Belief revision in the Situation Calculus without plausibility levels

Robert Demolombe[1] and Pilar Pozos Parra[2]

[1] ONERA Toulouse
France
Robert.Demolombe@cert.fr
[2] Universidad Tecnológica de la Mixteca, Mexico
pilar@mixteco.utm.mx

Abstract. The Situation Calculus has been used by Scherl and Levesque to represent beliefs and belief change without modal operators thanks to a predicate plays the role of an accessibility relation. Their approach has been extended by Shapiro et al. to support belief revision. In this extension plausibility levels are assigned to each situation, and the believed propositions are the propositions that are true in all the most plausible accessible situations.
Their solution is quite elegant from a theoretical point of view but the definition of the plausibility assignment, for a given application domain, raises practical problems.
This paper presents a new proposal that does not make use of plausibilities. The idea is to include the knowledge producing actions into the successor state axioms. In this framework each agent may have a different successor state axiom for a given fluent. Then, each agent may have his subjective view of the evolution of the world. Also, agents may know or may not know that a given action has been performed. That is, the actions are not necessarily public.

1 A brief introduction to the Situation Calculus

In the very beginning the Situation Calculus was introduced by McCarthy [4]. It has been used by several authors to propose solutions to the frame problem and a simple solution has been proposed by Reiter in [6]. Since several variants of the Situation Calculus have been presented in the literature, we refer to the Situation Calculus as it is defined in Reiter's book [7].

The Situation Calculus is a many sorted classical logic (most of it uses first order logic). There are two kinds of predicates. Those whose truth values may change when actions are performed, which are called "fluents", and those whose truth values do not change. The fluents have exactly one argument of the type situation which is the last argument. A similar deal is made for function symbols.

For instance, if we consider a fragile toy, the fluent $broken(s)$ can be used to represent the fact that in the situation s the toy is broken. The fluent $position(x, s)$ can represent the fact that in the situation s the toy is in the position x. If the color of the toy never changes, its color is not represented by a fluent. It is represented by a predicate. For instance, $color(x)$ which has no argument of the type situation, and whose meaning is that the toy color is x.

The situations are represented by terms that may be constants of the type situation, or complex terms formed with the designated binary function symbol do. If a is a term of the type action and s is of the type situation, then $do(a, s)$ is of the type situation. From an intuitive point of view, $do(a, s)$ represents the situation resulting from the performance of a in the situation s.

For example, the constant s_0 can be used to represent the initial situation, and, if $drop$ and $repair$ are of the type action, the terms: $do(drop, s_0)$, $do(repair, s_0)$ and $do(repair, do(drop, s_0))$, are of the type situation.

The solution to the frame problem is based on the assumption that we (the people who are modeling an application domain) know **all** the actions, and the contexts, that can change the truth value of each fluent.

The fact that the actions $drop$ and $repair$ cause respectively the fluent $broken(s)$ to be true and false is represented by the following axioms:

$\forall s \forall a(a = drop \rightarrow broken(do(a, s)))$

$\forall s \forall a(a = repair \rightarrow \neg broken(do(a, s)))$

The fact that there is no other action that can change the truth value of $broken(s)$ is represented by the "Successor State Axiom (SSA)":

$\forall s \forall a(broken(do(a, s)) \leftrightarrow a = drop \vee broken(s) \wedge \neg(a = repair))$

The general form of a SSA for a fluent p is[3]:

$(S_p)\ \ \forall s \forall a \forall \boldsymbol{x}(p(\boldsymbol{x}, do(a, s)) \leftrightarrow \Gamma_p^+(\boldsymbol{x}, a, s) \vee p(\boldsymbol{x}, s) \wedge \neg \Gamma_p^-(\boldsymbol{x}, a, s))$

where $\Gamma_p^+(\boldsymbol{x}, a, s)$ and $\Gamma_p^-(\boldsymbol{x}, a, s)$ are first order formulas whose free variables are: $\boldsymbol{x}$, a and s.

