stichting
mathematisch
centrum

MC

AFDELING INFORMATICA
(DEPARTMENT OF COMPUTER SCIENCE)

IW 116/79 SEPTEMBER

J.W. DE BAKKER & J.I. ZUCKER

DERIVATIVES OF PROGRAMS

Preprint

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O).

1980 Mathematics subject classification: 68B10, 68C05

ACM-Computing Reviews Category: 5.24

Derivatives of Programs*)

by

J.W. de Bakker & J.I. Zucker**)

ABSTRACT

The notions of *upper* and *lower derivatives* of a recursive (non-deterministic) program are defined, and used to characterize termination for such a program in terms of the *well-foundedness* of a function with respect to a predicate. This extends earlier work of Hitchcock and Park to the case of *nested recursions*, formulated in terms of a least-fixed-point construct. It is shown how this characterization can be interpreted as stating that a recursive procedure always terminates iff it exhibits neither *global* nor *local* nontermination.

KEY WORDS & PHRASES: *denotational semantics, derivative of a program, recursive procedure, termination, nontermination, global nontermination, local nontermination, divergence of a program.*

*) This report will be submitted for publication elsewhere.

**) Address of the second author after October 1979:
Dept. of Mathematics and Computer Science, Bar Ilan University,
Ramat Gan, Israel

1. INTRODUCTION

The notion of *derivative* of a program was introduced by Hitchcock and Park [H,P] as an aid to investigate properties of program termination. More specifically, they showed how termination of a recursive program scheme may be expressed through the well-foundedness of a relation involving the so-called upper and lower derivatives of the scheme. The framework in which this result is derived is a calculus of binary relations extended with recursion via the least-fixed-point construct $\mu X[\ldots]$. However, their main result was proved only under a number of restrictions: (i) only deterministic programs, (ii) no nested μ-constructs, (iii) some further technical restrictions. In De Bakker [dB1], it was shown how to generalize the theory of [H,P] in the framework of denotational semantics (using the Egli-Milner ordering to deal with nondeterminacy, thus lifting restriction (i)) in such a way that restriction (iii) also disappeared, but maintaining restriction (ii). The present paper gives the full story in that we now also deal with nested μ-constructs. This necessities a non-trivial extension of the definition of upper and lower derivatives (cf. 5.1c, 5.3c), and, accordingly, a considerably more intricate proof (surpassing in complexity all proofs in the μ-calculus we have had experience with) of the basic theorem (5.5) connecting these two notions.

Section 2 of this paper describes the syntax, section 3 provides the necessary background in denotational semantics, section 4 introduces a fundamental auxiliary result allowing us to syntactically reduce termination of a program (involving recursion) to termination of its components, and in section 5 we define the upper and lower derivatives of a program and state (without proof) the basic theorem relating the two. Finally, in section 6 we introduce the notion of a function being well-founded with respect to a predicate, thus refining an idea in [H,P], and prove as main theorem the announced extension of the result there. The section closes with an example illustrating how this result may be interpreted as stating that a recursive procedure terminates everywhere iff it exhibits neither *global* nor *local* nontermination.

A fuller exposition of this paper, with detailed proofs, is given in chapter 8 of [dB2].

2. SYNTAX

The definition in this and the next section, though to some extent variations on familiar themes in denotational semantics, also include some new ideas, e.g., role of $b \in$ *Stat*, of $\mu Z[p]$, and of $f_1 \to f_2$.

Convention. "Let $(\alpha\in)V$ be the set..." is short for "let V be the set..., with variable α ranging over V".

2.1. DEFINITIONS

"$\equiv$" denotes identity between syntactic constructs. Let $(n\in)$*Intc* be the set of *integer constants*. Let $(x,y\in)$*Intv*, $(X,Y\in)$*Stmv*, $(Z\in)$*Cndv* be the (infinite, well-ordered) sets of *integer-*, *statement-*, and *condition variables*. Let $(s\in)$ *Iexp* be the set of *integer expressions* defined by

$$s ::= x \mid n \mid s_1 + s_2 \mid \underline{\text{if}}\ b\ \underline{\text{then}}\ s_1\ \underline{\text{else}}\ s_2\ \underline{\text{fi}}$$

Let $(b\in)$ *Bexp* be the set of *boolean expressions* defined by

$$b ::= \underline{\text{true}} \mid s_1 = s_2 \mid \neg b \mid b_1 \supset b_2$$

Let $(S\in)$ *Stat* be the set of *statements* defined by

$$S ::= x := s \mid b \mid S_1 ; S_2 \mid S_1 \cup S_2 \mid X\ \ \mu X[S]$$

Let $(p,q\in)$ *Cond* be the set of *conditions* defined by

$$p ::= \underline{\text{true}} \mid s_1 = s_2 \mid \neg p \mid p_1 \supset p_2 \mid \exists x[p] \mid S\{p\} \mid S\langle p\rangle \mid Z \mid \mu Z[p]$$

Let $(f\in)$ *Afor* be the set of *atomic formulae* defined by

$$f ::= p \mid S_1 \sqsubseteq S_2 \mid f_1 \wedge f_2$$

Let $(g\in)$ *Form* be the set of *formulae* defined by

$$g ::= f_1 \to f_2$$

2.2. FREE AND BOUND VARIABLES; SUBSTITUTION

The variables x, X and Z are *bound* in $\exists x[p]$, $\mu X[S]$ and $\mu Z[p]$ respectively. *intv*(s), *stmv*(S), *cndv*(f), etc., denote the sets of *free* integer-, statement-, and condition variables in s, S, f etc. Constructs which differ at most in their bound (integer, statement or condition) variables are called *congruent* (denoted by "$\cong$").

p[s/x] denotes the result of substituting s for (free occurrences of) x in p; similarly for S[S'/X] and p[q/Z]. The usual precautions to avoid clashes between free and bound variables apply.

