Logical Theory of Concurrent Actions: Extended and Enlarged Abstract. *

Mikhail Soutchanski
Inst. for USA and Canada Studies
Lab. of Computer Science and Applied Research
2/3 Khlebny St., 121069 Moscow Russia
e-mail: usacanad@sovam.com (public mailbox)
fax: (+7095)-200-1207 , phone: (+7095)-405-9577

Eugenia Ternovskaia
Comp. Center of Russian Academy of Sciences
Artificial Intelligence Lab.
40 Vavilova St., 117967 Moscow Russia
fax: (+7095)-135-6159 , phone: (095)-425-3534

July 3, 1992
Revised: October 23, 1992; March 25, 1993; August 1993.

1 Introduction.

The ability to build plans - sequences or structures of the actions which transform an initial state of the world into the goal state is the necessary condition of intellectual behavior. For instance, we may consider a group of mobile robots in an environment. Each of them to achieve its aims should be able to plan its activity and to reason about result of joint actions.

The development of the logical formalizations of the reasoning about concurrent actions is important also because it is logical approach that will permit to unify the strategic analysis of the players' different interests, which is a part of the traditional game theory, with the analysis of the information exchange, in particular, with the analysis of argumentation, used by decision makers to overcome a conflict situation. This unification should encounter new perspectives for the research.

We proposed simple logical framework to represent knowledge about simultaneous execution of actions. As the starting point we used the situation calculus [1] which is the most known formalism for reasoning about sequential actions. Its reputation is fine owing to the clarity and the simplicity of the formalism. The negative attitude to the situation calculus in the artificial intelligence community was caused mainly by difficulties of dealing with concurrent actions/changes. We solved the following problem. How should causal

*The full (Russian) version will appear in the journal "Izvestia Rossi'skoi' Akademii Nauk, seria Tekhnicheskaia Kibernetika", 1993, N5. The translated (English) version of the paper will be published in the journal: "(Soviet) J. of Computer and System Sciences, 1993, V.31.

rules refer to simultaneous actions so that if we knew that some agents perform the group of actions which includes unspecified one, the appropriate causal rule nevertheless would be fired to derive default conclusions ? Such conclusions will agree with the intuition that the engagement in an activity (simultaneously with other actions) whose influence on the world is unknown, by default should not prevent us to make inferences about the results of other actions. These inferences will be defeasible and can be withdrawn in the presence of new causal rules with contrary specifications. For instance, in accordance with some ideological, teological or social doctrines all of us are unconsciously involved in a lot of activities. Nevertheless, it does not preclude us to reason about ourselves everyday actions.

The approach we proposed put no restrictions on how many such actions unspecified by causal rules are performed, and moreover, its flexibility is achieved without the complications of the causal rules. Our solution gives the possibility to derive easily new values of all properties of the world which can change after actions will be executed, in particular, values of those properties which were not influenced by actions. We stuck to the circumscription to deal with the *frame problem*. Our method solved also the *ramification problem*, because a theory may includes a number of static constraints between the properties of the world. As a consequence, all implicit results of actions can be found without applications of causal rules, i.e. without appeal to explicit dynamic constraints, but only from static constraints. We indicated also a not complicated way of coping with *qualification problem*, which is the problem how all sufficient preconditions for each action could be enumerated concisely.

Because this abstract is the excerpt from a paper, we would like to clarify the terminology we will keep in mind. This terminology originated from the ontological commitment to consider actions in the world from a finite automaton's point of view. In brief, we studied a multi-head, one-way deterministic automata with a finite number of states [2]. We adopted the following notions:

- **input symbol** on an automaton's input tape we named early and below as the action;
- **vector of input symbols** is a group of concurrent actions, equivalent term - an operator;
- **state** - circumstance, state-of-affairs, a set of properties whose status are considered as actual. The important difference between usual automaton's notion of state and the notion which is used in logics of actions is that for the state transition function of an automaton each state is indivisible entity. To the contrary, to reason about results of the actions each state is considered as a vector of boolean values such that each component corresponds to the value of a property. Because properties values are fluently alternate in dependence of actions on the input tape of an automaton, they named as the fluents in the situation calculus. This point of view allow to define the concept of the complexity of the frame problem solution. For this reason, we think that notion of state is very important (see the Appendix 1).
- **function of transitions between states** corresponds to a set of causal rules supplemented by static constraints on fluent values. Because we limit ourselves in this paper by consideration of deterministic automata only we will assume in the sequel that all actions have deterministic effect on the world. However, we expect that this restriction can be easily relaxed;
- **move** is an auxiliary temporal notion to enumerate in a sequence those operators which were accepted by an automaton. This notion is traditionally referred as a situation.

