FOCS 1982

A TEMPORAL LOGIC TO DEAL WITH FAIRNESS IN TRANSITION SYSTEMS

by J.P. Queille and J. Sifakis

Laboratoire IMAG, BP53X, 38041 Grenoble Cedex, France

Abstract

In this paper, we propose a notion of fairness for transition systems and a logic for proving properties under the fairness assumption corresponding to this notion. We consider that the concept of fairness which is useful is "fair reachability" of a given set of states P in a system, i.e. reachability of states of P when considering only the computations such that if, during their execution, reaching states of P is possible infinitely often, then states of P are visited infinitely often. This definition of fairness suggests the introduction of a branching time logic FCL, the temporal operators of which express, for a given set of states P, the modalities "it is possible that P" and "it is inevitable that P" by considering fair reachability of P. The main result is that, given a transition system S and a formula f of FCL expressing some property of S under the assumption of fairness, there exists a formula f' belonging to a branching time logic CL such that : f is valid for S in FCL iff f' is valid for S in CL. This result shows that proving a property under the assumption of fairness is equivalent to proving some other property without this assumption and that the study of FCL can be made via the "unfair" logic CL, easier to study and for which several results already exist.

I. Introduction

Fairness is a property whose study becomes important when non-deterministic models are used to represent systems, i.e. models such that from a state it is possible to execute different transitions (actions) and this choice is not specified. In such models there may exist infinite computation sequences such that during their execution some event becomes possible infinitely often but it has not an infinite number of occurrences. The preceding statement is an informal characterization of the unfair sequences of a system. In fact, the existence of such sequences may lead to unfair situations where, although an event is realizable infinitely often, it never occurs because conflicts are resolved in a non equitable manner.

The given informal definition is sufficient to apprehend intuitively the concept of fairness. However, many difficulties arise when a formal definition is to be given. In this case, the expression "some event becomes possible infinitely often" must be assigned a precise meaning : the class of the events which are of interest has to be characterized ; also, "becomes possible" and "infinitely often" must be defined in a given model.

In this paper, we define the notion of unfair computation sequence with respect to some given set of states. Roughly speaking, an infinite computation sequence is unfair with respect to a given set of states P, if the sequence s of the states visited during its execution is such that s contains,

- an infinite number of occurrence of states from which it is possible to reach states of P,
- only a finite number of occurrences of states of P.

Fairness relative to a set of states is, we believe, the notion which is important in practice since most of the "interesting" system properties express reachability relations of some set of states [16,17]. This notion is sufficiently general for covering the majority of the definitions proposed until now [5,6,8,10,12] as it is shown in [14].

The problem of fairness becomes crucial when a property is to be proved in formal systems based on non-deterministic models. In the descriptions with such models, are introduced computation sequences which are not feasible under the assumption of any "reasonable" scheduling policy. These sequences are difficult to distinguish from the feasible ones because, for infinite sequences, fairness cannot be decided by examining any prefix of them. As a result, proving properties valid under the assumption of fairness seems to be a non-trivial problem.

Two different approaches are possible to the problem of proving a property PR for some system S when considering only fair computation sequences.
1) Either transform S into S' such that S' is a system whose computation sequences are all the fair computation sequences of S, and prove the validity of PR for S'. S' can be obtained from S by adjoining to it a "fair scheduler" [2,11].

2) Or, give a method for associating to PR some other property PR' such thar : PR' is valid for S iff PR is valid for S under the assumption of fairness.

We consider the latter approach by defining a conditional branching time logic for fairness, FCL. In this logic temporal operators express, for a given set of states P, the modalities "it is possible that P" and "it is inevitable that P" by considering only the computation sequences which are fair with respect to P.

The introduction of FCL sets the problem of its comparison with the existing branching time logics [1,3,4,8,13]. We have shown that given a formula f of FCL it is possible to find a formula f' of a conditional time logic CL such that for any model S :

"f is valid for S in FCL" is equivalent to "f' is valid for S in CL".

This result is interesting from several points of view. In particular, it shows that the study of FCL can be made via "unfair" logics, easier to study in our opinion, and for which many results already exist (decision procedure, fixed point characterization of temporal operators [3,4,13]).

II. Fairness with respect to a set of states

Transition systems is the model used throughout this article.

Definitions 1 :
A transition system is a triple $S = (Q,T,\{\xrightarrow{t}\}_{t\in T})$

where,
- Q is a countable set of states
- $T = \{t_1,\ldots,t_m\}$ is a set of transitions
- $\{\xrightarrow{t}\}_{t\in T}$ is a set of binary relations on Q ($\xrightarrow{t} \subseteq Q\times Q$) in bijection with the transitions.

Transition systems are used to model discrete systems. Transitions are action names, the effect of which is described by the corresponding relations ; $q\xrightarrow{t}q'$ means that it is possible to execute t from the state q and the resulting state after its execution is q'. The state q' is said to be a direct successor of q.

In a transition system the relation $\rightarrow = \bigcup_{t\in T}\xrightarrow{t}$ is not necessarily total ; there may exist sink states i.e. states q such that $\nexists q'\in Q\ (q\rightarrow q')$. We denote by SINK the set of the sink states.

* For a sequence $\sigma\in T^*$, $\sigma = t_{i_1}t_{i_2}\ldots t_{i_s}$, the notation $q\xrightarrow{\sigma}q'$ is used to represent the fact $\exists q_0q_1\ldots q_s$, $q = q_0$, $q' = q_s$ and $q_{j-1}\xrightarrow{t_{i_j}}q_j$ for j=1,..s.
* We use T^∞ to represent the set of the infinite sequences on T. A sequence $\sigma\in T^*\cup T^\infty$ is said to be applicable from a state q_0 (this fact is denoted by $q_0\xrightarrow{\sigma}$) if there exists a sequence of states, $q_0q_1\ldots q_i\ldots$ such that $q_0\xrightarrow{\sigma_i}q_i$ where σ_i is the prefix of σ of length i.
* A computation sequence from a state q_0 is a sequence σ of $T^*\cup T^\infty$ such that σ is applicable from q_0 and if σ is finite then $\exists q'(q_0\xrightarrow{\sigma}q'$ and $q'\in SINK)$.
* An execution sequence from a state q_0 is a sequence of states visited when a computation sequence is executed from q_0.

