SUSANNE GRAF

On Lamport's comparison between linear and branching time temporal logic

RAIRO. Informatique théorique, tome 18, nº 4 (1984), p. 345-353
<http://www.numdam.org/item?id=ITA_1984__18_4_345_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(Vol. 18, n° 4, 1984, p. 345 à 353)

ON LAMPORT S COMPARISON BETWEEN LINEAR AND BRANCHING TIME TEMPORAL LOGIC (*)

by Susanne GRAF [1]

Communicated by K. APT

Abstract. — *We consider the problem of the comparison between a temporal logic of linear time TL_L and a temporal logic of branching time TL_B, already studied by Lamport, for a more restricted class of models. To this end, we define a common class of models for the two logics which are the transition systems. By adopting as criterion of comparison Lamport's strong equivalence, we obtain the incomparableness of the two logics considered, that means, in each one of the two logics there exists a formula which expresses a property, non-expressible in the other. From our proof of incomparableness we obtained a stronger result, stating that there exists no linear time logic more expressive than TL_B.*

Keywords: à venir

Résumé. — *Nous reprenons la comparaison effectuée par Lamport entre une logique temporelle du temps linéaire TL_L et une logique temporelle du temps arborescent TL_B, sur une classe de modèles plus restreinte. Pour cela, on définit une classe commune de modèles pour les deux logiques, qui sont les systèmes de transitions. En adoptant comme critère de comparaison l'équivalence forte de Lamport, on obtient l'incomparabilité des deux logiques considérées, c'est-à-dire dans chacune des deux logiques il existe des formules exprimant des propriétés non exprimables dans l'autre. A partir de notre preuve d'incomparabilité on a obtenu un résultat plus fort qui assure la non existence de logiques linéaires plus expressives que TL_B.*

Mots clés : à venir

1. INTRODUCTION

Temporal logic is an appropriate formalism to reason about concurrent programs [6]. There are two principal views of the underlying model of time [5]. One considers that time is *linear*, i. e. at each instant there is only one possible future. The other is that time is *branching*, i. e. at any instant, time may split into different possible futures. So, linear time logic describes events on a single time path and model is a set of paths. Branching time logic allows to reason about different possible futures and a model is a tree-like construction.

(*) Received in March 1983, revised in April 1983.
([1]) I.M.A.G., B.P. n° 68, 38402 Saint-Martin-d'Hères.

In [5] a comparison is effected between a linear time and a branching time logic, and their adequacy for the expression of the properties of concurrent systems is discussed. In this report the problem of comparison considered in [5] is studied, with the difference that the underlying class of models is restricted and we obtain a more general result. The same problem has been tackled in [4] where the same class of models has been considered but a comparison criterion different from the one adopted in this paper has been taken.

We proceed in the following way. In section 2 we define a common class of models for linear and branching time logics which are transition systems. In section 3 the syntax and the semantics of the two logics given in [5] are introduced. The comparison criterion between logics adopted and some results on it are presented in section 4. Finally, in section 5, we prove the incomparableness of the two logics considered and so confirm the result of [5] for a smaller class of models.

2. THE CLASS OF MODELS C

We define a common class of models C for the two logics to compare.

M is called a *model* over a set of propositional variables P, iff:

$M=(W, R, A)$, where

$W=\{w\}$, a countable set of *states*.

$R \subseteq W \times W$, a total binary relation on W, which represents *direct accessability* between states.

$A \in W \rightarrow 2^P$, a function from W into 2^P. A associates with each state w the subset of propositional variables that hold in w.

So a model can be considered as a transition system (W, R).

For a model M the set of *execution paths* EX_M is defined as,

$$EX_M := \{ s=s_0 s_1 \ldots \mid s_i \in W \wedge (s_i, s_{i+1}) \in R, \forall i \geqq 0 \}.$$

So is EX_M the set of the maximal paths produced by R. Furthermore we adopt the following conventions :

If $s=s_0 s_1 \ldots \in EX_M$ then:

$$s^+ := s_1 s_2 \ldots$$

$$s^{+n} := s_n s_{n+1} \ldots$$

$$\text{first}(s) := s_0.$$

We have the following properties of EX_M.