For instance, for the fluent $position(x, s)$ we have:

$\Gamma_{position}^+(x, a, s) = \exists y(a = move(y, x) \wedge position(y, s))$

In [8] Scherl and Levesque have extended the Situation Calculus to represent beliefs and belief changes. One of the basic ideas is to see situations like possible worlds in Kripke models[4]. The second idea is to introduce a fluent $K(s', s)$ which has the same intuitive meaning as an accessibility relation in doxastic or epistemic logics.

To represent what an agent believes in a given situation, say s_0, Reiter says in [7]: "we can picture this in terms of an agent inhabiting s_0, and imagining all possible alternative situations to the one he is in". Those imaginary situations are related with the real one through the fluent K, that is, they are the situations s' such that: $K(s', s_0)$.

In general, $Knows(\phi, s)$[5] is used as a notation to represent the fact in the situation s an agent believes ϕ. We have:

$Knows(\phi, s) \stackrel{\text{def}}{=} \forall s'(K(s', s) \rightarrow \phi[s'])$

---

[3] $\boldsymbol{x}$ is used as an abbreviation for the tuple of variables $x_1, \ldots, x_n$, and $\forall \boldsymbol{x}$ is an abbreviation for $\forall x_1 \ldots \forall x_n$.

[4] This is an intuitive analogy. In fact there are significant differences between possible worlds and situations. One of them is that situations represent histories.

[5] The notation $Knows(\phi, s)$ is a bit misleading because ϕ is not necessarily true in s.

where $\phi[s']$ is the formula obtained by replacing in each fluent the argument of type situation by s'.

To define belief changes **two questions** have to be answered:

(1) what are the new accessible situations after performance of an action a?

(2) what are the truth values of the fluents in the new accessible situations?

The answer to the first question depends on the type of action a. Two different types of actions are considered: those that are "knowledge producing actions" and those that are not. Each knowledge producing action informs the agent about the truth of a given proposition in the situation where the agent is.

For instance, we can have the action *look* that informs the agent about the truth of $broken(s)$, and another knowledge producing action to inform him about the toy position.

Given that each knowledge producing action informs about the truth value of a given proposition, after performance of a knowledge producing action, the new accessible situations s'' are the successor of the situations where this proposition has the same truth value as in real situation s. In other words, the imaginary situations that are **inconsistent** with what has been observed are **removed**.

In the case of a non knowledge producing action, the new accessible situations are the successors of the situations that were accessible before to perform a. That is, if s' was accessible from s, after performance of a the corresponding new accessible situation from $do(a, s)$ is $do(a, s')$.

In general, the new accessible situations are defined by the following axiom:

$$
\begin{aligned}
(S_K)\quad & \forall s \forall s'' \forall a (K(s'', do(a, s)) \leftrightarrow \exists s'(K(s', s) \land s'' = do(a, s') \land (\\
& (\neg(a = \alpha_1) \land \ldots \land \neg(a = \alpha_n)) \\
& \lor\ a = \alpha_1 \land (\phi_1(s) \leftrightarrow \phi_1(s')) \\
& \ldots \\
& \lor\ a = \alpha_n \land (\phi_n(s) \leftrightarrow \phi_n(s'))))
\end{aligned}
$$

where $\alpha_1, \ldots, \alpha_n$ is the set of all the knowledge producing actions, and $\phi_1, \ldots, \phi_n$ are their corresponding propositions.

The answer to the second question is that the truth values of the fluents in the new imaginary accessible situations are determined from the previous ones by the SSAs. For example, if we have $K(do(a, s'), do(a, s))$, the truth value of $broken(do(a, s'))$ is determined by the SSA of the fluent *broken*, and by the truth value of $broken(s')$. Then, for non knowledge producing actions we can say that, in some sense, the beliefs change in the same way the world does.

However, this formalization of belief change raises a big problem in the case of revision because if every imaginary situation is inconsistent with the situation s, they are all removed. The result is that in $do(a, s)$ every proposition is believed, and we have inconsistent beliefs.

To remedy this problem Shapiro et al. have proposed in [9] to assign plausibility levels to all the situations, and have defined the believed propositions as the propositions which are true in all the most plausible accessible situations.

