$
$$fuel(tank, s) \leq fuel\_capacity(tank)$$

Notice that fuel transfer from one tank to another is being facilitated by the previous paragraph, using *boat* for the consumer, and *tank* for the supplier.

Note: The use of rowing in the last paragraph emphasized a previously unnoticed problem. The problem is that objects other than cannibals and missionaries (e.g., the boat) can actually 'transfer' the boat from one bank to another.

11. *The destination is not directly across from the starting point, and the motor boat cannot carry enough fuel to make it in one trip. Therefore, fuel caches must be used. If they are far enough apart, the cannibals cannot run fast enough to gang up on the missionaries left at a different cache before the missionaries are reinforced or removed.* This problem is ill defined in my opinion, and should have explained the following:

    (a) Where is the cache?

    (b) Why should people not be left at the same place as the cache?

12. *One of the missionaries is Jesus Christ who can walk on water and is not incapacitated by being eaten. We include this fantastic elaboration, because for humans, the problem remains definite, even though no-one knows background details about how Jesus Christ is affected by being eaten by a cannibal. Note that a further elaboration might require such information in a way that would leave a vague problem.* Two parts to this elaboration. The first:

$$
\begin{array}{l}
(\exists p \in m0)(jesus(p)) \\
jesus(p) \rightarrow can(go(\{p\}, bank), s)
\end{array}
$$

    The second is slightly more complicated, since we did not ever mention the possibility of eating, but just said that it is wrong to have a possibility that is translated in our concepts as "eating".[2]

13. *Some event can occur in the boat....* Let us take two exmples: *The first, where the two occupants of the boat should exchange positions when the rowing one becomes tired of rowing.* We should split the cross of the river into a couple of sections, where after each section we may decide to change roles.

    Q: How do you take actions apart?
    Suggested Answer: Use the *Occurs* fluent. This is (seemingly) the only fluent that is concerned with concurrent events, and this is exactly what we need when we try to elaborate actions.

    EmphThe cannibals are all epileptic, and might require treatment, so the missionaries should keep rehearsing their medical knowledge. This seems very close to concurrent events.

[2] We should take a look at the "Disaster" article, and alter the definition of *ok* accordingly

Q: How do we treat that?
Suggested Answer: Use of the *Occurs* fluent again, is eminent. But, here we have something that requires more strength than that. *Sometime* during an action, something will occur. What tools do we have for that?

14. Consider an elaboration that requires considering concurrent events.

15. Consider an elaboration that requires considering continuous actions and other events. There are four cannibals and four missionaries, but if the strongest of the missionaries rows fast enough, the cannibals won't have gotten so hungry that they will eat the missionaries.

16. A missionary can deposit a cannibal at an arbitrary distance up the river except that fuel for the boat is limited. The missionaries then outnumber the cannibals until the marooned cannibal can walk back. Depending on the fuel available, this can make the problem solvable with more missionaries and cannibals.

17. Some may prefer a version in which what is to be avoided is the missionaries converting the cannibals to their religion, which will happen if missionaries outnumber cannibals on either bank. Adding this restriction without taking back the previous one makes the problem unsolvable.

18. Another direction of elaboration is to give more detail. Crossing the river involves some missionaries and cannibals getting into the boat, propelling the boat across the river, and some missionaries and cannibals getting out. Crossing the river can be further elaborated to take into account the order of entering and leaving the boat and auxiliary actions like tying and untying the boat to the bank or dock. When a formalism that elaborates unnecessarily is used, the formalization should admit collapsing sequences of actions to single actions so as to restore simplicity.

19. The obedience of a cannibal will last only for the time of two back-and-forth trips in the absence of a missionary.

20. Here are some more. Which missionaries actually rowed? When a missionary and a cannibal row together, which gets into the boat first and which gets out first? Perhaps it is safer if the cannibal gets into the boat first and the missionary gets out first.

21. When two row, does one do all the rowing or do they take turns?

22. Cannibals are initially not hungry, but rowing makes them hungry.

23. There is more than one boat.

5 Remarks and Problems

1. It is obvious to a human that in MC1 (the original problem) only the numbers matter. It should be obvious to the program as well.