2.3. REMARKS

2.3.1. Integer and boolean expressions are of no concern in our theory - as long as their evaluation always terminates - and are kept as simple as possible.

2.3.2. Let $S \equiv S(X)$. Then $\mu X[S(X)]$ corresponds to a call of the recursive procedure P declared by $P \Leftarrow S(P)$. The boolean expression b considered as a statement may be understood by the following correspondence with statements in more tradional syntaxes: <u>if</u> b <u>then</u> S_1 <u>else</u> S_2 <u>fi</u> $\sim b;S_1 \cup \neg b;S_2$, <u>while</u> b <u>do</u> S <u>od</u> $\sim \mu X[b;S;X \cup \neg b]$ ($X \notin \mathit{stmv}(S)$), and with Dijktra's "guarded commands" [D]: <u>if</u> $b_1 \rightarrow S_1 \square \ldots \square\, b_n \rightarrow S_n$ <u>fi</u> $\sim (b_1;S_1 \cup \ldots \cup b_n;S_n)$, <u>do</u> $b_1 \rightarrow S_1 \square \ldots \square\, b_n \rightarrow S_n$ <u>od</u> $\sim \mu X[(b_1;S_1 \cup \ldots \cup b_n;S_n);X \cup \neg b_1 \wedge \ldots \wedge \neg b_n]$ ($X \notin \mathit{stmv}(S_i), i=1,\ldots,n$).

2.3.3. S{p} and S<p> correspond to the *weakest precondition* for respectively *partial* and *total correctness* of S w.r.t. p.

2.3.4. In $\mu Z[p]$, p is assumed to be *syntactically monotonic* in Z, i.e., Z does not occur in p within the scope of an odd number of $\neg$-symbols (when $p_1 \supset p_2$ is rewritten as $\neg p_1 \vee p_2$). The construct $\mu Z[p]$ allows us to recursively define conditions, which then obtain meaning as the usual least fixed point of a suitable operator.

2.3.5. For $f_1 \rightarrow f_2$ cf. remark 3.6.7 below. A formula <u>true</u> $\rightarrow$ f will be abbreviated to f.

3. SEMANTICS

3.1. COMPLETE PARTIAL ORDERS AND COMPLETE LATTICES

A *complete partial order* or *cpo* $(x \in)C$ is a partially ordered set with a least element $\bot_C$ such that each (ascending) chain $\langle x_i \rangle_{i=0}^{\infty}$ has a lub $\bigsqcup_i x_i$. A *complete lattice* is a partially ordered set C in which *every* subset X has a lub $\bigsqcup X$ and (hence also) a glb $\bigsqcap X$; thus C is a cpo, with $\bot_c = \bigsqcap C$.

Let C_1 and C_2 be cpo's. A function $f: C_1 \rightarrow C_2$ is *strict* if $f(\bot_{C_1}) = \bot_{C_2}$, *monotonic* if $x_1 \sqsubseteq x_2 \Rightarrow f(x_1) \sqsubseteq f(x_2)$ and *continuous* if it is monotonic and also, for each chain $\langle x_i \rangle_i$ in C_1, $f(\bigsqcup_i x_i) \sqsubseteq \bigsqcup_i f(x_i)$ (or equivalently, $f(\bigsqcup_i x_i) = \bigsqcup_i f(x_i)$). If C_2 is a complete lattice, then $f: C_1 \to C_2$ is *anti-continuous* if for each chain $\langle x_i \rangle_i$ in C_1, $f(\bigsqcup_i x_i) = \sqcap_i f(x_i)$ (which implies that f is anti-monotonic, i.e. $x_1 \sqsubseteq x_2 \Rightarrow f(x_2) \sqsubseteq f(x_1)$).

The sets of all strict, monotonic and continuous functions from C_1 to C_2 are denoted, respectively by $C_1 \to_s C_2$, $C_1 \to_m C_2$ and $C_1 \to_c C_2$. These are all cpo's, when we define $f_1 \sqsubseteq f_2 \overset{df}{\Leftrightarrow} \forall x \in C_1 (f_1(x) \sqsubseteq f_2(x))$, and $\bot_{C_1 \to C_2} = \lambda x \in C_1 . \bot_{C_2}$.

A cpo C is *discrete* if for $x_1, x_2 \in C$, $x_1 \sqsubseteq x_2$ iff $x_1 = \bot_C$ or $x_1 = x_2$.

3.2. LEAST FIXED POINTS

If C is a cpo and $f: C \to_m C$ then the least fixed point of f, μf, may exist. If so, it is given by either of the formulas

$$\mu f = \sqcap \{x \mid f(x) = x\}$$

or

$$\mu f = \sqcap \{x \mid f(x) \sqsubseteq x\}.$$

The existence of μf is guaranteed by *either* of the following conditions:

(1) f is continuous,

(2) C is a complete lattice (Knaster-Tarski).

In the former case, μf is also given by the formula

$$\mu f = \bigsqcup_{i=0}^{\infty} f^i(\bot_C)$$

where $f^0(\bot_C) = \bot_C$ and $f^{i+1}(\bot_C) = f(f^i(\bot_C))$.

Two useful properties of the least fixed point (for monotonic f), to which we will refer later, are:

fpp ("fixed point property"): $f(\mu f) = \mu f$

lfp ("least fixed point"): $f(x) \sqsubseteq x \Rightarrow \mu f \sqsubseteq x$.

3.3. SOME SPECIFIC CPO'S

Let V_0 be the set of integers, and let $(\delta \in) W_0 = \{tt, ff\}$ be the set of truth-values. W_0 is a complete lattice, if we define $\bot_{W_0} = ff$. Let $(\alpha\in)V \overset{df}{=} V_0 \cup \{\bot_V\}$ and $(\beta\in)W \overset{df}{=} W_0 \cup \{\bot_W\}$. V and W are considered as discrete cpo's. (Note that $\bot_W \neq \bot_{W_0}$!)

