Formalization of the General Hoare Logic Laws

Aleksandar Kupusinac [1], Dusan Malbaski [1]

[1]*Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia*

***Abstract* – This paper presents a new approach to formalizing the general rules of the Hoare logic. Our way is based on formulas of the first-order predicate logic defined over the abstract state space of a virtual machine, i.e. so-called *S*-formulas. *S*-formulas are general tool for analyzing program semantics inasmuch as Hoare triples of total and partial correctness are not more than two *S*-formulas. The general rules of Hoare logic, such as the laws of consequence, conjunction, disjunction and negation can be derived using axioms and theorems of first-order predicate logic. Every proof is based on deriving the validity of some *S*-formula, so the procedure may be automated using automatic theorem provers. In this paper we will use `Coq`.**

***Keywords* – Program verification, program correctness, Hoare logic, first-order predicate logic, `Coq`.**

1. Introduction

Hoare logic incorporates the formulas of total and partial correctness, the assignment axiom and numerous rules [1]. The formulas of total and partial correctness are customarily denoted respectively by $\{P\}S\{Q\}$ and $P\{S\}Q$ and their meaning is given in a descriptive form. Instead of this, this paper introduces strict mathematical notation for both formulas treating them as two special *S*-formulas [2]. Formulas of the first-order predicate logic defined on the abstract state space we call briefly *S*-formulas [3].

We will show that the general rules of Hoare logic are theorems that can be derived using solely the axioms and theorems of predicate logic. It follows that Hoare logic is a special case of first-order predicate logic. Connecting Hoare's ideas with predicate logic is of significant importance [2, 4]. In such connection Hoare logic is an appropriate mechanism for describing program syntax, while in its background predicate logic stays with its powerful mathematical proving tools.

Accordingly, proving program correctness, as well as building new theorems conforms to the validity proofs of appropriate *S*-formulas. Based on that, we may conclude that for proving program correctness and new theorems we need rather uncomplicated mathematical tools such as axioms, theorems and proving procedures of first-order predicate logic. Moreover, the above-mentioned proofs can be automated by using theorem provers. We will demonstrate those possibilities using the prover `Coq` [5].

The `Coq` system is designed to develop mathematical proofs, and especially to write formal specifications, programs and to verify that programs are correct with respect to their specification. It provides a specification language named `GALLINA`. Terms of `GALLINA` can represent programs as well as properties of these programs and proofs of these properties. Clearly, `GALLINA` allows to develop mathematical theories (built from axioms, hypotheses, parameters, lemmas, theorems and definitions of constants, functions, predicates and sets) and to prove specifications of programs [6].

2. *S*-formulas

In this paper we use the following concepts and notation:

- The set of abstract states A,
- State variables (*S*-variables) $x, y, z, \dots$,
- State constants (*S*-constants) $s_1, s_2, s_3, \dots$,
- Unary *S*-formulas or *S*-predicates $P, Q, B, \dots$,
- Binary *S*-formulas or *S*-relations $S_1, S_2, S_3, \dots$,
- Program variables $a, b, c, \dots$,
- Program constants $c_1, c_2, c_3, \dots$.

Let $\{a_1, a_2, \dots, a_n\}$ be a set of program variables, which take values from sets $D_1, D_2, \dots, D_n$ respectively. Interpretation of the set A with respect to the set $\{a_1, a_2, \dots, a_n\}$ is a bijection that maps any *S*-constant from A to the appropriate vector of program constants from $D_1, D_2, \dots, D_n$ (usually called state vector). *S*-relation $S(x,y)$ contains ordered pairs (x,y), where $x \in A$ is the initial state and $y \in A$ is the final state. Interpreted *S*-relation on the set A is called syntactic unit on program variables $\{a_1, a_2, \dots, a_n\}$. A syntactic unit may be written in many different ways (program code is one of them), and it can refer to a statement, block, subprogram or program. This means that we observe two domains: the abstract state domain with S-constants, S-variables, S-predicates and S-relations and the interpretation domain with vectors of program constants, program variables, predicates and syntactic units. To simplify, S-constant is interpreted as a vector of program constants from the set D_1, D_2, ... , D_n, S-predicate is interpreted as a Boolean expression, and S-relation as a syntactic unit with program variables $\{a_1, a_2, \ldots, a_n\}$. Interpretation is denoted by ":". For example, x: $a>0 \wedge b=5$ means that S-variable x represents all states in which program variables a and b satisfy $a>0$ and $b=5$.