Thus, from the one hand, we distinguished between a state as a nontemporal entity, and a situation as an indicator of current position in an ordered sequence of activities, and, from the other hand, we distinguished between an ordinary, single action and an operator as a group of ordinary actions jointly performed.

It worth to mention, we followed a certain tradition in our terminology, whose sources will be specified in section 2. After a brief introduction of sorts for object variables, function and predicate symbols we demonstrate in section 3 our method of axiomatizing concurrent actions. For expository purposes we use slightly modified variant of the *Yale shooting* example [3]. We will focus our attention only on reasoning forward in time. In subsection 3.1 we outline the circumscriptive policies, subsection 3.2 contains hints how axioms should be formulate in general case. Section 4 compares our approach with related works. The last Section 5 notes some possible extensions and summarizes our paper.

2 Preliminaries.

Our approach based upon the idea of J.Weber [4]. He proposed to represent the dynamic component of the situational calculus ontological scheme by two distinct ontological entities. The first of them is a single action, second one is an operator. Roughly speaking, an operator is simultaneously performing actions, however it is a first class entity. Taken this step we can drastically simplify an axiomatization, because under this point of view on dynamics there is no need to add axioms about the inheritance of the properties of single actions by the operator which is composed from them. The operator is related with its compound actions by the binary predicate *Type*. Thus, J.Weber proposed to detach the type of an action from its name and make it into a predicate.

The predicate $Type$ will occur in causal rules and in the so-called scenario. The scenario will contain axioms describing which propositional fluents hold and which do not hold in the initial situation. It also will include axioms for defining each particular operator from the sequence of operators which are performed. To prevent undesirable initiations of causal rules $Type$ should be circumscribed. Without such circumscription it would be possible for some of the models to permit superfluous extensions of the predicate. After the circumscription each operator is composed only from the actions explicitly mentioned in the scenario.

The traditional axiom of inertion, proposed by J.McCarthy [5, 6], and his approach of minimizing abnormality to deal with the frame problem applies also to causal rules with operators. However, [4] noted that the circumscriptive policy known for him did not provide a correct solution to the classical *Yale shooting problem* (YSP) [3]. He coped with the frame problem using a method of L.Schubert [7], that is by adding several *explanation closure axioms*. But this method is applicable only in the presence of complete causal theory about the world. Moreover, the complexity of this solution is greater than the complexity of circumscriptive solution. For these reasons, we believed it worth to find such appropriate circumscriptive policy to deal with frame problem in the situation calculus with concurrent actions that is not subjected to YSP.

It turns out that the circumscriptive policy proposed by A.Baker [8], A.Baker&M.Ginsberg [9] as a solution of (YSP) can be adopted for our purpose. Their solution to the *Shooting* problem for standard situation calculus based on the idea that whether a situation is abnormal should not depend upon historical information about how the situation arose. We transformed their approach to situation calculus with operators constrained by action types. It was the paper [9] that first introduced very important notion of state and their method was named as state-based minimization approach. By following them we take

V.Lifschitz's [10] suggestion into consideration. He added to the persistence axiom an unary predicate $Frame$ that singles out so-called frame fluents to which the inertia will be limited.

The paper [11] also dealing with the representation of concurrent actions inside the framework of the situation calculus was issued practically simultaneously with the first version of our paper [12]. However, it seems that in contrast to the approach of [11], our method have some additional advantages such as flexibility and universality. We will perform thorough comparison of both alternatives in the section 4. We would like to emphasize here only that their circumscriptive policy for minimization abnormality also could be used for our case, because it solves YSP. Nevertheless, we look with favor on state-based minimization approach, due to importance of the notion of state. We revealed this importance for ourselves when we formulated the complexity criterion of the frame problem solution (see Appendix 1). In addition, we believed this notion to be an improvement of the initial ontological scheme of the situation calculus.

3 Language and Axioms

We will consider a subset of many-sorted first order language[1]. We use object variables of five sorts: for situation (s), for truth-valued fluents (f), for operators (o), for actions (a), and for circumstances (c) (or, equivalently, - states). Each state is a particular set of holding fluents, i.e. it designates a mapping from fluents to truth values. The object constants in the example are: the initial situation S_0; fluents $Loaded$ (gun) and $Alive$ (Fred); circumstances C_1, C_2, C_3, C_4 for the different combinations of fluents values; two particular operators which will be performed - O_1 and O_2; and actions $Shoot$, $Trigger$, Aim, $Wait$.

