Refinement in branching time semantics

R.J. van Glabbeek
W.P. Weijland
Centre for Mathematics and Computer Science
P.O.Box 4079, 1009 AB Amsterdam, The Netherlands

Abstract: In this paper we consider branching time semantics for finite sequential processes with silent moves. We show that MILNER's notion of *observation equivalence* is not preserved under refinement of actions, even when no interleaving operators are considered; however, the authors' notion of *branching bisimulation* is.

INTRODUCTION

Virtually all semantic equivalences employed in theories of concurrency are defined in terms of *actions* that concurrent systems may perform (cf. [1-7]). Mostly, these actions are taken to be *atomic*, meaning that they are considered not to be divisible into smaller parts. In this case, the defined equivalences are said to be based on *action atomicity*.

However, in the top-down design of distributed systems it might be fruitful to model processes at different levels of abstraction. The actions on an abstract level then turn out to represent complex processes on a more concrete level. This methodology does not seem compatible with non-divisibility of actions and for this reason, PRATT [7], LAMPORT [4] and others plead for the use of semantic equivalences that are not based on action atomicity.

As indicated in CASTELLANO, DE MICHELIS & POMELLO [2], the concept of action atomicity can be formalised by means of the notion of *refinement of actions*. A semantic equivalence is *preserved under action refinement* if two equivalent processes remain equivalent after replacing all occurrences of an atomic action a by a more complicated process r(a). In particular, r(a) may be a sequence of two actions a_1 and a_2. An equivalence is strictly based on action atomicity if it is not preserved under action refinement.

In a previous paper [3] the authors argued that MILNER's notion of observation equivalence [5] does not respect the branching structure of processes, and proposed the finer notion of *branching bisimulation equivalence* which does. In this paper we moreover find, that observation equivalence is not preserved under action refinement, whereas branching bisimulation equivalence is.

1. PROCESS GRAPHS

As a simple model, let us represent a process by a *state transition diagram* or *process graph*. Such a graph has a node for every one of the possible states of the process, and has arrows between nodes to indicate whether or not a state is accessible from another. Furthermore, these arrows (directed edges) are labelled, with labels from $A \cup \{\tau\}$, where $A = \{a,b,c,...\}$ is some set of *observable signals*, and τ stands for a *silent step* (cf. [5]).

DEFINITION 1.1 A *process graph* is a connected, rooted, edge-labelled and directed graph.

In an edge-labelled graph, one can have more than one edge between two nodes as long as they carry different labels. A rooted graph has one special node which is indicated as the root node. Graphs need not be finite, but in a connected graph one must be able to reach every node from the root node by following a finite path. If r and s are nodes in a graph, then $r \rightarrow^a s$ denotes an edge from r to s with label a (it is also used as a proposition stating that such an edge exists). In this paper we limit ourselves to processes represented by finite, non-trivial process graphs. A graph is *finite* if it is acyclic and contains only finitely many nodes and edges; it is *trivial* if it contains no edges at all. The set of non-trivial, finite process graphs will be denoted by G.

In order to turn G into an algebraic structure, it is possible to define binary operators '+' and '·' for alternative and sequential composition. For any two graphs g and h the process graph $(g + h)$ is obtained by simply identifying their root nodes, whereas $(g \cdot h)$ - often written as just (gh) - can be found by identifying the root node of h with all endnodes of g. Furthermore, constants from $A \cup \{\tau\}$ are interpreted as one-edge graphs, carrying the constant as their edge-label. The algebraic structure allows us to study equational theories that emerge from any defined equivalence on G. For instance, in branching time semantics, one often considers *observation congruence* (cf. MILNER [5]) - written as $\approx^c$ - as a deciding criterion for equality in observable behaviour. Let us write $r \Rightarrow r'$ for a path from r to r' consisting of an arbitrary number (≥ 0) of τ-edges. Then its definition can be rephrased as:

DEFINITION 1.2 Two graphs g and h are *observation equivalent* if there exists a symmetric relation $R \subseteq \text{nodes}(g) \times \text{nodes}(h) \cup \text{nodes}(h) \times \text{nodes}(g)$ (called a *τ-bisimulation*) such that:

1. The roots are related by R.
2. If $R(r,s)$ and $r \rightarrow^a r'$ $(a \in A \cup \{\tau\})$, then either $a = \tau$ and $R(r',s)$, or there exists a path $s \Rightarrow s_1 \rightarrow^a s_2 \Rightarrow s'$ such that $R(r',s')$.

Furthermore, g and h are *observation congruent* if we also have that

3. (root condition) Root nodes are related with root nodes only.