The symbol $\leftrightarrow$ means "abbreviation". If α is a token and F is an S-formula then $\alpha \leftrightarrow F$ means "α is an abbreviation for F". If F_1 and F_2 are two S-formulas with the same form, we say that F_1 is syntactically identical to F_2, and write $F_1=F_2$. If F_1 and F_2 have the same meaning but not the same form, they are semantically equivalent, denoted by $F_1 \equiv F_2$. When writing S-formulas we will obey the usual priority conventions, where the order of priority is: negation $\neg$, conjunction $\wedge$, disjunction $\vee$, implication $\Rightarrow$, equivalence $\Leftrightarrow$. The priority can be changed by using brackets ( ) and [ ].

We use strict mathematical notation for $\{P\}S\{Q\}$ and $P\{S\}Q$ formulas:

- Total correctness formula (**TCF**)

$\{P\}S\{Q\} \leftrightarrow \forall x[P(x) \Rightarrow (\exists y S(x,y) \wedge \forall z(S(x,z) \Rightarrow Q(z)))]$,

- Partial correctness formula (**PCF**)

$P\{S\}Q \leftrightarrow \forall x[(P(x) \wedge \exists y S(x,y)) \Rightarrow \forall z(S(x,z) \Rightarrow Q(z))]$.

Hoare's total correctness formula, denoted by $\{P\}S\{Q\}$, is defined by the statement "if the syntax unit S starts in a state satisfying the predicate P, then it terminates in a state satisfying the predicate Q" [1]. The connection between this sentence and the formula (**TCF**) is apparent: if for every state x the S-predicate P holds, then the S-formula $\forall x \exists y S(x,y) \wedge \forall z(S(x,z) \Rightarrow Q(z))$ is true. The state x is then called the initial state. The formula $\forall x\ \exists y S(x,y)$ means that for every initial state x there exists a state y such that $S(x,y)$. The state y is then called the final state. The meaning of the S-formula $\forall x \forall z(S(x,z) \Rightarrow Q(z))$ is the following: if for every initial state x and every state z it is true that $S(x,z)$, then in the state z the S-predicate Q is true.

Hoare's partial correctness formula, denoted by $P\{S\}Q$, is defined by the statement "if the syntax unit S starts in a state satisfying the predicate P and if it terminates then the final state satisfies the predicate Q" [1, 7]. In terms of our consideration, we assert: if for some state x the predicate P holds and if there exists a final state y such that $S(x,y)$, then the formula $\forall x \forall z(S(x,z) \Rightarrow Q(z))$ is true.

We will briefly cite some well-known theorems of predicate logic that will be needed for further proofs in this paper (the symbols F, G, H and K stand for S-formulas):