However, this new belief definition raises practical problems when we have to define the plausibility function *pl* for a given application domain. For example, it may be impossible to define the *pl* function by extension because the number of initial accessible situations may be infinite.

That is the case, for example, when the agent believes in the initial situation that the toy is not broken, and he ignores his position, and the number of possible position is infinite.

Fortunately, we can avoid having to give an extensional definition of *pl* by using the assumption: $Knows(\neg broken, s_0)$. Nevertheless, if we want to guarantee consistent beliefs after revision, we need to add the assumption: $broken(s_0) \rightarrow \exists s''(K(s'', s_0) \wedge \neg K_{max}(s'', s_0) \wedge broken(s''))$, where $K_{max}(s'', s_0)$ means that in the situation s_0, the situation s'' is one of the most plausible.

If the toy is broken, is this assumption sufficient to prove that in $do(look, s_0)$ the agent believes that the toy is broken? The answer is not obvious.

This simple example shows that it is definitely not easy to guarantee that no assumption is missing in the description of the initial situation. If in the application domain there are five, ten, or even more fluents, modeling the initial situation may be quite problematic.

To remove these difficulties, in [2] Demolombe and Pozos Parra have proposed a simple but restrictive solution to belief revision. The idea is to generalize the successor state axioms from literals to modal literals. Although in this solution the scope of the beliefs is limited to literals, Petrick and Levesque in [5] have analyzed its expressive power, and they have shown that its limitations are not so strong. In [1] Demolombe and Herzig have also shown that a very similar idea has been formalized in the Dependence Logic.

2 Belief revision without plausibility levels

Here a new solution is proposed[6] to belief revision. The most important feature is not to use plausibility levels and instead incorporate the sensing actions into the successor state axioms. Then the effect of a sensing[7] action is **not to remove** the situations that are inconsistent with the perceived information, as Scherl and Levesque do in [8], but to change the truth values of the fluents in the imaginary successor situations according to the perceived information, in a similar way as non knowledge producing actions do.

There are two other significant features in this new proposal. The first one is that we may have **distinct successor state axioms** for real situations and for agents' imaginary situations. The second one is that we have an explicit representation of the actions whose performance can be observed by an agent, and, if an agent has not observed the performance of an action, then this action does not change his beliefs.

Before to present a general framework we analyze a particular scenario and its formalization.

[6] A preliminary version of this work has been presented in [3].
[7] In the following "sensing action" will be used as a shorthand for "knowledge producing action".

Scenario.

In a room there is a young baby and his mother. It is assumed that the baby observes every action he performs iff he has his eyes open. The same applies for the mother.

In the situation s_0 (see Fig. 1)[8] the baby holds a fragile toy in his hands. The toy is not broken and both the mother and the baby believe that the toy is not broken. The baby and the mother have their eyes open.

In the situation s_1 the baby has dropped the toy, i.e. $s_1 = do(drop(b), s_0)$[9]. The mother believes that a fragile toy will break after it is dropped. It follows that she believes that the toy is broken. She does not need to look at the toy to hold this belief. However, the baby does not believe that dropping a fragile toy will cause it to break. So the baby in s_1 believes that the toy is not broken.

In the situation s_2 the baby has looked at the toy, i.e. $s_2 = do(look(b), s_1)$, and he observed that it is broken. Then, in s_2 he believes that the toy is broken. In s_2 the mother still believes that the toy is broken.

In the situation s_3 the baby has closed his eyes, i.e. $s_3 = do(close(b), s_2)$. The baby's beliefs and the mother's beliefs **remain unchanged**, except about the fact that this action has been performed.

In the situation s_4 the mother has repaired the toy, i.e. $s_4 = do(repair(m), s_3)$. Since the baby has closed his eyes he ignores that this action has been performed. Then, in s_4 he still believes that the toy is broken.

The main features of this scenario are the following ones.