The root condition was first formulated by BERGSTRA & KLOP [1], and serves to turn the notion of observation equivalence into a congruence with respect to the operators + and ·. It can be proved that observation equivalence and observation congruence are equivalence relations on G, and that the latter is the coarsest congruence contained in the former (cf. [5,1,3]). It was shown in [1] that with respect to closed terms the model $G/\leftrightarrows^c$ is completely axiomatized by the theory

$x + y = y + x$	A1	$x\tau = x$	T1
$x + (y + z) = (x + y) + z$	A2	$\tau x = \tau x + x$	T2
$x + x = x$	A3	$a(\tau x + y) = a(\tau x + y) + ax$	T3
$x(yz) = (xy)z$	A4		
$(x + y)z = xy + xz$	A5	$(a \in A \cup \{\tau\})$	

The τ-laws T1-T3 originate from MILNER [5], who gave a complete axiomatization for a similar model with prefixing instead of general sequential composition. From these axioms, it is easy to show why the notion of observation congruence is not preserved under refinement of actions: replacing the action a by the term bc, we obtain $bc(\tau x + y) = bc(\tau x + y) + bcx$ from T3, which obviously is not valid in $G/\leftrightarrows^c$. By T3, we *do* find $bc(\tau x + y) = b(c(\tau x + y) + cx)$ which unfortunately denotes a different process.
Apart from the problem with refinement, it was observed in VAN GLABBEEK & WEIJLAND [3] that observation equivalence does not strictly preserve the branching structure of processes. This is because an important feature of a bisimulation (cf. PARK [6]) is missing for τ-bisimulation, which is the property that any computation in the one process corresponds to a computation in the other, in such a way that all intermediate states of these computations correspond as well. However, in observation congruence, when satisfying the second requirement of definition 1.2 one may execute arbitrarily many τ-steps in a graph without worrying about the status of the nodes that are passed in the meantime.
In order to overcome this problem, in [3] a different notion was introduced, which yields a finer equivalence on graphs.

DEFINITION 1.3 Two graphs g and h are *branching equivalent* if there exists a symmetric relation $R \subseteq \text{nodes}(g) \times \text{nodes}(h) \cup \text{node}(h) \times \text{nodes}(g)$ (called a *branching bisimulation*) such that:

1. The roots are related by R
2. If $R(r,s)$ and $r \rightarrow^a r'$ $(a \in A \cup \{\tau\})$, then either $a=\tau$ and $R(r',s)$, or there exists a path of the form $s \Rightarrow s_1 \rightarrow^a s'$ such that $R(r,s_1)$ and $R(r',s')$.

Furthermore, g and h are *branching congruent* if we also have that

3. (root condition) Root nodes are related with root nodes only.

Let us write R: $g \leftrightarrows_b h$ if R is a branching bisimulation between g and h and R: $g \leftrightarrows_{rb} h$ if, in addition, R satisfies the root condition. One can prove that the same equivalence is defined when in definition 1.3 *all* intermediate nodes in $s \Rightarrow s_1$ are required to be related with r. Furthermore, observe that a branching bisimulation can also be defined as in definition 1.2, with as extra requirements that $R(r,s_1)$ and $R(r',s_2)$.

It can be proved that branching equivalence and branching congruence are equivalence relations on G. Furthermore, the latter is the coarsest congruence contained in the former. It was shown in [3] that with respect to closed terms, the model $G/\leftrightarrows_{rb}$ is completely axiomatized by the axioms A1-A5 together with

$$x\tau = x \qquad \text{B1}$$

$$x(\tau(y + z) + y) = x(y + z) \qquad \text{B2.}$$

Note that the axioms B1-B2 when applied from left to right only eliminate occurrences of τ's. Using this property, it can be shown that the associated term rewriting system on $G/\equiv_{A1\text{-}A5}$, i.e. G modulo equality induced by the axioms A1-A5, is confluent and terminating. So any two closed branching congruent terms can be reduced to the same normal form.

2. REFINEMENT

In this section we will prove that branching congruence *is* preserved under refinement of actions, and so it allows us to look at actions as abstractions of much larger structures. Consider the following definitions.

DEFINITION 2.1 (substitution)

Let r: $A \to G$ be a mapping from observable actions to graphs, and suppose $g \in G$. Then, the graph r(g) can be found as follows.

For every edge $r \to^a r'$ ($a \in A$) in g, take a copy $\underline{r(a)}$ of r(a) ($\in G$). Next, identify r with the root node of $\underline{r(a)}$, and r' with all endnodes of $\underline{r(a)}$, and remove the edge $r \to^a r'$.

Note that in this definition it is never needed to identify r and r', since graphs from G are non-trivial. This way, the mapping r is defined on the domain G. Note that since $\tau \notin A$, τ-edges cannot be substituted by graphs. Finally, observe that every node in g is a node in r(g).

DEFINITION 2.2 (preservation under refinement of actions)

An equivalence $\approx$ on G is said to be *preserved under refinement of actions* if for every mapping r: $A \to G$, we have: $g \approx h \Rightarrow r(g) \approx r(h)$.

In other words, an equivalence = is preserved under refinement if it is a congruence with respect to every substitution operator r.
Starting from a relation R: $g \leftrightarrows_{rb} h$, we construct a branching bisimulation relation r(R): $r(g) \leftrightarrows_{rb} r(h)$, proving that preserving branching congruence, every edge with a label from A can be replaced by a graph.

