Reasoning about Concurrent Actions within Features and Fluents

Choong-ho Yi
Department of Computer Science
University of Karlstad
S-651 88 Karlstad, Sweden
E-mail: Choong-ho.Yi@hks.se

Abstract

Sandewall proposed a systematic assessment method for temporal logics. In favour of the assessment of logics, we have introduced concurrency into his framework. The resulting formalism is capable of reasoning about interdependent as well as independent concurrent actions. We have then applied the entailment criteria PCM and PCMF to selecting intended models of common sense theories where concurrent actions are allowed, and proved that the criteria lead to only intended models for respective subsets of such theories.

1 Introduction

With restriction to the case where actions are assumed to occur sequentially, a number of nonmonotonic logics have been proposed in AI. In the meanwhile, research has advanced and one began to investigate model-theoretically whether a logic at hand produces conclusions correctly for a given theory. In his *Features and Fluents* [San94] Sandewall proposes a new approach in this context. For each of the major logics presented by then he identified, in a systematical way, a corresponding class of reasoning problems for which the logic is *proved* to obtain exactly the intended conclusions, i.e. the *range of applicability of the logic*.

On the other hand, logics have been suggested by, e.g. Kowalski & Sergot [KS86], Allen [All91], Pelavin [Pel91], Lansky [Lan90], and Georgeff [Geo86] which, directly or indirectly, allow concurrency. As was the case in reasoning with sequential actions, the importance and the need to identify the range of applicability of a given logic could not be emphasized too much even when concurrent actions are allowed. By this time, however, there has not been reported any systematic result in that direction. Actually, in contrast to the case for sequential actions where several entailment criteria, e.g. chronologically maximal ignorance [Sho88], have been proposed for selecting intended models, no such method has been tested for dealing with concurrent actions.

The work presented in this paper is an approach "in that direction" which has been done based directly on Sandewall. By making necessary generalizations we have introduced concurrent actions into his framework which was restricted to sequential actions. The resulting formalism is capable of reasoning about interdependent as well as independent concurrent actions. Then as a first step, we picked out the simplest entailment criteria PCM (prototypical chronological minimization of change) and PCMF (filtered PCM) of Sandewall, allowed independent concurrent actions into the classes of reasoning problems for which he had proven PCM and PCMF to be correct respectively, and proved that these criteria are still correct for the respective extended classes.

2 The Features and Fluents

In the Features and Fluents formalism Sandewall describes a *systematic approach* to common-sense reasoning, where he first defined several classes of reasoning problems, formalised different nonmonotonic logics, and then assessed the correctness of each logic on the classes of reasoning problems.[1] In his approach common-sense reasoning is understood on the basis of an *underlying semantics* which views the interaction between the ego of an intelligent agent and a world as a *game*, and which characterizes the actions the agent may evoke during the game in terms of a *trajectory semantics*. The *inertia problem* is approached by building inertia into the underlying semantics, i.e. the world has inertia so that features

[1] The presentation in this section is mainly based on [San93].

remain unchanged unless actions which override the inertia are performed.

2.1 The Game

The game is made in terms of a *finite development*

$$\langle \mathcal{B}, M, R, \mathcal{A}, \mathcal{C} \rangle.$$

$\mathcal{B}$ is a set of integers representing the time points at which the ego and the world alternate in the game and the largest member n of the set is "now". M assigns values to temporal constants and object constants. R is a mapping from a set $\{0, \dots, n\}$ of time points to a set $\mathcal{R}$ of states, i.e. R is a history of the world up to n. The pair

$$\langle M, R \rangle$$

then constitutes an interpretation for a given object domain $\mathcal{O}$. $\mathcal{A}$ is a set of tuples $\langle s, E, t \rangle$ where s and t are start respective end time of the action E and $s < t \leq n$, i.e. a set of actions which have been terminated at time n. $\mathcal{C}$ is a set of tuples $\langle s, E \rangle$ where s is start time and $s \leq n$, i.e. a set of actions which have been started but not terminated yet at time n. The tuple $\langle \mathcal{B}, M, R, \mathcal{A}, \mathcal{C} \rangle$ works as a "game board" in the game; the ego and the world alternate and extend it such that, roughly, the world executes the actions which are evoked by the ego.