- In s_4 the baby ignores that the action $repair(m)$ has been performed. Then, in s_4 the baby holds the same beliefs than in s_3, i.e. the baby's imaginary situations, like s'_3, **remain unchanged** after the execution of the unobserved action.
- The mother and baby have different beliefs about the evolution of the fluent *broken*. That is why, even when both of them know initially the toy is not broken, we have $broken(s''_1)$ and $\neg broken(s'_1)$ (see Fig. 1).
- In s_2 the action $look(b)$ has informed the baby about the truth value of the fluent *broken* in s_1. Though imaginary situations like s'_1 represent a belief that is false in s_1, the situation s'_1 is not removed, and instead the baby changes in s'_2 the belief that the toy is not broken because we have $broken(s_1)$.

Formalization.

The following notations are adopted.

$open(i)$: the agent i opens his eyes.
$close(i)$: the agent i closes his eyes.
$drop(i)$: the agent i drops the toy.
$look(i)$: the agent i looks whether or not the toy is broken.

[8] To make the figure easier to read we have respectively used the notations K_m and K_b for $K(m, -, -)$ and $K(b, -, -)$.

[9] In this section, since we may have several agents, action function symbols have an argument to make explicit the agent who is doing the action.

$repair(i)$: the agent i repairs the toy.
$broken(s)$: in the situation s the toy is broken.
$real(s)$: s is a real situation.

The person who is modeling an application defines which situations are real situations.
$observe(i, a, s)$: in the situation s, if the action a is performed, then after performance of a the agent i will be informed that a has been performed.

Note that $observe(i, a, s)$ does not mean that i has observed performance of a, but that i has the ability to observe performance of a **if** a is performed.
$K(i, s', s)$: in the situation s the agent i believes that s' is a situation where he might be in. That is, s' is compatible with what i believes in s.
$B(i, \phi(s', s), s', s)$: $\phi(s', s)$ holds in every situation s' related by K with i and s. In other words $\phi(s', s)$ is compatible with what i believes in s. Note that the free variables s' and s of ϕ must be the third and the fourth argument of B respectively, since s may not be a free variable of ϕ. In $B(i, \phi(s', s), s', s)$ the situation s may be a real situation or an imaginary situation.

Notice that B is an abbreviation and not a fluent. We have:

$B(i, \phi(s', s), s', s) \stackrel{\text{def}}{=} \forall s'(K(i, s', s) \rightarrow \phi(s', s))$

The formula $\phi(s', s)$ may be any formula in a first order language with the only restriction being that the fluent K only appears in formulas that represent beliefs.

Evolution of imaginary situations.

If the action a is performed in the situation s, the situations that are accessible from $do(a, s)$ are the successors of the situations accessible from s if the action a has been observed by i, and they are the same situations as the situations accessible from s if the action a has not been observed by i. Then, we have:

(EK) $\forall s \forall s'' \forall a \forall i (K(i, s'', do(a, s)) \leftrightarrow \exists s'(K(i, s', s) \wedge$
$((observe(i, a, s) \wedge s'' = do(a, s')) \vee (\neg observe(i, a, s) \wedge s'' = s'))))$

We can infer from (EK) that after an action has been performed an agent will believe it has been performed if he observes it.

We have to define a SSA for the fluent $observe(i, a, s)$. This SSA depends on each application domain. For the scenario we have presented here we have adopted the following SSA:

(OBS) $\forall s \forall a \forall a' \forall i (observe(i, a, do(a', s)) \leftrightarrow a' = open(i) \vee observe(i, a, s) \wedge$
$\neg(a' = close(i)))$

The intuitive meaning of (OBS) is that an agent has the capacity to observe any action provided he has opened his eyes and not subsequently closed them. Since s is not restricted to real situations, the SSA (OBS) is believed by every agent[10].

The SSAs for the fluents depend on the context where we are. We may have different SSAs for the same fluent depending on the fact that we are in a real situation or in some agent's imaginary situation.

[10] As a matter of simplification we have assumed that an agent performs no action, except open eyes if he has closed his eyes.