We use the ordinary situation-valued function symbol Res having operator term and situation term as arguments and the new unary function symbol St that has a situation as an argument and relates it to the corresponding state. The standard predicate constant symbol $Holds(f, s)$ asserts that a property f holds in a situation s. The binary predicate constant symbol $Type(o, a)$ mean that an operator o contains, in particular, an action a. Additionally, we use the unary predicate constant symbol $Frame(f)$ to express that f belongs to the frame, and the tern ary predicate constant symbol $Ab(f, o, c)$ to assert that a fluent f is abnormal after the performance of an operator o in a state c.

The predicate $Frame$ singles out primary fluents to which the inertia will be limited. This fluents are independent by analogy with coordinate frames [10]. These fluents do not change after actions which do not have the direct effect on them. All other fluents are secondary, i.e. their values are determined by those primary fluents which occur in static constraints.

As usually, whenever some circumscriptive policy is formulated we assume that sort and arity of each predicate variables exactly correspond to sort and arity of the initial predicate constant. We prove further that after the sequence of the circumscriptions of our initial theory we obtain a first-order theory.

We divide axioms on several groups in accordance with their intention. We will discuss general case and simultaneously the particular example.

The first group contains the only non-monotonic frame axiom with the predicate of abnormality with respect to a state:

$$Frame(f) \& \neg Ab(f, o, St(s)) \supset (Holds(f, s) \equiv Holds(f, Res(o, s))) \quad (1)$$

[1] Variables begin with lower-case letters. Constants, function symbols, predicate symbols begin with upper-case letters. Unbound variables are universally quantified.

This axiom says informally, that values of those primary properties f which are not abnormal in the state related with a current situation s will persist in the next situation that occurs as a result of performing an operator o. Roughly speaking, this axiom will force the persistence of as many properties as possible after the circumscription of the predicate Ab.

The second group is formed by axioms which describe the frame fluents. In our example we consider the following axioms:

$$Frame(Alive) \tag{2}$$

$$Frame(Loaded) \tag{3}$$

The notion of frame was restored by Lifschitz in [10] to allow principal interpretations of propositional fluents. Under this interpretation, propositional fluents are regarded as a subsets of the space of situations so the domain of propositional fluents is a power set of the domain of situation, and $Holds(f, s)$ is represented by the relation of membership s to f. It is such kind of interpretations that was intended for use, but omitting predicate $Frame$ from (1) in the earlier versions of situation calculus resulted in paradoxical conclusions.

The third group includes the following "existence of states" axiom:

$$\begin{aligned} \forall s\{((C_1 = St(s)) \equiv Holds(Alive, s)\&\ Holds(Loaded, s)) \\ \&((C_2 = St(s)) \equiv \neg Holds(Alive, s)\&\ Holds(Loaded, s)) \\ \&((C_3 = St(s)) \equiv Holds(Alive, s)\&\ \neg Holds(Loaded, s)) \\ \&((C_4 = St(s)) \equiv \neg Holds(Alive, s)\&\ \neg Holds(Loaded, s))\} \end{aligned} \tag{4}$$

Further we will formulate an axiom that stipulates uniqueness of names for fluents (i.e. $Alive \neq Loaded$). Axiom (4) ensures that for each of the four possible combinations of the fluent values, there is the unique corresponding circumstance (or state) C, such that for each situation s if just that combination takes place at s then $St()$ maps s to the state C. The reasons for formulation such kind of axiom were stressed by [8, 9]. It is this axiom that permits state-based minimization approach to overcome YSP. But unlike earlier formulation, we explicitly introduced object variables for circumstances, thus reifying those sets of fluents which holding in some situation make up a state. This axiom can be formulated alternatively in the second-order language [10, 13].

The forth group of axioms contains causal rules with predicate $Type$, that constrains operator variable of each causal rules. In our example we use one causal rule (note that we intentionally omit action type $Shoot$):

$$Holds(Loaded, s)\ \&\ Type(o, Trigger)\ \&\ Type(o, Aim) \supset \neg Holds(Alive, Res(o, s)) \tag{5}$$

The fifth group is comprised by unique names axioms:

$$Wait \neq Shoot \neq Aim \neq Trigger, \tag{6}$$

$$O_1 \neq O_2, \tag{7}$$

$$Alive \neq Loaded \tag{8}$$

where O_1 and O_2 - are particular operator constants from the scenario. We do not formulate domain-closure axiom for fluents, i.e. $f = Alive \vee f = Loaded$, because an inertion is applicable only to fluents included in the frame.