2.2 The Trajectory Semantics and Nondeterminism

The *trajectory semantics* characterizes actions in terms of two functions. The function

$$\mathtt{Infl}(E, r)$$

represents a set of features which may be affected if the action E is performed in the state r. The function

$$\mathtt{Trajs}(E, r)$$

represents a set of possible *trajectories* of E initiated in r, where a trajectory, written as v, is expressed as a finite sequence

$$\langle r'_1, \dots, r'_k \rangle$$

of partial states r'_i $(1 \leq i \leq k)$ each of which assigns values to exactly those features appearing in $\mathtt{Infl}(E, r)$. This sequence is a trajectory of the action of the form $[s, s+k]\, E$, where s is start point in time and $s+k$ end point, and describes the effects of E successively for ecah time point during the execution period. Therefore, in the trajectory semantics one cares not only about the *nondeterministic* results of an action, but also about its trajectories. Since it is the world which performs actions, the pair $\langle \mathtt{Infl}, \mathtt{Trajs} \rangle$ characterizes a world. Later, ego and world will be defined exactly in a trajectory semantics generalized for concurrency.

2.3 Commonsense Scenarios

A commonsense theory is expressed as a tuple

$$\langle \mathcal{O}, \Pi, \textsc{scd}, \textsc{obs} \rangle.$$

$\mathcal{O}$ is an object domain. Π is a set of formulae describing the effects of actions, e.g.

$$[s, t]\ \mathit{OpenWindow} \Rightarrow [t]\ \mathit{Window} \hat{=} \mathit{Open}$$

which means if the *OpenWindow* action happens over the time interval s to t, then the feature *Window* representing the openness of the window has value *Open* at time t. Actually, this set is an exhaustive description of Trajs in logical formulae. SCD represents the actions scheduled to be performed, and is a set of action statements, e.g.

$$[3, 5]\ \mathit{OpenWindow},$$

and time statements, e.g.

$$t_1 < s_2.$$

The effect

$$[5]\ \mathit{Window} \hat{=} \mathit{Open}$$

of performing $[3, 5]$ *OpenWindow* is then obtained by applying the action statement to Π. The result of replacing each action statement in SCD by the effects specified by Π will be written as $\Pi(\textsc{scd})$. OBS is a set of observation statements, i.e. any formulae not containing action statements.

2.4 Intended Models

If a scenario $\Upsilon = \langle \mathcal{O}, \Pi, \textsc{scd}, \textsc{obs} \rangle$ is given, then the set of *intended models* of Υ is defined as follows. First, select an arbitrary world which is exactly characterized by Π, select an arbitrary ego, an arbitrary initial state and an arbitrary initial mapping for temporal and object constants. Let

$$\mathit{Mod}(\Upsilon)$$

be a set of completed developments $\langle \mathcal{B}, M, R, \mathcal{A}, \mathcal{C} \rangle$ obtained from games between them over Υ such that there is a 1:1 correspondence between members of the set $\mathcal{A}$ and those of SCD (i.e. all of the scheduled actions have been performed successfully), and all formulae in SCD $\cup$ OBS are true in $\langle M, R\rangle$ having $\mathcal{O}$ as object domain. Then,

$$\{\langle M, R\rangle | \langle \mathcal{B}, M, R, \mathcal{A}, \mathcal{C}\rangle \in Mod(\Upsilon)\}$$

is the set of intended models of Υ.

2.5 Taxonomy of Reasoning Problems

One of the characteristics of Sandewall's systematic approach is the use of *taxonomy* of reasoning problems. The taxonomy is obtained by making explicitly *ontological assumptions* about actions and world and *epistemological assumptions* about knowledge about the actions and the world to be reasoned with. For example, the ontological characteristic **I** represents that inertia holds; **A** represents "alternative results", i.e. the effects of an action are conditional on the starting state; **C** represents that concurrent actions are allowed; **D** represents dependencies between features, i.e. change in one feature implies possibility of immediate change in another feature; and so on. The classical frame problem is then denoted as **IA**, and the ramification problem is in the **IAD** ontological family.