PROPOSITION 1: *The set of execution paths* EX_M *of a model* M *in* C *has the following properties:*

(1) *If* $s \in EX_M$ *then* $s^+ \in EX_M$ *(suffix closure).*

(2) *If* $t, t' \in W$, $w \in W$ *and* $s, s' \in EX_M$ *then* $tws \in EX_M$ *and* $t'ws' \in EX_M$ *implies* $tws' \in EX_M$ *(fusion closure).*

(3) *If* $\forall i \in \mathbb{N}$, $x_i \in W$ *and* $s_i \in EX_M$, *then* $x_0 s_0 \in EX_M$ *and* $\forall i > 0$, $x_0 \ldots x_i s_i \in EX_M$ *implies* $x_0 x_1 \ldots \in EX_M$ *(Koenig's closure).*

For a proof see in [2].

The only property required in [5] for the set of execution paths is (1). It is easy to see that the three properties above are independent. So, our class of models, which are transition systems, is in fact smaller. This restriction to transition systems seems to be quite reasonable as this model is at the basis of any realistic discrete system [7].

3. SYNTAX AND SEMANTICS OF THE TEMPORAL LOGICS TL_B AND TL_L

A. The temporal logic of branching time TL_B

The *set of wellformed formulas* F_B of TL_B is defined in the usual way over the set of propositional variables P with the logical operators and the unary modal operators ALL and $SOME$. The dual operators of ALL and $SOME$ are denoted by POT and $INEV$ respectively.

Interpretation of the formulas

We represent by $EX_M(w) := \{ s \in EX_M \mid \text{first}(s) = w \}$ the set of the execution paths starting from w and write $M, w \models_B f$ to express the fact that the *formula* f *is true in a state* w of a model M.

We define $\models_B$ inductively:

If $f, f' \in F_B$ then:

1. $M, w \models_B f$ if $f \in P$ and $f \in A(w)$.
2. $M, w \models_B \neg f$ iff $M, w \not\models_B f$.
3. $M, w \models_B f \vee f'$ iff $M, w \models_B f$ or $M, w \models_B f'$.
4. $M, w \models_B ALL\, f$ iff $\forall s \in EX_M(w)$, $\forall n \geqq 0\, M$, $\text{first}(s^{+n}) \models_B f$.
5. $M, w \models_B SOME\, f$ iff $\exists s \in EX_M(w)$, $\forall n \geqq 0\, M$, $\text{first}(s^{+n}) \models_B f$.

By dualisation of 4. and 5. we obtain:

6. $M, w \underset{B}{\models} POT\, f$ iff $\exists\, s \in EX_M(w)$, $\exists\, n \geqq 0\; M$, $\text{first}(s^{+n}) \underset{B}{\models} f$.

7. $M, w \underset{B}{\models} INEV\, f$ iff $\forall\, s \in EX_M(w)$, $\exists\, n \geqq 0\; M$, $\text{first}(s^{+n}) \underset{B}{\models} f$.

We say that a formula f is *true in M* ($M \underset{B}{\models} f$) iff it is true in all states of W, and f is *valid* ($\underset{B}{\models} f$) iff it is true in all models of C.

B. The temporal logic of linear time TL_L

The *set of wellformed formulas* F_L of TL_L is defined as F_B over the set of propositional variables P with the logical operators and the unary modal operator $\Box$. The dual operator of $\Box$ is denoted by $\Diamond$.

Interpretation of the formulas

We write $M, s \underset{L}{\models} f$ to express the fact that the *formula f is true on the execution path* $s \in EX_M$ of a model M and we define $\underset{L}{\models}$ inductively.

If $f, f' \in F_L$ then:

1. $M, s \underset{L}{\models} f$ if $f \in P$ and $f \in A$ (first (s)).

2. $M, s \underset{L}{\models} \neg f$ iff $M, s \underset{L}{\not\models} f$.

3. $M, s \underset{L}{\models} f \vee f'$ iff $M, s \underset{L}{\models} f$ or $M, s \underset{L}{\models} f'$.