2. In the present formulation of MCP $ok(s)$ is given as a precondition for the actions. It would be better if the possibility of missionaries being eaten were explicitly stated. This reduces the number of missionaries on the bank where this happens to 0, we provide no way for new missionaries to appear, so an induction tells us that a state in which there are three missionaries and three cannibal on the right bank can no longer be achieved.

   We aren't up to that yet, but here's a start.

$$0 < m(s, bank) < m(c, bank) \rightarrow occurs(eat(\{m|location(m, s) = bank\}), s)$$

$$\begin{aligned} &card(\{m|location(m, \\ &result(eat(\{m|location(m, s) = bank\}), s)) = bank\}) = 0 \end{aligned} \tag{25}$$

3. It is easy to see what happens to the people when they go to the opposite bank. How should the fact that the boat goes too be stated? How about

$$\begin{aligned} &occurs(using(boat, go(g1, bank)), s) \rightarrow occurs(go(g1, bank), s) \\ &occurs(using(boat, go(g1, bank)), s) \rightarrow occurs(go(boat, bank), s) \end{aligned} \tag{26}$$

4. One of the corollaries of the Elaboration analysis, is noticing that anything we try to "cut" in our formalism at the first stage, retaliates when we try to elaborate on something close (e.g., the "eating" principle versus the "ok" definition).

5. Another corollary is that definitions should be made in a nonmonotonic fasion as possible. The "direction" of this nonmonotonic definition should be determined, and should probably be a major issue in constructing the system, because ... it gives us only one direction of freedom.

6 Restricting the Elaborated

6.1 3-tuples

In this section we try to show how we can use the elaboration tolerant formalism we mentioned previously, to form the 3-tuples States mentioned in section 3.1 of the main article. For now, we will call that formulation, *the restricted formulation* or the *3-tuples*.

We will start constructing the sentences of 3-tuples one by one, pointing out the problems for each one of the sentences, and possible solutions, if such are visible.

1. $States = Z_4 \times Z_4 \times Z_2$.

   In our formalism we used situations, and not states. To restrict the situations to be these states, we need something like

   $$ok(s) \rightarrow f(s) \in Z_4 \times Z_4 \times Z_2 \tag{27}$$

   where f is a (partial) function mapping situations to states. Lemma (6.1) at the end of this section, guarantees the soundness of the use of such a function.

2. $ok(state) \equiv ok1(p1(state), p2(state)) \wedge ok1(3 - p1(state), 3 - p2(state))$
   $ok1(m, c) \equiv m \in Z_4 \wedge c \in Z_4 \wedge (m = 0 \vee m \geq c)$.

   Recall the definition of m and c, the cardinality of missionaries and cannibals respectively, in a given situation, on a given bank, defined in (15). We can formulate the following:

   $$p1 = m\restriction_{lb}$$
   $$p2 = c\restriction_{lb}$$

   Now the two upper sentences, from the original 3-tuple formalization, is equivalent to (16) and (27):

   $$ok(s) \rightarrow \neg(0 < m(s, lb) < c(s, lb)) \wedge \neg(0 < m(s, rb) < c(s, rb))$$

   To achieve a better result we need the following two observations:

   (a) The *Closed World Assumption* will help us infer that we have exactly 3 missionaries and 3 cannibals.

   (b) The *Inertia* axiom schema will help us infer that the total number of missionaries and the total number of cannibals will not change unless something will cause it to change.

   Now we can infer the following formalization:

   $$ok(s) \rightarrow m(s, lb) = 3 - m(s, rb) \wedge c(s, lb) = 3 - c(s, rb)$$

   which will enable us to infer

   $$ok(s) \rightarrow \neg(0 < m(s, lb) < c(s, lb)) \wedge \neg(0 < 3 - m(s, lb) < 3 - c(s, lb))$$

3. $moves = \{(1,0),(2,0),(0,1),(0,2),(1,1)\}$

   We formalize the available moves by forming a set of possible moves (we should probably move that formalization to section 2.2).

$$moves = \left\{ action \;\middle|\; \begin{array}{c} action = using(boat, go(Group_a, bank)) \\ (\exists s, bank, boat)(can(action, s)) \end{array} \wedge \right\}$$

   The equivalence is implied by (8) and (7).