And finally, the sixth group of axioms is formed by scenario's assertions and, consequently, is specific only for the example under consideration:

$$Holds(Alive, S_0) \tag{9}$$

$$Holds(Loaded, S_0) \tag{10}$$

$$Type(O_1, Wait) \tag{11}$$

$$Type(O_2, Trigger) \ \& \ Type(O_2, Shoot) \ \& \ Type(O_2, Aim) \tag{12}$$

3.1 Circumscriptive policies and effects of circumscriptions.

As usually (see [14, 15]), we will denote by $Circum(A(P,Z); P; Z)$ the *global circumscription of the predicate* P *in the formula* $A(P,Z)$ *with* Z *allowed to vary* to represent the formula:

$$A(P,Z) \ \& \ \neg\exists p, z \ (A(p,z) \ \& \ p < P) \tag{13}$$

where z is a list of predicate and/or function variables whose arities equal to arities of corresponding letters from the tuple Z, and $p < P$ denotes

$$\forall x(p(x) \supset P(x)) \ \& \ \neg\exists x(P(x) \supset p(x))$$

.

We remind that the corresponding form of *pointwise circumscription*

$$A(P) \ \& \ \neg\exists x, z \ [P(x) \ \& \ A(\lambda y(P(y) \& \ x \neq y), z)]$$

is denoted by $C_P \ (A(P,Z); Z)$.

We remind also that [15] defined *the pointwise circumscription of P in A(P,Z) with the predicate Z allowed to vary only on the part V of its domain*, where V is a λ-expression $\lambda u V(u)$ of the same arity as $Z(u)$, which gas no parameters and contains neither P nor Z:

$$A(P) \ \& \ \neg\exists x, z \ [P(x) \ \& \ \forall u(\neg V(u) \supset z(u) \equiv Z(u)) \ \& \ A(\lambda y(P(y) \& \ x \neq y), z)].$$

In the sequel we will need in more flexible circumscription policies that also were defined by [15]. Let V be λ-expression $\lambda x u V(x,u)$ whose arity equals the sum of the arities of P and Z, and V_x is the function $\lambda u V(x,u)$ which maps every value of x into the set of all values of u satisfying $V(x,u)$. Then, whenever Z may vary only on that part V_x of its domain which depends of the point x where P is minimized, this circumscription is denoted by $C_P \ (A(P,Z); Z/V)$):

$$A(P) \ \& \ \neg\exists x, z \ [P(x) \ \& \ \forall u(\neg V_x \supset z(u) \equiv Z(u)) \ \& \ A(\lambda y(P(y) \& \ x \neq y), z)]. \tag{14}$$

If instead of Z we may vary some values of the predicate P itself while minimizing its value at some point, then this *pointwise circumscription of P in A(P) with P itself allowed to vary on the domain* V_x is $C_P(A(P); P/V)$:

$$A(P) \ \& \ \neg\exists x, p \ [P(x) \ \& \ \neg p(x) \ \& \ \forall u(\neg V_x \supset p(u) \equiv P(u)) \ \& \ A(p)]. \tag{15}$$

Here p is a predicate variable similar to P, V is a λ-expression $\lambda x u V(x,u)$ which has no parameters and does not contain P. The second term of (15) says that if $P(x)$ is *true* it is impossible to change its value to *false* without losing the property $A(P)$, even if the values of P will change arbitrary on the domain V_x.

We should circumscribe $Frame$ with Ab varying, in accordance with (13),i.e. $Frame$ is minimized at a higher priority than Ab. This is because we prefer to regard of the set of primary fluents (the extent of the predicate $Frame$) as already fixed when the inertion of fluents is described [10, 13, 6]. Afterwards, we will pointwisely minimize $Type$ in the theory resulted from the above circumscription in accordance with the policy defined in (15), with $Type$ itself allowed to vary anywhere. And finally, we will minimize Ab in the resulting theory with $Holds$ varying. Formally,

$$Circum(A; Frame; Ab) = A', \tag{16}$$

$$C_{Type}\ (A'; Type/V_1) = A'', \tag{17}$$

$$Circum(A''; Ab; Holds), \tag{18}$$

where A is the conjunction of the axioms (1)–(12), and V_1 is *true*.

The circumscription of predicate $Type$ eliminates all unforeseen characteristics of operators. After this circumscription, group of actions that is referred to as the operator O_1 includes neither of the actions $Trigger$, $Shoot$, Aim. In addition, group of actions that is referred to as the operator O_2 does not contain the action $Wait$. We could not be sure that the performance O_1 will not activate the causal rule (5) if we were not to make the extent $Type$ more exact.