In addition, for a more precise specification, he provides *sub-characteristics* which are additional constraints within characteristics and which are written with small letters. For example, **Is** represents the subfamily of **I** where all actions take a single time step; **An** denotes the subfamily of **A** where all features which are allowed to be influenced as result of an action in a given state should change their value if the action is performed in that state: if a feature with three possible values red, yellow, green is influenced by an action, then the action is allowed to nondeterministically change the value from red to yellow or red to green, but it is not allowed to choose between switching from red to green or keeping it red; and so on. All of the sub-characteristics can be defined precisely in terms of the trajectory semantics.

In order to characterize the epistemological assumptions, a list of epistemological characteristics is provided. For example, $\mathcal{K}$ denotes complete and correct knowledge about actions. $\mathcal{K}$p represents that in addition there are no observations about any time point after the initial one. Therefore $\mathcal{K}$p denotes pure prediction problem.

The ontological and the epistemological descriptors are then combined and characterize a class of systems or reasoning problems. For example, the combination $\mathcal{K}$p$-$**IsAn** represents a set of reasoning problems satisfying the restrictions **IsAn** and $\mathcal{K}$p. Such combinations are used for identifying the applicability of different logics. That is, the correctness of a logic is defined for a class of reasoning problems in terms of equality between the set of intended models and the set of preferred models.

2.6 PCM and PCMF

The entailment criteria PCM has been formalized by Sandewall as follows. Let $I = \langle M, R\rangle$ be an interpretation, then the *breakset* of I at time t is defined as a set of features which change value from time $t-1$ to t; formally

$$breakset(I, t) = \{f_i | \; R(f_i, t-1) \neq R(f_i, t)\}.$$

Definition [San94] Let $I = \langle M, R\rangle$ and $I' = \langle M', R'\rangle$ be interpretations, then I is said to be *PCM-preferred over* I', written as $I \ll_{pcm} I'$, iff $M = M'$ and there is some time point $\mathtt{t}$ such that

- $R(f, t) = R'(f, t)$ for all features f in ν and for all time points $t < \mathtt{t}$, and
- $breakset(I, \mathtt{t}) \subset breakset(I', \mathtt{t})$. □

Sandewall has shown that PCM guarantees only intended models for reasoning problems within the class $\mathcal{K}$p$-$**IsAn** described above.

However, the applicability can be improved by combining PCM with the entailment technique *filtering* into the new criterion PCMF. The idea with filtering [San89] is to separate the premises in the sets SCD and OBS. Let $[\![\Gamma]\!]$ denote the set of classical models for the set Γ, then the set of PCM-minimal models is

$$Min(\ll_{pcm}, [\![\Pi(\text{SCD}) \cup \text{OBS}]\!]).$$

The PCMF-minimal models set is instead

$$Min(\ll_{pcm}, [\![\Pi(\text{SCD})]\!]) \cap [\![\text{OBS}]\!],$$

so that the PCM-minimization is performed before the observations. PCMF is correct for a larger class $\mathcal{K}-$**IsAn** of reasoning problems, i.e. the restriction that only initial observations are allowed, is now removed. For the detailed discussion and the full proofs, please refer to [San94].

3 Concurrency in the Trajectory Semantics

In Sandewall [San94] the trajectory semantics was defined with the restriction that only sequential actions are allowed, i.e. at most one action is considered at a time. However, in dealing with concurrent actions new problems arise which were not there for sequential actions. Concurrent actions imply that at least two actions are involved at a given time, and, consequently, that interactions may arise between them. Therefore, the semantics must be modified.

3.1 Concurrent Interactions

In a broad sense, concurrent actions may be *interacting* or *noninteracting*, and, if they interact, they may interact *interferingly* or *noninterferingly*. Given two or more actions, one cannot say unconditionally whether they interact or not, and if they do, whether they interfere or not. The behaviour of individual actions in these respects is dependent on their *start states* and the *trajectories chosen* for them. The set of features influenced by executing an action in a state may be different if the action occurs in another state. What it means is that any two overlapping actions E_1 and E_2 which influence no feature in common if they start in state r_1 and r_2 respectively, i.e.

$$\mathtt{Infl}(E_1, r_1) \cap \mathtt{Infl}(E_2, r_2) = \emptyset,$$

can easily show different effects if performed in different states, e.g. such that

$$\mathtt{Infl}(E_1, r_1) \cap \mathtt{Infl}(E_2, r_3) \neq \emptyset.$$

For another example, let

$$\{v_1, v_2\} \subseteq \mathtt{Trajs}(E_1, r_i)$$
$$\{v_3, v_4\} \subseteq \mathtt{Trajs}(E_2, r_j).$$

Then it may be the case that the trajectory v_1 interacts with v_3, and, also with v_4 but differently than with v_3, while there is no interaction between v_2 and v_4.

