A history-oriented view of causality

Sanjai Narain
Jeff Rothenberg

RAND Corporation
1700 Main Street
Santa Monica, CA 90406

The causality relation yields a simple basis for computing event occurrences in dynamic systems using the rule:

occurs(F) if causes(E,F) & occurs(E)

Here *causes(E,F)* means that event *E* brings about or is directly responsible for event *F* and *occurs(E)* means that event *E* occurs. However, it can be quite difficult to compute the causality relation when statements about it refer to the past or the future of causing events. A natural example of such a statement, and its formalized version are:

An event of aircraft taking off from an aiport towards a radar causes an event of detection by that radar in the future provided an event of its changing course does not occur in between, and provided in the past an event of its being instructed to fly low has not occurred.

causes(flies_towards(aircraft(A),radar(R),T),in_range(aircraft(A),radar(R),T1)) if
detection_time(aircraft(A),radar(R),T,T1) &
not exists(X). X<T & occurs(told_to_fly_low(aircraft(A),X)) &
not exists(X).T<X<T1 & occurs(changes_course(aircraft(A),X)).

Here *flies_towards(A,R,T)* denotes an event of object *A* taking off towards object *R* at time *T*. Similarly for *in_range(A,R,T), told_to_fly_low(A,T), changes_course(A,T). detection_time(A,R,T,T1)* computes the time *T1* at which the flight path of *A*, starting at *T*, intersects coverage of *R*.

The difficulty arises from two sources. First, there is mutual recursion between this statement about *causes* and the above definition of *occurs*. This can be quite difficult to control. If occurring events are computed in increasing (decreasing) order of time how do we evaluate reference to the future (past)? Second, reference can also be to states. As time can be real-valued the number of distinct states, e.g. positions of a moving object, can be non-computably infinite. How do we keep a record of all these?

We present a formalism called DMOD and use it to propose solutions to these problems. More generally, it can be used to model and simulate dynamic systems. It is based upon the following fundamental assumption:

If we know the history of a system till time T, i.e. the sequence of events which occurred till T, we can compute the value of every state parameter at any point of time till T.

Let *F* be a condition about the system which takes *m+1* arguments, *m>=0*, and the last argument ranges over *real-valued* time. Let *F* be defined for entities *a1,...,am,t*. Then the proposition *F(a1,...,am,t)* is said to become true if $F(a1,...,am,t)$ is true but there exists a finite time interval immediately preceding t such that for each time instant X within it $F(a1,...,am,X)$ is false. Thus, we distinguish between a proposition being true and becoming true.

Let $F1,F2,...,Fk$ be a special set of conditions about the system called event-defining conditions. Each Fi takes $m+1$ arguments, $m>=0$, and the last argument ranges over real-valued time. Where Fi is defined for entities $a1,...,am,t$, the proposition $Fi(a1,...,am,t)$ is called an event and t is called its time-stamp. If the event $Fi(a1,...,am,t)$ becomes true then it is said to occur at t. Thus, we distinguish between an event and its occurrence.

Now, the set of event-defining conditions must be chosen in such a way that the fundamental assumption of DMOD is satisfied. We assume that rules can be written down which, given the history till time T, compute the state of the system at time T. These are called state-computation rules. Using these, the history till T can be regarded as a representation of all the states till T even if they are non-computably infinite.

To compute which events occur we view causality from a different point of view. We regard it as a ternary relation between two events and a time-ordered sequence of events in which they appear. To avoid confusion with the traditional view we call the new relation *causal_connection*. The sequence is regarded as a context and we are to specify whether two events are causally connected within this context. Causal connectedness is similar to connectedness between nodes in a network. Two nodes may be connected in one network but not in another. *Note that events in the context do not have to occur*. This is the basis for avoiding the mutual recursion above. The *causal_connection* relation is defined using causality rules of the form:

causal_connection(E,HE,F,HEF) if causality_predicted(E,HE,F) & prediction_unfalsified(E,HE,F,HEF).

```
               E                        F
o---o---o---o--o-------o-----o---o----o...o        (S)
<---HE-->        <-----HEF---->
```

In the figure above S is a sequence of events sorted in increasing order of time stamps. E appears before F in S, HE is the sequence of all events in S upto but not including E, and HEF is the sequence of all events in S between E and F but not including either. The above rule is to be read as:

> If from the information available till E it is predicted that there is a causal connection between E and F, and this prediction is not falsified by information collected between E and F then E is causally connected to F.