Definitions 2 : (Relative fairness [14])
An infinite computation sequence σ applicable from a state q_0 is,

2a) unfair with respect to a transition t if there is an infinity of sequences σ', prefixes of σ, such that for each σ' there exists $\sigma''\in T^*$, $q_0\xrightarrow{\sigma'\sigma''t}$ and t has a finite number of occurrences in σ.

2b) unfair with respect to a given set of states P if there exists an infinity of sequences σ', prefixes of σ, such that for each σ' there exists $\sigma''\in T^*$, $q_0\xrightarrow{\sigma'\sigma''}q$, $q\in P$, while there is only a finite number of occurrences of states of P in the execution sequence corresponding to σ.

Relative fairness allows to focus on the possibility for a particular class of events to occur. It is, we believe, the useful concept in practice since all the "interesting" system properties express reachability of some set of states. In particular, fair termination [6] can be defined in terms of fairness with respect to HALT (HALT is the set of the termination states) i.e. a program terminates fairly if all its infinite computation sequences are unfair with respect to HALT. Furthermore, the property of fairness derived from definition 2a) by considering sequences which are fair with respect to any transition, is strong fairness [8,10,12]. All the known types of fairness such as the finite delay property [7] (or weak fairness [5,8,10,12] or justice [9]) and impartiality [9] can be considered as particular cases of it.

Notice that the same execution sequence s can correspond to two different computation sequences σ_1 and σ_2 the one being fair and the other unfair with respect to a given transition. On the contrary, all the computation sequences corresponding to the same execution sequence s, are either fair or unfair with respect to any given set of states. By extending 2b), we say that an execution sequence is fair with respect to a set P iff one of the corresponding computation sequences is fair with respect to P.

The result given hereafter shows that the property of fairness with respect to a transition in a transition system S can be considered as a property of fairness with respect to a set of states in some other transition S_T.

Given $S = (Q,T,\{\xrightarrow{t}\}_{t\in T})$, define $S_T = (Q\times T,T,\{\rightsquigarrow\}_{t\in T})$ such that, $(q,t)\overset{t''}{\rightsquigarrow}(q',t')$ iff $q\xrightarrow{t''}q'$ and t'=t".
i.e. S_T is a transition system having the same computation sequences as S with the difference that the only transition leading to state (q,t) is transition t.

For a transition t, we define the set of the states,

$after(t) = \{(q,t) \mid \exists q'\ q'\xrightarrow{t}q\}$,

i.e. after(t) is the set of the states of S_T reached just after (and only after) the trnasition t is executed.

The function after has already been used to express system properties in temporal logic [13,15].

Proposition 1 :
A computation sequence σ is unfair in S with respect to t iff σ is an unfair computation sequence in S_T with respect to after(t).

III. A temporal logic to deal with fairness

In this section a logic for dealing with the properties of the fair functioning of transition systems is given. The expressibility of this logic is compared with the expressibility of existing branching time logics. The main result is that there exists a conditional branching time logic CL [1,4] such that for a given transition system (model) S and for each formula f' of the introduced logic, there exists a formula f of CL for which : f' is valid for S iff f is valid for S in CL ; i.e. f' under the assumption of fairness is equivalent to f. This result allows the application to the introduced logic of well-known results for CL (decision procedure, fixed point characterization of the temporal operators).

III.1 The conditional branching time logic CL

We introduce a conditional branching time logic CL obtained by augmenting the propositional calculus in the following manner.

The formulas of CL are built from a set of propositional variables V and the constants <u>true</u> and <u>false</u> by using the logical connectives $\vee$, $\wedge$, $\neg$, $\Rightarrow$ and two binary temporal operators POT and INEV.
Each one of the arguments of the temporal operators play very different roles ; in order to better distinguist them, we prefer writing $POT[f_1](f_2)$ and $INEV[f_1](f_2)$ instead of $POT(f_1,f_2)$ and $INEV(f_1,f_2)$. Also, the abbreviations $ALL[f_1](f_2)$ and $SOME[f_1](f_2)$ are used for respectively $\neg POT[f_1](\neg f_2)$ and $\neg INEV[f_1](\neg f_2)$.

The class of the models for CL is defined in terms of transition systems as follows :

* Given a transition system $S = (Q,T,\{\xrightarrow{t}\}_{t\in T})$, represent by EX(q) the set of all the execution sequences starting from state q. In order to simplify the notations, for a sequence s of EX(q) we represent by s(k) the k-th element of it, if it is defined ; if not, we take $s(k) = \omega$ where ω represents some fictitious non accessible state adjoined to Q. Thus, the relation $s(0) \xrightarrow{k} s(k)$ is satisfied iff $s(k) \neq \omega$.
* Given CL and a transition system $S = (Q,T,\{\xrightarrow{t}\}_{t\in T})$, an interpretation of CL is a function $|\,|$ associating to each formula of CL a subset of Q such that :
  - $\forall f\in V,\ |f| \subseteq Q$,
  - $|\underline{true}| = Q$,
  - $\forall f\in CL,\ |\neg f| = Q - |f|$,
  - $\forall f_1, f_2 \in CL,\ |f_1 \wedge f_2| = |f_1| \cap |f_2|$,
  - $\forall f_1, f_2 \in CL$,
  $q\in |POT[f_1](f_2)| \equiv \exists s\in EX(q)\ \exists k\in \mathbb{N}\ [q\xrightarrow{k}s(k)\ \underline{and}\ \forall_{i=0}^{k-1} s(i)\in|f_1|\ \underline{and}\ s(k)\in|f_2|]$,
  - $\forall f_1, f_2 \in CL$,
  $q\in |INEV[f_1](f_2)| \equiv \forall s\in EX(q)\ \exists k\in \mathbb{N}\ [q\xrightarrow{k}s(k)\ \underline{and}\ \forall_{i=0}^{k-1} s(i)\in|f_1|\ \underline{and}\ s(k)\in|f_2|]$.

