SPECIFICATION OF SCHEDULERS
WITH ALGEBRAIC SPECIFICATION TECHNIQUES

Kouichi WADA  Kenichi HAGIHARA
Toshiro ARAKI  Nobuki TOKURA

Department of Information and Computer Sciences
Faculty of Engineering Science
Osaka University
Toyonaka, Osaka 560 Japan

ABSTRACT

The specification of a class of single resource schedulers is presented. A class of these schedulers is regarded as a parameterized data type whose formal parameter is a scheduling strategy. The specification is given by using the concept of parameterization in algebraic techniques. The specification of each scheduler instance can be directly obtained only by replacing the formal parameter with a concrete scheduling strategy. The assignment of an actual parameter to a formal parameter is given by a mechanism called extended parameter morphisms. This specification technique is applicable to many other objects.

I. INTRODUCTION

To specify an operating system (for short OS) formally is a very important problem both in practice and in theory. But there has been few works on specifying the total OS formally. Because OS is too complex and has many machine-dependent parts, and we do not have a framework of formal specification enough to describe large objects such as OS.

An algebraic specification technique has been investigated by many researchers, Guttag (1975), ADJ (1976,1979), Ehrig (1977) and Ehrich (1979). This technique is considered a promising method for formal specification. However, this method has been also lack of a framework to treat large objects. Recently by ADJ (1979) and Ehrig (1979) the concept of parameterization in algebraic specification is presented. In those papers a precise mathematical treatment is shown. This concept is considered to be the one step specifying large objects.

In the part, with the algebraic technique a specification for a file management system is presented (Kasami et al (1978)). In this paper by using the concept of parameterization in an algebraic specification technique, some specifications of schedulers are described. A scheduler is a kernel of a process management system, which is one of the most significant parts in OS. To specify schedulers advances works on specifying total OS formally.

In section 2 we summarize some definitions in algebraic specification needed in this paper. In section 3 we specify a class of single resource schedulers as a parameterized
data type, i.e., a scheduler with scheduling strategy as a formal parameter, and show that the specification is correct. If we consider a scheduler as a parameterized data type and we assign a concrete scheduling strategy to the scheduler, we can obtain a correct specification of the scheduler with the concrete scheduling strategy. In section 4 we give a specification of schedulers with a concrete scheduling strategy and we give a mechanism called extended parameter morphisms in order that we can get a specification of schedulers with more complex scheduling strategy without changing the specification given in section 3. And in section 5 as an example of an extended parameter morphism, the specification of schedulers with a more complex scheduling strategy is presented.

2. PRELIMINARIES

we shall assume the algebraic and categorical background of ADJ (1973,1976) or of Ehrig and Kreowski (1977). Some definitions from category theory will be found in Appendix. The basic algebraic background is as follows. We start with a set $S$ of names of data types ($S$ is called sort set).

**Definition 2.1.** An $S$-sorted signature $\Sigma$ is a family $\Sigma_{w,s}$ of sets, for $s \in S$ and $w \in S^*$. Call $\sigma \in \Sigma_{w,s}$ an operation symbol and $w$ is its source and $s$ is its target.

**Definition 2.2.** Let $\Sigma$ be an $S$-sorted signature. Then an $<S,\Sigma>$-algebra $A$ consists of a set $A_s$ for each $s \in S$ (called the carrier of $A$ of sort $S$) and a function

$$\sigma^A: A_{S_1} \times A_{S_2} \times \cdots \times A_{S_n} \rightarrow A_s$$

for each $\sigma \in \Sigma_{w,s}$ with $w=s_1s_2\cdots s_n$ (called the operation of $A$ named by $\sigma$). For $\sigma \in \Sigma_{w,s}$ we have $\sigma^A \in A_s$ (also written $\sigma^A : A_s$), that is, $\Sigma_{w,s}$ is the set of (names of) constants of $A$ of sort $S$.

**Definition 2.3.** Let $A$ and $B$ be $<S,\Sigma>$-algebras, $A \Sigma$-homomorphism $h: A \rightarrow B$ is a family of functions $<h_s: A_s \rightarrow B_s> s \in S$ that preserve the operations, i.e., that satisfy

(h0) if $\sigma \in \Sigma_{w,s}$ then $h_s(\sigma^A) = \sigma^B$;

(h1) if $\sigma \in \Sigma_{s_1s_2\cdots s_n}$ and $<a_1, \ldots, a_n> \in A_{s_1} \times \cdots \times A_{s_n}$ then

$$h_s(\sigma^A(a_1, \ldots, a_n)) = \sigma^B(h_{s_1}(a_1), \ldots, h_{s_n}(a_n))$$

A data type is regarded as (the isomorphism class of) a minimal $<S,\Sigma>$-algebra (an algebra $A$ is minimal iff the only subalgebra of $A$ is $A$ itself). The pair $<S,\Sigma>$ determines the category $\text{Alg}_{<S,\Sigma>}$ of all $S$-sorted $\Sigma$-algebras with $\Sigma$-homomorphisms between them.

A signature describes the syntactic part of a specification. And the semantic part of the specification is described by $\Sigma$-axioms (for short axioms) which in the most general case are universally quantified implication:

$$(a-1) \ e_1 \wedge e_2 \wedge \cdots \wedge e_n \implies e_{n+1}$$

where each $e_i$ is an equation or inequation between $\Sigma$-terms (expressions over $\Sigma$ with variables). When all the antecedents of (a-1) are equations and the consequent is either an equation or an inequation, the axiom is called universal Horn. For Universal