INFORMATIQUE THÉORIQUE ET APPLICATIONS

DILIAN GUROV
BRUCE KAPRON

A note on negative tagging for least fixed-point formulae

Informatique théorique et applications, tome 33, nº 4-5 (1999), p. 383-392

<http://www.numdam.org/item?id=ITA_1999__33_4-5_383_0>

Theoretical Informatics and Applications
Theoret. Informatics Appl. **33** (1999) 383–392

A NOTE ON NEGATIVE TAGGING FOR LEAST FIXED-POINT FORMULAE*

DILIAN GUROV[1] AND BRUCE KAPRON[2]

Abstract. Proof systems with sequents of the form $U \vdash \Phi$ for proving validity of a propositional modal μ-calculus formula Φ over a set U of states in a given model usually handle fixed-point formulae through unfolding, thus allowing such formulae to reappear in a proof. Tagging is a technique originated by Winskel for annotating fixed-point formulae with information about the proof states at which these are unfolded. This information is used later in the proof to avoid unnecessary unfolding, without having to investigate the history of the proof. Depending on whether tags are used for acceptance or for rejection of a branch in the proof tree, we refer to "positive" or "negative" tagging, respectively. In their simplest form, tags consist of the sets U at which fixed-point formulae are unfolded. In this paper, we generalise results of earlier work by Andersen *et al.* which, in the case of least fixed-point formulae, are applicable to singleton U sets only.

AMS Subject Classification. 03B70, 68Q60.

1. INTRODUCTION

The propositional modal μ-calculus is a particularly expressive logic for reasoning about branching-time properties of communicating systems. Many other logics, like dynamic logic and CTL, have uniform encodings in this logic [4, 8]. Over the last decade, many proof systems for checking validity of μ-calculus formulae over given states in a model have been proposed, *e.g.* in [1, 3, 5, 7, 9] among others. The main difficulty in devising such proof systems lies in the handling of fixed-point formulae. These are usually unfolded during proof construction,

* *The first author was partially supported by a Swedish Foundation for Strategic Research Junior Individual Grant.*

[1] Swedish Institute of Computer Science, Box 1263, SE-164 29 Kista, Sweden; e-mail: `dilian@sics.se`

[2] Department of Computer Science, University of Victoria, Victoria, B.C., Canada V8W 3P6; e-mail: `bmkapron@csc.uvic.ca`

thus allowing them to reappear in a proof. One therefore needs conditions for terminating the proof search process based on identifying certain "loops" in a proof. Important techniques for dealing with fixed-point formulae are the subformula condition of Streett and Emerson [10], the constants of Stirling and Walker [9], the tags of Winskel [11], and the ordinal variables of Dam *et al.* [6]. The tagging approach is appealing in that it allows all reasoning to be performed using local rules only, and also in that it has a simple semantic justification.

Of the two kinds of fixed-point formulae, the least fixed-point ones are more difficult to handle in general, usually requiring some sort of Noetherian induction over some well-founded set [1, 3, 7]. When model checking finite-state systems, however, it is sufficient to perform simple unfolding. In this case, inductive reasoning can reduce the size of a proof significantly, but makes proof search far more complicated. Even if no induction is employed, it still makes sense to record the states at which a least fixed-point formula has been unfolded, since this information can be used to reject a branch. For example, the proof system presented in [2] has a rule of the shape:

$$(\mu)\ \frac{s \vdash \Phi[\mu Z\{s, L\}.\Phi/Z]}{s \vdash \mu Z\{L\}.\Phi}\ s \notin L$$

which prevents least fixed-point formulae from being unfolded more than once at the same state. Such a rule can be justified semantically by defining tags L to denote sets of states, and by defining the denotation of tagged least fixed-point formulae as follows:

$$\|\mu Z\{L\}.\Phi\|_{\mathcal{V}} \overset{\Delta}{\equiv} \mu X.\Big(\|\Phi\|_{\mathcal{V}[Z\mapsto X]} - L\Big).$$

Rule (μ) is sound and reversible due to the following equivalence, known as the Reduction lemma (Kozen [8], Winskel [11]):

$$s \in \mu X.f(X) \equiv s \in f(\mu X.(f(X) - \{s\})) \tag{1}$$

which holds for any monotone mapping $f : \wp(S) \rightarrow \wp(S)$. We refer to tagging used in this way as *negative tagging*, since tags are in some sense negative assumptions: we assume that the states in the tag do *not* belong to the denotation of the tagged least fixed-point formula.

