Processes and formalisms for unbounded choice

Wim H. Hesselink

Rijksuniversiteit Groningen, Department of Computing Science, P.O. Box 800, 9700 AV Groningen, Netherlands

Communicated by J.W. de Bakker
Received November 1989
Revised June 1990

Abstract

Hesselink, W.H., Processes and formalisms for unbounded choice, Theoretical Computer Science 99 (1992) 105–119.

In the field of program refinement a specification construct has been proposed that does not have a standard operational interpretation. Its weakest preconditions are monotone but not necessarily conjunctive. In order to develop a corresponding calculus we introduce specification algebras. These algebras may have two choice operators: demonic choice and angelic choice. The wish to allow unbounded choice, of both modalities, leads to the question of defining and constructing completions of specification algebras. It is shown that, in general, a specification algebra need not have a completion. On the other hand, a formalism is developed that allows for any specific combination of unbounded demonic choice, unbounded angelic choice and sequential composition. The formalism is based on transition systems. It is related to the processes of De Bakker and Zucker.

0. Introduction

In the field of program refinement, several authors (cf. [2, 9]) have proposed a specification construct that represents an angelic choice between an unbounded set of commands. We use the notation $(\Diamond i :: C.i)$ for the angelic choice between commands $C.i$. It is specified by its weakest precondition for postcondition P,

$$wp.(\Diamond i :: C.i).P = (\exists i :: wp.(C.i).P).$$

On the other hand, the demonic choice $([]i :: C.i)$ is given by

$$wp.([]i :: C.i).P = (\forall i :: wp.(C.i).P).$$

In the case of a choice between two commands, the symbols "$\Diamond$" and "$[]$" are used as infix operators, so that

$$wp.(C \Diamond D).P = wp.C.P \vee wp.D.P,$$

$$wp.(C\ []\ D).P = wp.C.P \wedge wp.D.P.$$

The angelic choice has rather strange properties. If, for example, i is an integer program variable, then command $C=(i:=0 \diamond i:=1)$ is easily seen to satisfy both

$$wp.C.(i=0)=true, \quad \text{and}$$

$$wp.C.(i=1)=true, \quad \text{although}$$

$$wp.C.(i=0 \wedge i=1)=false.$$

So the command seems to guess the postcondition. For this reason we prefer to speak of specification C instead of command C. The angelic properties of specification C prohibit an operational model in the usual style. In this special case, $wp.C$ can be expressed by means of the diamond operator of dynamic logic, cf. [4], but that formalism cannot express mixtures of demonic choice and angelic choice. In [2], a general specification C is regarded as a game between a demon and an angel: predicate $wp.C.P$ means that the angel has a winning strategy to establish postcondition P. We refer to [8] for a more extensive discussion.

For a general specification C, the predicate transformer $wp.C$ is montone, but it may fail to be conjunctive (as is shown in the above example).

The aim of refinement calculus in the sense of [2, 9, 10] is to calculate with specifications that satisfy some algebraic laws. Therefore, we would like to consider specifications as forming a certain algebra, which is an abstraction of the algebra of the monotone predicate transformers. This algebra should have three operators: composition ";", demonic choice "$[\!]$", and angelic choice "$\diamond$".

In this paper, we want to point out an obstruction in combining the composition with the unbounded demonic choice in the presence of angelic commands. We define a concept of a *specification algebra*, with operators ";" and "$[\!]$". A specification algebra is called *complete* if it allows unbounded demonic choice. We define the concept of a *completion* of a specification algebra. The problem is that a simple specification algebra may fail to have a completion; or rather, the "completion" is too big to be a set, it is a proper class in the sense of set theory, cf. [11]. As a remedy, we introduce transition systems with termination and both angelic and demonic choice to represent the elements of the "completion".

The motivation for the paper came from refinement calculus in sequential programming. The proposed transition systems and their wp-interpretation, however, may be useful for concurrency as well.

In each section, the formulae are numbered consecutively. For reference to formulae of other sections, we use $i(j)$ to denote formula (j) of Section i.

1. Process algebras and specification algebras

We start with the (basic) process algebras of Bergstra et al., cf. [1]. A *process algebra* is defined to be a triple $\langle A, [\!], ;\rangle$, where A is a set and "$[\!]$" and ";" are binary operators on A. The operator "$[\!]$" stands for demonic nondeterminate choice. So its interpretation differs from the interpretation of the operator "+" of [1]. The operator

";" stands for sequential composition. The following axioms are postulated.

(0) $a \mathbin{[\,]} a = a,$

(1) $a \mathbin{[\,]} b = b \mathbin{[\,]} a,$

(2) $(a \mathbin{[\,]} b) \mathbin{[\,]} c = a \mathbin{[\,]} (b \mathbin{[\,]} c),$

(3) $(a;\, b);\, c = a;\, (b;\, c),$

(4) $(a \mathbin{[\,]} b);\, c = a;\, c \mathbin{[\,]} b;\, c.$

In (4), and henceforth, we give the operator ";" a higher priority than "[]". In fact, ";" is regarded as a multiplication and "[]" is regarded as an addition operator. If no ambiguity can arise, we speak of the process algebra A instead of $\langle A, [\,], ;\rangle$.