However, as will be discussed when we define concurrent interactions formally, the interactions are relative to the *start time points* of actions as well.

3.2 Trajectory Preserving Condition

As mentioned in our previous discussion, the function $\mathtt{Trajs}(E, r)$ captures the set of possible trajectories of the action E w.r.t. its starting state r. In the case of sequential actions, it was enough to say merely that there are several ways for a given action to go. In discussing about concurrent actions, one also needs to know the *conditions under which each trajectory proceeds as such*, since, unless these conditions are available, interactions between trajectories of concurrent actions cannot be represented and reasoned about effectively. What is missing in this function is to represent such conditions for every trajectory $v \in \mathtt{Trajs}(E, r)$.

Definition Let E be an action and r a state, then each trajectory $v \in \mathtt{Trajs}(E, r)$ is defined now as a pair

$$\langle\langle r'_1, \ldots, r'_k\rangle, \langle r''_1, \ldots, r''_k\rangle\rangle$$

of finite sequences of partial states where

$$\langle r'_1, \ldots, r'_k\rangle$$

is a *trajectory description* and is our "old" trajectory, and

$$\langle r''_1, \ldots, r''_k\rangle$$

is *trajectory preserving condition*. If the action E is started at time point s, each r''_i $(1 \leq i \leq k)$ in the trajectory preserving condition specifies conditions to hold at time point $s + i$ in order for the action E to proceed as described by $\langle r'_1, \ldots, r'_k\rangle$. □

The trajectory description will be written as d, and the trajectory preserving condition as pc. The trajectory preserving condition is a generalization of the *prevail condition* of Sandewall & Rnnquist [SR86] in that both two refer to conditions that should hold during an action performance. The difference is that their prevail condition only represents conditions that should hold during the *whole duration* of an action, while the trajectory preserving condition can freely refer to conditions not solely for the whole duration but also for some parts of it, or even for a time point.

3.3 Concurrent Interactions in terms of Infl and pc

In our formalism, concurrent interactions are considered at the level of *concurrent trajectories*, i.e. the trajectories of concurrent actions. Concurrent trajectories can interact in two ways, namely, by influencing some feature in common, or by influencing a feature which appears in the preserving condition of the other trajectory. The formal definition follows. For the forthcoming discussion, we introduce some notations first. For a given trajectory

$$\langle d, pc\rangle = \langle\langle r'_1, \ldots, r'_m\rangle, \langle r''_1, \ldots, r''_m\rangle\rangle,$$

$d(k)$ and $pc(k)$ shall be the k:th $(1 \leq k \leq m)$ element of the trajectory description d and that of the trajectory preserving condition pc, respectively. Similarly, $d(f, k)$ and $pc(f, k)$ will be used to express the value of a feature f defined in $d(k)$ respectively $pc(k)$. Let $v = \langle d, pc\rangle$, then $length(v)$ shall be the length of time period over which the trajectory v proceeds, i.e. m. In addition, by $\mathcal{F}(r)$ we denote the set of features which are defined in a given state r.

Definition Given two arbitrary actions of the form $[s_1, t_1]E_1$ and $[s_2, t_2]E_2$ such that $max(s_1, s_2) < min(t_1, t_2)$, i.e. they are concurrent actions, let R be an arbitrary history defined over $[0, s]$ where $s \geq max(s_1, s_2)$, and, for $1 \leq i \leq 2$, let $R(s_i) = r_i$, let $v_i = \langle d_i, pc_i \rangle$ be a member of $\mathtt{Trajs}(E_i, r_i)$ of length $t_i - s_i$, let $x_i = max(s_1, s_2) - s_i + 1$ and $y_i = min(t_1, t_2) - s_i$, i.e. x_i and y_i are intended to represent the first respectively the last moment at which v_i might interact with the other trajectory. Then v_1 and v_2 are said to