For x_1, x_2 in a cpo C, let <u>if</u> β <u>then</u> x_1 <u>else</u> x_2 <u>fi</u> $\overset{df}{=}$ $<\bot_C$ if $\beta = \bot_W$, x_1 if $\beta = tt$, x_2 if $\beta = ff>$.

Let $(\sigma\in)\Sigma \overset{df}{=} (\mathit{Intv} \to V_0) \cup \{\bot_\Sigma\}$ be the set of *states*. Again, this is a discrete cpo. We will abbreviate $\bot_\Sigma$ to $\bot$. Let $T \overset{df}{=} \{\tau \subseteq \Sigma \mid \tau$ is finite or $\bot \in \tau\}$. T is a cpo, where we define (Egli-Milner) $\tau_1 \sqsubseteq \tau_2$ iff $<\bot \in \tau_1$ and $\tau_1 \backslash \{\bot\} \subseteq \tau_2$, or $\tau_1 = \tau_2>$, and $\bot_T \overset{df}{=} \{\bot\}$.

Let $(\phi\in)M \overset{df}{=} \Sigma \to T$, and $(\pi\in)\Pi \overset{df}{=} \Sigma \to_s W_0$. M is the set of (nondeterministic) *state transformations*, and Π is the set of *predicates* on Σ. Note that Π is a complete lattice (since W_0 is). Let $(\gamma\in)\Gamma \overset{df}{=} (\mathit{Stmv} \to M) \cup (\mathit{Cndv} \to \Pi)$.

Variants of states etc.: We define $\sigma\{\alpha/x\}$ to be the state σ' such that $\sigma' = \bot$ if $\sigma = \bot$, and otherwise $\sigma'(y) = <\sigma(y)$ if $y \not\equiv x$, α if $y \equiv x>$. $\gamma\{\phi/X\}$ and $\gamma\{\pi/Z\}$ are defined similarly.

3.4. COMPOSITION OF STATE TRANSFORMATIONS AND PREDICATES

3.4.1. We define:

a. $\phi_1 \circ \phi_2 \overset{df}{=} \lambda\sigma\cdot \bigcup\{\phi_1(\sigma') \mid \sigma' \in \phi_2(\sigma)\}$,

b. $\pi \circ \phi \overset{df}{=} \lambda\sigma\cdot \sqcap\{\pi(\sigma') \mid \sigma' \in \phi(\sigma)\}$, and

c. $\pi \,\square\, \phi \overset{df}{=} \lambda\sigma\cdot (\sigma \neq \bot \wedge \sqcap\{\pi(\sigma') \mid \sigma' \in \phi(\sigma)\backslash\{\bot\}\})$.

The first "∘" is used to define the meaning of $S_1;S_2$ (3.5c below), while the second "∘" and "□" are used to define the meanings of S<p> and S{p} respectively (3.5d).

3.4.2. *Remark*

"∘" (in both definitions) is monotonic and continuous in both arguments, while "□" is monotonic, but not continuous, in its first argument, and anti-continuous (and hence anti-monotonic) in its second.

3.5. DEFINITIONS

The functions $\mathcal{V}$: $\mathit{Iexp} \to (\Sigma\to V)$, $\mathcal{W}$: $\mathit{Bexp} \to (\Sigma\to W)$, $\mathcal{M}$: $\mathit{Stat} \to (\Gamma\to M)$, $\mathcal{T}$: $\mathit{Cond} \to (\Gamma\to\Pi)$, $\mathcal{F}$: $\mathit{Afor} \to (\Gamma\to\Pi)$ are defined by:

a. $V(s)(\bot) = \bot_V$, and, for $\sigma \neq \bot$, $V(x)(\sigma) = \sigma(x),\ldots,V(\underline{\text{if}}\ b\ \underline{\text{then}}\ s_1\ \underline{\text{else}}\ s_2\ \underline{\text{fi}})(\sigma) = \underline{\text{if}}\ W(b)(\sigma)\ \underline{\text{then}}\ V(s_1)(\sigma)\ \underline{\text{else}}\ V(s_2)(\sigma)\underline{\text{fi}}$

b. $W(b)(\bot) = \bot_W$, and, for $\sigma \neq \bot$, $W(\underline{\text{true}})(\sigma) = tt,\ldots,\ W(b_1 \supset b_2)(\sigma) = (W(b_1)(\sigma) \Rightarrow W(b_2)(\sigma))$

c. $M(x:=s)(\gamma) = \lambda\sigma\cdot\{\sigma\{V(s)(\sigma)/x\}\}$, $M(b)(\gamma) = \lambda\sigma\cdot\ \underline{\text{if}}\ W(b)(\sigma)\ \underline{\text{then}}\ \{\sigma\}\ \underline{\text{else}}\ \emptyset\ \underline{\text{fi}}$, $M(S_1;S_2)(\gamma) = M(S_2)(\gamma)\circ M(S_1)(\gamma)$, $M(S_1 \cup S_2)(\gamma) = M(S_1)(\gamma) \cup M(S_2)(\gamma)$, $M(X)(\gamma) = \gamma(X)$, $M(\mu X[S])(\gamma) = \mu[\lambda\phi\cdot M(S)(\gamma\{\phi/X\})]$.

d. $T(\underline{\text{true}})(\gamma) = \lambda\sigma\cdot(\sigma \neq \bot),\ldots,T(\exists x[p])(\gamma) = \lambda\sigma\cdot\exists\alpha[T(p)(\gamma)(\sigma\{\alpha/x\})],T(S\{p\})(\gamma) = T(p)(\gamma)\Box M(S)(\gamma),T(S\langle p\rangle)(\gamma) = T(p)(\gamma)\circ M(S)(\gamma),T(Z)(\gamma) = \gamma(Z)$, $T(\mu Z[p])(\gamma) = \mu[\lambda\pi\cdot T(p)(\gamma\{\pi/Z\})$.

e. $F(p)(\gamma) = T(p)(\gamma),F(S_1 \sqsubseteq S_2(\gamma) = \lambda\sigma\cdot((\sigma \neq \bot)\wedge(M(S_1)(\gamma)(\sigma) \sqsubseteq M(S_2)(\gamma)(\sigma)))$, $F(f_1 \wedge f_2)(\gamma) = F(f_1)(\gamma)\wedge F(f_2)(\gamma)$.