Note that reference to events *and* states in the past of E is resolved using HE and to those in the future of E upto F using HEF. Reference beyond F is not available as it ought not to influence causal connectedness between E and F. A DMOD program consists of a set of causality rules, and a set of state-computation rules. For example, DMOD rules expressing the above rule are:

```
causal_connection(E,HE,F,HEF) if
        E=flies_towards(aircraft(X),radar(R),CT) &
        F=in_range(aircraft(X),radar(R),FT) &
        causality_predicted(E,HE,F) &
        prediction_unfalsified(E,HE,F,HEF).

causality_predicted(E,HE,F,HEF) if
        E=flies_towards(aircraft(X),radar(R),CT) &
        F=in_range(aircraft(X),radar(R),FT) &
        detection_time(aircraft(X),radar(R),[E/HE],FT),
        not member(told_to_fly_low(aircraft(X),_),HE).

prediction_unfalsified(E,HE,F,HEF) if
        E=flies_towards(aircraft(X),radar(R),CT) &
        F=in_range(aircraft(X),radar(R),FT) &
        not member(changes_course(aircraft(X),_),HEF).
```

detection_time(A,R,H,T) is similar to that above although its third argument is the history upto the causing event. Relevant information about positions and velocities can be retrieved from it.

We now show how to use causality rules to compute which events occur in a system, i.e. its history. Let *S=E0,E1,E2,...* be a sequence of events sorted in increasing order of time-stamps. Then *S* is said to satisfy causal-soundness if for each *j*, $j \neq 0$, there exists *i*, $i<j$ such that *causal_connection(Ei,HEi,Ej,HEiEj)* holds, where *HEi* is the sequence *E0,...,Ei-1* and *HEiEj* is the sequence *Ei+1,...,Ej-1*. Note that the initial event *E0* is exempt from requiring a cause. Intuitively, a sequence is causally-sound if every event in it, except the first one, has a cause in it. Causal-soundness is similar to weak-causality property of de Kleer & Brown.

Let *S=E0,E1,E2,...* be a sequence of events sorted in increasing order of time-stamps. Then *S* is said to satisfy causal-completeness if for each *i*, *j*, if there is an event *G* such that *causal_connection(Ei,HEi,G,HEiEj)* then *G* also appears in *S*, where *HEi* is the sequence *E0,...,Ei-1* and *HEiEj* is the sequence *Ei+1,...,Ej-1*. Intuitively, a sequence is causally-complete if it contains all the events whose occurrence is required by the occurrence of the initial event and the causality rules.

Let *E0* be a special initial event for the system. Assume that *E0* has occurred. A history of the system is defined to be a sequence of events starting at *E0* which is both causally-sound and causally-complete. Intuitively, it contains all of the events whose occurrence is required by the occurrence of the initial event and the causality rules, and only these events. It can be shown that there is exactly one history in which each event possesses a distinct time-stamp. It can represent the history of a non-concurrent system.

To compute a history, let the initial event *E0* occur. Suppose the history *HF=E0,E1,...,Em* till a certain point of time has been computed. We need to compute the next event *Em+1*. Let *Sm={F1,F2,...}* be the set of events where for each *Fi*, there exists an *Ei* such that *causal_connection(Ei,HEi,Fi,HEiFi)* holds, where *HEi* is the sequence *E0,E1,...,Ei-1* and *HEiFi* is the sequence *Ei+1,...,Em*. Then, as the history must be causally-complete, the next event, *Em+1* must be the event in *Sm* with the least time-stamp. If *Em+1* cannot be computed the algorithm halts. As there may be more than one event in *Sm* with least time-stamp the algorithm is non-deterministic. A different history would be computed for each choice of *Em+1* signifying that the system is concurrent.