The effect of circumscriptions (16)–(18) is equivalent to adding of the following formulas to the theory (1)–(12):

$$Frame(f) \equiv [f = Alive \vee f = Loaded] \tag{19}$$

$$Type(o, a) \equiv [o = O_1 \ \& \ a = Wait \vee o = O_2 \ \& \ a = Trigger \ \vee o = O_2 \ \& \ a = Aim \ \vee \\ o = O_2 \ \& \ a = Shoot] \tag{20}$$

$$Ab(f, o, c) \equiv [f = Alive \ \& \ o = O_2 \ \& \ c = C_1] \tag{21}$$

As the result we have:

$$A' = Circum(A; Frame; Ab) = A \ \& \ (19), \tag{22}$$

$$A'' = C_{Type}(A'; Type/V_1) = A' \ \& \ (20), \tag{23}$$

$$Circum(A''; Ab; Holds) = A'' \ \& \ (21) \tag{24}$$

To prove (22) we can perform an well-known transformation for the elimination of varying predicate at the cost of introduction second-order formulae with its subsequent simplification which result in the ordinary circumscription of $Frame$ in the conjunction of (2) and (3) (see [14, 16]). The proof of (24) can be obtained as a direct generalization of [8, 10] and have been given both in the preliminary version [12] and in the full version of this paper. This proof shows that we actually coped with YSP. To prove (23) we can use the similar arguments taken into account that the minimality should be considered pointwisely.

3.2 Remark and addendum to axiomatics.

We intentionally chose very simple example to demonstrate that our approach is YSP-free. However, if one needs to axiomatize more complicated domains he/she will encounter the complicated issue of concurrent actions which cancel out each other's effects. We propose to classify the concurrent performance of actions according to this issue:

- All actions are simple (i.e. each single action produce its own effect) and each one is independent of others.

- Some actions have cooperative effect, which they could not produce if they were to perform one by one (for instance, actions $Trigger$ and Aim in (5)), but neither actions cancel out each other's effects.

- Some actions are simple, some actions have cooperative effect, and additionally there is an action (a group of actions) such that would it be performed either another action could not be performed concurrently or some group of actions could not produced its normal effect.

It is not hard to imagine a lot of real conflict situations which can justify these cases.

The main advantage of our method of reasoning about concurrent actions is that we can represent each of the cases above very naturally. So, suppose that we know about a simple action that it can be executed only if some other actions $\{A_1, ..., A_n\}$ will not be executed concurrently. Then we simply have to add conjunctively to the left side of the causal rule formulas $\neg Type(o, A_i)$ $(i = 1, .., n)$, to say explicitly what actions the performing operator should not contain. Note that in the paper [11] this issue is much more complicated (see section 4).

As an additional argument of the flexibility and universality of our representational apparatus we would like to show that all sufficient preconditions for each action could be characterized without great complications. Because the modified YSP story we used for expository purposes is too simple to demonstrate how the qualification problem can be solved, we will discuss further another version of the story.

Let us assume that instead of axiom (10) the scenario will include the following axiom: $\neg Holds(Loaded, S_0)$. In that case, the operator O_2 even if it would be performed at first, did not fired the causal rule (5), and hence, the conclusion $\neg Holds(Alive, Res(O_2, S_0))$ could not be drawn. It is obvious, that the circumscription of the predicate $Ab(f, o, St(s))$ in that theory is equivalent to the assertion about emptiness of its extent. Nevertheless, as following from the axiom of inertion, we could derived

$$\neg Holds(Loaded, Res(O_2, S_0)),$$
$$Holds(Alive, Res(O_2, S_0))).$$

If we will look at these formulas carefully, we will see that they are meaningless: they say about the values of certain fluents at the situation which have been resulted from the execution of the operator O_2, but this operator cannot be executed due to the absence of its preconditions and, consequently, the situation its performance have been resulted in does not exist.

To eliminate such "ghost" situations several authors have introduced the binary predicate $Poss$ with an action and a situation as its arguments [13, 17]. We adopt this idea, but to conform with concurrent actions we will use the binary predicate symbol $Possible(a, c)$ having action term and state term as arguments. The predicate $Possible(a, St(s))$ asserts that at a state $St(s)$ corresponding to a situation s there are the prerequisites to perform an action a. Now to formulate effect axioms, each causal rule should be split along the implication sign:

$$Holds(Loaded, s) \supset Possible(Trigger, St(s)) \ \& \ Possible(Aim, St(s))$$

$$Possible(Trigger, St(s)) \ \& \ Possible(Aim, St(s)) \ \& \ Type(o, Trigger) \ \& \ Type(o, Aim)$$
$$\supset \neg Holds(Alive, Res(o, s))$$