The SSA for the real situations is:

(SSA_r) $\forall s \forall a(real(s) \rightarrow$
$(broken(do(a,s)) \leftrightarrow \exists i(a = drop(i)) \vee broken(s) \wedge \neg(\exists i(a = repair(i)))))$

The SSA for the mother's imaginary situations is:

(SSA_m) $\forall s \forall s' \forall a(real(s) \rightarrow (K(m,s',s) \rightarrow$
$broken(do(a,s')) \leftrightarrow \exists i(a = drop(i)) \vee (a = look(m) \wedge broken(s)) \vee broken(s') \wedge$
$\neg(\exists i(a = repair(i)) \vee (a = look(m) \wedge \neg broken(s)))))$

Notice that the fluent broken comes to be true (resp. false) in a mother's imaginary situation $do(a,s')$ if the mother has looked at the toy (action $look(m)$) and the toy is broken (resp. not broken) in the real situation s.

The SSA for the baby's imaginary situations is:

(SSA_b) $\forall s \forall s' \forall a(real(s) \rightarrow (K(b,s',s) \rightarrow$
$broken(do(a,s')) \leftrightarrow (a = look(b) \wedge broken(s)) \vee broken(s') \wedge \neg(\exists i(a = repair(i)) \vee$
$(a = look(b) \wedge \neg broken(s)))))$

If we adopt the notations:

$ssa(b,s',s) \stackrel{\text{def}}{=} broken(do(a,s')) \leftrightarrow (a = look(b) \wedge broken(s)) \vee broken(s') \wedge$
$\neg(\exists i(a = repair(i)) \vee (a = look(b) \wedge \neg broken(s)))$

$ssa(m,s',s) \stackrel{\text{def}}{=} broken(do(a,s')) \leftrightarrow \exists i(a = drop(i)) \vee (a = look(m) \wedge broken(s)) \vee$
$broken(s') \wedge \neg(\exists i(a = repair(i)) \vee (a = look(m) \wedge \neg broken(s)))$

the axioms SSA_b and SSA_m can be reformulated as:

(SSA_b) $\forall s \forall a(real(s) \rightarrow B(b, ssa(b,s',s), s', s))$

(SSA_m) $\forall s \forall a(real(s) \rightarrow B(m, ssa(m,s',s), s', s))$

It is worth noting that, depending on the application, it can be assumed that an agent is informed about the SSAs believed by the other agents.

For example, it may be assumed that the baby believes that his mother believes, like him, that the toy is not broken if it is dropped, and also that the mother believes that her baby believes that the toy is not broken if it is dropped, even if she does not hold that belief.

That can be formally represented by the following axioms.

(SSA_{bm}) $\forall s \forall a(real(s) \rightarrow B(b, B(m, ssa(b,s'',s'), s'', s'), s', s))$

(SSA_{mb}) $\forall s \forall a(real(s) \rightarrow B(m, B(b, ssa(b,s'',s'), s'', s'), s', s))$

Successor State Axioms in general.

In general, to define the "real" evolution of fluents, for each fluent p we have the SSA:

(SSA_p) $\forall s \forall a \forall \boldsymbol{x}(real(s) \rightarrow (p(\boldsymbol{x}, do(a,s)) \leftrightarrow \Gamma_p^+(\boldsymbol{x},a,s) \vee p(\boldsymbol{x},s) \wedge$
$\neg \Gamma_p^-(\boldsymbol{x},a,s)))$

To define the "subjective" evolution of fluents, for an agent i and a fluent p we have the SSA:

$(SSA_{i,p})$ $\forall s \forall s' \forall a \forall \boldsymbol{x}(real(s) \rightarrow (K(i,s',s) \rightarrow$
$(p(\boldsymbol{x}, do(a,s')) \leftrightarrow \Gamma_{i,p}^+(i,\boldsymbol{x},a,s') \vee (a = sense_p(i) \wedge p(\boldsymbol{x},s)) \vee p(\boldsymbol{x},s') \wedge$
$\neg(\Gamma_{i,p}^-(i,\boldsymbol{x},a,s') \vee (a = sense_p(i) \wedge \neg p(\boldsymbol{x},s))))))$

It is assumed that the successors of the real situations are real situations. Then, we have:

(SSA_R) $\forall s \forall a(real(do(a,s)) \leftrightarrow real(s))$

Progression and regression.