A formula $g \equiv f_1 \rightarrow f_2$ is called *valid* (denoted by $\models g$) if $\forall\gamma[\forall\sigma\neq\bot\ [F(f_1)(\gamma)(\sigma)] \Rightarrow \forall\sigma \neq \bot[F(f_2)(\gamma)(\sigma)]]$, and an *inference* $\frac{g_1,\ldots,g_n}{g}$ is called *sound* if $\models g_1,\ldots,\models g_n$ implies $\models g$.

3.6. REMARKS

3.6.1. $\Phi \overset{df}{=} \lambda\phi\cdot M(S)(\gamma\{\phi/X\}) \in M \rightarrow_c M$, $\Psi \overset{df}{=} \lambda\pi\cdot T(p)(\gamma\{\pi/Z\}) \in \Pi \rightarrow_m \Pi$, hence the least fixed points $\mu\Phi$, $\mu\Psi$ do exist (cf. parts d and e of definition 3.5).

3.6.2. $\models p \supset S\{q\}$ iff S is *partially correct* w.r.t. p,q (often written $\models \{p\}S\{q\}$). $\models p \supset S\langle q\rangle$ iff S is *totally correct* w.r.t. p,q (sometimes written $\models [p]S[q]$).

3.6.3. We have the familiar properties of $S\{q\}$: $\models (S_1;S_2)\{q\} = S_1\{S_2\{q\}\}$, $\models S\{q_1 \wedge q_2\} = S\{q_1\}\wedge S\{q_2\}$, $\models (S_1 \cup S_2)\{q\} = S_1\{q\}\wedge S_2\{q\}$, etc., and similarly for $S\langle q\rangle$.

3.6.4. $\models S\langle\underline{\text{true}}\rangle$ holds iff execution of S always terminates (i.e. $\bot \notin M(S)(\gamma)(\sigma)$ for all γ,σ).

3.6.5. Hence $\models S\langle p\rangle = S\langle\underline{\text{true}}\rangle \wedge S\{p\}$.

3.6.6. $S\langle p\rangle$ is monotonic in both S and p, but $S\{p\}$ is anti-monotonic in S (i.e., $\models (S_1 \sqsubseteq S_2) \rightarrow (S_2\{p\} \supset S_1\{p\})$). (Cf. 3.4.2.)

3.6.7. Observe that $\models f_1 \rightarrow f_2$ is a stronger fact than soundness of $\frac{f_1}{f_2}$. The meaning of the former is of the form $\forall\gamma[1\Rightarrow 2]$, of the latter $\forall\gamma[1] \Rightarrow \forall\gamma[2]$.

3.7. FIXED POINT PROPERTIES FOR STATEMENTS AND CONDITIONS

We re-state the fixed point properties given above (in 3.2).

fpp $\models \mu X[S] = S[\mu X[S]/X]$

lfp $\models (S[S_1/X] \sqsubseteq S_1) \Rightarrow (\mu X[S] \sqsubseteq S_1)$,

and similarly for $\mu Z[p]$.

3.8. CONTINUITY AND ANTI-CONTINUITY OF CONDITIONS; SCOTTS INDUCTION RULE

3.8.1. We say that p is *continuous* in X, or *anti-continuous* in X, if $\lambda\phi\cdot\tau(p)(\gamma\{\phi/X\})$ $(\in M \to \Pi)$ is continuous or anti-continuous respectively.

3.8.2. *Examples*. If X does not occur free in p or q, then (by 3.4.2) $\{X\}p$ is anti-continuous in X, $\langle X\rangle p$ is continuous in X and (hence) $(\langle X\rangle p) \supset q$ is anti-continuous in X.

3.8.3. Below (in 4.3) we will use the following version of Scott's induction rule: The inference

$$\frac{p[\Omega/X], (p \wedge (X \sqsubseteq \mu X[S])) \to p[S/X]}{p[\mu X[S]/X]}$$

is sound, provided p is anti-continuous in X.

4. TERMINATION

In this section we study the construct S<true>. By remark 3.6.4, we have that the validity of S<true> amounts to termination of S (for all γ,σ). We are now interested in a *syntactic* decomposition of S<true>, determined by the structure of S. More specifically, we want to define a *condition* $\tilde{S}$ by induction on the complexity of S, such that

(*) $\models \tilde{S} = S\langle \underline{\text{true}}\rangle$.

We will show how to define "~" by induction on the complexity of S, such that (*) is indeed satisfied. Now for $S \equiv X \in \mathit{Stmv}$, there is no possibility of syntactically reducing S, so we *extend* the class of conditions *Cond* with an additional clause $p::=\ldots|\tilde{X}$, and correspondingly extend the definition of T by: $T(\tilde{X})(\gamma)(\sigma) = (\perp \notin \gamma(X)(\sigma))$.

We first give the definition of $\tilde{S}$, and then an explanation of it. (A substitution of the form $p[q/\tilde{X}]$, occurring below, is defined in a natural way; e.g. $\tilde{Y}[q/\tilde{X}] = \langle q \text{ if } X \equiv Y, \tilde{Y} \text{ otherwise}\rangle$.)

4.1. DEFINITION

a. $(x:=s)^\sim \equiv \underline{true}$, $\tilde{b} \equiv \underline{true}$

b. $(S_1;S_2)^\sim \equiv \tilde{S}_1 \wedge S_1\{\tilde{S}_2\}$, $(S_1 \cup S_2)^\sim \equiv \tilde{S}_1 \wedge \tilde{S}_2$

c. $\mu X[S]^\sim \equiv \mu Z[\tilde{S}[\mu X[S]/X][Z/\tilde{X}]]$, where Z is (for definiteness) the first condition variable.