On a process algebra A we define the binary relation $\leqslant$ by

(5) $a \leqslant b \;\equiv\; a = a \mathbin{[\,]} b.$

As is well known (and easily verified), axioms (0), (1), (2) imply that $\leqslant$ is a (partial) order on A. Relation $\leqslant$ is called the *order of determinacy*. In fact, if $a \leqslant b$, cf. (5), then b is a possible choice for a; in other words, a is less determinate than b.

One easily verifies that $a \mathbin{[\,]} b$ is the greatest lower bound of a and b in the ordered set $(A, \leqslant)$, and that

(6) $a \leqslant b \wedge c \leqslant d \;\Rightarrow\; a \mathbin{[\,]} c \leqslant b \mathbin{[\,]} d.$

The composition operator ";" is monotone with respect to its first argument:

(7) $a \leqslant b \;\Rightarrow\; a;\, c \leqslant b;\, c.$

This is proved in

$$\begin{aligned}
& a;\, c \leqslant b;\, c \\
\equiv\ & \{(5)\}\ a;\, c = a;\, c \mathbin{[\,]} b;\, c \\
\equiv\ & \{(4)\}\ a;\, c = (a \mathbin{[\,]} b);\, c \\
\Leftarrow\ & \{(5)\}\ a \leqslant b.
\end{aligned}$$

A *specification algebra* is defined to be a process algebra in which the composition operator ";" is also monotone with respect to its second argument:

(8) $b \leqslant c \;\Rightarrow\; a;\, b \leqslant a;\, c.$

Postulate (8) can be replaced by the equivalent axiom

(9) $a;\, (b \mathbin{[\,]} c) = a;\, (b \mathbin{[\,]} c) \mathbin{[\,]} a;\, c.$

In fact, the equivalence is proved in

$$\begin{aligned}
& (\forall b :: b \leqslant c \;\Rightarrow\; a;\, b \leqslant a;\, c) \\
\equiv\ & \{(5)\} \\
& (\forall b :: b = b \mathbin{[\,]} c \;\Rightarrow\; a;\, b = a;\, b \mathbin{[\,]} a;\, c) \\
\equiv\ & \{\text{take } b := b \mathbin{[\,]} c \text{ and use } b \mathbin{[\,]} c = b \mathbin{[\,]} (b \mathbin{[\,]} c)\} \\
& (\forall b :: a;\, (b \mathbin{[\,]} c) = a;\, (b \mathbin{[\,]} c) \mathbin{[\,]} a;\, c).
\end{aligned}$$

Notice that we do not postulate $a;\ (b \,[]\, c) = a;\ b \,[]\, a;\ c$, see formula (8) in Section 2 below.

A function $w: A \to B$ between specification algebras A and B is called a *homomorphism* if and only if it satisfies

(10) $\quad w.(p \,[]\, q) = w.p \,[]\, w.q, \qquad w.(p;\ q) = w.p;\ w.q.$

For any nonempty finite subset E of A, the repeated choice $([]x \in E :: x)$ can be defined in the obvious way. By axiom (4), the nonempty finite repeated choice satisfies the left-$[]$-distributive law

(11) $\quad ([]x \in E :: x);\ y = ([]x \in E :: x;\ y).$

One can verify that $([]x \in E :: x)$ is the greatest lower bound of set E in A with respect to the order of determinacy (5).

2. Complete specification algebras

The specification algebra A is called *complete* if and only if *every subset* E has a greatest lower bound. In a complete specification algebra we use the notation $([]x \in E :: x)$ to denote the greatest lower bound of an arbitrary subset E.

A homomorphism $w: A \to B$ between complete specification algebras A and B is called $[]$-*complete* if and only if for every subset E of A

(0) $\quad w.([]x \in E :: x) = ([]x \in E :: w.x).$

We do not go into the axiomatisation of angelic choice for arbitrary specification algebras. Let M be a complete specification algebra. Since it has arbitrary greatest lower bounds, it also has arbitrary least upper bounds: the least upper bound of a subset E of M is (as is well known, and easily verified) the greatest lower bound of its upper bounds. Using the symbol $\Diamond$ for the least upper bound, we have

$$(\Diamond x \in E :: x) = ([]y \in M : (\forall x \in E :: x \leq y) : y).$$

Notice that M has a biggest element $\top$ and a smallest element $\bot$ given by

$$\top = ([]x \in \emptyset :: x),$$

$$\bot = (\Diamond x \in \emptyset :: x) = ([]y \in M :: y).$$

The complete specification algebra M is called *completely left-distributive* if and only if for *every* subset E of M and every element $y \in M$

(1) $\quad ([]x \in E :: x);\ y = ([]x \in E :: x;\ y), \quad$ and

$\qquad (\Diamond x \in E :: x);\ y = (\Diamond x \in E :: x;\ y).$

Notice that, in the analogous formula 1(11), the set E is supposed to be nonempty and finite. Notice also that formula (1) with $E=\emptyset$ implies

(2) $\top; y=\top, \qquad \bot; y=\bot \qquad$ for all $y\in M$.