Unfortunately, equivalence (1) holds only for single states, and not for sets of states in general [1]. Rule (μ) would in general be unsound in a proof system with sequents of the shape $U \vdash \mu Z\{L\}.\Phi$ where U is a set of states, and where validity of sequents is understood as set inclusion.

In this paper, we investigate for what semantics of tags and tagged formulae, and for what relationship $\bowtie$ between a set of states U and a tag L, one could justify a rule of the shape

$$(\mu')\ \frac{U \vdash \Phi\,[\mu Z\{U,L\}.\Phi/Z]}{U \vdash \mu Z\{L\}.\Phi}\ U \bowtie L.$$

The paper is organised as follows. First, we present the syntax and semantics of the propositional modal μ-calculus. In the following section we motivate a way of tagging least fixed-point formulae, and propose a suitable semantics for tagged least fixed-point formulae, giving rise to a sound and reversible inference rule. Section 4 presents a proof system in which this proof rule fits naturally. Finally, some conclusions are drawn in the last section.

2. Propositional modal μ-calculus

This section presents briefly the usual notions and notation for the modal μ-calculus used in the sequel.

2.1. Syntax

Formulae Φ of the logic are generated by the grammar:

$$\Phi \ ::= \ Z \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid [a]\,\Phi \mid \langle a\rangle\,\Phi \mid \nu Z.\Phi \mid \mu Z.\Phi$$

where Z ranges over a set of propositional variables, and a ranges over a non-empty set $\mathcal{L}$ of labels.

2.2. Semantics

Modal μ-calculus formulae are usually interpreted as sets of states in transition systems.

Definition 2.1 (Transition System). A *transition system* is a pair $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{L}\})$ where $\mathcal{S}$ is a non-empty set of states, $\mathcal{L}$ is a non-empty set of labels, and for each $a \in \mathcal{L}$, $\xrightarrow{a}\subseteq \mathcal{S} \times \mathcal{S}$.

Definition 2.2 (Model). A *model* for a (possibly open) modal μ-calculus formula is a pair $\mathcal{M} = (\mathcal{T}, \mathcal{V})$, where $\mathcal{T}$ is a transition system, and $\mathcal{V}$ is a valuation taking propositional variables to subsets of states of $\mathcal{T}$.

The semantics of a modal μ-calculus formula Φ in a model $\mathcal{M} = (\mathcal{T}, \mathcal{V})$ is given by its denotation $||\Phi||^{\mathcal{T}}_{\mathcal{V}}$ (we shall sometimes omit the superscript).

Definition 2.3 (Denotation). The *denotation* $||\Phi||^{\mathcal{T}}_{\mathcal{V}}$ of a modal μ-calculus formula Φ is defined inductively as follows:

$$\begin{aligned}
\|Z\|^{\mathcal{T}}_{\mathcal{V}} &\triangleq \mathcal{V}(Z) \\
\|\Phi_1 \wedge \Phi_2\|^{\mathcal{T}}_{\mathcal{V}} &\triangleq \|\Phi_1\|^{\mathcal{T}}_{\mathcal{V}} \cap \|\Phi_2\|^{\mathcal{T}}_{\mathcal{V}} \\
\|\Phi_1 \vee \Phi_2\|^{\mathcal{T}}_{\mathcal{V}} &\triangleq \|\Phi_1\|^{\mathcal{T}}_{\mathcal{V}} \cup \|\Phi_2\|^{\mathcal{T}}_{\mathcal{V}} \\
\|[a]\,\Phi\|^{\mathcal{T}}_{\mathcal{V}} &\triangleq \|[a]\|^{\mathcal{T}}\, \|\Phi\|^{\mathcal{T}}_{\mathcal{V}} \\
\|\langle a\rangle\,\Phi\|^{\mathcal{T}}_{\mathcal{V}} &\triangleq \|\langle a\rangle\|^{\mathcal{T}}\, \|\Phi\|^{\mathcal{T}}_{\mathcal{V}} \\
\|\nu Z.\Phi\|^{\mathcal{T}}_{\mathcal{V}} &\triangleq \nu X.\, \|\Phi\|^{\mathcal{T}}_{\mathcal{V}[Z\mapsto X]} \\
\|\mu Z.\Phi\|^{\mathcal{T}}_{\mathcal{V}} &\triangleq \mu X.\, \|\Phi\|^{\mathcal{T}}_{\mathcal{V}[Z\mapsto X]}
\end{aligned}$$

where we refer to the predicate transformers

$$\begin{aligned}
\|[a]\|^{\mathcal{T}} &\triangleq \lambda X.\{s \in \mathcal{S} \mid \forall s' : s \xrightarrow{a} s'.\, s' \in X\} \\
\|\langle a\rangle\|^{\mathcal{T}} &\triangleq \lambda X.\{s \in \mathcal{S} \mid \exists s' : s \xrightarrow{a} s'.\, s' \in X\}.
\end{aligned}$$