Obviously, $|POT[f_1](f_2)|$ represents the set of the states q of S such that there exists an execution sequence from q containing some state q' which belongs to $|f_2|$, and all the states between q and q' (excluding q') belong to $|f_1|$. We say that $|POT[f_1](f_2)|$ is the set of the states from which (some state of) $|f_2|$ is <u>potentially reachable under</u> the condition $|f_1|$. In the same way, $|INEV[f_1](f_2)|$ is the set of the states from which $|f_2|$ is <u>inevitably reachable under the condition</u> $|f_1|$ in the sense that every execution sequence, starting from a state q of this set, contains some state q' of $|f_2|$ and all the states between q and q' (excluding q') belong to $|f_1|$.

The interpretation of the dual operators ALL and SOME is, $\forall f_1, f_2 \in CL$,

- $q\in|ALL[f_1](f_2)| \equiv \forall s\in EX(q)\ \forall k\in\mathbb{N}\ [q\xrightarrow{k}s(k)\ \underline{and}\ \forall_{i=0}^{k-1} s(i)\in|f_1|\ \underline{implies}\ s(k)\in|f_2|]$,
- $q\in|SOME[f_1](f_2)| \equiv \exists s\in EX(q)\ \forall k\in\mathbb{N}\ [q\xrightarrow{k}s(k)\ \underline{and}\ \forall_{i=0}^{k-1} s(i)\in|f_1|\ \underline{implies}\ s(k)\in|f_2|]$.

$|ALL[f_1](f_2)|$ represents a set of states such that if q belongs to $|f_1|$ then every possible successor of q belongs to $|f_2|$ <u>as long as</u> f_1 remains true. In an analoguous manner, $|SOME[f_1](f_2)|$ represents a set of states such that if q belongs to $|f_1|$ then there exists an execution sequence applicable from q whose states belong to $|f_2|$ <u>as long as</u> f_1 remains true. The formulas $\neg f_1 \vee ALL[f_1](f_2)$ and $\neg f_1 \vee SOME[f_1](f_2)$ express for branching time logic a similar notion as the <u>until</u> operator for linear time logic [8,15] ; both of them express the fact that "f_2 remains true until f_1 becomes false".

The logic CL constitutes a natural generalization of the branching time logic L [13] obtained by adjoining to the propositional calculus the unary temporal operators POT and INEV such that $POT = \lambda f.POT[\underline{true}](f)$, $INEV = \lambda f.INEV[\underline{true}](f)$.
The logic CL has been introduced in [1] and studied in [4] where a decision procedure is given. Moreover, a logic similar to L has been studied in [3].

III.2 Interpretation of the temporal operators as invariants and trajectories

The following results constitute a generalization of well-known results for L [13,17] and show that the interpretation of the temporal operators of CL can be expressed as least or greatest fixed points of predicate transformers and correspond to the well-known notions of invariant and trajectory. Reasoning in terms of these notions makes easier the understanding of the properties of CL. A similar fixed point characterization of the operators of CL is given in [4].

Let $S = (Q,T,\{\xrightarrow{t}\}_{t\in T})$ be a transition system. Represent by $(2^Q,\cup,\cap,\bar{\ })$ the lattice of the subsets of Q and $[2^Q \to 2^Q]$ the set of the internal mappings of 2^Q. For $h,g\in[2^Q\to 2^Q]$, $h\cup g$, $h\cap g$, $\bar{g}$, $\tilde{g}$ and Id denote the functions $h\cup g \equiv \lambda x.h(x)\cup g(x)$, $h\cap g = \lambda x.h(x)\cap g(x)$, $\bar{g} = \lambda x.\overline{g(x)}$, $\tilde{g} = \lambda x.\bar{g}(\bar{x})$, $Id = \lambda x.x$. We also introduce the notations gfpx.g(x) and lfpx.g(x) to denote the greatest and the least fixed point of the monotonic function $g = \lambda x.g(x)$.

<u>Definition</u> :

Given $S = (Q,T,\{\xrightarrow{t}\}_{t\in T})$, $P\subseteq Q$ and $q\in Q$, pre is a function of $[2^Q\to 2^Q]$ such that,

$q\in pre(P) \equiv \exists q'\ (q\to q'$ <u>and</u> $q'\in P)$

<u>Definitions</u> :

Given a transition system $S = (Q,T,\{\xrightarrow{t}\}_{t\in T})$ and $C\subseteq Q$,

a) an <u>invariant</u> of S is a subset J of Q, such that, $\forall q,q'\in Q\quad (q\in J$ <u>and</u> $q\to q'$ <u>implies</u> $q'\in J)$.

b) <u>a conditional invariant of S under the condition C</u> is a subset J of Q such that, $\forall q,q'\in Q\quad (q\in J\cap C$ <u>and</u> $q\to q'$ <u>implies</u> $q'\in J)$.

c) a <u>trajectory</u> of S is a subset W of Q such that, $\forall q\in Q\ (q\in W\text{-}SINK$ <u>implies</u> $\exists q'\ (q\to q'$ <u>and</u> $q'\in W))$.

d) <u>a conditional trajectory of S under the condition C</u> is a subset W of Q such that, $\forall q\in Q\ (q\in W\cap C\text{-}SINK$ <u>implies</u> $\exists q'(q\to q'$ <u>and</u> $q'\in W)$.

<u>Proposition 2</u> :

Given a transition system S and a subset C of Q,

a) J is an invariant iff $J = J\cap \widetilde{pre}(J)$,

b) J is a conditional invariant under C iff $J = J\cap(\bar{C}\cup\widetilde{pre}(J))$.