In addition, we have to include in the theory a new group of axioms. Each axiom from this seventh group will assert that actions which have not been mentioned in causal rules can be performed unconditionally:

$$Possible(Shoot, St(s))$$

$$Possible(Wait, St(s))$$

And, finally we have to rewrite the axiom of inertion:

$$[Type(o,a) \supset Possible(a, St(s))] \supset$$

$$Frame(f) \ \& \ \neg Ab(f,o,St(s)) \ \& \ \supset (Holds(f,s) \equiv Holds(f, Res(o,s))$$

It is not hard to see that such axiom will allow to derive facts about new situations only if all actions composing an operator are individually possible.

We will circumscribe *Possible* pointwisely in the theory A'' resulted from (23) with *Holds* allowed to vary on the domain $V_2 = \lambda f, s'(s' \neq s)$, which depends from the point $(a, St(s))$ of minimization, i.e. we may vary *Holds* only at those situations s', which differ from situations s corresponding to the current state $St(s)$. Note that such circumscription policy was defined in (14). If we denote the effect of the circumscription $C_{Possible}$ $(A''; Holds/V_2)$, as the theory A''', then Ab have to be circumscribed in A''' with *Holds* varying.

4 Discussion.

One could attempt to incorporate concurrent actions into the situation calculus by means of the function *Compose* (from *action* $*$ *action* into *action*):

$$Holds(Loaded, s) \supset \neg Holds(Alive, Res(Compose(Aim, Trigger), s))$$

A similar approach is used in [18, 19]. But in that case one need to consider a great deal of new causal rules, which specify, for example, combinations of the actions that affect something in the world with an action that does not affect anything. In the one imaginary extension of *Shooting story* there can be many spectators watching the Fred's murder. No *watch* action affects anything. But to take into account the results of all composed actions, a lot of new causal rules should be formulated. Otherwise intuitively expected conclusion could not be drawn.

Although Schubert [7] composed actions a_1, a_2 by function $Costart(a_1, a_2)$, his method avoids this pitfall, because he can easy prove fluents persistence without circumscription. To do that job he formulated so-called *explanation closure axioms*. But tracking change in his framework is overly complicated because he added to the left side of effect axioms (causal rules) predicate $compatible(a,p)$, where a is an action and p is a plan (actions combination). That will hinder default conclusions about performing of unspecified action from being inferred. Our modification of YSP shows that appropriate conclusions nevertheless can be drawn despite the causal rule (5) does not mention *Shoot* at all. Moreover, it is unclear how to cope with qualification problem inside his framework. The advantage of Schubert's method is the easy of explanation generation. But this research avenue is beyond the scope of our work.

The method for reasoning about concurrent actions was proposed in papers [20, 11]. Their approach is similar to our one to some extent, because it is based also on the situation calculus. Authors used the binary predicate $In(a, \{A_1, ...A_n\})$ to express that primitive action a belongs to global action $\{A_1, ...A_n\}$. This predicate plays the same role as our predicate *Type*. Because global actions which will be executed are described by explicit enumeration of what primitive actions they contain as the components, there is no need to circumscribe *In*.

After the closer comparison of two approaches it turns out that for [11] the issue of concurrent actions which cancel out each other's effects is much more complicated then for us. They have to ensure that compound actions inherit the effects of their components [21]. Moreover, they concerned themselves with the problem how to take into account the fact that for an action A_{10} to override the effect of some global action $\{A_1, A_2\}$ itself must not be overridden by another action A_{23}. We have achieved the same result without compound actions, because we divide the dynamic component of the situational ontology on two distinct primitives.

Their causal rules are similar to our ones with the only exception. To forbid combinations of the interfering actions, at the left sides of their rules have to occur (under negation sign) the ternary predicate $Canceled(g_1, g_2, s)$. This predicate is true if "normal" effect of a global action g_1 is cancelled out by some other actions from a global action g_2. This representational mechanism have several shortcomings. First, as was mentioned by the authors in [11] they have to use sophisticated nonmonotonic tactics to take into account interactions and/or interference between actions. Second, let us suppose that we would like to formulate the causal rule for joint actions (such as our (5)) with cooperative effect. If we will follow their methodology, we have to write something like

$$\forall sg Holds(Loaded, s) \ \& \ In(Trigger, g) \ \& \ In(Aim, g) \ \&$$
$$\neg Canceled(Trigger, g, s) \ \& \ \neg Canceled(Aim, g, s) \supset \neg Holds(Alive, Res(g, s))$$

As the result of the circumscription of *Canceled* we find two different minimal models. At the first model: $Canceled(Trigger, \{Trigger, Aim\})$ is true; at the second model $Canceled(Aim, \{Trigger, Aim\})$ is true. This situation is fraught with unpleasant consequences: they might draw undesired conclusions. Our solution avoids these drawbacks and, moreover, it is more simple, because we does not need in the auxiliary predicate *Canceled*.