This definition uses the fact that the logic is in positive form, and hence the predicate transformers $\lambda X.\, \|\Phi\|^{\mathcal{T}}_{\mathcal{V}[Z\mapsto X]}$ are monotone w.r.t. set inclusion and are guaranteed to have greatest and least fixed points, denoted $\nu X.\, \|\Phi\|^{\mathcal{T}}_{\mathcal{V}[Z\mapsto X]}$ and $\mu X.\, \|\Phi\|^{\mathcal{T}}_{\mathcal{V}[Z\mapsto X]}$, respectively.

We shall also need the notion of Knaster-Tarski fixed-point approximants of monotone mappings over $\wp(\mathcal{S})$.

Definition 2.4 (Fixed-Point Approximants). Let $f : \wp(\mathcal{S}) \to \wp(\mathcal{S})$ be monotone, let *Ord* denote the class of all *ordinals*, and let γ and λ range over ordinals and limit ordinals, respectively. Fixed-point approximants are defined inductively as follows:

$$\begin{aligned}
\mu^0 f &\triangleq \emptyset & \nu^0 f &\triangleq \mathcal{S} \\
\mu^{\gamma+1} f &\triangleq f(\mu^\gamma f) & \nu^{\gamma+1} f &\triangleq f(\nu^\gamma f) \\
\mu^\lambda f &\triangleq \textstyle\bigcup_{\gamma<\lambda} \mu^\gamma f & \nu^\lambda f &\triangleq \textstyle\bigcap_{\gamma<\lambda} \nu^\gamma f.
\end{aligned}$$

3. Negative tagging for sets of states

Let us start by analysing why it is that the equivalence (1) fails for sets of states. If we adopt the notation $\mu X\{U\}.f(X)$ for $\mu X.(f(X) - U)$, this equivalence could be rewritten as:

$$s \in \mu X.f(X) \;\equiv\; s \in f(\mu X\{s\}.f(X)).$$

Consider the following LTS:

$$s_3 \xrightarrow{a} s_2 \xrightarrow{a} s_1 \xrightarrow{a} s_0$$

and the formula $\mu Z.\,[a]\,Z$, the denotation of which is the least fixed-point μf of the state transformer $f \triangleq \lambda X.\, \|[a]\|\, X$. We have $\mu X\{s_2\}.f(X) = \{s_0, s_1\}$ and hence $f(\mu X\{s_2\}.f(X)) = f(\{s_0, s_1\}) = \{s_0, s_1, s_2\}$ includes s_2. In terms of fixed-point approximants, $\mu X\{s\}.f(X)$ contains $\mu^\alpha f$ for the greatest ordinal α such that $\mu^\alpha f$ does not include s, since this is the first point in the iterative construction of the fixed-point where s comes into play[1]. In this example α equals two. Since f is monotone, $s \in \mu f$ implies:

$$s \in \mu^{\alpha+1} f = f(\mu^\alpha f) \subseteq f(\mu X\{s\}.f(X))$$

and therefore $s \in f(\mu X\{s\}.f(X))$. This is exactly the point where we cannot extend this reasoning to an arbitrary set of states U: if α is the greatest ordinal[2] for which $\mu^\alpha f$ does not intersect U, then $U \subseteq \mu^{\alpha+1} f$ is guaranteed only when U is a singleton set. For example, for $U = \{s_1, s_2\}$ we have $\mu X\{U\}.f(X) = \{s_0\}$ and hence $f(\mu X\{U\}.f(X)) = \{s_0, s_1\}$ which includes s_1 but does not include s_2. On the other hand, the following observation can be made: a relationship of the shape

$$U \subseteq \mu^{\alpha+1} f = f(\mu^\alpha f) \subseteq f(\mu X\{U\}.f(X))$$

would still hold if we redefined:

- α to be the greatest ordinal (if there is such) so that $\mu^\alpha f$ does not contain (rather than "does not intersect") U. Then $U \subseteq \mu^{\alpha+1}$.
- Tags to be sets of states U denoting not themselves, but rather those elements of U only which are not in $\mu^\alpha f$. Then $\mu^\alpha f \subseteq \mu X\{U\}.f(X)$ and therefore $f(\mu^\alpha f) \subseteq f(\mu X\{U\}.f(X))$.