- `Infl`-*interact* iff
  - $\mathtt{Infl}(E_1, r_1) \cap \mathtt{Infl}(E_2, r_2) \neq \emptyset$;
- *pc-interact* iff
  - $\mathtt{Infl}(E_1, r_1) \cap \mathcal{F}(pc_2(k_2)) \neq \emptyset$ for some k_2 where $x_2 \leq k_2 \leq y_2$ or
  - $\mathtt{Infl}(E_2, r_2) \cap \mathcal{F}(pc_1(k_1)) \neq \emptyset$ for some k_1 where $x_1 \leq k_1 \leq y_1$. □

Notice that $\mathtt{Infl}(E_1, r_1)$ and $\mathtt{Infl}(E_2, r_2)$ represent the set of features defined in the elements of d_1 respectively d_2, and that the interactions in the above definition are relative to the choice of s_1 and s_2. Additionally, let

$$
\begin{aligned}
f_1 &\in \mathtt{Infl}(E_1, r_1) \cap \mathtt{Infl}(E_2, r_2), \\
f_2 &\in (\mathtt{Infl}(E_1, r_1) \cap \mathcal{F}(pc_2(k_2))) \cup \\
&\quad\ (\mathtt{Infl}(E_2, r_2) \cap \mathcal{F}(pc_1(k_1))).
\end{aligned}
$$

Then, for any f_1 and f_2, v_1 and v_2 are said to `Infl`-interact through f_1 and *pc*-interact through f_2, respectively. Therefore, concurrent trajectories may `Infl`-, or *pc*-interact, or both of the two.

Based on these concepts, we can go on and identify clearly interferences too. `Infl`-interacting trajectories interfere iff they assign at some overlapping time point different values to some feature through which they `Infl`-interact. And the *pc*-interacting interfere iff one trajectory assigns at some overlapping time point a different value from the trajectory preserving condition of the other to some feature through which they *pc*-interact. The precise definition is as follows.

Definition Let the same assumptions be given as in the previous definition, and let k_1 and k_2 be any time points satisfying $x_1 \leq k_1 \leq y_1$, $x_2 \leq k_2 \leq y_2$ and $s_1 + k_1 = s_2 + k_2$. Then the trajectories v_1 and v_2 are said to

- `Infl`-*interfere* iff
  - they `Infl`-interact through some feature f and
  - $d_1(f, k_1) \neq d_2(f, k_2)$ for some k_1 and k_2;
- *pc-interfere* iff
  - they *pc*-interact through some feature f and
  - $(d_1(f, k_1) \neq pc_2(f, k_2) \vee d_2(f, k_2) \neq pc_1(f, k_1))$ for some k_1 and k_2. □

In addition, independent actions are defined trivially such that they neither `Infl`-interact nor *pc*-interact.

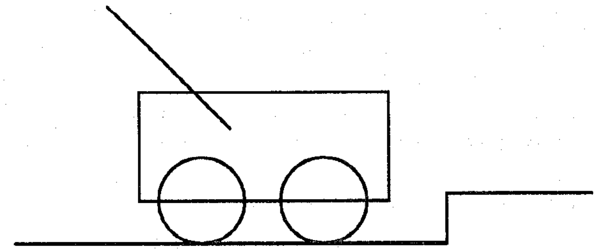

Figure 1: A cart in front of the curb.

3.4 An Example

As an example, consider a situation shown in Figure 1 where a cart is standing in front of the curb, and we want to move it over the curb. In order to do that, you should first press down the handle while the front wheels go over the curb and then lift the handle while the back wheels go over. The vertical position of the front wheels and the back wheels is represented in relation to the curb by the features *VpF* repectively *VpB* which have as their value domain

$$\{on, lifted\}.$$

Therefore, the wheels may be *on* the ground, under or over the curb, or *lifted* higher than the curb so as to pass it freely. Similarly, HpF and HpB whose value domain is

$$\{before, passed, elsew\},$$

state the horizontal position of the front and the back wheels in relation to the curb. *before* means that the front or the back wheels are directly before the curb and ready to go over, *passed* they passed it, and *elsew* the wheels are elsewhere before the curb but not directly in front of it. Let PH denote the action "press handle" whose intended effect is to lift the front wheels. The action PC, "push cart", has the effect of moving the cart over the curb while its wheels are lifted. By restricting to these actions and some states which would characterize the actions well, let us briefly consider about concurrent interactions in the trajectory semantics.