4. $result(move, state) = mkstate(p1(state) - (2p3(state)-1) * p1(move),$
   $p2(state) - (2p3(state)-1) * p2(move),$
   $1 - p3(state))$

   We should make some observations before we tackle this one. First - the operators p1,p2 are overloaded. Thus when formalizing our equivalent definitions, we will use our own similar operators. We use $p1, p2$ as defined above, as operators on situations. We want something like

$$\begin{array}{l} location(boat, state) = opp(destination(move)) = lb \wedge \\ move \in moves \rightarrow \end{array}$$
$$\begin{aligned} result(move, state) \quad = \quad & (p1(state) - m\text{-}in\text{-}boat(move), \\ & p2(state) - c\text{-}in\text{-}boat(move), \\ & opp(location(boat, state))) \end{aligned} \tag{28}$$

$$\begin{array}{l} location(boat, state) = opp(destination(move)) = rb \wedge \\ move \in moves \rightarrow \end{array}$$
$$\begin{aligned} result(move, state) \quad = \quad & (p1(state) + m\text{-}in\text{-}boat(move), \\ & p2(state) + c\text{-}in\text{-}boat(move), \\ & opp(location(boat, state))) \end{aligned} \tag{29}$$

$$\begin{array}{l} location(boat, state) = location(group(move), state) = \\ = destination(move)) \wedge move \in moves \rightarrow \end{array}$$
$$result(move, state) = state \tag{30}$$

   m-in-boat and *c-in-boat* should be defined as:

$$\begin{array}{l} destination(move) = \text{the target bank} \\ group(move) = \text{the moving group} \\ m\text{-}in\text{-}boat(move) = card(group(move) \cap missionaries) \\ c\text{-}in\text{-}boat(move) = card(group(move) \cap cannibals) \end{array} \tag{31}$$

   and $location(boat, state)$ should be defines as

$$location(boat, state) = \left\{ \begin{array}{ll} lb & p3(state) = 1 \\ rb & p3(state) = 0 \end{array} \right\}$$

   We should note a couple of things about this formalism:

- In our definition of result, we use only states, and no situations.
- The construction here is somewhat more elaborated than I originally intended to. It happens that we have problems eliminating the case where the *move* does not move anything. This is something that is not defined anywhere in the $3-tuples$ formalism, but is somewhat hard not to consider here. The problem is mostly caused by the Homomorphism lemma that follows.
- There are some more further remarks on the paper I used in the group meeting.

5. $attainable((3,3,1))$
$attainable(state) \wedge ok(result(move, state)) \rightarrow attainable(result(move, state))$

   In this case, we can use the formalization formed earlier, to *define* the *attainable* relation in our formalism. Notice the "define" emphasis.

$$\begin{aligned} & attainable(s0) \\ & attainable(s) \rightarrow attainable(s') \wedge ok(s') \wedge \\ & (\exists move \in moves)(s = result(move, s)) \end{aligned}$$

   We will also need the corresponding 'trace' of our elaborated formalism? We should add that as well.

As we argued above, we use a mapping from a subset of the possible Situations Set, to the set of possible states, in $Z_4 \times Z_4 \times Z_2$. The following lemma shows that solving the *3-tuples* problem, will give us a solution for the Situation Calculus formalism.

