Chapter 1

Introduction and Motivations

1.1 Software Specification, Binary Relations and Fork

Fork algebras —the subject of this book— have their origin as the foundation of a framework for software specification, verification and derivation. In our view, specification languages —as modern graphical notations like UML [G. Booch et al. (1998)]— must allow for a modular description of the different aspects that comprise a system. These aspects include structural properties, dynamic properties, temporal properties, etc. Different formalisms allow us to specify each one of these aspects, namely,

- first-order classical logic for structural properties
- propositional and first-order dynamic logic for dynamic properties,
- different modal logics for temporal properties.

Many of the previously mentioned formalisms have complete deductive systems. Nevertheless, reasoning across formalisms may be difficult if not impossible. A possible solution in order to solve this problem consists on finding an amalgamating formalism satisfying at least the following:

- the formalism must be expressive enough to interpret the specification formalisms,
- the formalism must have very simple semantics, understandable by non mathematicians,
- the formalism must have a complete and simple deductive system.

In this book we propose the formalism called *fork algebras* to this end. The formalism is presented in the form of an equational calculus, which reduces reasoning to substitution of equals by equals. The calculus is complete with respect to a very simple semantics in terms of *algebras of binary relations*.

Algebras of binary relations, such as the ones to be used in this book, have as domain a set of binary relations on some set (let us say A). Among the operations that can be defined on such domain, consider the following:

- the empty binary relation $\emptyset$,
- complement of a binary relation x with respect to a largest relation E, i.e., $\overline{x}$ —as the complement of x is denoted— is defined as $E \setminus x$,
- union of binary relations —denoted by $\cup$—, and
- intersection of binary relations —denoted by $\cap$.

Notice that the previous operations are defined on arbitrary sets, independently of whether these are binary relations or not. Actually, a set of binary relations closed under these operations is an example of *set Boolean algebra*. However, there are other operations that operate naturally on binary relations but are not defined on arbitrary sets. Among these we can mention:

- the identity binary relation on A —denoted by Id—,
- composition of binary relations —denoted by $\circ$—, and
- transposition of the pairs of a binary relation —denoted by $^{\smile}$.

Unfortunately, a class of algebras containing these operations cannot be axiomatized by a finite number of equations [D. Monk (1964)]. In order to overcome this important drawback, we add an extra binary operation on relations called *fork*. Addition of fork has two main consequences. First, the class of algebras obtained can be axiomatized by a finite (and small) number of equations. Second, addition of fork induces a structure on the domain on top of which relations are built, i.e., rather than being the arbitrary set A, it is a set A^* closed under a binary function $*$. The definition of the operation fork (denoted by $\underline{\nabla}$) is then given by:

$$R \underline{\nabla} S = \{ \langle x, y * z \rangle : x R y \ \wedge \ x S z \} \ .$$

The definition of $\underline{\nabla}$ is depicted in Fig. 1.1. Whenever x and y are related via R, and x and z are related via S, x and $y * z$ are related via $R \underline{\nabla} S$. Notice that the definition strongly depends on the function $*$. Actually, the definition of fork evolved around the definition of the function $*$. From 1990

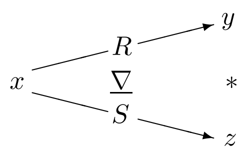

Fig. 1.1 Fork of binary relations R and S.

(when the first class of fork algebras was introduced) until now, different alternatives were explored with the aim of finding a framework which would satisfy our needs. In the definition of the first class of fork algebras [P. Veloso et al. (1991)], function $*$ produced true set theoretical pairs, i.e., when applied to values a and b, $*(a, b)$ returned the pair $\langle a, b\rangle$. Mikulás, Sain, Simon and Németi showed in [S. Mikulás et al. (1992); I. Sain et al. (1995)] that this class of fork algebras was not finitely axiomatizable. This was done by proving that a sufficiently complex theory of natural numbers can be interpreted in the equational theory of these fork algebras, and thus leads to a non recursively enumerable equational theory. Other classes of fork algebras were defined, in which $*$ was binary tree formation or even concatenation of sequences, but these were shown to be non finitely axiomatizable too. It was in [M. Frias et al. (1995)a] where the class of fork algebras to be used in this book came up. The only requirement placed on function $*$ was that it had to be injective. This was enough to prove in [M. Frias et al. (1997)b] that the newly defined class of fork algebras was indeed finitely axiomatizable by a set of equations.