5 Conclusion.

We proposed the logical theory of concurrent actions that allow to draw conclusions about the execution of actions not all of which are specified in the causal rules. We coped with the *fluent oriented frame problem* and the complexity of our solution is less than the complexity of monotonic solution. We proposed the monotonic way to tackle the *action-oriented frame-problem*, but it seems that the complexity of our solution is lower than the complexity of the nonmonotonic method developed in [11]. We believed that the problem of complete characterization of the preconditions (*qualification problem*) can be overcome by the circumscription of the predicate *Possible*. Our approach also dealt with *ramification problem*), because implicit effects of actions can be found from constraints on the values of fluents. This effect is achieved due to varying *Holds* when *Ab* is circumscribed.

In our opinion, the concurrent actions representation will be used for reasoning about changes extended in time because it may happen that continuous changes are overlapped at a time [22]. It is important to note that correspondence between the situation calculus with action types and the explicit time-line temporal calculus established by Weber [4] facilitates future progress.

One of the future extensions may allow action types hierarchy, for example

$$\forall o(Type(o, Drive) \supset Type(o, Move))$$

A matter of prime concern is how to compute circumscription algorithmically. This point will be addressed in our future works.

Acknowledgements: discussions with Andrei Bondarenko, Charles Elkan, Vladimir Sazonov gave rise to clarification of our statement. M.Shanahan sent us helpful comments on the preliminary version of this paper. We are grateful to anonymous referees for their criticisms. We are greatly indebted to all scientists who have sent us their papers.

References

[1] J.McCarthy, P.Hayes *Some philosophical problems from the standpoint of artificial intelligence.* In: B.Meltzer and D.Michie (eds.), Machine Intelligence, v. 4, Edinburgh University Press, 1969, p. 463–502

[2] A.V.Aho, J.D.Ullman *The theory of parsing, translation and compiling.* "Prentice-Hall", Englewood Cliffs (N.J.), 1972.

[3] S.Hanks, D.McDermott. *Nonmonotonic logics and temporal projection.* Artif. Intell. v.33, 1987, p. 379–412.

[4] J.C.Weber *On the representation of concurrent actions in the situational calculus.* In: Proc. of the 8th Biennual Confer. of Canadian Society of Computat. Study of Intelligence, Ottava, 22-23 May, 1990, 28-32.

[5] J.McCarthy *Circumscription - a form of nonmonotonic reasoning.* In: Artific. Intell. v.13, 1980, p. 27–39

[6] J.McCarthy *Applications of circumscription to formalizing commonsense nonmonotonic reasoning.* In: Artific. Intell. v.28, 1986, p. 89–118

[7] L.Schubert *Monotonic solution of the frame problem in the situation calculus: an efficient method for worlds with fully specified actions.* In: H.E.Kyburg,Jr. et al. (eds.), Knowledge Representation and Defeasible Reasoning, Kluwer Acad. Publ., Dordrecht, 1990, p. 23-67

[8] A.B.Baker. *Nonmonotonic reasoning in the framework of situation calculus.* Artif. Intell. v.49, 1991, p.5–23

[9] A.B.Baker and M.L.Ginsberg. *Temporal projection and explanation.* In: Proc. of the 11th Int. Joint Conf. on Artif. Intell., Detroit (MI), August 1989, 906–911

[10] V.Lifschitz *Frames in the space of situations.* Artif. Intell. v.46, N 3, 1990,p. 365–376.

[11] F.Lin, Y.Shoham. *Concurrent actions in the situation calculus.* In: Amer. Assoc. of Artif. Intell., 10th National Conference on Artif. Intell., 1992, v.1, p. 590-595

[12] M.E.Soutchanski, E.A.Ternovskaia *Situation calculus is still alive, but Fred is not: reconciliation with concurrent actions* The paper has been accepted for presentation at the workshop on "Logic and Change" at GWAI'92 (Bonn, Germany), held September 1-3, 1992.

