МАТЕМАТИКА И МАТЕМАТИЧЕСКО ОБРАЗОВАНИЕ, 2001
MATHEMATICS AND EDUCATION IN MATHEMATICS, 2001
Proceedings of Thirtieth Spring Conference of
the Union of Bulgarian Mathematicians
Borovets, April 8–11, 2001

AXIOMATIC REPRESENTATION OF THE METHODS FOR SORTING – QUICKSORT, INSERTION SORT AND SELECTION SORT

Irena L. Atanasova

In [3] the basic methods for sorting – quicksort, insertion sort, selection sort are described. These methods are presented axiomatic in this article by language of predicate logic, described in [1] and [2]. The functions quicksort, inssort and selsort describe methods quicksort, insertion sort and selection sort. The most important properties of these functions are showed. The theorem proves the equivalence of these methods.

1. Notation. A tuple is a finite collection of elements, called atoms. The set of atoms may be either finite or infinite. The empty tuple, denoted by "$<>$", has no atoms at all. The unary predicate symbol *atom* (x) is true if x denotes an atom, and false otherwise. The unary predicate symbol *list* (x) is true if x denotes a tuple. The value of the binary function "$\bullet$" is the tuple obtained by concatenation of the atom a and the tuple $x - a \bullet x$.

The concept **tuple** is defined by the following axioms:

1. $list\ (<>)$;
2. $(\forall\ atom\ a)(\forall\ list\ x)(\text{not } ((a \bullet x) =<>))$;
3. $(\forall\ atom\ a)(\forall\ listx)(list(a \bullet x))$;
4. $(\forall\ atom\ a, b)(\forall\ list\ x, y)(\text{if } a \bullet x = b \bullet y \text{ then } a = b \text{ and } x = y)$;
5. $(\text{if } (F[<>] \text{ and } (\forall\ atom\ a)(\forall\ list\ x)(\text{ if } F[x] \text{ then } F[a \bullet x]) \text{ then } (\forall\ list\ x)F[x])$, where a does not occur free in $F[x]$.

The fifth axiom is called induction principle. The formula $F[x]$ is called an inductive formula. The subformula $F[<>]$ is called a base case of the induction. The subformula

$$(\forall\ atom\ a)(\forall\ list\ x)(\text{ if } F[x] \text{ then } F[a \bullet x]) \tag{1}$$

is called an inductive step. The subformula $F[x]$ of (1) is called induction hypothesis, and $F[a \bullet x]$ is called condition. The variable x is called an inductive variable.

The **subtuple** is defined by the following axioms:

1. $(\forall\ list\ x)(<> \prec x)$;
2. $(\forall\ atom\ a)(\forall\ list\ x)(\text{not } ((a \bullet x) \prec <>))$;
3. $(\forall\ atom\ a)(\forall list\ x, y)(a \bullet x \prec a \bullet y \equiv x \prec y)$;
4. $(\forall\ atom\ a, b)(\forall\ list\ x, y)(\text{if not } (a = b) \text{ then } a \bullet x \prec b \bullet y \equiv a \bullet x \prec y)$;

A tuple y is said to be a permutation of another tuple x if each can be obtained from the other simply by rearranging the elements.

The permutation relation $perm(x,y)$ is defined by the following axioms:

1. $perm(<>,<>)$;
2. $(\forall\ atom\ a)(\forall\ list\ x_1,x_2,y_1,y_2)(perm(x_1\bullet <a> \bullet x_2, y_1\bullet <a> \bullet y_2)$
$\equiv perm(x_1 \bullet x_2, y_1 \bullet y_2))$;
3. $(\forall\ list\ x,y)(\text{if } perm(x,y) \text{ then } (\forall\ atom\ a)(a\in x \equiv a \in y))$.