We now proceed to formalise the above intuitive ideas. Let $\mathcal{S}$ be a set (of states), and let $f : \wp(\mathcal{S}) \to \wp(\mathcal{S})$ be monotone.

Definition 3.1. Let $U \subseteq \mathcal{S}$ be a set of states. The *closure ordinal* $co_f U$ and *closure elements* $ce_f U$ of U w.r.t. f are defined as follows:

$co_f U \triangleq$ the least ordinal α such that $U \cap \mu f \subseteq \mu^\alpha f$
$ce_f U \triangleq U - \bigcup_{\beta < co_f U} \mu^\beta f$.

Note 3.2. $\mu^{co_f U} f$ can be partitioned into $ce_f U \cap \mu f$ and $\bigcup_{\beta < co_f U} \mu^\beta f$, the latter set being equal to $\mu^\alpha f$ whenever $co_f U$ is the successor ordinal of α.

Proposition 3.3. *Let $U \subseteq \mathcal{S}$ be a set of states. Then:*
(i) $(U \cap \mu f) \subseteq \mu^{co_f U} f$.
(ii) $ce_f U \cap \mu f \neq \emptyset$ if and only if $co_f U$ is a successor ordinal.
(iii) If U is finite, then $co_f U$ is not *a limit ordinal.*
(iv) If $s \in \mathcal{S}$, then $ce_f\{s\} = \{s\}$.

Proof. These properties are established as follows.

(i) Follows directly from the definition of $co_f U$.

[1] Or alternatively, $\alpha + 1$ is the least ordinal such that $\mu^{\alpha+1} f$ includes s.

[2] It should also be noted here, that such a greatest ordinal is guaranteed to exist only when U is finite.

(ii) We have:

$$\begin{array}{lll} ce_f U \cap \mu f \neq \emptyset & \equiv \bigcup_{\beta < co_f U} \mu^{\beta} f \neq \mu^{co_f U} f & \{\text{Note 3.2}\} \\ & \equiv co_f U \text{ is a successor ordinal} & \{\text{Def. 2.4}\}. \end{array}$$

(iii) From the definition of fixed-point approximants follows immediately that the closure ordinal for singleton sets is not a limit ordinal. If U is finite, the closure ordinals of the singletons formed by the elements of U have a greatest element α which is not a limit ordinal. This ordinal is also the closure ordinal of U.

(iv) This is a direct consequence of (iii).

□

Definition 3.4. Let $U \subseteq S$. We define *tagged mappings* as follows:

$$f_{\{U\}} \triangleq \lambda X.(f(X) - ce_f U)$$

and use the notation $f_{\{U,V_1,\dots,V_n\}}$ for $(f_{\{V_1,\dots,V_n\}})_{\{U\}}$.

Note 3.5. In the chosen notation $\mu f_{\{U\}}$ equals $\mu X\{ce_f U\}.f(X)$. Because of Proposition 3.3 (iv), this semantics of tags coincides with the one already given in the Introduction for the case of singleton sets.

Proposition 3.6. *Let $U \subseteq S$ be a set of states. Then:*

(i) $\mu f_{\{U\}} \subseteq \mu f$

(ii) if $co_f U$ is the successor of some ordinal α, then $\mu^{\alpha} f = \mu^{\alpha} f_{\{U\}}$.

Proof. These properties are established as follows.

(i) Follows directly from the well-known equation:

$$\mu g = \bigcap \{X \mid g(X) \subseteq X\}$$

since $f(X) \subseteq X$ implies $f(X) - ce_f U \subseteq X$.

(ii) Let $co_f U = \alpha + 1$. Then $ce_f U \cap \mu^{\alpha} f = \emptyset$ by Definition 3.1 and Note 3.2. From Definitions 2.4 and 3.4 it follows that $\mu^{\beta} f_{\{U\}} \subseteq \mu^{\beta} f$ for any β, and consequently $ce_f U \cap \mu^{\beta} f_{\{U\}} = \emptyset$ holds for all $\beta \leq \alpha$, implying the result.