For reasoning about actions and beliefs we define a Basic Action and Belief Theory T which is composed of the following axioms.

1. Σ: the foundational axioms for situations (see [7]).
2. T_{SS}: a set of successor state axioms of the form[11] (SSA_p) and $(SSA_{i,p})$.
3. T_K: the successor state axiom (EK) for the fluent $K(i, s', s)$.
4. T_O: the successor state axiom for the fluent $observe(i, a, s)$ which depends on the application domain and has a form similar to (OBS).
5. T_R: the successor state axiom (SSA_R).
6. T_{AP}: a set of precondition axioms (see [7]).
7. T_{UNA}: a set of unique name axioms for actions (see [7]).
8. T_0: a set of first order sentences defining the initial situation (including initial beliefs).

Progression.

The progression problem is to infer from the initial situation s_0 consequences about properties in a further situation s_i. So, to solve the progression problem for $\phi(s_i)$, where s_i is a ground term, we have to prove that:

$\vdash T \rightarrow \phi(s_i)$

That can be proved with any standard automated theorem proving technique for classical first order logics with equality.

Regression.

The regression problem is to find a transformation R_T that transforms a formula $\phi(do(a, s))$ about $do(a, s)$ into a formula $\psi(s)$ about s (i.e. $R_T(\phi(do(a, s))) = \psi(s)$), and such that:

$\vdash T \rightarrow (\phi(s_i) \leftrightarrow R^*_T(\phi(s_i)))$

where s_i is a ground term and R^*_T represents an iterated application of R_T until the result of R_T application remains unchanged (in that case $R^*_T(\phi(s_i))$ is a sentence about s_0) (see [7]).

We say that R_T is **sound** if we have: $\vdash T \rightarrow R^*_T(\phi(s_i)) \Rightarrow \vdash T \rightarrow \phi(s_i)$.

We say that R_T is **complete** if we have: $\vdash T \rightarrow \phi(s_i) \Rightarrow \vdash T \rightarrow R^*_T(\phi(s_i))$.

At the present time we have not solved the regression problem in general. However, we can easily show that, if the SSAs are the same in every situations, a "rewriting" technique very close to the one defined in [8] can be proved to be sound and complete.

If the SSAs are different in imaginary situations, the problem is more complex and requires further investigations.

3 Conclusion

We have presented a general framework for belief revision in the Situation Calculus that does not require the definition of a plausibility distribution. The key idea is to have for each agent i SSAs for beliefs of the form of $SSA_{i,p}$.

[11] We omit for simplicity the axioms for functional fluents.

These axioms define how the truth values of the fluents change in imaginary situations when a sensing action is performed, and we do not have to remove the imaginary situations that are inconsistent with the observations as Scherl and Levesque do in [8]. Also, it is worth noting that in the proposed framework iterated belief revisions do not raise specific problems.

The framework is dramatically simplified when the SSAs are independent of the context. In that case it is simpler to define regression.

In this approach the sensing actions provide information about the truth of a fluent and not about the truth of a general formula, like in [8]. Is this a strong limitation? We do not think so, because the information that is directly delivered by a sensor is always an atomic information, whose interpretation, in the theory of the application domain, may be represented by a general formula.