<u>Proposition 3</u> :

Given a transition system S and a subset C of Q,

a) W is a trajectory iff $W = W\cap(pre(W)\cup\widetilde{pre}(W))$,

b) W is a conditional trajectory under C iff $W = W\cap(\bar{C}\cup pre(W)\cup\widetilde{pre}(W))$.

<u>Proposition 4</u> :

Let f_1, f_2 be two formulas of CL, S a transition system and $|\ |$ an interpretation of CL in S.

a) $|POT[f_1](f_2)| = lfpx.\ |f_2| \cup |f_1| \cap pre(x)$ and dually, $|ALL[f_1](f_2) = gfpx.\ |f_2| \cap (|\neg f_1| \cup \widetilde{pre}(x))$,

b) $|INEV[f_1](f_2)| = lfpx.|f_2|\cup|f_1|\cap(pre(x)\cap\widetilde{pre}(x))$ and dually,

$|SOME[f_1](f_2) = gfpx.|f_2|\cap(|\neg f_1|\cup pre(x)\cup\widetilde{pre}(x))$.

According to proposition 4, $|ALL[f_1](f_2)|$ and $|SOME[f_1](f_2)|$ are respectively the greatest conditional invariant and trajectory under $|f_1|$ contained in $|f_2|$. In particular, $|ALL[\underline{true}](f_2)|$ and $|SOME[\underline{true}](f_2)|$ are respectively the greatest invariant and trajectory contained in $|f_2|$. These results suggest a proof method in CL by iterative evaluation of the formulas, which is used in [13].

III.3 <u>The logic FCL</u>

III.3.1. <u>Preliminary results</u>

Given a transition system S, a state q_0 of S, and a countable set A of subsets of Q, we define $FEX(q_0,A)$ to be the set of the execution sequences starting from q_0 and which are fair with respect to any member a of A i.e. $FEX(q_0,A) = \bigcap_{a\in A} FEX(q_0,\{a\})$

<u>Proposition 5</u> :

For every transition system S, any state q of it, and any countable set A of subsets of Q, the set $FEX(q_0,A)$ is not empty.

<u>Proof</u> :

a) Suppose that A is finite, $A = \{a_1,..,a_s\}$. Then, a computation sequence σ, fair with respect to any member of A, can be iteratively computed as follows:

$q:=q_0$; $\sigma:=\lambda$; (λ is the empty word of T^*)

<u>while</u> true <u>do</u>
  <u>for</u> j:=1 <u>to</u> |A| <u>do</u>
    <u>if</u> $\exists q_j$ and σ_j such that $q\xrightarrow{\sigma_j}q_j$ and $q_j\in a_j$
      <u>then</u> $q:=q_j$; $\sigma:=\sigma\sigma_j$
    <u>fi</u> ;
    <u>if</u> $q\in SINK$ <u>then</u> <u>stop</u> <u>fi</u>
  <u>od</u>
<u>od</u>

The sequence σ defined in this manner is of the form,

$\sigma=\sigma_{11}\sigma_{12}\cdots\sigma_{1|A|}\sigma_{21}\sigma_{22}\cdots\sigma_{2|A|}\cdots\sigma_{k1}\cdots\sigma_{k|A|}\cdots$,

where σ_{wr} represents the sequence concatenated to the variable σ for j=r during the w-th iteration (we take $\sigma_{wr}=\lambda$ if the condition "$\exists q_r$ and σ_r such that $q\xrightarrow{\sigma_r}q_r$ and $q_r\in a_r$" is not satisfied).

The sequence σ is fair with respect to any element of A : suppose that some a_u is reachable infinitely often during the execution of σ but its states are not infinitely often visited. Then there exists some integer k such that $\forall m$, $m\geq k$ <u>implies</u> $\sigma_{mu}=\lambda$; furthermore, the state q_{u-1} reached after the application of the sequence, $\sigma_{11}\ldots\sigma_{1|A|}\cdots\sigma_{k1}\quad\sigma_{ku-1}$ is such that it is not possible to reach from it any state of a_u. This implies that from every possible successor of q_{u-1} it is not possible to reach some state of a_u. Thus σ is fair with respect to a_u (contradiction).

b) Suppose that A has an infinity of elements. In this case a fair computation sequence with respect to any member of A, can be defined in the following manner :

$q:=q_0$; $\sigma:=\lambda$; (is the empty word of T^*)

<u>for</u> i:=1 <u>to</u> infinity <u>do</u>
  <u>for</u> j:=1 <u>to</u> i <u>do</u>
    <u>if</u> $\exists q_j$ and σ_j such that $q\xrightarrow{\sigma_j}q_j$ and $q_j\in a_j$
      <u>then</u> $q:=q_j$; $\sigma:=\sigma\sigma_j$
    <u>fi</u> ;
    <u>if</u> $q\in SINK$ <u>then</u> <u>stop</u> <u>fi</u>
  <u>od</u>
<u>od</u>

The sequence σ defined in this manner is of the form, $\sigma\sigma_{11}\sigma_{21}\sigma_{22}\sigma_{31}\sigma_{32}\sigma_{33}\ldots\sigma_{k_1}\sigma_{k_2}\ldots\sigma_{kk}\ldots$ where σ_{wr} represents the sequence concatenated to σ for i=w and j=r. For the same reasons as in the case where A is a finite, σ is fair with respect to any member of A. □

Corollary :
Given a countable set A of subsets of Q, $q_0 \in Q$ and an execution sequence s, $s \notin FEX(q_0,A)$, it is possible for any integrer k to find a sequence s', $s' \in FEX(q_0,A)$ such that the sequence s(o)...s(k) is a prefix of s'.

III.3.2. Obtaining the logic FCL

Given CL and an interpretation ot it, a formula f represents a property of both the fair and unfair sequences of a transition system. If one is interested in the property expressed by this formula f when restricting to fair functioning, the following approach can be adopted.