A tuple is said to be ordered if its elements are ordered into increasing order. The unary predicate symbol $order(x)$ is true if the atoms are ordered into increasing order. $order(x)$ is defined by the following axioms:

1. $order(<>)$;
2. $(\forall\ atom\ a)(order(<a>)$;
3. $(\forall\ atom\ a,b)(\forall\ list\ x)(order(a\bullet(b\bullet x)) \equiv <a> \prec <b> \text{ and } order(b\bullet x))$.

2. The methods for sorting.

2.1. Quicksort. The method quicksort is realized by the unary function $quicksort(x)$. $Quicksort(x)$ produces a tuple, whose elements are into increasing order.

$$(\forall\ list\ x)(order(list(quicksort(x)))) \tag{2}$$

The function *quicksort* is defined by the axioms:

1. $quicksort(<>) = <>$;
2. $(\forall\ atom\ a)(\forall list\ x)(quicksort(a\bullet x) = quicksort$
$(split(a, Before)\bullet <a> \bullet split(a, After))))$.

The function *split* returns subtuple $Before$ and $After$. $Before$ contains those elements of x less than or equal to a. $After$ contains those elements of x strictly greater than a.

The auxiliary function *split* is defined by the axioms:

1. $(\forall\ atom\ a)(split(a,<>) = <>)$;
2. $(\forall\ atom\ a,b)(\forall list x)(split(a, b\bullet x) = \text{ if (not } (<b> \prec <a>)$
$\text{then } b\bullet split(a, After) \text{ else } split(a, Before))$;
3. $(\forall\ atom\ a,b)(\forall\ list\ x)(split(a, b\bullet x) = \text{if } (<b> \prec <a>)$
$\text{then } split(a, Before)\bullet b \text{ else } split(a, After))$.

The function *split* has following properties:

1.1 $(\forall\ atom\ a)(\forall\ list\ x)(list(split(a, Before)))$;
1.2 $(\forall\ atom\ a)(\forall\ list\ x)(list(split(a, After)))$;
2.1 $(\forall\ atom\ a,b)(\forall\ list\ x)(\text{if } b\in split(a, Before) \text{ then } <b> \prec <a>)$;
2.2 $(\forall\ atom\ a,b)(\forall list x)(\text{if } b \in split(a, After) \text{ then not } (<b> \prec <a>))$;
3. $(\forall\ atom\ a)(\forall\ list\ x)(perm(x, split(a, Before)\bullet split(a, After)))$.

2.2 Insertion sort. The unary function symbol *inssort* (x) produces a tuple whose elements are the same as those of x but rearranged into increasing order.

$$(\forall\ list\ x)(order(inssort(x))). \tag{3}$$

The function *inssort* (x) is defined by the axioms:

1. $inssort(<>) = <>$;
2. $(\forall\ atom\ a)(\forall\ list\ x)(inssort(a\bullet x) = insert(a, inssort(x)))$.

The auxiliary function *insert* is defined by the axioms:

1. $(\forall\ atom\ a)(insert(a,<>) = <a>)$;
2. $(\forall\ atom\ a,b)(\forall list\ x)(insert(a, b\bullet x) = (\text{if } <a> \prec <b> \text{ then } a\bullet(b\bullet x)$
$\text{else } b\bullet insert(a\bullet x)))$.

The functions *inssort* and *insert* always yield a tuple:
1. $(\forall\ list\ x)(list(inssort(x)))$;
2. $(\forall\ atom\ a)(\forall\ list\ x)(list(insert(a,x)))$.
The functions *inssort* and *insert* have following properties:
1. $(\forall\ atom\ a)(\forall\ list\ x)(\text{if } order(x) \text{ then } order(insert(a,x)))$;
2.1 $(\forall\ atom\ a)(\forall\ list\ x)(perm(a \bullet x, insert(a,x)))$;
2.2 $(\forall\ list\ x)(perm(x, inssort(x)))$;
3. $(\forall\ list\ x,y)(\text{if } order(y) \text{ and } perm(x,y) \text{ then } y = inssort(x))$;
4. $(\forall\ list\ x)(inssort(inssort(x)) = inssort(x))$.