□

The following proposition will be used to justify the side condition of the new proof rule (μ').

Proposition 3.7. *For any finite non-empty set U,*

$$U \not\subseteq \mu f_{\{V_1,\dots,U,\dots,V_n\}}.$$

Proof. By induction on n. The base case (*i.e.*, empty tag) holds vacuously. The induction hypothesis assumes the property for an arbitrary k. Assume U is a finite non-empty set. If $U = V_i$ for some i such that $2 \leq i \leq k+1$ then the property holds, since $\mu f_{\{V_2,\ldots,V_{k+1}\}} \subseteq \mu f_{\{V_1,V_2,\cdots,V_{k+1}\}}$ by Proposition 3.6 (i) and $U \not\subseteq \mu f_{\{V_2,\ldots,U,\ldots,V_{k+1}\}}$ by the induction hypothesis. The case that remains to be considered is $U = V_1$. Let g denote $\mu f_{\{V_2,\ldots,V_{k+1}\}}$. We have to show that $U \not\subseteq \mu g_{\{U\}}$. This obviously holds when $U \not\subseteq \mu g$, so assume $U \subseteq \mu g$ instead. Since U is non-empty and finite, it follows from Propositions 3.3 (ii) and (iii) that $ce_g U$ is also non-empty. But $ce_g U \subseteq U$ by Definition 3.1, and $\mu g_{\{U\}} \cap ce_g U = \emptyset$ by Definition 3.4, implying the desired $U \not\subseteq \mu g_{\{U\}}$. □

The following lemma plays the same rôle as Kozen's Reduction lemma.

Lemma 3.8 (Reduction lemma). *For any set $U \subseteq S$ the following equivalence holds:*

$$U \subseteq \mu f \equiv U \subseteq f(\mu f_{\{U\}}).$$

Proof. The two directions are established as follows:

($\Leftarrow$) this direction holds simply because $f(\mu f_{\{U\}}) \subseteq f(\mu f) = \mu f$.

($\Rightarrow$) If $ce_f U \cap \mu f$ is empty, then the implication holds trivially since in this case $\mu f = \mu f_{\{U\}} = f(\mu f) = f(\mu f_{\{U\}})$. If $ce_f U \cap \mu f$ is not empty, then by Proposition 3.3 (ii) $co_f U$ is the successor of some ordinal α. Then:

$$\begin{array}{llll}
U \subseteq \mu f & \equiv & U \subseteq \mu^{co_f U} f & \{\text{Prop. 3.3(i)}\} \\
 & \equiv & U \subseteq \mu^{\alpha+1} f & \{co_f U = \alpha + 1\} \\
 & \equiv & U \subseteq f(\mu^{\alpha} f) & \{\text{Def. fixed} - \text{point approximants}\} \\
 & \equiv & U \subseteq f(\mu^{\alpha} f_{\{U\}}) & \{\text{Prop. 3.6(ii)}\} \\
 & \Rightarrow & U \subseteq f(\mu f_{\{U\}}) & \{\mu^{\alpha} f_{\{U\}} \subseteq \mu f_{\{U\}}\}.
\end{array}$$

□

We are now ready to give a suitable semantics to formulae tagged with lists of sets of states.

Definition 3.9. The *denotation* of negatively tagged formulae is defined as follows:

$$\|\mu Z\{V_1,\ldots,V_n\}.\Phi\|_{\mathcal{V}}^{\mathcal{T}} \triangleq \mu f_{\{V_1,\ldots,V_n\}}, \text{ where } f = \lambda X.\, \|\Phi\|_{\mathcal{V}[Z\mapsto X]}^{\mathcal{T}}.$$

Due to Note 3.5 this semantics is equivalent to the one already given in the Introduction for the case when the tag sets are singletons, and is hence a proper generalisation of the latter. It gives rise to the following inference rule:

$$(\mu')\ \frac{U \vdash \Phi\,[\mu Z\{U,L\}.\Phi/Z]}{U \vdash \mu Z\{L\}.\Phi}\ \ U \textit{ finite} \Rightarrow \forall V \in L.\, V \not\subseteq U.$$

In general, a proof rule is called *sound* if it preserves validity, *i.e.*, whenever the premises to the rule are valid and the side-condition holds, then the conclusion is also valid. If the opposite holds, the rule is called *reversible.* In the rule above, the purpose of the side-condition is somewhat unusual, since it is not needed to ensure soundness, but rather to avoid unnecessary application of the rule in case the conclusion is invalid. Reversibility of the rule ensures that validity of the conclusion implies the side-condition; in fact we use, and prove, the counterpositive statement.