Let f be a formula of CL written in such a form that only the temporal operators POT and INEV occur in it and let $\{f_1,..f_s\}$ be the set of the sub-formulas of f which are second arguments of POT and INEV. We put $A(f) = \{|f_i|\}_{i=1}^{s}$.
Given that POT and INEV express reachability of their second argument, we try to obtain a formula f', expressing the same property as f under the assumption of fairness, by restricting to the functioning corresponding to the set of the execution sequences which are fair with respect to any member of A(f). To do this, we define for any countable set A of subsets of Q, and any pair of formulas f_1, f_2 of CL such that $|f_2| \in A$, the operators F_APOT and F_AINEV,

$q \in |F_A POT[f_1](f_2)| \equiv \exists s \in FEX(q,A)\ \exists k \in \mathbb{N}$
$[q \overset{k}{\to} s(k)$ and $\forall_{i=0}^{k-1} s(i) \in |f_1|$ and $s(k) \in |f_2|]$,

$q \in |F_A INEV[f_1](f_2)| \equiv \forall s \in FEX(q,A)\ \exists k \in \mathbb{N}$
$[q \overset{k}{\to} s(k)$ and $\forall_{i=0}^{k-1} s(i) \in |f_1|$ and $s(k) \in |f_2|]$.

i.e. these operators express respectively the fact that $|f_2|$ is potentially or inevitably reachable under $|f_1|$ by using only the sequences which are fair with respect to any member of A.

Thus, the formula f' expressing the same property as f under the assumption of fairness, is obtained by uniform substitution of the operators POT and INEV in f by the operators $F_{A(f)}$POT and $F_{A(f)}$INEV.

Proposition 6 :
For any pair of formulas f_1, f_2 of CL and any countable set A of subsets of Q such that $|f_2| \in A$,

$|F_A POT[f_1](f_2)| = |POT[f_1](f_2)|$.

Proof :
Given that $FEX(q,A) \subseteq EX(q)$, we have
$F_A POT[f_1](f_2) \Rightarrow POT[f_1](f_2)$.

Suppose that $\exists q \in Q\ q \in |POT[f_1](f_2)|$ and $q \notin |F_A POT[f_1](f_2)|$. This implies that there exists a sequence s, $s \notin FEX(q,A)$ and $s \in EX(q)$ such that,
$\exists k \in \mathbb{N}\ [q \overset{k}{\to} s(k)$ and $\forall_{i=0}^{k-1} s(i) \in |f_1|$ and $s(k) \in |f_2|]$.
Then it is possible to find, according to the corollary of proposition 5, a sequence s' = s(0)...s(k)s" such that s' is fair with respect to each one of the members of A. This implies $q \in |F_A POT[f_1](f_2)|$ (contradiction). □

Proposition 7 :
For any pair of formulas f_1 and f_2 of CL and any countable set A of subsets of Q such that $|f_2| \in A$,

$|F_A INEV[f_1](f_2)| = |F_{\{|f_2|\}} INEV\ [f_1](f_2)|$

Proof :
$FEX(q,A) \subseteq FEX(q,\{|f_2|\})$ because $|f_2| \in A$; consequently we have,
$F_{\{|f_2|\}} INEV[f_1](f_2) \Rightarrow F_A INEV[f_1](f_2)$.

Suppose that for some state q, $q \in |F_A INEV[f_1](f_2)|$ and $q \notin |F_{\{|f_2|\}} INEV[f_1](f_2)|$.

This implies that there exists a sequence s, $s \in FEX(q,\{|f_2|\})$ such that, $\forall k \in \mathbb{N}$,
$q \overset{k}{\to} s(k)$ implies $(\exists_{i=0}^{k-1} s(i) \in |\neg f_1|$ or $s(k) \in |\neg f_2|)$,
and there exists a member of $A-\{|f_2|\}$ with respect to which s is not fair. Thus s is an infinite sequence.

a) Suppose that for $\forall i \in \mathbb{N}\ s(i) \in |f_1|$. This implies $\forall i \in \mathbb{N}\ s(i) \in |\neg f_2|$.

But since s is fair with respect to $|f_2|$ there exists some $j_0 \in \mathbb{N}$ such that $\forall j$, $j \geq j_0$ implies $s(j) \in |\neg POT(f_2)|$.
Then, according to the corollary of proposition 5, it is possible to find a sequence s', fair with respect to $A-\{|f_2|\}$ starting from $s(j_0)$. The sequence $s(o)..s(j_0)s'$ is fair with respect to any element of A and all its elements satisfy $\neg f_2$. Contradiction, since we supposed that $q \in |F_A INEV[f_1](f_2)|$.

b) Suppose that $\exists i \in \mathbb{N}\ s(i) \in |\neg f_1|$ and put $i_0 = Min\{i | s(i) \in |\neg f_1|\}$.
Then, for $0 \leq i < i_0$ $s(i) \in |f_1|$ and
for $0 \leq i \leq i_0$ $s(i) \in |\neg f_2|$.

Consider a sequence $s(o)...s(i_0)s'$, where s' is such that it is fair with respect to any member of A. The fact that this sequence belongs to FEX(q,A), $s(i_0) \in |\neg f_1|$ and $s(i) \in |\neg f_2|$ for $0 \leq i \leq i_0$ contradicts $q \in |F_A INEV[f_1](f_2)|$. □

The results of proposition 6 and 7 considerably simplify the method for obtaining a formula f' expressing the same property as a formula f of CL under the assumption of fairness (f is supposed to be written in such a form that the operator SOME does not occur in it). For this, one has to substitute every occurrence of $\mathrm{INEV}[f_1](f_2)$ by

$F_{\{|f_2|\}}\mathrm{INEV}[f_1](f_2)$.

The simpler notation $\mathrm{FINEV}[f_1](f_2)$ will be adopted in the sequel for $F_{\{|f_2|\}}\mathrm{INEV}\ [f_1](f_2)$. We call FCL the logic obtained by adjoining to the propositional calculus the binary temporal operators POT and FINEV, their duals being respectively denoted by ALL and FSOME. Also, we call FL the sub-logic of FCL constructed from the unary temporal operators $\mathrm{POT}=\lambda f.\mathrm{POT}[\underline{true}](f)$ and $\mathrm{FINEV}=\lambda f.\mathrm{FINEV}[\underline{true}](f)$.