2.3 Selection sort. The unary function $selsort(x)$ is defined by the axioms:
1. $selsort(<>) =<>$;
2.$(\forall\ atom\ a)(\forall\ list\ x,y)(\text{if } a = sel(x\bullet < a > \bullet y)$
$\text{then } selsort(x\bullet < a > \bullet y) = a \bullet selsort(x \bullet y)))$.

The unary function *sel* produces the least element of nonempty tuple *x*. This function is defined by the axioms:
1. $(\forall\ atom\ a)(sel(< a >) = a)$;
2. $(\forall\ atom\ a)(\forall\ list\ x)(\text{if not } (x =<>) \text{ then } (sel(a \bullet x) =$
$\text{if } < a > \prec\ sel(x) \text{ then } a \text{ else } sel(x)))$;
sel (x) is indeed the least element of the nonempty tuple x, that is:
$(\forall\ list\ x)(\text{if not } (x =<>) \text{ then}(sel(x) \in x \text{ and } (\forall\ atom\ a) \text{ if } a \in x \text{ then } sel(x) \prec< a >)))$.

The function *selsort* has the following properties:

$$(\forall\ list\ x)(order(selsort(x))); \tag{4}$$

$$(\forall\ list\ x)(perm(x, selsort(x))). \tag{5}$$

Theorem *(equivalence of methods for sorting)*
1. $(\forall\ list\ x)(inssort(x) = selsort(x)))$*;*
2. $(\forall\ list\ x)(inssort(x) = quicksort(x)))$*;*
3. $(\forall\ list\ x)(quicksort(x) = selsort(x)))$.

Proof. Let x is a tuple. *perm* $(x,\ y)$ returns a tuple y. y contains same atoms as x. order (x) returns true if x is ordered into increasing order.

The functions *quicksort*, *inssort* and *selsort* have properties *perm* and *order*.

Perm $(y,\ quicksort(x))$ returns a tuple that is the result of function $quicksort(x)$. This tuple y is a permutation of tuple x: $y{=}quicksort(x)$. *order* (y) is true. Therefore y and x have exactly the same atoms, but ordered in different way. Atoms of y are into increasing order. The tuple $z{=}inssort(x)$ and $t{=}selsort(x)$ have exactly the same atoms as x too. z and t are into increasing order (by (2), (3), (4)). Therefore y, z and t are one tuple - ordered tuple x.

REFERENCES

[1] Mendelson E. Introduction to Mathematical Logic. D. Van Nostrand Company, Inc. 1964.
[2] Takeuti G. Proof Theory. North-Holland Publishing Company – Amsterdam. London American Elsevier Publishing Company, Inc. New York, 1975.
[3] Азълов П. Програмиране. Основен курс I част. Увод в програмирането. Асио, София, 1995.

Irena Atanasova
Dept. of Mathematics and Computer Sciences
"Neofit Rilski" South-West University
2700 Blagoevgrad, Bulgaria
e-mail: `irenatm@aix.swu.bg`

АКСИОМАТИЧНО ПРЕДСТАВЯНЕ НА МЕТОДИТЕ ЗА СОРТИРАНЕ – „БЪРЗО СОРТИРАНЕ", „СОРТИРАНЕ ЧРЕЗ ВМЪКВАНЕ" И „СОРТИРАНЕ ЧРЕЗ СЕЛЕКЦИЯ"

Ирена Любомирова Атанасова

В [3] са описани основните методи за сортиране – бързо сортиране, сортиране чрез вмъкване, сортиране чрез селекция. Тези метода са представени аксиоматично в тази статия чрез езика на предикатната логика, описан в [1] и [2]. Методите за сортиране – бързо сортиране, сортиране чрез вмъкване, сортиране чрез селекция са описани чрез функциите quicksotr, inssort и selsort. Показани са най-важните свойства на тези функции. Теоремата доказва еквивалентността на тези методи.