Note. One can verify that, for all X and S, $\tilde{S}$ is syntactically monotonic in $\tilde{X}$, and hence clause c is well-formed (cf. 2.3.4).

4.2. DISCUSSION OF THE ABOVE DEFINITION

We want to see that (*) holds for $\tilde{S}$ as defined above. This is given by theorem 4.3 below, but a few heuristic remarks on the definition should be helpful now.

Clauses a and b should be clear. (a) Since $x:=s$ and b always terminate, (*) holds for these two types of S. (b) We show that (*) is preserved for these cases: $\models (S_1;S_2)\langle\underline{true}\rangle = S_1\langle S_2\langle\underline{true}\rangle\rangle =$ (ind.hyp) $S_1\langle\tilde{S}_2\rangle =$ (by 3.6.5) $S_1\langle\underline{true}\rangle \wedge S_1\{\tilde{S}_2\} =$ (ind.hyp.) $\tilde{S}_1 \wedge S_1\{\tilde{S}_2\}$. Similarly for the case $S \equiv S_1 \cup S_2$.

Clause c deserves some explanation. We anticipate a result (step b in the course of proving theorem 4.3); viz., for each S and S_0,

$$(**) \qquad \models S[S_0/X]^\sim = \tilde{S}[S_0/X][\tilde{S}_0/\tilde{X}].$$

(A simpler guess for expressing $S[S_0/X]^\sim$ in terms of $\tilde{S}$ and $\tilde{S}_0$, namely $\models S[S_0/X]^\sim = \tilde{S}[\tilde{S}_0/\tilde{X}]$, can be seen to be false by considering e.g. the case $S \equiv X;S_1$ with $X \notin stmv(S_1)$.)

Now taking $S_0 \equiv \mu X[S]$ in (**), and applying *fpp* (3.7), we obtain

$$\models \underbrace{\mu X[S]^{\sim}} = \tilde{S}[\mu X[S]/X][\underbrace{\mu X[S]^{\sim}}/\tilde{X}].$$

Thus $\mu X[S]^{\sim}$ satisfies the above fixed point relationship, making plausible definition 4.1c (which gives it as the *least* such fixed point).

4.3. THEOREM. $\models \tilde{S} = S\langle\underline{true}\rangle$.

PROOF. The proof is fairly involved, and only sketched here. ($i \in \{1,\ldots,n\}$, $n \geq 0$).

a. $S \cong S' \Rightarrow \tilde{S} \cong \tilde{S}'$. This is shown by simultaneously proving, by induction on the complexity of S, that

(i) $S \cong S' \Rightarrow \tilde{S} \cong \tilde{S}'$

(ii) $S[X'/X]^{\sim} \cong \tilde{S}[X'/X][\tilde{X}'/\tilde{X}]$

b. $S[S_i/X_i]_i^{\sim} \cong \tilde{S}[S_i/X_i]_i[\tilde{S}_i/\tilde{X}_i]_i$

Induction on the complexity of S, using part a.

c. $\models \tilde{S}[S_i/X_i]_i[S_i\langle\underline{true}\rangle/\tilde{X}_i]_i \supset S[S_i/X_i]_i\ \langle\underline{true}\rangle$

(Taking n = 0, we infer that $\models \tilde{S} \supset S\langle\underline{true}\rangle$)

d. $\models (S_i' \sqsubseteq S_i'')_i \wedge (q_i' \supset q_i'')_i \rightarrow \tilde{S}[S_i'/X_i]_i[q_i'/\tilde{Y}_i]_i \supset \tilde{S}[S_i''/X_i]_i[q_i''/\tilde{Y}_i]_i$

I.e., $\tilde{S} \equiv \tilde{S}(X,\tilde{Y})$ is monotonic in both X and $\tilde{Y}$. Proved by induction on the complexity of S. The case $S \equiv S_1;S_2$ is not obvious, since then $\tilde{S} \equiv \tilde{S}_1 \wedge S_1\{\tilde{S}_2\}$, and $S_1\{\tilde{S}_2\}$ is not monotonic in S_1 (cf. 3.6.6). But here we use the equivalence $\models \tilde{S}_1 \wedge S_1\{\tilde{S}_2\} = \tilde{S}_1 \wedge S_1\langle\tilde{S}_2\rangle$, (from part c, with n = 0), and note that $S_1\langle\tilde{S}_2\rangle$ *is* monotonic in S_1.

e. $\models S\langle\underline{true}\rangle \supset \tilde{S}$. Induction on the complexity of S. If $S \equiv \mu X[S_0]$, apply Scott's induction rule (3.8.3) with $p \equiv (X\langle\underline{true}\rangle) \supset \mu X[S_0]$ (cf. 3.8.2), using the induction hypothesis and parts c,d.

5. DERIVATIVES

We will define the upper and lower derivatives of a statement S, and state a fundamental theorem connecting these two notions. Before giving the exact definitions, we make some introductory remarks.

The *upper derivative* of S w.r.t. X, written $\frac{dS}{dX}$, is an element of *Stat*, and has the following intended meaning: Dropping the γ-arguments for simplicity, we have that $\sigma' \in M(\frac{dS}{dX})(\sigma)$ iff execution of S for input state σ

leads to σ' as an intermediate state just before execution of X starts. E.g., if $S \equiv S_1;X;S_2;X;S_3 \cup S_4$, $X \notin stmv(S_i)$, $i = 1,\ldots,4$, then $\frac{dS}{dX} \equiv S_1 \cup S_1;X_1;S_2$. For statements without recursion, we may also briefly say that $\frac{dS}{dX}$ is the union of all prefixes of X in S.