Theorem 3.10. *Rule (μ') is sound and reversible.*

Proof. As a straightforward consequence of Definition 3.9 and the Reduction lemma, validity of the premise implies validity of the conclusion, and *vice versa.* Now assume the side condition does not hold, *i.e.*, U is finite and some set V_i in the tag is a subset of U. Then V_i is also finite, and hence, due to Proposition 3.7, the sequent $V_i \vdash \mu Z\{V_1, \dots, V_n\}.\Phi$ is invalid, and hence $U \vdash \mu Z\{V_1, \dots, V_n\}.\Phi$ is invalid as well. □

Rule (μ') is easily seen to be a proper generalisation of rule (μ) presented in the Introduction. The most interesting question that offers itself immediately is whether finiteness of U is really relevant for rejecting a branch in a proof tree. This turns out to be the case, as Example 4.2 in the next section shows.

4. Applications

The proof rule (μ') can be plugged into any standard proof system for establishing satisfaction between a set of states U in a model and a modal μ-calculus formula. In Figure 1 below we present one such proof system, borrowed from Andersen [1], in which rule (μ') replaces the rules for least fixed-point formulae of the original proof system. In these rules the following notation is used:

$$(\xrightarrow{a} U) \triangleq \{s \in \mathcal{S} \mid \exists s' \in U.\, s \xrightarrow{a} s'\}$$

$$(U \xrightarrow{a}) \triangleq \{s \in \mathcal{S} \mid \exists s' \in U.\, s' \xrightarrow{a} s\}.$$

A proof for a sequent is a proof tree with this sequent at the root and axiom leaves only. Proof trees are constructed in a goal-oriented fashion, beginning with the sequent to be proved at the bottom and applying the rules upwards.

Example 4.1. Consider a LTS with two states s_1 and s_2 and two labelled transitions $s_1 \xrightarrow{a} s_1$ and $s_1 \xrightarrow{a} s_2$. State s_1 can engage in an infinite a-sequence, and therefore the attempt of proving the opposite, namely $\{s_1\} \vdash \mu Z.\, [a]\, Z$, fails:

$$\cfrac{\cfrac{\{s_1, s_2\} \vdash \mu Z\{s_1\}.\, [a]\, Z}{\{s_1\} \vdash [a]\, \mu Z\{s_1\}.\, [a]\, Z}([\,])}{\{s_1\} \vdash \mu Z.\, [a]\, Z}(\mu').$$

$$(\emptyset)\;\frac{}{\emptyset \vdash \Phi}$$

$$(\wedge)\;\frac{U \vdash \Phi_1 \qquad U \vdash \Phi_2}{U \vdash \Phi_1 \wedge \Phi_2} \qquad (\vee)\;\frac{U_1 \vdash \Phi_1 \qquad U_2 \vdash \Phi_2}{U_1 \cup U_2 \vdash \Phi_1 \vee \Phi_2}$$

$$([\,])\;\frac{(U\xrightarrow{a}) \vdash \Phi}{U \vdash [a]\,\Phi} \qquad (\langle\rangle)\;\frac{U \vdash \Phi}{U' \vdash \langle a\rangle\,\Phi}\;(\xrightarrow{a}U) \supseteq U'$$

$$(\nu 0)\;\frac{}{U \vdash \nu Z\{V\}.\Phi}\;U \subseteq V \qquad (\nu 1)\;\frac{U \vdash \Phi\,[\nu Z\{U \cup V\}.\Phi/Z]}{U \vdash \nu Z\{V\}.\Phi}\;U \not\subseteq V$$

$$(\mu')\;\frac{U \vdash \Phi\,[\mu Z\{U, L\}.\Phi/Z]}{U \vdash \mu Z\{L\}.\Phi}\;U\ \mathit{finite} \Rightarrow \forall V \in L.\, V \not\subseteq U$$

FIGURE 1. An example proof system.

At the leaf sequent $\{s_1, s_2\} \vdash \mu Z\{s_1\}.\,[a]\,Z$, rule μ' is not applicable since its side-condition is violated, indicating that the leaf sequent is invalid. A proof-search mechanism would use such information to enforce backtracking in such points in the proof.