(**T₁**) $\forall x \forall y F \Leftrightarrow \forall y \forall x F$,
(**T₂**) $\exists x \exists y F \Rightarrow \exists y \exists x F$,
(**T₃**) $\forall x F \Leftrightarrow F$,
(**T₄**) $\forall x(F \wedge G) \Leftrightarrow \forall x F \wedge \forall x G$,
(**T₅**) $\forall x F \vee \forall x G \Rightarrow \forall x(F \vee G)$,
(**T₆**) $\neg \forall x F \Leftrightarrow \exists x \neg F$,
(**T₇**) $\forall x F \Leftrightarrow \forall x(F \Leftrightarrow \tau)$,
(**T₈**) $\forall x(\tau \Rightarrow F) \Leftrightarrow \forall x F$,
(**T₉**) $\forall x \neg F \Leftrightarrow \forall x(F \Leftrightarrow \varphi)$,
(**T₁₀**) $\forall x(F \Leftrightarrow F \wedge F)$,
(**T₁₁**) $\forall x(F \Leftrightarrow F \vee F)$,
(**T₁₂**) $\forall x(\neg F \vee \neg G) \Leftrightarrow \forall x \neg(F \wedge G)$,
(**T₁₃**) $\forall x(\neg F \wedge \neg G) \Leftrightarrow \forall x \neg(F \vee G)$,
(**T₁₄**) $\forall x(F \Rightarrow G) \Rightarrow (\forall x F \Rightarrow \forall x G)$,
(**T₁₅**) $\forall x(F \Rightarrow G) \Leftrightarrow \forall x(\neg F \vee G)$,
(**T₁₆**) $\forall x[(F \Rightarrow H) \wedge (H \Rightarrow G)] \Rightarrow \forall x(F \Rightarrow G)$,
(**T₁₇**) $\forall x[(F \Rightarrow G) \wedge (H \Rightarrow K)] \Rightarrow \forall x[(F \vee H) \Rightarrow (G \vee K)]$,
(**T₁₈**) $\forall x[(F \Rightarrow G) \wedge (H \Rightarrow K)] \Rightarrow \forall x[(F \wedge H) \Rightarrow (G \wedge K)]$,
(**T₁₉**) $\forall x[(F \Rightarrow G) \wedge (F \Rightarrow H)] \Leftrightarrow \forall x(F \Rightarrow G \wedge H)$,
(**T₂₀**) $\forall x[(F \Rightarrow G) \vee (F \Rightarrow H)] \Leftrightarrow \forall x(F \Rightarrow G \vee H)$,
(**T₂₁**) $\forall x[(F \Rightarrow H) \wedge (G \Rightarrow H)] \Leftrightarrow \forall x(F \vee G \Rightarrow H)$,
(**T₂₂**) $\forall x[(F \Rightarrow G) \vee (H \Rightarrow K)] \Rightarrow \forall x[(F \wedge H) \Rightarrow (G \vee K)]$,

where
$\forall x\ \tau(x) \equiv \top$,
$\forall x\ \varphi(x) \equiv \perp$.

Program correctness or a new theorem are proven by proving the validity of an appropriate S-formula. This needs a modest mathematical apparatus e.g. the axioms, theorems and proof procedures of the first-order predicate logic. Firstly, we prove a useful theorem which provides an alternative form of the total correctness formula (**TCF**).

Theorem 1. $\{P\}S\{Q\} \Leftrightarrow \forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x \forall z\, [P(x) \wedge S(x,z) \Rightarrow Q(z)]$.

Proof. Since:
$\{P\}S\{Q\} \leftrightarrow \forall x[P(x) \Rightarrow (\exists y S(x,y) \wedge \forall z(S(x,z) \Rightarrow Q(z)))]$,
by the Theorem ($\mathbf{T_{19}}$), the right side becomes:
$\forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x \forall z[P(x) \Rightarrow (S(x,z) \Rightarrow Q(z))]$
and by the Theorem ($\mathbf{T_{15}}$), it becomes:
$\forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x \forall z[P(x) \Rightarrow (\neg S(x,z) \vee Q(z))]$
$\equiv \forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x \forall z[\neg P(x) \vee (\neg S(x,z) \vee Q(z))]$.
After that, by the Theorem ($\mathbf{T_{12}}$), we obtain:
$\forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x \forall z[\neg(P(x) \wedge S(x,z)) \vee Q(z)]$
and finally, by the Theorem ($\mathbf{T_{19}}$), it becomes:
$\forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x \forall z[P(x) \wedge S(x,z) \Rightarrow Q(z)]$.

Q.E.D.

3. General Laws of the Hoare Logic

In this section we will consider the general laws of Hoare logic such as the laws of consequence, disjunction, conjunction and negation [1]. While the Hoare logic treats these laws as rules, we will treat them as theorems. Some of them will be proven using Coq automatic prover.

3.1 Laws of Consequence

Theorem 2. $\forall x(P(x) \Rightarrow R(x)) \wedge \{R\}S\{Q\} \Rightarrow \{P\}S\{Q\}$.