Let r_1 be any state which satisfies

$$r_1 \supset \{VpF \hat{=} on,\ VpB \hat{=} on\},$$

and let

$$\mathtt{Infl}(PH, r_1) = \{VpF\}.$$

It means, pressing down the handle of the cart when the wheels are on the ground, can influence the vertical position of the front wheels. Then consider a trajectory

$$v_1 = \langle d_1, pc_1 \rangle \in \mathtt{Trajs}(PH, r_1)$$

where

$$\begin{aligned} d_1 &= \langle \{VpF \hat{=} lifted\}, \{VpF \hat{=} lifted\}, \\ &\qquad \{VpF \hat{=} lifted\}, \{VpF \hat{=} lifted\} \rangle \\ pc_1 &= \langle \emptyset, \emptyset, \emptyset, \emptyset \rangle. \end{aligned}$$

That is, a possible trajectory of the action PH initiated in r_1 is that it proceeds over 4 time units and holds the front wheels lifted over the interval, i.e.

$$d_1(VpF, 1) = \ldots = d_1(VpF, 4) = lifted.$$

No trajectory preserving condition is required. Of course, $\mathtt{Trajs}(PH, r_1)$ may also contain other trajectories. For convenience, however, we assume that v_1 is its only member.

Next, suppose a state r_2 such that

$$\{VpF \hat{=} lifted, HpF \hat{=} before, VpB \hat{=} on, HpB \hat{=} elsew\}.$$

Then, for the action PC we consider in a similar way a trajectory

$$v_2 = \langle d_2, pc_2 \rangle \in \mathtt{Trajs}(PC, r_2)$$

where

$$\begin{aligned} &\mathtt{Infl}(PC, r_2) = \{HpF, HpB\} \\ &d_2 = \langle \{HpF \hat{=} before, HpB \hat{=} elsew\}, \\ &\qquad \{HpF \hat{=} passed, HpB \hat{=} elsew\}, \\ &\qquad \{HpF \hat{=} passed, HpB \hat{=} before\} \rangle, \\ &pc_2 = \langle \{VpF \hat{=} lifted\}, \{VpF \hat{=} lifted\}, \emptyset \rangle. \end{aligned}$$

According to the trajectory v_2, pushing the cart in r_2 would proceed as follows; if the front wheels continue to be held lifted over the curb over $[s+1, s+2]$ where s is start time of PC, i.e.

$$pc_2(VpF, 1) = pc_2(VpF, 2) = lifted,$$

then the cart rolls on back wheels so that the front wheels pass the curb at $[s+2]$, i.e.

$$d_2(HpF, 2) = passed,$$

and the back wheels reach the curb at $[s+3]$, i.e.

$$d_2(HpB, 3) = before.$$

Here, too, v_2 is assumed to be the only member of $\mathtt{Trajs}(PC, r_2)$.

Now, let r be the initial state

$$\{VpF \hat{=} on, HpF \hat{=} before, VpB \hat{=} on, HpB \hat{=} elsew\}$$

pictured in Figure 1, and let 0 be initial time point. In addition suppose that we press the handle of the cart over the interval $[0, 4]$ and push the cart concurrently over $[1, 4]$. Since r satisfies the condition of r_1, $v_1 \in \mathtt{Trajs}(PC, r)$. Let v_1 be chosen for $[0, 4]\,PH$. (We need to and will discuss in detail about choosing trajectories for given actions in next section.) By starting the trajectory v_1 from time 0, the front wheels are lifted at succeeding time point 1, and this is the only change caused by v_i at time 1; recall $d_1(1) = \{VpF \hat{=} lifted\}$. Therefore the state of the world is changed from r to r_2 over $[0, 1]$. And so, let the trajectory v_2 be selected for $[1, 4]\,PC$. The concurrent trajectories v_1 and v_2 interact, namely *pc*-interact through the feature VpF, but not interfere. Actually, v_1 enables v_2 such that the trajectory preserving condition pc_2, i.e. $pc_2(1)$ and $pc_2(2)$, is satisfied by $d_1(2)$ and $d_1(3)$.

4 Trajectory Semantics World and Ego

For dealing with concurrency in the trajectory semantics, the *single-timestep ego-world game* where the world advances time by exactly one time step at a time, is adopted. It offers a clear and simple underlying semantics, and reduces the technical complexity. Following Sandewall, trajectory semantics world and ego are defined as follows. Formal definitions are given in [Yi95].