The set of the monotone predicate transformers is our prototype of a complete and completely left-distributive specification algebra. This algebra is constructed as follows: Let $\mathbb{B}$ be the set of the two boolean values *true* and *false*. Let S be a set, the so-called state space. We write $\mathbb{B}^S$ to denote the set of the boolean functions on S. By abuse of language, the elements of $\mathbb{B}^S$ are called *predicates*, even though we do not impose any syntax. We write

(3) $[p\Rightarrow q] \equiv (\forall s\in S :: p.s\Rightarrow q.s)$.

A function $f:\mathbb{B}^S\to\mathbb{B}^S$ is called a *predicate transformer*. It is called *monotone* if and only if

(4) $(\forall p, q\in\mathbb{B}^S : [p\Rightarrow q] : [f.p\Rightarrow f.q])$.

Let MT be the set of the monotone predicate transformers. It is made into a specification algebra by defining

(5) $(f \,[]\, g).p.s \equiv f.p.s \wedge g.p.s, \qquad (f;\, g).p = f.(g.p)$

for all $f,\ g\in MT,\ p\in\mathbb{B}^S,\ s\in S$. One can easily verify that $\langle MT, [], ;\rangle$ is a process algebra. Relation 1(5) specialises to

(6) $f\leqslant g \equiv (\forall p\in\mathbb{B}^S :: [f.p\Rightarrow g.p])$.

Now it follows from (4), (5), (6), that for all $f,\ g,\ h\in MT$

$$f\leqslant g \Rightarrow h; f\leqslant h;\, g,$$

so that MT is a specification algebra indeed. Actually, MT is complete and completely left-distributive. For a subset E of MT, the predicate transformers $([]f\in E :: f)$ and $(\Diamond f\in E :: f)$ are given by

(7) $([]f\in E :: f).p.s \equiv (\forall f\in E :: f.p.s)$, and

$(\Diamond f\in E :: f).p.s \equiv (\exists f\in E :: f.p.s)$.

The remaining proofs are left to the reader.

Remark. In [7], we define a *command algebra* to be a process algebra that satisfies the right-distributive law for "[]":

(8) $a; (b \,[]\, c) = a;\, b \,[]\, a;\, c.$

It is easy to see that every command algebra is a specification algebra. The algebra MT is not a command algebra, as is shown in the following example.

Let S be the set of the integers and let variable i range over S. We consider the specifications

$$a=(i:=0 \Diamond i:=1), \qquad b=(i:=i+1) \quad \text{and} \quad c=skip,$$

and the postcondition $P:(i=1)$. One easily verifies that

$$\begin{aligned} & wp.(a;\,(b \,[]\, c)).P \\ = \; & wp.a.(wp.(b \,[]\, c).(i=1)) \\ = \; & wp.a.\mathit{false} \\ = \; & \mathit{false} \end{aligned}$$

whereas

$$\begin{aligned} & wp.(a;\,b \,[]\, a;\,c).P \\ = \; & wp.(a;\,b).(i=1) \wedge wp.(a;\,c).(i=1) \\ = \; & wp.a.(i=0) \wedge wp.a.(i=1) \\ = \; & \mathit{true}. \end{aligned}$$

Thus, if we identify *a*, *b*, *c* with their predicate transformers *wp.a*, etc., in *MT*, then

$$a;\,(b \,[]\, c) \neq a;\,b \,[]\, a;\,c \text{ in } MT.$$

Similarly, the right-distributive law for $\diamond$ is not valid in *MT*, as is shown by taking *P*, *b*, *c* as above and

$$d = (i := 0 \,[]\, i := 1),$$

in which case one can verify that

$$\begin{aligned} & wp.(d;\,(b \diamond c)).P = \mathit{true}, \\ & wp.(d;\,b \diamond d;\,c).P = \mathit{false}. \end{aligned}$$

The failure of right distributivity suggests some connection with the semantics of process algebras, cf. [1, p. 132].

3. Completion of specification algebras

Let a specification algebra A be given. Since A need not allow infinite choice, we want to embed A into a complete specification algebra A^{c}. Moreover, we expect that any semantic interpretation of A has some (preferably unique) extension to A^{c}. Therefore, we impose the condition that every homomorphism $w: A \to M$ of A to a complete (and possibly completely left-distributive) specification algebra M has some extension to a $[]$-complete homomorphism $w^{c}: A^{c} \to M$. Such an algebra A^{c} would be called a *completion* of A.