DEFINITION 2.3 Let r: $A \rightarrow G$ be a mapping from observable actions to graphs, $g,h \in G$ and R: $g \leftrightarrows_{rb} h$. Now r(R) is the smallest relation between nodes of r(g) and r(h), such that:

1. $R \subseteq r(R)$.
2. If $r \rightarrow^a r'$ and $s \rightarrow^a s'$ ($a \in A$) are edges in g and h such that R(r,s) and R(r',s'), and both edges are replaced by copies $\underline{r(a)}$ and $\overline{r(a)}$ of r(a) respectively, then nodes from $\underline{r(a)}$ and $\overline{r(a)}$ are related by r(R), only if they are copies of the same node in r(a).

Edges $r \rightarrow^a r'$ and $s \rightarrow^a s'$ ($a \in A$) such that R(r,s) and R(r',s'), will be called *related* by R, as well as the copies $\underline{r(a)}$ and $\overline{r(a)}$ that are substituted for them. Observe, that on nodes from g and h the relation r(R) is equal to R. Note that if r(R)(r,s), then r is a node in g iff s is a node in h.

THEOREM (refinement)
Branching congruence is preserved under refinement of actions.

PROOF We prove that R: $g \leftrightarrows_{rb} h \Rightarrow$ r(R): $r(g) \leftrightarrows_{rb} r(h)$ by checking the requirements.

1. The root nodes of r(g) and r(h) are related by r(R).
2. Assume r(R)(r,s) and in r(g) there is an edge $r \rightarrow^a r'$. Then there are two possibilities (similarly in case $r \rightarrow^a r'$ stems from r(h)):
(i) The nodes r and s originate from g and h. Then R(r,s), and by the construction of r(g) we find that either $a=\tau$ and $r \rightarrow^\tau r'$ was already an edge in g, or g has an edge $r \rightarrow^b r^*$ and $r \rightarrow^a r'$ is a copy of an initial edge from r(b). In the first case it follows from R: $g \leftrightarrows_{rb} h$ that either R(r',s) - hence r(R)(r',s) - or in h there is a path $s \Rightarrow s_1 \rightarrow^\tau s'$ such that $R(r,s_1)$ and R(r',s'). By definition, the same path also exists in r(h), and we have $r(R)(r,s_1)$ and r(R)(r',s'). In the second case there must be a path $s \Rightarrow s_1 \rightarrow^b s^*$ in h such that $R(r,s_1)$ and R(r*,s*). Then, in r(h) we find a path $s \Rightarrow s_1 \rightarrow^a s'$ (by replacing $\rightarrow^b$ by r(b)) such that $r(R)(r,s_1)$ and r(R)(r',s').
(ii) The nodes r and s originate from related copies $\underline{r(b)}$ and $\overline{r(b)}$ of a substituted graph r(b) (for some $b \in A$), and are no copies of root or endnodes in r(b). Then $r \rightarrow^a r'$ is an edge in $\underline{r(b)}$. From r(R)(r,s) we find that r and s are copies of the same

node from r(b). So, there is an edge $s \rightarrow^a s'$ in $\overline{r(b)}$ where s' is a copy of the node in r(b), corresponding with r'. Clearly r(R)(r',s').

3. Since for nodes from g and h we have r(R)(r,s) iff R(r,s), the root condition is satisfied. □

With respect to closed terms, the refinement theorem can be proved much easier by syntactic analysis of proofs, instead of working with equivalences between graphs. For observe that the axioms A1-A5 + B1-B2, that form a complete axiomatization of branching congruence for closed terms, do *not* contain any occurrences of (atomic) actions from A. Now assume we have a proof of some equality s=t between closed terms, then this proof consists of a sequence of applications of axioms from A1-A5 + B1-B2. Since all these axioms are universal equations without actions from A, the actions from s and t can be replaced by general variables, and the proof will still hold. Hence, every equation is an instance of a universal equation *without* any actions. Immediately we find that we can substitute arbitrary closed terms for these variables, obtaining refinement for closed terms.

Nevertheless, the semantic proof of the refinement theorem is important as one may wish to generalize the result to models of larger graphs than just finite ones from ***G***.

REFERENCES

[1] J.A.BERGSTRA & J.W.KLOP, *Algebra of communicating processes with abstraction*, TCS 37 (1), pp.77-121, 1985.

[2] L.CASTELLANO, G.DE MICHELIS & L.POMELLO, *Concurrency vs Interleaving: an instructive example*, Bulletin of the EATCS 31, pp.12-15, 1987.

[3] R.J.VAN GLABBEEK & W.P.WEIJLAND, *Branching time and abstraction in bisimulation semantics* (extended abstract), Report CS-R8911, Centrum voor Wiskunde en Informatica, Amsterdam 1989, to appear in: proc. IFIP 11th World Computer Congress, San Francisco 1989.

[4] L.LAMPORT, *On interprocess communication. Part 1: Basic formalism*, Distributed Computing 1 (2), pp.77-85, 1986.

[5] R.MILNER, *A calculus of communicating systems*, Springer LNCS 92, 1980.

[6] D.PARK, *Concurrency and automata on infinite sequences*, proc. 5th GI conf. on Th. Comp. Sci. (P.Deussen ed.), Springer LNCS 104, pp.167-183, 1981.

[7] V.R.PRATT, *Modelling concurrency with partial orders*, International Journal of Parallel Programming 15 (1), pp.33-71, 1986.