Lemma 6.1 (Homomorphism lemma) *Let f : Situations $\rightarrow$ States, be a partial function defined as follows:*

$$f(s) = (m(s, lb), c(s, lb), \left\{ \begin{array}{ll} 1 & location(Boat, s) = lb \\ 0 & location(Boat, s) = rb \end{array} \right\})^3$$

Let g : Actions $\rightarrow$ Moves, be a partial function defined as follows:

$$g = \left\{ (a, a) \;\middle|\; \begin{array}{l} a = using(Boat_a, go(Group_a, Bank_a)) \\ for\ some\ objects\ Boat_a, Group_a, Bank_a \end{array} \right\}$$

Then,

$$\begin{aligned} & result(\ g(using(boat, go(group, bank))),\ f(s)) = \\ & \qquad f(\ result(using(boat, go(group, bank)), s)\ ) \end{aligned} \tag{32}$$

[3] notice that f is undefined when the location of the boat, or $m(s, lb)$ or $c(s, lb)$ are undefined

Proof The lemma follows directly from the way we defined the *result* operator for the state-calculus formalism.

Fix a situation s. Let $move = using(boat, go(group, bank))$. If $move \notin moves$, then both sides of the equation are not defined, and thus can be considered equal. Therefor, let us assume $move \in moves$.

First case: Assume $location(boat, s) = opp(destination(move) = lb$. Then, $location(boat, f(s)) = lb$. Thus, by (28), the left side of equality (32) is

$$(p1(state) - m\text{-}in\text{-}boat(move), p2(state) - c\text{-}in\text{-}boat(move), 0)$$

For the right side of equality (32), let us examine the outcome:

$$result(using(boat, go(group, bank)), s).$$

Since the move is feasible, (12) and (9) implies

$$location(boat, result(using(boat, go(group, bank)), s)) = bank$$

We can apply (13) and (9) again, to get

$$location(group, result(using(boat, go(group, bank)), s)) = bank$$

Since $location(group, s)$ is defined,

$$\begin{aligned}&location(missionaries(group), s') = \\ &\quad location(cannibals(group), s') = location(group, s'),\end{aligned}$$

where $s' = result(using(boat, go(group, bank)), s)$. Recall that in the current case, $bank = opp(location(group, s))$. Now all that remains is counting.

$$\begin{aligned}&m(s', lb) = \\ &\quad m(s, lb) \\ &\quad -card((inhabitants(lb, s) \setminus inhabitants(lb, s')) \cap missionaries) \\ &\quad +card((inhabitants(lb, s') \setminus inhabitants(lb, s)) \cap missionaries)\end{aligned} \tag{33}$$

Inertia axiom schema (from the above), plus nonmonotonic reasoning (from down below) ensures us that the boat is the only mean of adding and removing items from each of the banks. Thus using (9), we have in this case

$$group(move) = (inhabitants(lb, s) \setminus inhabitants(lb, s'))$$

and by (31)

$$m\text{-}in\text{-}boat(move) = card(group(move) \cap missionaries)$$

Combining these two, with the above remark, we get

$$m(s', lb) = m(s, lb) - m\text{-}in\text{-}boat(move) \tag{34}$$

The same process will lead us to

$$c(s', lb) = c(s, lb) - c\text{-}in\text{-}boat(move) \tag{35}$$

Eventually applying f, we get to the above conclusion, i.e.,

$$(p1(state) - m\text{-}in\text{-}boat(move), p2(state) - c\text{-}in\text{-}boat(move), 0)$$

Second case: $location(boat, s) = opp(destination(move) = rb$. The treatment is similar.

Third case: $destination(move) = location(boat, s) = location(group, s)$. Here we apply the third part of the definition of *result* for states, and get that the left hand side of equation (32) is $f(s)$.

On the other hand, considering the right hand side of (32), observe that by (9) the *group* gets to keeps its location, when we apply the *move* to the situation s. Thus, as we noticed earlier

$$\begin{aligned} &m(s', lb) = \\ &\quad m(s, lb) \\ &\quad -card((inhabitants(lb, s) \setminus inhabitants(lb, s')) \cap missionaries) \\ &\quad +card((inhabitants(lb, s') \setminus inhabitants(lb, s)) \cap missionaries) \end{aligned} \tag{36}$$

and using the Inertia schema, we get that

$$m(s, lb) = m(s', lb)$$

(recall that $s' = result(move, s)$). Since the boat keeps its place as well, we get that

$$f(s) = f(s')$$

And thus our proof is done. ■

6.2 Infering the restriction

Let us go back to the formalism mentioned in the previous sub-section. Observe (2). Here we encounter the first serious problem. When we defined $ok(s)$ in (16), we defined it by means of the objects lb (left bank) and rb (right bank). There is no obvious relation between the two. Only a mathematical analysis of the actions in section 2.4 and (15) might yield the above formulas.