We now show that not every specification algebra has a completion. Let B be the free specification algebra over a single indeterminate t. It consists of all specification algebra expressions in t, modulo the equalities induced by the axioms 1(0)–1(4) and 1(9). It is characterised by the "universal" property that for every specification algebra M and every $m \in M$ there is precisely one homomorphism of specification algebras $wp: B \to M$ with $wp.t = m$.

Here, the name "*wp*" is chosen for the analogy with weakest preconditions: if B is the programming language, M is an algebra of predicate transformers, t is a command specified by $wp.t = m$, then indeed function "*wp*" gives the weakest preconditions of the commands in B.

We shall prove that B does not have a completion B^c. To do so, we assume that B^c exists and we prove that the cardinality of B^c exceeds the cardinality of an arbitrary ordinal number, cf. [11].

Let λ be an arbitrary ordinal number. As is usual, λ is identified with the set of the ordinals $\sigma < \lambda$. This set is treated as the state space of our programs. Therefore, the set of the "predicates" is $\mathbb{B}^\lambda$, the set of the boolean functions on λ. Let MT be the set of the monotone functions $\mathbb{B}^\lambda \to \mathbb{B}^\lambda$, i.e. the algebra of the monotone predicate transformers of λ. It is a complete and completely left-distributive specification algebra. We define $m \in MT$ by

(0) $\quad m.p.\sigma \;\equiv\; (\exists \mu \in \lambda : \mu < \sigma : p.\mu)$

for any predicate $p \in \mathbb{B}^\lambda$ and any state $\sigma \in \lambda$. The predicate transformer m represents the unbounded angelic choice of a smaller ordinal. Notice that m is monotone, as required. By the universal property of B, there is a unique homomorphism of specification algebras $wp : B \to MT$ with $wp.t = m$. By the defining property of B^c, there is a $[]$-complete homomorphism $wp^c : B^c \to MT$ that extends wp. We claim that for every ordinal $\nu \in \lambda$ there is a specification $e.\nu \in B^c$ with

(1) $\quad wp^c.(e.\nu).true.\sigma \;\equiv\; \nu < \sigma.$

The specifications $e.\nu$ are constructed by transfinite induction. Let $\xi \in \lambda$ be an ordinal such that $e.\nu$ is constructed for all $\nu < \xi$. Then we define

(2) $\quad e.\xi = t; ([]\nu : \nu < \xi : e.\nu).$

We verify that for any σ

$$\begin{aligned}
& wp^c.(e.\xi).true.\sigma \\
\equiv\; & \{(2),\ wp^c \text{ homomorphism}\} \\
& (wp^c.t;\ wp^c.([]\nu : \nu < \xi : e.\nu)).true.\sigma \\
\equiv\; & \{wp^c.t = m \text{ and ``;'' in } MT \text{ is composition}\} \\
& m.(wp^c.([]\nu : \nu < \xi : e.\nu).true).\sigma \\
\equiv\; & \{(0)\} \\
& (\exists \mu : \mu < \sigma : wp^c.([]\nu : \nu < \xi : e.\nu).true.\mu) \\
\equiv\; & \{(1),\ wp^c \text{ is } []\text{-complete}\} \\
& (\exists \mu : \mu < \sigma : (\forall \nu : \nu < \xi : \nu < \mu)) \\
\equiv\; & \{\text{ordinal calculus}\} \\
& (\exists \mu : \mu < \sigma : \xi \leq \mu) \\
\equiv\; & \{\text{ordinal calculus}\} \\
& \xi < \sigma.
\end{aligned}$$

This implies that $e.\xi$ satisfies the induction hypothesis (1), thus concluding the inductive construction (one may notice that the induction starts with $e.0 = (t; \top)$). It follows from (1) that for any $\mu, \nu \in \lambda$

$$\mu \neq \nu \;\Rightarrow\; e.\mu \neq e.\nu.$$

This proves that the cardinality of B^c is at least the cardinality of λ. As λ was arbitrary, this proves that B^c is not a set, cf. [11], though it may exist as proper class.

We suspect that a similar argument can be given if the monotone predicate transformers are replaced by the conjunctive ones, but we know that in the case of the *positively* conjunctive predicate transformers a completion exists. Here, a predicate transformer f is called conjunctive if and only if it satisfies $f.(p \wedge q) = f.p \wedge f.q$. It is called positively conjunctive if and only if $f.(\bigwedge p \in E :: p) = (\bigwedge p \in E :: f.p))$ for any nonempty set of predicates E.

4. Transition systems as specifications

Now that we know that the completion A^c cannot be constructed as a set, we may try and construct A^c as a class. So the aim is to construct objects that represent elements of A^c. We call our constructs "D-specifications". In a modification of the construction the objects are called "AD-specifications" and both angelic and demonic choice are admitted. A D-specification is a kind of formal expression with unbounded choice. It is formalised as a transition system with termination point, i.e. a kind of nondeterministic, not necessarily finite, automaton. This formalisation was inspired by [3] and [5].