Remark :
Due to the result of proposition 7 and by dualization one obtains,

$$q\in|\mathrm{FSOME}[f_1](f_2)| \equiv \exists s\in \mathrm{FEX}(q,\{|\neg f_2|\})\ \forall k\in\mathbb{N}$$
$$[q\xrightarrow{k}s(k)\ \underline{\text{and}}\ \forall_{i=0}^{k-1}\ s(i)\in|f_1|\ \underline{\text{implies}}\ s(k)\in|f_2|].$$

Notice that for the definition of the interpretation of $\mathrm{FSOME}[f_1](f_2)$ only the fair execution sequences with respect to $|\neg f_2|$ are considered. This is not surprising because $\mathrm{FSOME}[f_1](f_2)$ express the possibility of "remaining fairly in $|f_2|$" i.e. without using unfair sequences with respect to $|\neg f_2|$ in order to remain in $|f_2|$.

III.4 Comparison of the logics CL and FCL

In this section FCL and CL are compared with respect to their capabilities to express properties of transition systems. This comparison is made in a progressive manner by considering successively FL and L, FL and CL, FCL and CL. To do this, we consider the logics CL(V) and FCL(V) defined on the same set of propositional variables V and such that the restrictions of their interpretation functions agree on V.

The problem studied is the search for a meaning preserving homomorphism μ of FCL into CL i.e. a function μ such that if $\mu(f_1)=f_2$ then the statements "f_1 is valid in FCL for S" and "f_2 is valid in CL for S" are equivalent (for any S). This means that the interpretations of f_1 and $\mu(f_2)$ are the same for any f_1 of FCL ($|f_1|=|\mu(f_1)|$). Obviously, if such a function μ exists, then proving the validity of a formula f_1 in FCL amounts to proving the validity of $\mu(f_1)$ in CL.

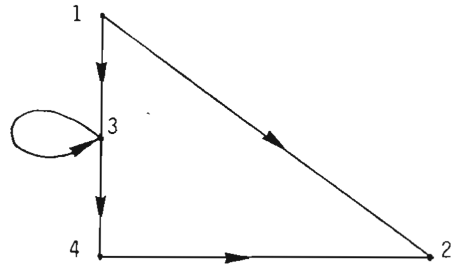
In order to simplify the notations, we omit the interpretation function and we use subsets of Q instead of formulas, whenever there is no risk of confusion. For example, we write $\mathrm{POT}[f_1](f_2)$ to represent the set of the states $|\mathrm{POT}[f_1](f_2)|$.

III.4.1 Comparison of L(V) and FL(V)

Consider L(V) and FL(V), the sub-logics obtained from CL(V) and FCL(V) by taking the conditions of the temporal operators to be equal to $\underline{true}$ (POT, INEV, FINEV stand respectively for $\mathrm{POT}[\underline{true}]$, $\mathrm{INEV}[\underline{true}]$, $\mathrm{FINEV}[\underline{true}]$).

Proposition 8 :
The operator FINEV of FL(V) cannot be expressed in L(V).

Proof :
Consider the following transition system.



If FINEV can be expressed by means of some formula of L(V), then the set FINEV{4}={3,4} can be obtained from {4} by carrying out a finite number of times the operations $\wedge$, $\neg$, POT and INEV. But this is not possible as shown hereafter.

1) From {4} by carrying out logical operations, one obtains the sets,
   $\emptyset$, {4}, {1,2,3}, {1,2,3,4}.
2) The application of POT and INEV to these sets respectively gives,
   $\emptyset$, {1,3,4}, {1,2,3,4}, {1,2,3,4}
   $\emptyset$, {4} , {1,2,3,4}, {1,2,3,4}
3) The combination of the obtained sets via logical operators gives,
   {2}, {4}, {1,3}, {2,4}, {1,2,3}, {1,3,4}, {1,2,3,4}.

No new set can be obtained by application of POT and INEV. □

The following proposition gives upper and lower approximations of FINEV(f) by formulas of L(V).

Proposition 9 :
For any set of states f, $\mathrm{INEV}(f)\vee\mathrm{ALLPOT}(f)\Rightarrow\mathrm{FINEV}(f)$ and $\mathrm{FINEV}(f)\Rightarrow\mathrm{INEV}(f)\vee\mathrm{SOMEPOT}(f)$.

Proof :
a) $\mathrm{INEV}(f)\vee\mathrm{ALLPOT}(f)\Rightarrow\mathrm{FINEV}(f)$.

Obviously, $\mathrm{INEV}(f)\Rightarrow\mathrm{FINEV}(f)$.
Suppose that $q\in\mathrm{ALLPOT}(f)$ and $q\notin\mathrm{FINEV}(f)$.
$q\notin\mathrm{FINEV}(f)$ is equivalent to $q\in\mathrm{FSOME}(\neg f)$ which means that there exists an execution sequence s, starting from q, whose states belong to $\neg f$ and which is fair with respect to f. Remark that, due to the fact that $q\in\mathrm{ALLPOT}(f)$, s is infinite, since there is no sink state in $\mathrm{POT}(f)\wedge\neg f$. Thus, s is not fair with respect to f as all its states belong to POT(f) (contradiction).

b) $\mathrm{FINEV}(f)\Rightarrow\mathrm{INEV}(f)\vee\mathrm{SOMEPOT}(f)$.