One 'magical' solution to the problem, is just avoid noticing the right (destination) bank, and calculating everything through the view of the left (source) bank. It will work of course, but will require the 'god' playing the game, to notice the redundancy of the right bank, in the posed problem. Since we did not supply this god the necessary formalism for that, it is unlikely that it will infer such a thing so quickly.

We try to solve this problem somehow in the next sub-section.

6.3 6-tuples

Some of the problems met in the 3-tuples structure can be resolved by a 6-tuples structure:

$$Z_4 \times Z_4 \times Z_2 \times Z_4 \times Z_4 \times Z_2$$

which represents the states of both banks. This structure can eliminate some of the problems posed by the algebra related with the 3-tuples.

I will discuss only the majorly changed formulas and principles. They are referenced by their ordinal number in the 3-tuples section.

2. We can now formalize the 3-tuples idea as follows:

$$\begin{aligned}
&ok(state) \equiv ok1(p1(state), p2(state)) \wedge ok1(p4(state), p5(state)) \\
&ok1(m, c) \equiv m \in Z_4 \wedge c \in Z_4 \wedge (m = 0 \vee m \geq c)
\end{aligned}$$

This representation is much easier for our god to understand, and thus it might come with the following sub-formalization of the elaborated one:

$$\begin{aligned}
&ok(s) \rightarrow s \in Z_4 \times Z_4 \times Z_2 \times Z_4 \times Z_4 \times Z_2 \\
&ok(s) \rightarrow \neg(0 < p1(s) < p2(s)) \wedge \neg(0 < p4(s) < p5(s))
\end{aligned}$$

Which is obviously equivalent to the above one.

4. The result of moving, changes. We now have to apply the move to both banks (which is somewhat what happens in reality).

$$\begin{aligned}
&result(movable\text{-}group, state) = s \wedge \\
&\qquad location(boat, state) = lb \wedge \\
&\qquad movable\text{-}group \in moves \rightarrow \\
&\qquad\qquad s = (p1(state) - m\text{-}in\text{-}boat(movable\text{-}group), \\
&\qquad\qquad\quad p2(state) - c\text{-}in\text{-}boat(movable\text{-}group), 0, \\
&\qquad\qquad\quad p4(state) + m\text{-}in\text{-}boat(movable\text{-}group), \\
&\qquad\qquad\quad p5(state) + c\text{-}in\text{-}boat(movable\text{-}group), 1)
\end{aligned} \quad (37)$$

$$\begin{aligned}
&result(movable\text{-}group, state) = s \wedge \\
&\qquad location(boat, state) = rb \wedge \\
&\qquad movable\text{-}group \in moves \rightarrow \\
&\qquad\qquad s = (p1(state) + m\text{-}in\text{-}boat(movable\text{-}group), \\
&\qquad\qquad\quad p2(state) + c\text{-}in\text{-}boat(movable\text{-}group), 1, \\
&\qquad\qquad\quad p4(state) - m\text{-}in\text{-}boat(movable\text{-}group), \\
&\qquad\qquad\quad p5(state) - c\text{-}in\text{-}boat(movable\text{-}group), 0)
\end{aligned} \quad (38)$$

6.4 Restrictions: Conclusions

The following are conclusions from the restriction experiments, that should be applied to the elaborated formalism.

1. We should add some mathematical formulation, so that the effects of 'go' could be translated into mathematical terms.
2. The relations Situation vs State should be studied.
3. Actions should be generalized.
4. We should add the 'attainable' predicate to our elaborated formalism.
5. Many senctences in the 'elaborated' formalism, should be represented as Nonmonotonic Reasoning ones (e.g., *ok*).

This is /u/eyal/ongo/mcp/jmc/m4.tex.