References

1. R. Demolombe and A. Herzig. Obligation change in Dependence Logic and Situation Calculus. In A. Lomuscio and D. Nute, editors, *Proceedings of the 7th International Workshop on Deontic Logic in Computer Science*. Springer, LNAI 3065, 2004.
2. R. Demolombe and M. P. Pozos-Parra. A simple and tractable extension of situation calculus to epistemic logic. In Z. W. Ras and S. Ohsuga, editors, *Proc. of 12th International Symposium ISMIS 2000*. Springer. LNAI 1932, 2000.
3. R. Demolombe and M. P. Pozos-Parra. Belief change in the situation calculus: a new proposal without plausibility levels. In A. Herzig and H. van Ditmarsch, editors, *Workshop on Belief Revision and Dynamic Logic, European Summer School on Logic, Language and Information*, 2005.
4. J. McCarthy. Situations, actions and causal laws. In M. Minski, editor, *Semantic Information Processing*. The MIT press, 1968.
5. R. Petrick and H. Levesque. Knowledge equivalence in combined action theories. In *Proceeding 8th International Conference on Principles of Knowledge Representation and Reasoning*, 2002.
6. R. Reiter. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 359–380. Academic Press, 1991.
7. R. Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.
8. R. Scherl and H. Levesque. The Frame Problem and Knowledge Producing Actions. In *Proc. of the National Conference of Artificial Intelligence*. AAAI Press, 1993.
9. S. Shapiro, M. Pagnuco, Y. Lespérance, and H. Levesque. Iterated belief change in the situation calculus. In *Proc. of the 7th Conference on Principles on Knowledge Representation and Reasoning (KR2000)*. Morgan Kaufman Publishers, 2000.

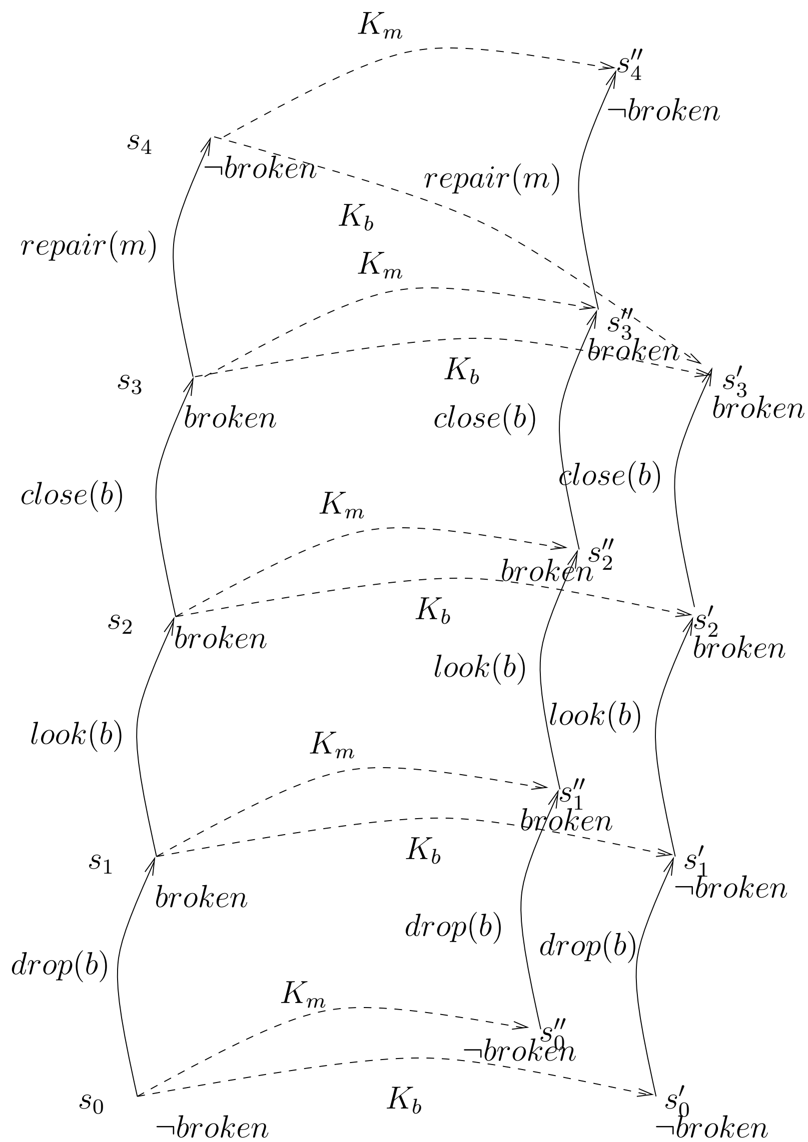

Fig. 1. Mother and baby beliefs' evolution.