Suppose that for some state q, $q\in\mathrm{FINEV}(f)$ and $q\notin\mathrm{INEV}(f)$. This is equivalent to $q\notin\mathrm{FSOME}(\neg f)$ and $q\in\mathrm{SOME}(\neg f)$. Thus, there is no fair execution sequence with respect to f, starting from q and contained in $\neg f$ but there is an execution sequence s starting from q and contained in $\neg f$. Thus, s is unfair with respect to f. So it is possible to reach f from every state of s and consequently all its states belong to POT(f). SOMEPOT(f) being the greatest trajectory in POT(f) all the states of s belong to it. In particular, q∈SOMEPOT(f). □

Remark :
The implications are strict due to proposition 8. For the transition system given in its proof we have for f = {4}
INEV(f) ={4}, ALLPOT(f)=∅, FINEV{f}={3,4}, SOMEPOT(f)={1,3}.

Corollary :
For any set of states f,

a) If SOMEPOT(f) => INEV(f) then FINEV(f)≡INEV(f)

b) If FINEV(f)≡INEV(f) then ALLPOT(f) => FINEV(f).

Proposition 10 :
For every transition system S and any set of states f of S, ¬SOME(¬f∧POT(f)) is a valid formula iff every sequence is fair with respect to f.

Proof :
It is sufficient to show that,
∃q∈SOME(¬f∧POT(f)) iff there exists some sequence which is unfair with respect to f.

a) If there exists q∈SOME(¬f∧POT(f)) then the trajectory SOME(¬f∧POT(f)) is non-terminating because there is no sink state in ¬f∧POT(f). Thus, there exists an infinite execution sequence whose states do not belong to f but f is reachable from every state of this sequence. So, it is unfair with respect to f.

b) If there exists an unfair execution sequence s with respect to f, it is possible to find some suffix of it s' such that the corresponding trajectory is in ¬f. The sequence s' is also unfair with respect to f and all its states belong to POT(f). Thus, the geatest trajectory contained in ¬f∧POT(f) is a non-empty set. □

III.4.2. Comparison of CL(V) and FCL(V)

Lemma 1 :
For any set of states f, INEV(f) => ALL[¬f](POT(f)).

Proposition 11 :
For any set of states f, FINEV(f)≡ALL[¬f](POT(f)).

Proof :
a) ALL[¬f](POT(f)) => FINEV(f)).
Suppose that q∈ALL[¬f]POT(f) and q∉FINEV(f). q∉FINEV(f) is equivalent to q∈FSOME(¬f), which means that there exists an execution sequence s, fair with respect f, starting from q, the states of which are in ¬f. Remark that s is infinite because as long as f is not reached the system is at some state of ¬f∧POT(f).

For s to be fair with respect to f, there must be a suffix s' of it, the states of which do not belong to POT(f). This means that there exists an execution sequence in ¬f starting from q and leading to some state of ¬POT(f). But this fact contradicts q∈ALL[¬f]POT(f), equivalent to q∉POT[¬f]¬POT(f), which means that ¬POT(f) is not reachable from q under the condition ¬f.

b) FINEV(f) => ALL[¬f]POT(f).

Suppose that for some q, q∈FINEV(f), and q∉ALL[¬f]POT(f). This implies q∉FSOME(¬f) and q∈SOME(¬f) (by lemma 1) which means that there are only unfair sequences with respect to f, starting from q and having all their states in ¬f.
This contradicts q∉ALL[¬f]POT(f) which is equivalent to q∈POT[¬f]ALL(¬f) i.e. there exists a finite sequence s, starting from q, leading to some state of ALL[¬f] and whose all states are in ¬f. Every sequence having s as prefix is fair with respect to f. □

This result shows that uniform substitution of FINEV(f) and FSOME(f) in a formula of FL(V) by respectively ALL[¬f]POT(f) and POT[f]ALL(f) gives an equivalent formula of CL(V). As a consequence we have that FL(V) is equivalent to the sub-logic of CL(V) generated by adjoining to the propositional calculus on V the operators λf.POT(f) and λf.ALL[¬f]POT(f).

III.4.3. Comparison of CL(V) and FCL(V)

Lemma 2 :
For any pair of sets of states f_1 and f_2,
$INEV[f_1](f_2)$ => $ALL[¬f_2]POT[f_1](f_2)$.

Proposition 12 :
For any pair of sets of states f_1 and f_2,
$FINEV[f_1](f_2)$≡$ALL[¬f_2]POT[f_1](f_2)$.

Proof :
a) $ALL[¬f_2]POT[f_1](f_2)$ => $FINEV[f_1](f_2)$.

Suppose that for some state q, q∈$ALL[¬f_2]POT[f_1](f_2)$ and q∉$FINEV[f_1](f_2)$. q∉$FINEV[f_1](f_2)$ is equivalent to q∈$FSOME[f_1](¬f_2)$ which means that there exists an execution sequence s fair with respect to f_2, starting from q, the states of which are in $¬f_2$. Remark that s is infinite because $¬f_2∧POT[f_1](f_2)$ does not contain sink states. For s to be fair with respect to f_2, it must have a suffix whose all the states belong to $¬POT(f_2)$. Thus,

q∈$POT[¬f_2]$ $¬POT(f_2)$ which implies,

q∈$POT[¬f_2]$ $¬POT[f_1](f_2)$

and this contradicts q∈$ALL[¬f_2]POT[f_1](f_2)$.

b) $FINEV[f_1](f_2)$ => $ALL[¬f_2]POT[f_1](f_2)$.

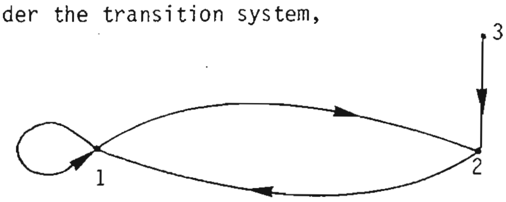
Suppose that, q∈$FINEV[f_1](f_2)$ and q∉$ALL[¬f_2]POT[f_1](f_2)$ for some state q. This implies,
q∉$FSOME[f_1](¬f_2)$ and q∈$SOME[f_1](¬f_2)$ (by lemma 2) which means that all the conditional trajectories under f_1 which are contained in $¬f_2$ and start from q, are unfair with respect to f_2. These trajectories being unfair with respect to f_2 (when only execution paths in f_1 are considered) all their

states belong to $POT[f_1](f_2)$.