4. $M, s \underset{L}{\models} \Box f$ iff $\forall\, n \geqq 0$, $M, s^{+n} \underset{L}{\models} f$.

By dualisation of 4. we obtain:

5. $M, s \underset{L}{\models} \Diamond f$ iff $\exists\, n \geqq 0$, $M, s^{+n} \underset{L}{\models} f$.

We say that a formula f is *true in M* ($M \underset{L}{\models} f$) iff it is true on all execution paths of EX_M, and f is *valid* ($\underset{L}{\models} f$) iff it is true in all models of C.

4. A CRITERION OF COMPARISON OF LOGICS

Comparing the "expressive power" of two logics $L1$ and $L2$, the set of *w.f.f.* of which are respectively denoted by F_{L1} and F_{L2}, consists in verifying that for any formula $f \in F_{L1}$ there exists a formula $g \in F_{L2}$ that has the "same meaning" as f. To compare the meanings of the formulas of $L1$ and $L2$ it is necessary that these logics have the same class of models (up to isomorphism). This allows to compare two formulas by comparing the sets of the models in which they are true.

DEFINITION 1: Let $L1$, $L2$ be logics on a class of models C, $f \in F_{L1}$ and $g \in F_{L2}$. Then we say that f and g are equivalent ($f \equiv g$) iff:

$$\forall M \in C, \qquad M \underset{L1}{\models} f \Leftrightarrow M \underset{L2}{\models} g.$$

DEFINITION 2 (strong equivalence [5]): Let $L1$, $L2$ be logics on a class of models C. Then we say that $L1$ is *less expressive than* $L2$ ($L1 \leqq L2$) iff:

$$\forall f \in F_{L1}, \qquad \exists g \in F_{L2}, f \equiv g.$$

We think that strong equivalence is the appropriate comparison criterion because in practice formulas are used to express specifications of programs, i. e. global properties of models. So we do not want to define a finer equivalence relation by comparing the formulas with respect to their capabilities to express properties of states (in branching time logic) or paths (in linear time logic) as it is done in [4], even if there are satisfiable formulas (for instance the formula $\neg p \wedge POT(p)$) strongly equivalent to false. However such such a formula cannot be used to describe global properties.

PROPOSITION 2: *If $L1$, $L2$ are logics on a class of models C, then:*

$$\exists f \in F_{L1}, \qquad \exists M1, M2 \in C, (M1 \underset{L1}{\not\models} f \wedge M2 \underset{L1}{\not\models} f)$$
$$\wedge \forall g \in F_{L2}. (M2 \underset{L2}{\models} g \Rightarrow M1 \underset{L2}{\models} g)$$

implies $L1 \nleqq L2$.

Proof: obvious.

5. COMPARISON OF TL_B AND TL_L

THEOREM 1: $TL_B \nleqq TL_L$.

Proof: We consider the formula $f = POTp \in F_B$, where $p \in P$, and two models $M1$, $M2$ such that:

$$\text{(I)} \qquad M1 \underset{B}{\not\models} POTp \wedge M2 \underset{B}{\models} POTp$$

and

(II) $$\forall g \in F_L, M2 \underset{L}{\models} g \Rightarrow M1 \underset{L}{\models} g.$$

Thus by proposition 2 we obtain the proof [5].

It remains to find $M1$, $M2$ satisfying (I) and (II).

We define the models $M1$, $M2$ over $P = \{p\}$, as:

$-M1 = (W_1, R_1, A_1)$

$= (\{w_0, w_1\}, \{(w_0, w_0), (w_1, w_1)\}, A_1 : A_1(w_0) = \emptyset$ and $A_1(w_1) = \{p\})$

represented by:

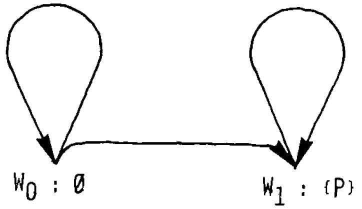

Figure 1

We have for $M1$, $EX_{M1} = \{w_0^\infty, w_1^\infty\}$.

$$-M2 = (W_2, R_2, A_2) = (W_1, R_1 \cup \{(w_0, w_1)\}, A_1)$$

represented by:

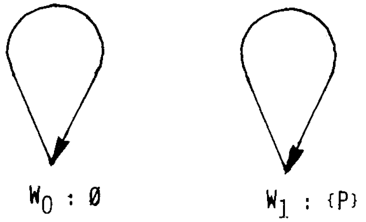

Figure 2

We have for $M2$, $EX_{M2} = EX_{M1} \cup \{w_0^n w_1^\infty \mid n \geqq 0\}$.