Let $\mathcal{X} \subseteq \mathit{Stmv}$. The *lower derivative* of S w.r.t. $\mathcal{X}$, written $\delta_{\mathcal{X}}(S)$, is an element of *Cond*, and has the intended meaning: $\delta_{\mathcal{X}}(S)$ is true in a state whenever S terminates in σ *provided* that, for each $X \in \mathcal{X}$, execution of X for all states σ' in $M(\frac{dS}{dX})(\sigma)$ terminates. (Hence, $\delta_{\emptyset}(S) \equiv \tilde{S}$.)

(This is essentially the idea as introduced in [H,P] for statements without inner μ-terms. The novelty of our definition lies in clauses c of definitions 5.1 and 5.3.)

Combining the two intended meanings of $\frac{dS}{dX}$ and $\delta_{\mathcal{X}}(S)$, we expect that the following result holds: For each $X \notin \mathcal{X}$,

$$\models \delta_{\mathcal{X}}(S) = \frac{dS}{dX}\{\tilde{X}\} \wedge \delta_{\mathcal{X}\cup\{X\}}(S).$$

Let us give the verbal transliteration of this for the case that $\mathcal{X} = \emptyset$:

S terminates in σ iff both (i) and (ii) are satisfied:

(i) Execution of X terminates for all $\sigma'(\neq\bot)$ in $M(\frac{dS}{dX})(\sigma)$,

(ii) S terminates in σ *provided* execution of X for all $\sigma'(\neq\bot)$ in $M(\frac{dS}{dX})(\sigma)$ terminates.

(Note that a more naive equivalence: $\models \tilde{S} = \tilde{X} \wedge \delta_{\{X\}}(S)$ would not work, since termination of X is required for the wrong states.)

5.1. DEFINITION (upper derivative).

a. $$\frac{dx:=s}{dX} \equiv \underline{\text{false}},\quad \frac{db}{dX} \equiv \underline{\text{false}},\quad \frac{dY}{dX} \equiv \begin{cases}\underline{\text{true}}, & \text{if } X \equiv Y\\ \underline{\text{false}}, & \text{if } X \not\equiv Y\end{cases}$$

b. $$\frac{dS_1;S_2}{dX} \equiv \frac{dS_1}{dX} \cup S_1;\frac{dS_2}{dX},\quad \frac{d(S_1 \cup S_2)}{dX} \equiv \frac{dS_1}{dX} \cup \frac{dS_2}{dX}$$

c. $$\frac{d\mu Y[S]}{dX} \equiv \begin{cases}\underline{\text{false}}, & \text{if } X \equiv Y\\ \mu X_1[(\frac{dS}{dX} \cup \frac{dS}{dY};X_1)[\mu Y[S]/Y]], & \text{if } X \not\equiv Y, \text{ where } X_1\\ \text{is the first statement variable} \notin stmv(X,Y,S). & \end{cases}$$

5.2. REMARKS

5.2.1. By way of comment to clause 5.1c, we offer the following: We expect that (*): $\models \frac{dS_1[S_2/Y]}{dX} = \frac{dS_1}{dX}[S_2/Y] \cup \frac{dS_1}{dY}[S_2/Y];\frac{dS_2}{dX}$. In words (first forgetting about the substitutions on the right-hand side): Prefixes of X in $S_1[S_2/Y]$ are obtained either as prefixes of X in S_1, or by composing prefixes of Y in S_1 on the right with prefixes of X in S_2. Supplementing this description with the indicated substitutions then explains the plausibility of (*). Taking $S_1 \equiv S$, $S_2 \equiv \mu Y[S]$, and applying *fpp*, we obtain as property of $\frac{d\mu Y[S]}{dX}$: $\models \frac{d\mu Y[S]}{dx} = \frac{dS}{dX}[\mu Y[S]/Y] \cup \frac{dS}{dY}[\mu Y[S]/Y];\frac{d\mu Y[S]}{dX}$. We see that $\frac{d\mu Y[X]}{dX}$ satisfies a fixed point relationship, and, since our fixed points are usually *least*, one may now understand clause 5.1c.

5.2.2. If $X \notin stmv(S)$ then $\models \frac{dS}{dX} = \underline{false}$.

5.3. DEFINITION (lower derivative).

a. $\delta_{\mathcal{X}}(x:=s) \equiv \underline{true}$, $\delta_{\mathcal{X}}(b) \equiv \underline{true}$, $\delta_{\mathcal{X}}(X) \equiv \begin{cases}\underline{true}, & X \in \mathcal{X}\\ \tilde{X}, & X \notin \mathcal{X}\end{cases}$

b. $\delta_{\mathcal{X}}(S_1;S_2) \equiv \delta_{\mathcal{X}}(S_1) \wedge S_1\{\delta_{\mathcal{X}}(S_2)\}$, $\delta_{\mathcal{X}}(S_1 \cup S_2) \equiv \delta_{\mathcal{X}}(S_1) \wedge \delta_{\mathcal{X}}(S_2)$

c. $\delta_{\mathcal{X}}(\mu X[S]) \equiv \mu Z[\delta_{\mathcal{X}\setminus\{X\}}(S)[\mu X[S]/X][Z/\tilde{X}]]$, where Z is the first condition variable.

5.4. REMARKS

5.4.1. The definitions of $\delta_{\emptyset}(s)$ and $\tilde{S}$ (4.1) coincide.

5.4.2. $\delta_{\mathcal{X}}(S_1;S_2;\ldots;S_n) \equiv \delta_{\mathcal{X}}(S_1) \wedge S_1\{\delta_{\mathcal{X}}(S_2)\} \wedge S_1;S_2\{\delta_{\mathcal{X}}(S_3)\} \wedge \ldots$
$\ldots \wedge S_1;S_2;\ldots;S_{n-1}\{\delta_{\mathcal{X}}(S_n)\}$.

5.4.3. $X \notin stmv(S) \Rightarrow \delta_{\mathcal{X}}(S) \equiv \delta_{\mathcal{X}\setminus\{X\}}(S)$.