4.1 Trajectory Semantics World in a Single-timestep Ego-World Game

As mentioned previously, the ego-world game is performed in terms of a finite development $\langle \mathcal{B}, M, R, \mathcal{A}, \mathcal{C} \rangle$. Let a world description $\langle \mathtt{Trajs}, \mathtt{Infl} \rangle$ be given, and let $\langle \mathcal{B}, M, R, \mathcal{A}, \mathcal{C} \rangle$ be a development given for a single-timestep game between a trajectory semantics world and a trajectory semantics ego where the "now" time, i.e. $max(\mathcal{B})$, is n, and the history R is defined over $[0, n]$. Assume that the world takes over the control now, and that the world modifies the development into $\langle \mathcal{B}', M', R', \mathcal{A}', \mathcal{C}' \rangle$. As we will see, the modification is made differently according to whether the current action set $\mathcal{C}$ is empty or not. However, the following hold irrespective of it; $\mathcal{B}' = \mathcal{B} \cup n+1$, i.e. the now-time is increased by one time point, $M \subseteq M'$, and the restriction of R' to the period $[0, n]$ equals R.

If $\mathcal{C} = \emptyset$, then it means that no action is going on at time n. It's because all actions being processed have "died out", and because the ego has decided not to start any new action at this moment. In this case, the world extends history such that $R'(n+1) = R(n)$, and $\mathcal{A}'$ and $\mathcal{C}'$ are set to $\mathcal{A}$ and $\emptyset$ respectively.

On the other hand, at n there may be an arbitrary number of actions to be considered, i.e. $\mathcal{C}$ has an arbitrary number of members $\langle s_i, E_i \rangle$ where s_i is start time of E_i and $s_i \leq n$. Here, it's not sure whether all members of $\mathcal{C}$ can be performed concurrently. For example, it may be that some actions are evoked at n but interfere with other actions which have been started previously. When conflict arises between concurrent actions, one may choose to perform as many actions as possible, or break the game there, or abandon all interfering ones, or save earlier actions first, and so on. However, rather than choose a specific "policy" among them, we leave our underlying semantics open and more general on the question, and simply find some combination of compatible actions containing some or all members of $\mathcal{C}$. Actually, it's a combination of mutually compatible trajectories which are in accordance with the history R. There may be more than one such combination. The precise definition follows.

Definition Let a development $\langle \mathcal{B}, M, R, \mathcal{A}, \mathcal{C} \rangle$ be given where $max(\mathcal{B}) = n$ and R is defined over $[0, n]$. Then a *compatible-trajectories combination*, written as c, for $\mathcal{C}$ is defined as any set of trajectories $v_i = \langle d_i, pc_i \rangle \in \mathtt{Trajs}(E_i, R(s_i))$ for some or all members $\langle s_i, E_i \rangle$ of $\mathcal{C}$ which satisfy

- $length(v_i) > n - s_i$,
- if $n > s_i$, then $d_i(k) \cup pc_i(k) \subseteq R(s_i + k)$ for all $1 \leq k \leq n - s_i$, i.e. v_i agrees with the previous history,
- v_i interferes in no way with any other member of c and
- the trajectory preserving condition pc_i is satisfied by other trajectories in c or by applying inertia. □

Let c be a compatible-trajectories combination selected by the world. Then the world extends the history as follows.

$$R'(n+1) = R(n) \oplus \bigcup_{\langle d_i, pc_i \rangle \in \mathsf{c}} d_i(n - s_i + 1)$$

where $\oplus$ is Sandewall's "override" operation over states such that the value of a feature f in $[r \oplus r']$ equals that in state r' if f is defined there, otherwise that in state r. Using this $\oplus$ operation inertia is interwoven into the semantics. Notice that, since the trajectories in c do not interfere, the partial states at time $n+1$, $d_i(n - s_i + 1)$, which are obtained from the trajectories, can be "put together" into a union without causing any conflict. On the other hand, the trajectory preserving conditions at that time point, $pc_i(n - s_i + 1)$, do not participate in the history extension. They are expected to be satisfied by other trajectories $d_j(n - s_j + 1)$ or inertia.