We need two auxiliary symbols, a symbol $\tau \notin A$ to denote a silent action and a symbol ε to denote a termination state. We write $A^\tau = A \cup \{\tau\}$ and $P^\varepsilon = P \cup \{\varepsilon\}$.

A *D-specification* over a set A is a triple $\langle P, p, \pi\rangle$, where P is a set with $\varepsilon \notin P$, and $p \in P$ is a constant, and $\pi \subset P \times A^\tau \times P^\varepsilon$ is a ternary relation such that

(0) $\langle x, a, y\rangle \in \pi \;\Rightarrow\; \neg(a = \tau \wedge y = \varepsilon)$.

If no ambiguity can arise, we speak of D-specification P instead of $\langle P, p, \pi\rangle$.

The intutition is that $x \in P$ corresponds to the demonic choice between composition $(a; y)$ with $\langle x, a, y\rangle \in \pi$. The distinguished element p serves as the initial point which represents the whole D-specification. A triple $\langle x, a, \varepsilon\rangle \in \pi$ corresponds to the call of a followed by termination. We introduce the concept of extension to formalise this intuition.

Let P be a D-specification over a set A and let $f : A \to M$ be a function from A to a complete (and preferably completely left-distributive) specification algebra M. A function $w : P \to M$ is called an *extension* of f to P if and only if for every $x \in P$

```
(1)    $w.x = ([\!]a, y : \langle x, a, y\rangle \in \pi :$
           **if** $a \neq \tau \wedge y \neq \varepsilon$ **then** $f.a;\ w.y$
           $[\!]a = \tau$ **then** $w.y$
           $[\!]y = \varepsilon$ **then** $f.a$ **fi**).
```

Remark. The righthand side of (1) is well defined because of condition (0). In the absence of (0), equation (1) would need an extra clause

$$[]a = \tau \wedge y = \varepsilon \textbf{ then } skip,$$

where *skip* would have to be a unit element of the composition of M. Since we do not want to postulate the existence of *skip* in every specification algebra, we impose condition (0).

It is a straightforward application of the theorem of Knaster–Tarski to show that every function $f: A \to M$ has a unique smallest extension and a unique biggest extension. In fact, let M^P be the set of the functions $P \to M$. An element $w \in M^P$ is an extension if and only if it is a fixpoint of the function $E: M^P \to M^P$ given by

(2) $E.w.x = ([]a, y : \langle x, a, y\rangle \in \pi :$

$\quad \textbf{if } a \neq \tau \wedge y \neq \varepsilon \textbf{ then } f.a;\ w.y$

$\quad []a = \tau \textbf{ then } w.y$

$\quad []y = \varepsilon \textbf{ then } f.a \textbf{ fi})$

for all $w \in M^P$ and $x \in P$. It follows from the formulae 1(8) and 1(6) that function E is monotone with respect to the induced order of M^P. Therefore, by the theorem of Knaster–Tarski, cf. [12], function E has a smallest and a biggest fixed point.

Let f^P and f_P be the biggest and smallest extension of f to P. The meaning of D-specification P itself is determined by the elements

(3) $f^c.P = f^P.p, \qquad f_c.P = f_P.p \qquad (\in M).$

D-specifications are most useful if the extension is necessarily unique. This leads to the following considerations. For a D-specification $P = \langle P, p, \pi\rangle$, *the binary relation* "$\lhd$" on P is defined by

(4) $y \lhd x \ \equiv\ (\exists a \in A^\tau :: \langle x, a, y\rangle \in \pi).$

The D-specification P is called *well-founded* if and only if the relation "$\lhd$" is well-founded, i.e. for every subset U of P

(5) $U = P \ \equiv\ (\forall x \in P :: (\forall y \in P : y \lhd x : y \in U) \Rightarrow x \in U).$

For a well-founded D-specification P every function $f: A \to M$ has precisely one extension. This is proved by the following standard argument. Let v and w be extensions of f. Let U be defined as the subset of P where v and w are equal. For any $x \in P$ we have

$$\begin{aligned}
&(\forall y \in P : y \lhd x : y \in U)\\
\equiv\ &\{(4) \text{ and definition } U\}\\
&(\forall y \in P,\ a \in A^{\tau} : \langle x, a, y\rangle \in \pi : v.y = w.y)\\
\Rightarrow\ &\{(1),\ v \text{ and } w \text{ are extensions}\}\\
&v.x = w.x\\
\equiv\ &\{\text{definition } U\}\\
&x \in U.
\end{aligned}$$

By (5), this proves that $U = P$ so that $v = w$.