Proof. Since:
$\{R\}S\{Q\} \leftrightarrow \forall x[R(x) \Rightarrow (\exists y S(x,y) \wedge \forall z(S(x,z) \Rightarrow Q(z)))]$,
the left side of the implication can be written as:
$\forall x(P(x) \Rightarrow R(x)) \wedge \forall x[R(x) \Rightarrow (\exists y S(x,y) \wedge \forall z(S(x,z) \Rightarrow Q(z)))]$
and by the Theorem ($\mathbf{T_{16}}$), we obtain:
$\forall x[P(x) \Rightarrow (\exists y S(x,y) \wedge \forall z(S(x,z) \Rightarrow Q(z)))]$,
i.e.
$\{P\}S\{Q\}$.

Q.E.D.

Theorem 3. $\{P\}S\{R\} \wedge \forall z(R(z) \Rightarrow Q(z)) \Rightarrow \{P\}S\{Q\}$.

Proof. By the Theorem 1, the left side of the implication can be written as:
$\forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x \forall z[P(x) \wedge S(x,z) \Rightarrow R(z)] \wedge \forall z(R(z) \Rightarrow Q(z))$
and by the Theorem ($\mathbf{T_{16}}$), we obtain:
$\forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x \forall z[P(x) \wedge S(x,z) \Rightarrow Q(z)]$,
i.e.
$\{P\}S\{Q\}$.

Q.E.D.

Theorem 4. $\forall x(U(x) \Rightarrow P(x)) \wedge \forall z(Q(z) \Rightarrow V(z)) \wedge \{P\}S\{Q\} \Rightarrow \{U\}S\{V\}$.

Proof. By the Theorem 2, the left side of the implication can be written as:
$\forall z(Q(z) \Rightarrow V(z)) \wedge \{U\}S\{Q\}$
and by the Theorem 3, we obtain:
$\{U\}S\{V\}$.

Q.E.D.

Finally, using by the Theorem ($\mathbf{T_3}$), from Theorems 2, 3 and 4 we can obtain the well-known Hoare's rules of consequence [1]:

$$\frac{(P \Rightarrow R),\ \{R\}S\{Q\}}{\{P\}S\{Q\}}\ ,$$

$$\frac{\{P\}S\{R\},\ (R \Rightarrow Q)}{\{P\}S\{Q\}}\ ,$$

$$\frac{(U \Rightarrow P),\ (Q \Rightarrow V),\ \{P\}S\{Q\}}{\{U\}S\{V\}}\ .$$

The Theorems 2, 3 and 4 also can be proven using automatic prover Coq:

```
Variable A: Set.
Variables P Q R: A->Prop.
Variable S: A->A->Prop.
Theorem t2 : ((forall x:A,(P x->R x)) /\
(forall x:A, (R x ->((exists y:A, S x
y)/\(forall z:A, (S x z ->Q z )))))) ->
(forall x, (P x ->((exists y, Sxy)/\(forall
z, (S x z ->Q z ))))).
firstorder.
```

```
Variable A: Set.
Variables P Q R: A->Prop.
Variable S: A->A->Prop.
Theorem t3 : ((forall x:A, (P x ->((exists
y:A, S x y)/\(forall z:A,(S x z ->Q z )))))
/\ (forall z:A,(Q z->R z))) -> (forall x,
(P x ->((exists y, S x y)/\(forall z, (S x
z ->R z ))))).
firstorder.
```

```
Variable A: Set.
Variables P Q U V: A->Prop.
Variable S: A->A->Prop.
Theorem t4  :  ((forall x:A,(U x->P x)) /\
(forall x:A, (P x ->((exists y:A, S x
y)/\(forall z:A,(S x z->Q z))))) /\ (forall
z:A,(Q z->V z)))->(forall x,(U x ->((exists
y, S x y)/\(forall z, (S x z ->V z ))))).
firstorder.
```

3.2 Laws of Conjunction

Theorem 5. $\{P\}S\{Q\} \wedge \{R\}S\{W\} \Rightarrow \{P \vee R\}S\{Q \vee W\}$.