[13] V.Lifschitz *Toward a metatheory of action.* In: Proc. of the "2nd Int'l Conf. on Principles of Knowledge Representation and Reasoning", J.Allen, R.Fikes, E.Sandewall (Eds), 1991, p.376-387

[14] V.Lifschitz *Computing circumscription.* In: Proc. of the 9th Int. Joint Conf. on Artif. Intell., Los Angeles (CA), August 1985, v.1, 121–127

[15] Lifschitz *Pointwise circumscription: preliminary report.* In: Amer. Assoc. of Artif. Intell., 5th National Conference on Artif. Intell., 1986, v.1, p. 406-410

[16] Lifschitz V. *Circumscriptive theories: a logic-based framework for knowledge representation.* In: R.H.Thomason (ed.), Philosophical Logic and AI, Kluwer Academic Publ., Amsterdam, 1989, p. 109–159

[17] R.Reiter *The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression.* In: Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy, Academic Press, San Diego (CA),1991, p. 359–380.

[18] G.Große and R.Waldinger. *Towards a theory of simultaneous actions.* In: Proceedings the European workshop on planning - 91, (Sankt Augustin FRG), held 18–19 March, 1991. In: Lect. Notes in AI, N 522, Springer Verlag, Berlin, 1991, 78–87.

[19] M.Gelfond, V.Lifschitz, A.Rabinov *What are the limitations of the situation calculus ?* In: Essays for Bledsoe, Ed. R.Boyer, Kluwer Academic, 1991, p.167–177

[20] F.Lin, Y.Shoham. *Provable correct theories of action* In: Amer. Assoc. of Artif. Intell., 9th National Conference on Artif. Intell., 1991, v.1, p. 349-354

[21] M.Shanahan *E-mail message N frigate.do.395.* Date: Wednesday, 20th January 1993, 13:12:01.

[22] Soutchanski M.E. *Qualitative Reasoning about Physical Systems.* In: "Izvestia Rossi'skoi' Akademii Nauk, seria Tekhnicheskaia Kibernetika", 1992, N5. The translated (English) version of the paper will be published in the journal: (Soviet) J. of Computer and System Sciences, 1992, V.30.

Appendix 1.

We can consider a state as a boolean vector, whose dimension equals the number of fluents $| F |$. Let $C(s) = (v_1(s), \ldots, v_{|F|}(s))$, where s is a situation, be the state which corresponds to a situation s, and $v_1(s), \ldots, v_{|F|}(s)$ be the values of primary fluents in s ($v_i(s) \in \{0, 1\}$, for all i). Let us assume that the result of execution of an action a in a situation s is new situation $s' = res(a, s)$.

It is clear that there is unique boolean vector $p(s) = (p_1(s), \ldots, p_{|F|}(s))$ such that

$$C(s') = C(res(a, s)) = (v_1(s) + p_1(s)\ (mod\ 2), \ldots, v_{|F|}(s)) + p_{|F|}(s)\ (mod\ 2))$$

Thus, the boolean vector $p(s)$ completely characterize the situation which result from the execution of an action a in the initial situation s. Moreover, for each i if ith component of this vector $p_i(s)$ is 1, then ith fluent change its value, otherwise ($p_i(s) = 0$) this fluent is not affected by the action a.

To solve the fluent-oriented frame problem [2] one should determine "transformation vector" $p(s)$ for each action by adding to the theory several new axioms. Let k be a least natural number such that $2^k \geq | F |$. We can redefine boolean vector $p(s)$ as a boolean function having k variables (we will put values in the range between $p_{|F|}(s)$ and $p_{2^k}(s)$ arbitrary, for instance, let all these values be 0).

The main statement of this appendix is that there is 1–1 correspondence between the way of frame problem solution and the way this boolean function is defined. Because a boolean function can be equivalently defined by different ways, there are a lot of approaches to cope with the frame problem. As the first idea, we can try to construct disjunctive normal form for this function, i.e. we will explicitly enumerate only those k-tuples of boolean values $p_i(s)$ ($i = 1, \ldots, 2^k$) where the boolean function $p(x_1, \ldots, x_k)$ equals 1. Then such way of determination of the function $p(s)$ corresponds to circumscriptive approach. Alternatively, if we will give this function pointwisely enumerating all its values in sequence, then we will stick to the initial monotonic solution of the frame problem proposed in [1]. The further generalizations are straightforward.

From the considerations above follows the definition: one solution of the frame problem has lower complexity then another solution iff the boolean function $p(x_1, \ldots, x_k)$ that corresponds to the first solution has less boolean connectives (variables,...) than the boolean function that corresponds to the second solution.

[2] the distinction between terms fluent-oriented and action-oriented frame problem was introduced in F.Lin&Y.Shoham's paper [11]