Every element $a \in A$ has an associated well-founded D-specification

(6) $\quad i.a = \langle \{p\}, p, \{\langle p, a, \varepsilon\rangle\}\rangle,$

which is easily seen to satisfy

(7) $\quad f^{c}.(i.a) = f.a.$

5. Choice and composition of D-specifications

The *unbounded demonic choice* $([]i :: P.i)$ of D-specifications $\langle P.i, p.i, \pi.i\rangle$ is defined as $P = \langle P, p, \pi\rangle$ where p is a new symbol and the set P is the union of $\{p\}$ with the disjoint union of the sets $P.i$. Relation π is the union of the set of the silent transitions $\langle p, \tau, p.i\rangle$ with the union of the transition relations $\pi.i$ of the constituents. It is left to the reader to prove that

(0) $\quad f^{c}.P = ([]i :: f^{c}.(P.i)),$

and similarly for f_c. It is easy to see that P is well-founded if and only if all constituent D-specifications $P.i$ are well-founded.

The *composition* of a D-specification $P = \langle P, p, \pi\rangle$ and a D-specification $Q = \langle Q, q, \psi\rangle$ is defined as the D-specification $(P;\ Q) = \langle R, p, \rho\rangle$, where R is the disjoint union $P \,\dot{\cup}\, Q$ of the sets P and Q, and ρ is given by

(1) $\quad \langle x, a, y\rangle \in \rho \ \equiv\ (\langle x, a, y\rangle \in \pi \wedge y \neq \varepsilon) \vee (\langle x, a, \varepsilon\rangle \in \pi \wedge y = q)$
$\qquad\qquad\qquad\qquad \vee \langle x, a, y\rangle \in \psi.$

Again it is easy to see that $(P;\ Q)$ is well-founded if and only if both P and Q are well-founded.

It is the aim of the construction that the composition should satisfy the property $f^{c}.(P;\ Q) = f^{c}.P;\ f^{c}.Q$, and similarly for f_c. Our proof of these facts is unexpectedly difficult. It is here that we need the assumption that M be completely left-distributive.

Let P be a D-specification and let $m \in M$. We define function $F.m : M^P \to M^P$ by

(2) $F.m.w.x = ([]a, y : \langle x, a, y\rangle \in \pi :$

if $a \neq \tau \wedge y \neq \varepsilon$ **then** $f.a;\ w.y$

$[]a = \tau$ **then** $w.y$

$[]y = \varepsilon$ **then** $f.a;\ m$ **fi**)

for all $w \in M^P$ and $x \in P$. It follows from 4(1), (1) and (2) that we have the next lemma.

(3) **Lemma.** *A function $u \in M^{P \cup Q}$ is an extension of f to $(P;\ Q)$ if and only if $u \mid P$ is a fixpoint of $F.(u.q)$ and $u \mid Q$ is an extension of f to Q.*

This result leads to consideration of the fixpoint equation for $F.m$ with $m \in M$.

(4) **Lemma.** *Let M be completely left-distributive. Then $F.m$ has a biggest fixpoint $g \in M^P$, and it satisfies*

$$(\forall x \in P :: g.x = ((f^P.x);\ m)).$$

Proof. Let $G : (M^P)^2 \to (M^P)^2$ be given by

(5) $\quad G.\langle u, v\rangle = \langle E.u, F.m.v\rangle.$

We use the componentwise ordering of $(M^P)^2$. Therefore, $(M^P)^2$ has greatest lower bounds, which can be calculated componentwise. We use $[]$ to denote greatest lower bounds in $(M^P)^2$.

Function G is easily seen to be monotone. Let Y be the subset of $(M^P)^2$ given by

(6) $\quad \langle u, v\rangle \in Y \ \equiv\ (\forall x \in P :: v.x = (u.x;\ m)).$

We claim that Y is G-invariant and closed under greatest lower bounds in $(M^P)^2$. In fact, for any $\langle u, v\rangle \in Y$ and any $x \in P$ we have

(7) $\quad F.m.v.x$

$= \{(2), (6)\}$

$([]a, y : \langle x, a, y\rangle \in \pi :$

if $a \neq \tau \wedge y \neq \varepsilon$ **then** $f.a;\ u.y;\ m$

$[]a = \tau$ **then** $u.y;\ m$

$[]y = \varepsilon$ **then** $f.a;\ m$ **fi**)

$= \{2(1)$, M is completely left-distributive; $4(2)\}$

$E.u.x;\ m.$

It follows that Y is G-invariant, since for any pair $\langle u, v\rangle$ we have

$$\begin{aligned}
&G.\langle u, v\rangle \in Y\\
\equiv\ &\{(5), (6)\}\\
&(\forall x \in P :: F.m.v.x = E.u.x;\ m)\\
\Leftarrow\ &\{(7)\}\\
&\langle u, v\rangle \in Y.
\end{aligned}$$

The subset Y is closed under greatest lower bounds, since for any family $(i \in I :: \langle u.i, v.i\rangle)$ in $(M^P)^2$ we have

$$\begin{aligned}
&([]i :: \langle u.i, v.i\rangle) \in Y\\
\equiv\ &\{(6) \text{ and definition } [] \text{ in } (M^P)^2\}\\
&(\forall x \in P :: ([]i :: v.i).x = (([]i :: u.i).x;\ m))\\
\equiv\ &\{\text{definition } [] \text{ in } M^P\}\\
&(\forall x \in P :: ([]i :: v.i.x) = (([]i :: u.i.x);\ m)))\\
\equiv\ &\{2(1),\ M \text{ is completely left-distributive}\}\\
&(\forall x \in P :: ([]i :: v.i.x) = ([]i :: (u.i.x;\ m)))\\
\Leftarrow\ &\{(6)\}\\
&(\forall i :: \langle u.i, v.i\rangle \in Y).
\end{aligned}$$