For some trajectory $\langle d_j, pc_j \rangle \in \mathsf{c}$, if $d_j(n - s_j + 1)$ is the last element of d_j, then it means that the trajectory v_j has been performed successfully and is terminated at $n+1$. Therefore $\mathcal{C}'$ is obtained by first making a set of corresponding tuples $\langle s_i, E_i \rangle$ for each member v_i of c and then removing from the set all of the "terminated" members $\langle s_j, E_j \rangle$. On the other hand, tuples of the form $\langle s_j, E_j, n+1 \rangle$ are added to the past action set $\mathcal{A}$ for the completed actions E_j, and $\mathcal{A}'$ is set to the resulting set.

4.2 Trajectory Semantics Ego

In the game it is the ego that activates one or more actions in its turn. Let a development $\langle \mathcal{B}, M, R, \mathcal{A}, C \rangle$ be given where $max(\mathcal{B}) = n$, then for each action E_i which is started by the ego at time n, a corresponding tuple $\langle n, E_i \rangle$ shall be added to the C component. If the ego passes on the control to the world without evoking any new actions, then no change is made for C. This definition does not need to be restricted to the single-timestep games.

5 Reapplying PCM and PCMF to Concurrent Actions

As a subset of concurrent actions, we have defined into Sandewall's taxonomy the class **Ci** where all trajectories of concurrent actions are mutually independent. Then we have introduced **Ci** into the classes $\mathcal{K}$**p−IsAn** and $\mathcal{K}$**−IsAn** and extended them to $\mathcal{K}$**p−IsAnCi** respectively $\mathcal{K}$**−IsAnCi**. By generalizing Sandewall's proofs of the correctness of PCM for $\mathcal{K}$**p−IsAn** and PCMF for $\mathcal{K}$**−IsAn**, we have proven that PCM and PCMF are still correct for the new classes $\mathcal{K}$p−**IsAnCi** and $\mathcal{K}$−**IsAnCi**, respectively. For want of space, the full proofs and details are reserved in [Yi95].

6 Conclusion and Future work

This work gives a base for analyzing the range of applicability of logics for concurrency. The result of our work implies that Sandewall's systematic approach can easily be extended to concurrency, and that most of the results shown by him for the case of sequential actions may be reobtained similarly for concurrent actions as well after necessary modifications. Considering the necessity of concurrent actions in common sense reasoning, and also, the importance of identifying the applicability of a given logic, it is urgently required to extend such assessment for concurrent actions. In addition, we need to lift the restriction of independent actions in next step, so that even interdependent actions are considered when assessing applicabilities.

References

[All91] James F. Allen. Temporal reasoning and planning. In J. F. Allen, H. A. Kautz, R. N. Pelavin, and J. D. Tenenberg, editors, *Reasoning about Plans*, pages 1–68. Morgan Kaufmann, 1991.

[Geo86] Michael P. Georgeff. The representation of events in multigent domains. In *AAAI*, pages 70–75, 1986.

[KS86] Robert Kowalski and Marek Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.

[Lan90] Amy L. Lansky. Localised representation and planning. In J. Allen, J. Hendler, and A. Tate, editors, *Readings in Planning*, pages 670–674. Morgan Kaufmann, 1990.

[Pel91] Richard N. Pelavin. Planning with simultaneous actions and external events. In J. F. Allen, H. A. Kautz, R. N. Pelavin, and J. D. Tenenberg, editors, *Reasoning about Plans*, pages 127–211. Morgan Kaufmann, 1991.

[San89] Erik Sandewall. Filter preferential entailment for the logic of action in almost continuous worlds. In *IJCAI*, 1989.

[San93] Erik Sandewall. The range of applicability of nonmonotonic logics for the inertia problem. In *IJCAI*, 1993.

[San94] Erik Sandewall. *Features and Fluents, A Systematic Approach to the Representation of Knowledge about Dynamical Systems.* Oxford University Press, 1994.

[Sho88] Y. Shoham. *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence.* MIT Press, Massachusetts, 1988.

[SR86] Erik Sandewall and Ralph Rönnquist. A representation of action structures. In *AAAI*, pages 89–97, 1986.

[Yi95] Choong-Ho Yi. Reasoning about concurrent actions in the trajectory semantics. Licentiate Thesis, 1995. Department of Computer and Information Science, Linköping University.