5.4.4. $\tilde{X}$ free in $\delta_{\mathcal{X}}(S) \Rightarrow X \in stmv(S)\setminus\mathcal{X}$.

5.5. THEOREM. *For* $X \notin \mathcal{X}$, $\models \delta_{\mathcal{X}}(S) = \frac{dS}{dX}\{\tilde{X}\} \wedge \delta_{\mathcal{X}\cup\{X\}}(S)$.

PROOF. Induction on the complexity of S. The only interesting case is that

$S \equiv \mu Y[S_0]$, $Y \not\equiv X$. We have to show that

$$\begin{aligned}&\mu Z[\delta_{X\setminus\{Y\}}(S_0)[S/Y][Z/\tilde{Y}]]\\ \models\ =\ &\\ &\mu X_1[(\frac{dS_0}{dX} \cup \frac{dS_0}{dY}\ ;\ X_1)[S/Y]]\{\tilde{X}\} \wedge \mu Z[\delta_{X\cup\{X\}\setminus\{Y\}}(S_0)[S/Y][Z/\tilde{Y}]].\end{aligned}$$

The proof - omitted here - involves fairly complicated manipulations in the μ-calculus, using *fpp* and *lfp* and properties of $S\{q\}$ (cf. 3.6.3).

5.6. COROLLARY. *For* $X \notin X$, $\models\ \delta_X(S) = \frac{dS}{dX} <\tilde{X}> \wedge\ \delta_{X\cup\{X\}}(S)$.

PROOF. It appears that, in the proof of theorem 5.5, $\{p\}$ may be replaced everywhere by $<p>$.

6. DERIVATIVES AND TERMINATION

We express termination of a recursive procedure $\mu X[S]$ in terms of the so-called *well-foundedness* of a function with respect to a predicate (involving $\frac{dS}{dX}$ and $\delta_{\{X\}}(S)$, respectively.)

6.1. DEFINITION. ϕ is called well-founded in σ w.r.t. π if

(i) There exists no infinite sequence $\sigma_0 = \sigma, \sigma_1, \ldots$, such that $\sigma_{i+1} \in \phi(\sigma_i)$, $i = 0,1,\ldots$

(ii) There exists no finite sequence $\sigma_0 = \sigma, \sigma_1, \ldots, \sigma_k$ such that $\sigma_{i+1} \in \phi(\sigma_i)$, $i = 0,\ldots,k$, $\sigma_k \neq \perp$, and $\pi(\sigma_k) = ff$.

6.2. REMARKS

6.2.1. By strictness, ϕ is not well-founded in $\perp$ w.r.t. any π.

6.2.2. If, for each $\sigma' \in \phi(\sigma)$, ϕ is well-founded in σ' w.r.t. π, and moreover, $\pi(\sigma) = tt$, then ϕ is well-founded in σ w.r.t. π.

6.3. LEMMA. *For each* ϕ,σ,π

a. $\mu[\lambda\pi'\cdot((\pi'\circ\phi)\wedge\pi)](\sigma) = tt \Rightarrow \phi$ *is well-founded in* σ *w.r.t.* π

b. ϕ *is well-founded in* σ *w.r.t.* $\pi \Rightarrow \mu[\lambda\pi'\cdot((\pi'\square\phi)\wedge\pi)](\sigma) = tt$.

PROOF.

a. Let $\pi_1 \overset{df}{=} \mu[\lambda\pi'\cdot((\pi'\circ\phi') \wedge \pi)]$, and let $\pi_{\phi,\pi}$ denote the predicate which, for each σ, expresses that ϕ is well-founded in σ w.r.t. π. We show that $\pi_1 \sqsubseteq \pi_{\phi,\pi}$, or, by *lfp*, that $(\pi_{\phi,\pi}\circ\phi) \wedge \pi \sqsubseteq \pi_{\phi,\pi}$. Now this is immediate by 6.2.2.

b. Let $\pi_2 \overset{df}{=} \mu[\lambda\pi'\cdot((\pi'\square\phi) \wedge \pi)]$. Assume that ϕ is well-founded in σ w.r.t. π, but $\pi_2(\sigma) = ff$. Clearly, $\sigma \neq \bot$. By *fpp*, then $((\pi_2\square\phi) \wedge \pi)(\sigma) = ff$. Thus, either $\pi(\sigma) = ff$, contradicting definition 6.1 (ii), or there exists $\sigma' \in \phi(\sigma)$, $\sigma' \neq \bot$, such that $\pi_2(\sigma') = ff$. Thus, again by *fpp*, either $\pi(\sigma') = ff$, contradicting 6.1 (ii), or we obtain $\sigma'' \neq \bot$ such that $\sigma'' \in \phi(\sigma')$ and $\pi_2(\sigma'') = ff$. Repeating the argument, either we find a finite sequence $\sigma_0 = \sigma,\ldots,\sigma_k$ $(k\geq 0)$ such that $\sigma_{i+1} \in \phi(\sigma_i)$, $i = 0,\ldots,k-1$, $\sigma_k \neq \bot$, and $\pi(\sigma_k) = ff$, or we obtain an infinite sequence $\sigma_0 = \sigma,\sigma_1,\sigma_2,\ldots$, such that $\sigma_{i+1} \in \phi(\sigma_i)$, $i = 0,1,\ldots$. In both cases, we have found a contradiction.

6.4. DEFINITION. S is called well-founded w.r.t. p if for all γ,σ, $M(s)(\gamma)$ is well-founded in σ w.r.t. $T(p)(\gamma)$.

6.5. COROLLARY.

a. $\models \mu Z[S\langle Z\rangle \wedge p] \Rightarrow$ S *is well-founded w.r.t.* p

b. S *is well-founded w.r.t.* p $\Rightarrow\ \models \mu Z[S\{Z\} \wedge p]$.