Proof. By the Theorem 1, the left side of the implication can be written as:
$\forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x \forall z[P(x) \wedge S(x,z) \Rightarrow Q(z)] \wedge \forall x[R(x) \Rightarrow \exists y S(x,y)] \wedge \forall x \forall z[R(x) \wedge S(x,z) \Rightarrow W(z)]$.

By the Theorem ($\mathbf{T_{17}}$), we obtain:
$\forall x[(P(x)\vee R(x)) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[((P(x)\wedge S(x,z)) \vee (R(x)\wedge S(x,z))) \Rightarrow (Q(z) \vee W(z))]$
$\equiv \forall x[(P(x)\vee R(x)) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[((P(x)\vee R(x)) \wedge(S(x,z)\vee S(x,z))) \Rightarrow (Q(z) \vee W(z))]$
$\equiv \forall x[(P(x) \vee R(x)) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[(P(x) \vee R(x)) \wedge S(x,z) \Rightarrow (Q(z) \vee W(z))]$,
i.e.
$\{P\vee R\}S\{Q\vee W\}$.

Q.E.D.

Theorem 6. $\{P\}S\{Q\}\wedge\{R\}S\{W\} \Rightarrow \{P\wedge R\}S\{Q\wedge W\}$.

Proof. By the Theorem 1, the left side of the implication can be written as:
$\forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[P(x)\wedge S(x,z) \Rightarrow Q(z)] \wedge \forall x[R(x) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[R(x)\wedge S(x,z) \Rightarrow W(z)]$.
By the Theorem ($\mathbf{T_{18}}$), we obtain:
$\forall x[(P(x)\wedge R(x)) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[((P(x)\wedge S(x,z)) \wedge (R(x)\wedge S(x,z))) \Rightarrow (Q(z)\wedge W(z))]$
$\equiv \forall x[(P(x)\wedge R(x)) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[(P(x)\wedge R(x)) \wedge S(x,z) \Rightarrow (Q(z)\wedge W(z))]$,
i.e.
$\{P\wedge R\}S\{Q\wedge W\}$.

Q.E.D.

The Theorems 5 and 6 can be proven using `Coq`:

```
Variable A: Set.
Variables P R Q W: A->Prop.
Variable S: A->A->Prop.
Theorem t5 : ((forall x:A, (P x -> ((exists y:A,S x y)/\(forall z:A,(Sxz->Q z ))))) /\ (forall x:A, (R x -> ((exists y:A, S x y)/\(forall z:A,(Sxz->W z)))))) -> (forall x:A, ((P x \/ R x) -> ((exists y:A, S x y)/\(forall z:A, (S x z->(Q z \/ W z)))))).
firstorder.
```

```
Variable A: Set.
Variables P Q R W: A->Prop.
Variable S: A->A->Prop.
Theorem t6 : ((forall x:A, (P x -> ((exists y:A, S x y)/\(forall z:A, (S x z->Q z ))))) /\ (forall x:A, (R x -> ((exists y:A, S x y)/\(forall z:A, (S x z->W z )))))) -> (forall x:A, ((P x /\ R x) -> ((exists y:A, S x y)/\(forall z:A, (S x z->(Q z /\ W z)))))).
firstorder.
```

3.3 Law of Resolution

Corollary 1. $\{P\}S\{Q\} \wedge \{\neg P\}S\{W\} \Rightarrow \{\tau\}S\{Q\vee W\}$.

Proof. If we substitute R with $\neg P$ in the Theorem 5 we obtain:
$\{P\}S\{Q\} \wedge \{\neg P\}S\{W\} \Rightarrow \{P\vee\neg P\}S\{Q\vee W\}$,
$\equiv \{P\}S\{Q\} \wedge \{\neg P\}S\{W\} \Rightarrow \{\tau\}S\{Q\vee W\}$.

Q.E.D.