1. It is easy to see that (I) holds for the $M1$ and $M2$ given.
2. From $A_1 = A_2$ and $EX_{M1} \subseteq EX_{M2}$ we obtain,

$$\forall g \in F_L, \qquad \forall s \in EX_{M1}, M2, s \underset{L}{\models} g \Rightarrow M1, s \underset{L}{\models} g.$$

This proves (II). □

This proof is similar to that given in [5], adapted to our class of models. Furthermore, notice that the proof of theorem 1 does not depend on the modal operators of the linear time logic considered. That means, that even if other temporal operators are added to TL_L in order to increase its expressive power, the resulting linear time logic will not be more expressive than the branching time logic considered. So we obtain the proposition:

PROPOSITION 3: $LT_B \not\leq LT'_L$ *for any linear logic* LT'_L, *extension of* LT_L.

THEOREM 2: $TL_L \not\leq TL_B$.

Proof: We consider the formula $f = \Box a \vee \Box b \vee \Diamond c \in F_L$, where a, b, $c \in P$, and two models $M1$, $M2$ such that:

(I) $$M_1 \models_L f \wedge M_2 \not\models_L f$$

and

(II) $$\forall g \in F_B, (M_1 \models_B g \Leftrightarrow M_2 \models_B g).$$

Thus by proposition 2 we obtain the proof.

It remains to find $M1$, $M2$ satisfying (I) and (II).

We define the models $M1$, $M2$ over $P = \{a, b, c\}$ as:

$M1 = (W_1, R_1, A_1)$, $M2 = (W_2, R_2, A_2)$.

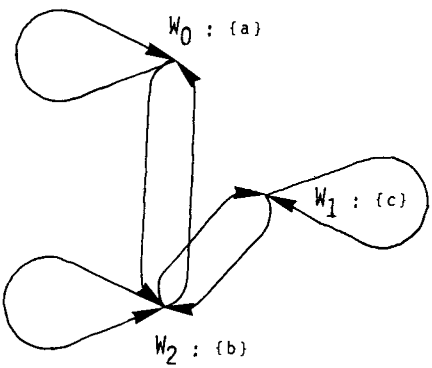

Figure 3

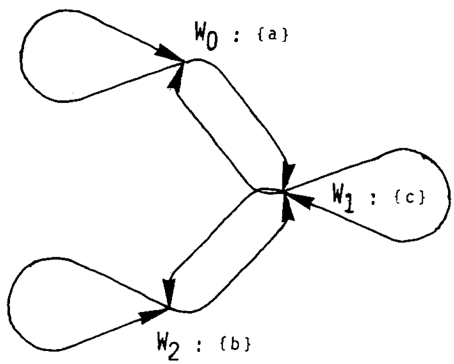

Figure 4

Obviously, for any execution path $s \in EX_{M1}$ we have:

$$\text{if } M1, s \not\models_L \Box a \text{ and } M1, s \not\models_L \Box b \text{ then } M1, s \not\models_L \Diamond c.$$

Thus we have: $M1 \models_L f$.

For the path $s = w_0 w_2^\infty \in EX_{M2}$ we have : $M2, s \not\models_L f$.

So we obtain (I).

We prove (II) by induction on the structure of formulas in primitive form.

(1) If $g \in P$ then $(M1, w \underset{B}{\models} g \Leftrightarrow M2, w \underset{B}{\models} g, \forall w \in W := W_1)$ because $W_1 = W_2$ and $A_1 = A_2$.