This shows that Y is G-invariant and closed under greatest lower bounds in $(M^P)^2$. By the extended theorem of Knaster-Tarski, i.e. the dual of the theorem in [6, Section 1.3], it follows that G has a biggest fixpoint in $(M^P)^2$ and that this fixpoint is element of Y.

Since function G acts diagonally, cf. (5), the biggest fixpoint of G is a pair of biggest fixpoints of E and $F.m$, respectively. The biggest fixpoint of E is f^P. Let g be the biggest fixpoint of $F.m$. Since $\langle f^P, g\rangle \in Y$, we have $g.x = (f^P.x;\ m)$ for all $x \in P$. □

(8) **Theorem.** *Let M be completely left-distributive. Then* $f^c.(P;\ Q) = f^c.P; f^c.Q$.

Proof. Let $m = f^Q.q$ in M. Let $g \in M^P$ be the biggest fixpoint of $F.m$. By Lemma (4) we have

(9) $\quad (\forall x \in P :: g.x = (f^P.x; f^Q.q)).$

Recall that $(P;\ Q)$ is the D-specification $\langle R, p, \rho\rangle$ where R is the disjoint union of the sets P and Q, and where ρ is given by (1). The union $g \cup f^Q$ is a function on R. By Lemma (3), it is an extension of f to R. Therefore, we have $g \cup f^Q \leq f^R$, or equivalently

(10) $\quad g \leq f^R | P, \qquad f^Q \leq f^R | Q.$

By Lemma (3), function $F^R|Q$ is an extension of f to Q. This implies $f^R|Q \leqslant f^Q$, and hence $f^R|Q = f^Q$ by (10). This proves that $f^R.q = m$. From Lemma (3) and $f^R.q = m$, it follows that $f^R|P$ is a fixpoint of $F.m$, so that $f^R|P \leqslant g$. With (10), this proves that $f^R|P = g$. By (9), this implies

$$f^c.(P;\ Q) = f^R.p = g.p = (f^P.p; f^Q.q) = (f^c.P; f^c.Q). \quad \square$$

Remark. If P and Q are well-founded, the proof of Theorem (8) can be simplified by using well-founded induction instead of fixpoint induction. One still needs complete left-distributivity of M as used in calculation (7).

(11) **Corollary.** *Let M be completely left-distributive. Then $f_c.(P;\ Q) = f_c.P;\ f_c.Q$.*

Proof. The proof is almost identical to the proof of the theorem, if everywhere the word "biggest" is replaced by "smallest". In the analogue of Lemma (4), however, one needs the second formula of 2(1) to show that the set Y is closed under least upper bounds. $\square$

We finally come to the construction of a completion A^c of specification algebra A. Let *WFD* be the class of well-founded D-specifications over A. We define the equivalence relation $\approx$ on *WFD* by

$$P \approx Q \ \equiv \ (\forall f :: f^c.P = f^c.Q)$$

where f ranges over the homomorphisms $f: A \to M$ from A to complete and completely left distributive specification algebras M. It follows from formula (0) that for families of D-specifications $P.i$ and $Q.i$ with i ranging over some set, we have

$$(\forall i :: P.i \approx Q.i) \ \Rightarrow \ ([]i :: P.i) \approx ([]i :: Q.i).$$

Similarly, it follows from Theorem (8) that for D-specifications P, P', Q, Q'

$$P \approx P' \wedge Q \approx Q' \ \Rightarrow \ P;\ Q \approx P';\ Q'.$$

This shows the class of equivalence classes ($WFD/\approx$) can be equipped with induced operators "[]" and ";" to yield an "algebra"

$$A^c = \langle WFD/\approx, [], ;\rangle.$$

It can be shown that A^c satisfies the axioms of a complete specification algebra. Every homomorphism $f: A \to M$ from A to a complete and completely left distributive specification algebra induces a function f^c: $WFD \to M$, which factors over a []-complete homomorphism $f^c: A^c \to M$. This shows that A^c is a completion in the sense of Section 3. By the result of Section 3, it follows that ($WFD/\approx$) is not a set, but a proper class, cf. [11]. Note that relation "$\approx$" can be compared with bisimilarity, although its definition is not based on the existence of a bisimulation, but on a universal quantification over homomorphisms.

Remark. In this construction of A^c, we have restricted ourselves to well-founded D-specifications. The reason for consideration of general D-specifications is that they may be useful for reactive systems where one has to use other semantical formalisms such as traces, failure sets, or temporal logic. We leave this as a subject of future research.