Example 4.2. Consider the infinite-state LTS with states S:

$$\cdots \xrightarrow{a} s_3 \xrightarrow{a} s_2 \xrightarrow{a} s_1 \xrightarrow{a} s_0$$

and the formula $\mu Z.\,[a]\,Z$. Consider the following derivation:

$$\dfrac{\dfrac{S \vdash \mu Z\{S\}.\,[a]\,Z}{S \vdash [a]\,\mu Z\{S\}.\,[a]\,Z}([\,])}{S \vdash \mu Z.\,[a]\,Z}(\mu').$$

While in this case it still makes sense to backtrack at the leaf sequent, since there is nothing to be gained from repeating the above steps, it is unsound to conclude that this sequent is invalid.

This proof system is complete for finite-state systems and tag-free closed formulae (*i.e.*, tags only emerge during proof construction). To see this, first observe that the only rules which do not increase the size of formulae are the tagging rules (*i.e.*, the rules for unfolding fixed-point formulae), and that tags can only be of finite length with the chosen tagging discipline enforced by the side-conditions. Proof tableaux are hence of finite size only. On the other hand, it can easily be shown that every valid sequent can be derived from some (possibly empty) set of valid sequents. Together, these two observations imply that for every valid sequent there is a finished tableau, *i.e.* a finite tableau with axiom leaves only. A formal proof of completeness can easily be obtained along the lines of the completeness proof for the original proof system [1].

5. Conclusion

In this paper we present a way of tagging, together with a suitable semantics, for least fixed-point formulae of the propositional modal μ-calculus. These are used to justify a proof rule for unfolding, combined with tagging, of such formulae in proof systems with sequents of the shape $U \vdash \Phi$ where U is a set of states, and Φ is a formula. The proof rule is plugged into a standard proof system for model checking, yielding a complete proof system for finite-state systems.

The result is an extension of previous results on negative tagging to the case of sets of states. This suggests that it can be used for devising similar proof rules in other settings. For example, formulae can be understood as sets of states, and so can parametrised processes, and consequently, proof systems with sequents of the shape $\Phi \vdash \Psi$ or $P(x) \vdash \Psi$ can benefit from the proposed negative tagging technique to provide additional termination conditions, thus aiding both proof search and the theoretical investigation of these proof systems.

We would like to thank Mads Dam and Lars-åke Fredlund, as well as the anonymous referee for valuable comments on the manuscript.

References

[1] H.R. Andersen, *Verification of Temporal Properties of Concurrent Systems.* Ph.D. Thesis, Computer Science Department, Aarhus University, Denmark (1993).

[2] H.R. Andersen, C. Stirling and G. Winskel, A compositional proof system for the modal mu-calculus, in *Proc. of LICS'94* (1994).

[3] J. Bradfield, *Verifying Temporal Properties of Systems.* Birkhauser (1992).

[4] M. Dam, CTL* and ECTL* as fragments of the modal μ-calculus. *Theoret. Comput. Sci.* **126** (1994) 77–96.

[5] M. Dam, Proving properties of dynamic process networks. *Inform. and Comput.* **140** (1998) 95–114.

[6] M. Dam, L. Fredlund and D. Gurov, Toward parametric verification of open distributed systems, H. Langmaack, A. Pnueli and W.-P. De Roever, Eds., *Compositionality: The Significant Difference.* Springer, *Lecture Notes in Comput. Sci.* **1536** (1998) 150–158.

[7] D. Gurov, S. Berezin and B. Kapron, A modal μ-calculus and a proof system for value passing processes. *Electron. Notes Theoret. Comput. Sci.* **5** (1996).

[8] D. Kozen, Results on the propositional μ-calculus. *Theoret. Comput. Sci.* **27** (1983) 333–354.

[9] C. Stirling and D. Walker, Local model checking in the modal mu-calculus. *Theoret. Comput. Sci.* **89** (1991) 161–177.

[10] R.S. Streett and E.A. Emerson, An automata theoretic decision procedure for the propositional mu-calculus. *Inform. and Comput.* **81** (1989) 249–264.

[11] G. Winskel, A note on model checking the modal nu-calculus. *Theoret. Comput. Sci.* **83** (1991) 157–167.

Received October 30, 1998. Revised June 1, 1999.