(2) Let g_1, g_2 be formulas such that:

$$(\alpha)\ M1, w \underset{B}{\models} g_i \Leftrightarrow M2, w \underset{B}{\models} g_i, \quad \forall w \in W, \quad \text{for } i=1, 2.$$

Then one obtains for any formula $g \in F_B$:

(*a*) if $g = \neg g_1$ then $\forall w \in W$, $M1, w \underset{B}{\models} g \Leftrightarrow M2, w \underset{B}{\models} g$ straightforward by definition of $\underset{B}{\models}$ and (α);

(*b*) if $g = g_1 \vee g_2$ then $\forall w \in W$, $M1, w \underset{B}{\models} g \Leftrightarrow M2, w \underset{B}{\models} g$ straightforward by definition of $\underset{B}{\models}$ and (α);

(*c*) if $g = ALL\, g_1$ then $\forall w \in W$,

$$\begin{aligned} M1, w \underset{B}{\models} g &\Leftrightarrow \forall w' \in W,\ M1, w' \underset{B}{\models} g_1, \text{ because } R_1 \text{ is strongly connected,}\\ &\Leftrightarrow \forall w' \in W,\ M2, w' \underset{B}{\models} g_1, \text{ by } (\alpha),\\ &\Leftrightarrow M2, w \underset{B}{\models} g, \text{ because } R_2 \text{ is strongly connected;} \end{aligned}$$

(*d*) if $g = SOME\, g_1$ then $\forall w \in W$,

$$\begin{aligned} M1, w \underset{B}{\models} g &\Leftrightarrow M1, w \underset{B}{\models} g_1, \text{ because } w^\infty \in EX_{M1}(w),\\ &\Leftrightarrow M2, w \underset{B}{\models} g_1, \text{ by } (\alpha),\\ &\Leftrightarrow M2, w \underset{B}{\models} g, \text{ because } w^\infty \in EX_{M2}(W). \end{aligned}$$

From (1) and (2) we obtain $\forall w \in W$, $\forall g \in F_B$:

$$M1, w \underset{B}{\models} g \Leftrightarrow M2, w \underset{B}{\models} g.$$

Which implies (II). □

The proof given in [5] does not go through in our class of models because it depends on the fact, that the sets of execution paths choosen do not satisfy the Koenig's closure property.

Notice that the proof of theorem 2 depends essentially on the operators of the branching time logic considered. Indeed, if operators "until" like in [1] are added to TL_B then our proof is no longer valid (e. g. we have $M1 \models E(a\ \mathcal{U}\ c) \vee b$ but $M2, w_0 \not\models E(a\ \mathcal{U}\ c) \vee b$).

CONCLUSION

Even by restricting the class of models to transition systems, we proved a in [5] the incombarableness of the logics TL_B and TL_L. Furthermore, there i no linear time logic which is more expressive than TL_B because formulas of the form *POTp* cannot be expressed. We did not prove the same general result for branching time logic, and we think that it is interesting to find a branching time logic, extension of TL_B, which is more expressive than TL_L, or to prove that such a logic cannot exist. We have the strong feeling that a pure branching time logic, that means a logic where not, as in [3], formula on paths are introduced, cannot express all formulas of TL_L.

REFERENCES

1. E. M. Clarke and E. A. Emerson, *Design and synthesis of synchronization skeletons using branching time logic*, Harvard University TR-12-81.
2. E. A. Emerson, *Alternative semantics for temporal logics*, University of Texas at Austin, TR-182.
3. E. A. Emerson and J. Y. Halpern, *Decision procedure and expressiveness in temporal logic of branching time*, 14th Annual ACM Symp. on Theory of Computing, 1982.
4. E. A. Emerson and J. Y. Halpern, *"Sometimes" and "Not Never" revisited: on branching versus linear time*, ACM Symp. on Principals of Programming Languages.
5. L. Lamport, *"Sometimes" is sometimes "Not Never"*, 7th Annual ACM Symp. on Principals of Programming Languages, 1980.
6. A. Pnueli, *The temporal logic of concurrent Programs*, 19th Annual Symp. on Foundations on Computer Science, 1971.
7. J. P. Queille and J. Sifakis, *Specification and verification of concurrent systems in CESAR*, 5th International Symp. on Programming, 1982.