6.6. DEFINITION. $\mathring{S} \equiv (\frac{dS}{dX})[\mu X[S]/X]$,

$$\underset{\circ}{S} \equiv \delta_{\{X\}}(S)[\mu X[S]/X].$$

We now come to main theorem of the paper (an intuitive explanation of which is given afterwards).

6.7. THEOREM. *The following two facts are equivalent:*

a. $\models \mu X[S]$ <u><true></u>

b. $\mathring{S}$ *is well-founded w.r.t.* $\underset{\circ}{S}$.

PROOF. We have successively:

a. $\models \tilde{S} = \frac{dS}{dX}\{\tilde{X}\} \wedge \delta_{\{X\}}(S)$ (by 5.5 and 5.4.1)

b. $\models \tilde{S}[\mu X[S]/X] = \mathring{S}\{\tilde{X}\} \wedge \underset{\circ}{S}$ (subst. $\mu X[S]$ for X)

c. $\models \tilde{S}[\mu X[S]/X][Z/\tilde{X}] = \mathring{S}\{Z\} \wedge \underset{\circ}{S}$ (subst. Z for $\tilde{X}$)

d. $\models \mu Z[\tilde{S}[\mu X[S]/X][Z/\tilde{X}]] = \mu Z[\mathring{S}\{Z\} \wedge \underset{\circ}{S}]$ (prefixing μZ)

e. $\models \mu X[S]$ <u><true></u> $= \mu Z[\mathring{S}\{Z\} \wedge \underset{\circ}{S}]$ (4.1, 4.3)

f. $\models \mu X[S]$ <u><true></u> $= \mu Z[\mathring{S}\langle Z\rangle \wedge \underset{\circ}{S}]$ (as a-e, starting from 5.6).

(Note: in c, we use that $\tilde{X}$ is not free in $\delta_{\{X\}}(S)$ by 5.4.4, hence also not in $\underset{\circ}{S}$.)

The theorem now follows from e,f and corollary 6.5.

6.8. DISCUSSION

We have derived the following result: A recursive procedure $\mu X[S]$ terminates for all input states $\neq \bot$ iff $\mathring{S}$ is well-founded w.r.t. $\underset{\circ}{S}$. How should one understand this proposition? Let us consider e.g. the procedure $\mu \overset{df}{\equiv} \mu X[S]$, where $S \equiv S_1;X;S_2;X;S_3 \cup S_4$, with $X \notin$ *stmv*(S_i), $i = 1,\ldots,k$. Then $\models \mathring{S} = S_1 \cup S_1;\mu;S_2$ (using 5.2.2). Also $\models \delta_{\{X\}}(S) = \tilde{S}_1 \wedge S_1;X\ \{\tilde{S}_2\} \wedge S_1;X;S_2;X\ \{\tilde{S}_3\} \wedge \tilde{S}_4$ (using 5.4.2, 5.4.3, 5.4.1), and so $\models \underset{\circ}{S} = \tilde{S}_1 \wedge S_1;\mu\{\tilde{S}_2\} \wedge S_1;\mu;S_2;\mu\{\tilde{S}_3\} \wedge \tilde{S}_4$. Forgetting about the γ-arguments, we have that for all σ:

a. There exists no infinite sequence $\sigma_0 = \sigma,\sigma_1,\ldots$, such that $\sigma_{i+1} \in M(S_1 \cup S_1;\mu;S_2)(\sigma_i)$, $i = 0,1,\ldots$. Since $\mathring{S}$ is nothing but the statement executed between a call of μ at a certain level of recursion depth, and a call at the next deeper level, we see that the non-existence of such an infinite sequence amounts to the absence of infinite recursion, i.e., it is not possible that the procedure goes on calling itself indefinitely.

b. There exists no finite sequence $\sigma_0 = \sigma,\ldots,\sigma_k$, such that $\sigma_{i+1} \in M(\mathring{S})(\sigma_i)$, $i = 0,\ldots,k-1$, $\sigma_k \neq \bot$, and $T(\underset{\circ}{S})(\sigma_k) = ff$. Assume that, contrariwise, such a sequence would exist. This would mean that, at a certain level of recursion depth, we have obtained an intermediate state $\sigma_k \neq \bot$ such that $T(\underset{\circ}{S})(\sigma_k) = ff$. By the definition of $\underset{\circ}{S}$ this means that either

(i) S_1 does not terminate in σ_k, or

(ii) There exists some $\sigma' \neq \bot$ such that $\sigma' \in M(S_1;\mu)(\sigma_k)$ and S_2 does not terminate in σ', or

(iii) There exists some $\sigma'' \neq \bot$ such that $\sigma'' \in M(S_1;\mu;S_2;\mu)(\sigma_k)$ and S_3 does not terminate in σ'', or

(iv) S_4 does not terminate in σ_k.
Altogether, we see that $\underset{\circ}{S}$ is false in $\sigma_k \neq \perp$ precisely when there is some instance of *local* nontermination stemming from σ_k, i.e., nontermination which is not due to infinite recursion of μ, but to nontermination of one of the S_i-components of μ.

Combining results a and b, we see that $\mu X[S]$ terminates everywhere whenever, for all σ, there is neither the possibility of infinite recursion (global nontermination), nor the possibility of the computation reaching some intermediate state which leads to local nontermination.

References

[dB1] DE BAKKER, J.W., *Semantics and termination of nondeterministic recursive programs,* in Proc. 3rd Coll. Automata, Languages and Programming (S. Michaelson & R. Milner, eds), pp.435-477, Edinburgh University Press (1976).

[dB2] DE BAKKER, J.W., *Mathematical theory of program correctness.* To appear.

[D] DIJKSTRA, E.W., *A Dicipline of Programming,* Prentice-Hall (1976).

[H,P] HITCHCOCK, P. & D.M.R. PARK, *Induction rules and proofs of termination,* in Proc. 1st Coll. Automata, Languages and Programming (M. Nivat, ed.), pp.225-251, North-Holland (1973).