6. Transition systems with angelic choice

The formalism for D-specifications is chosen in such a way that it is easy to incorporate angelic choice. We only have to add a subset $P_\Diamond$ of P to indicate the states where the choice is to be angelic.

We define an *AD-specification* to be a quadruple $\langle P, p, \pi, P_\Diamond\rangle$ where $\langle P, p, \pi\rangle$ is a D-specification and $P_\Diamond$ is a subset of P.

Let $f: A \to M$ be a function from A to a complete and completely left-distributive specification algebra M. A function $w: P \to M$ is defined to be an *extension* of f to the AD-specification P if and only if for every $x \in P \backslash P_\Diamond$ formula 4(1) holds and for every $x \in P_\Diamond$ we have

$$\begin{aligned} w.x = (\Diamond a, y : \langle x, a, y\rangle \in \pi :\ & \\ &\textbf{if } a \neq \tau \wedge y \neq \varepsilon \textbf{ then } f.a;\ w.y \\ &[]\, a = \tau \textbf{ then } w.y \\ &[]\, y = \varepsilon \textbf{ then } f.a \textbf{ fi}). \end{aligned}$$

The existence of extreme extensions is proved in the same way as for D-specifications. Again, we write f^P and f_P to denote the extreme extensions of f to P, and we write $f^c.P = f^P.p$ and $f_c.P = f_P.p$. If P is a well-founded AD-specification, then $f^c.P = f_c.P$ as before.

The construction of the composition is a straightforward adaptation of the case of the D-specifications: take $(P;\ Q)_\Diamond = P_\Diamond \cup Q_\Diamond$. In the construction of the demonic choice between AD-specifications, one defines $P_\Diamond$ as the union of the sets $P_\Diamond.i$. The angelic choice is completely analogous, but with the new initial state p added to $P_\Diamond$. The main result is

Theorem. (a) *For any family* $(i :: P.i)$ *of AD-specifications*

$$f^c.([]i :: P.i) = ([]i :: f^c.(P.i))$$

and similarly for the combinations $\langle f^c, \Diamond\rangle$ *and* $\langle f_c, []\rangle$ *and* $\langle f_c, \Diamond\rangle$.

(b) *For AD-specifications* P *and* Q *we have* $f^c.(P;\ Q) = f^c.P;\ f^c.Q$, *and similarly for* f_c.

7. Concluding remarks

A lot remains to be done. We have not axiomatised noncomplete specification algebras with both demonic and angelic choice. It may be important to characterise the order of determinacy for AD-specifications. Actually, this is a preorder and it will induce an equivalence relation that must be related to bisimulation. A complicating factor is that if A is itself a specification algebra, then every finite well-founded AD-specification should be equivalent to an element of *A*. This problem is related to the encapsulation problem in bisimulation.

References

[1] J.C.M. Baeten, J.A. Bergstra and J.W. Klop, On the consistency of Koomen's fair abstraction rule, *Theoret. Comput. Sci.* **51** (1987) 129-176.

[2] R.J.R. Back and J. von Wright, A lattice-theoretical basis for a specification language, in: J.L.A. van de Snepscheut, ed., *Mathematics of Program Construction*, Lecture Notes in Computer Science, Vol. 375 (Springer, Berlin, 1989) 139-156.

[3] J.W. de Bakker, J.I. Zucker, Processes and a fair semantics for the ADA rendez-vous, in: *Proc. 10th ICALP*, Lecture Notes in Computer Science, Vol. 154 (Springer, Berlin, 1983) 52-66.

[4] D. Harel, Dynamic logic, in: D. Gabbay, F. Guenthner, eds., *Handbook of Philosophical Logic, Vol. 2* (Reidel, Dordrecht, 1984) 497-604.

[5] W.H. Hesselink, Deadlock and fairness in morphisms of transition systems, *Theoret. Comput. Sci.* **59** (1988) 235-257.

[6] W.H. Hesselink, Predicate transformer semantics of general recursion, *Acta Inform.* **26** (1989) 309-332.

[7] W.H. Hesselink, Command algebras, recursion and program transformation, *Formal Aspects Comput.* **2** (1990) 60-104.

[8] W.H. Hesselink, Modalities of nondeterminacy, in: W.H.J. Feijen et al., eds, *Beauty is our Business, a Birthday Salute to Edsger W. Dijkstra* (Springer, Berlin, 1990) 182-192.

[9] C. Morgan and P. Gardiner, Data refinement by calculation, Manuscript 1988.

[10] J.M. Morris, A theoretical basis for stepwise refinement and the programming calculus. *Sci. Comput. Programming* **9** (1987) 287-306.

[11] G. Takeuti and W.M. Zaring, *Introduction to axiomatic set theory*, Graduate texts in mathematics 1 (Springer, Berlin, 1971).

[12] A. Tarski, A lattice theoretical fixpoint theorem and its applications, *Pacific J. Math.* **5** (1955) 285-309.