But this contradicts $q \notin ALL[\neg f_2]POT[f_1](f_2)$ which means that there exists a possible successor q' of q such that $q' \in \neg POT[f_1](f_2)$. □

This result shows that FCL is the conditional time logic obtained by adjoining to the propositional calculus the operator POT. This logic is less expressive than CL as it is shown by the following counter example :

Consider the transition system,



If INEV(f) is expressible by a formula of FCL(V), then it is possible to compute INEV{2} = {2,3} by applying to {2} logical and/or temporal operators of FCL.

It is easy to verify that this is not possible :

1) From 2 by application of logical operators we obtain,

   $\emptyset$, {2} , {1,3} , {1,2,3}

2) Application of the conditional temporal operator POT gives (x successively takes the values, $\emptyset$, {2}, {1,3}, {1,2,3}).

POT[{1,2,3}](x) : $\emptyset$, {1,2,3}, {1,2,3}, {1,2,3}

POT[{2}](x) : $\emptyset$, {2}, {1,2,3}, {1,2,3}

POT[{1,3}](x) : $\emptyset$, {1,2,3}, {1,3}, {1,2,3}

Since there is no new set generated we conclude that for this transition system it is not possible to express INEV{2} in FCL.

IV. Conclusion

We have proposed a notion of fairness for transition systems and a logic for proving properties under the fairness assumption induced from this notion.

In part II we have proposed a definition of fairness which is sufficiently general for covering the majority of the definitions given until now [14]. However, a precise comparison is not always possible either because fairness is sometimes introduced by informal discussion or because it is defined on models of higher level than transition systems (for example, models where the notions of process and parallel composition are primitive [12]).

The approach proposed for proving a property under our fairness assumption has the advantage of avoiding the complexification of the system under study by adjoining a fair scheduler to it. In fact, a fairness assumption characterizes the set of all the possible (fair) scheduling policies. In our approach this assumption is incorporated with the formula to prove, without modifying the model. This result is especially interesting as the logic is relatively simple and sufficiently well-studied.

The reader may be astonished that it is possible to prove a property under our assumption of fairness by proving another property without this assumption, i.e. that proving under the fairness assumption is not more difficult than proving without it. In fact, this is not surprising since for the temporal logics considered, the interpretations of the operators are sets of states which can be computed without computing the set of the execution sequences in which these states are involved.

Acknowledgements : We wish to thank Krzysztof Apt, Pedro Guerreiro and Jacques Voiron for their helpful comments on the results of this paper.

References

1. K. ABRAHAMSON "Modal logic of concurrent non deterministic programs" Semantics of concurrent computation, Lecture Notes in Computer Science, Vol. 70, Springer Verlag, 1979, pp. 21-33.

2. K.R. APT, A. PNUELI and J. STAVI "Fair termination revisited - with delay" unpublished abstract, October 1981.

3. M. BEN-ARI, Z. MANNA and a. PNUELI "The temporal logic of branching time" 8th Annual ACM Symp. on principles of Programming Languages, January 1981, pp. 164-176.

4. E.M. CLARKE and E.A. EMERSON "Design and synthesis of synchronization skeletons using branching time temporal logic" TR-12-81, Aiken Computation Laboratory, Harvard University, 1981.

5. E.A. EMERSON and E.M. CLARKE "Characterizing correctness properties of parallel programs using fixpoints" Proc. ICALP80, Lecture Notes in Comp. Science Vol. 85, Springer Verlag 1980, pp. 169-181.

6. O. GRUMBERG, N. FRANCEZ, J.A. MAKOWSKY and W.P. de ROEVER "A proof rule for fair termination of guarded commands" Technical Report RUU-CS-81-2 Univ. of Utrecht, Dpt. of Computer Science, January 1981.

7. R.M. KARP and R.E. MILLER "Parallel Program Schemata" Journal of Computer and System Sciences" Vol. 3, May 1969, pp. 147-195.

8. L. LAMPORT ""Sometime" is sometimes "not never" - on the temporal logic of programs" Proc. 7th Annual ACM Symp. on Principles of programming Languages, Las Vegas, January 1980, pp. 174-185.

9. D. LEHMANN, A. PNUELI and J. STAVI "Impartiality, justice and fairness : the ethics of concurrent termination" ICALP 1981, LNCS Vol. 115, pp. 264-277.

10. D. PARK "On the semantics of fair parallelism" Abstract Software Specifications, Lecture Notes in Computer Science Vol. N° 86, Springer Verlag, 1980, pp. 504-524.

11. D. PARK "A predicate transformer for weak fair iteration" Proc. of the Sixth IBM Symp. Math. Foundations of Comp. Science, May 1981.

12. G.D. PLOTKIN "A powerdomain for countable non-terminism" unpublished draft, Dpt. of Computer Science, Univ. of Edinburgh, Nov. 1981.

13. J.P. QUEILLE and J. SIFAKIS "Specification and verification of concurrent systems in CESAR" International Symp. on Programming, Lecture Notes on Computer Science, Vol. 137, pp. 337-351, Springer Verlag, April 1982.

14. J.P. QUEILLE and J. SIFAKIS "Fairness and related properties in transition systems - A time logic to deal with fairness" Research report N° 292, IMAG, Grenoble, March 1982, revised June 1982.

15. R.L. SCHWARTZ and P. MELLIAR-SMITH "Temporal logic specification of distributed systems" Proc. 2nd Int. Conf. on Distributed Computing Systems, April 1981, pp. 446-454.

16. J. SIFAKIS "Deadlocks and livelocks in transition systems" Mathematical foundations of Computer Science, Lecture Notes in Computer Science Vol. 88, Springer Verlag, 1980, pp. 587-600.

17. J. SIFAKIS "A unified approach for studying the properties of transition systems" Theoretical Computer Science, 18 (1982), pp. 227-258.