3.4 Law of Disjunction

Theorem 7. $\{P\}S\{Q\}\vee\{R\}S\{W\} \Rightarrow \{P\wedge R\}S\{Q\vee W\}$.
Proof. By the Theorem 1, the left side of the implication can be written as:

$(\forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[P(x)\wedge S(x,z) \Rightarrow Q(z)]) \vee (\forall x[R(x)\Rightarrow\exists y S(x,y)] \wedge \forall x\forall z[R(x)\wedge S(x,z) \Rightarrow W(z)])$
$\equiv (\forall x[P(x)\Rightarrow\exists y S(x,y)]\vee\forall x[R(x)\Rightarrow\exists y S(x,y)]) \wedge (\forall x\forall z [P(x)\wedge S(x,z) \Rightarrow Q(z)] \vee \forall x\forall z[R(x)\wedge S(x,z) \Rightarrow W(z)])$.
By the Theorem ($\mathbf{T_{22}}$), we obtain:
$\forall x[(P(x)\wedge R(x)) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[((P(x)\wedge S(x,z)) \wedge (R(x)\wedge S(x,z)) \Rightarrow (Q(z)\vee W(z))]$
$\equiv \forall x[(P(x)\wedge R(x)) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[(P(x)\wedge R(x)) \wedge S(x,z) \Rightarrow (Q(z)\vee W(z))]$,
i.e.
$\{P\wedge R\}S\{Q\vee W\}$.

Q.E.D.

The proof of Theorem 7 in `Coq` is following:

```
Variable A: Set.
Variables P Q R W: A->Prop.
Variable S: A->A->Prop.
Theorem t7 : ((forall x:A, (P x -> ((exists y:A, S x y)/\(forall z:A, (S x z->Q z ))))) \/ (forall x:A, (R x -> ((exists y:A, S x y)/\(forall z:A, (S x z->W z )))))) -> (forall x:A, ((P x /\ R x) -> ((exists y:A, S x y)/\(forall z:A, (S x z->(Q z \/W z)))))).
firstorder.
```

3.5 Laws of Conjunction and Disjunction

Theorem 8. $\{P\vee R\}S\{Q\} \Leftrightarrow \{P\}S\{Q\} \wedge \{R\}S\{Q\}$.

Proof. The left side of the eqivalence can be written as:
$\forall x[(P(x)\vee R(x)) \Rightarrow \exists y S(x,y)\wedge\forall z(S(x,z) \Rightarrow Q(z))]$
and by the Theorem ($\mathbf{T_{21}}$), we obtain:
$\forall x[(P(x) \Rightarrow \exists y S(x,y) \wedge \forall z(S(x,z)\Rightarrow Q(z))) \wedge (R(x) \Rightarrow \exists y S(x,y) \wedge \forall z(S(x,z)\Rightarrow Q(z)))]$,
i.e.
$\{P\}S\{Q\} \wedge \{R\}S\{Q\}$.

Q.E.D.

Theorem 9. $\{P\}S\{Q\wedge R\} \Leftrightarrow \{P\}S\{Q\} \wedge \{P\}S\{R\}$.

Proof. By the Theorem 1, the left side of the eqivalence can be written as:
$\forall x[P(x)\Rightarrow\exists y S(x,y)] \wedge \forall x\forall z[P(x)\wedge S(x,z) \Rightarrow Q(z)\wedge R(z)]$
and by the Theorem ($\mathbf{T_{18}}$), we obtain:
$\forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[P(x)\wedge S(x,z) \Rightarrow \mathrm{Q}(z)] \wedge \forall x\forall z[P(x)\wedge S(x,z) \Rightarrow R(z)]$
$\equiv \forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[P(x) \wedge S(x,z) \Rightarrow Q(z)] \wedge \forall x[P(x) \Rightarrow \exists y S(x,y)] \wedge \forall x\forall z[P(x)\wedge S(x,z) \Rightarrow R(z)]$,

i.e.
$\{P\}S\{Q\} \wedge \{P\}S\{R\}$.

Q.E.D.

Theorem 10. $\{P \vee U\}S\{Q \wedge W\} \Leftrightarrow \{P\}S\{Q\} \wedge \{U\}S\{W\} \wedge \{P\}S\{W\} \wedge \{U\}S\{Q\}$.

Proof. By the Theorem 8, from the left side of the equivalence we obtain:
$\{P \vee U\}S\{Q \wedge W\} \Leftrightarrow \{P\}S\{Q \wedge W\} \wedge \{U\}S\{Q \wedge W\}$
and by the Theorem 9, we obtain:
$\{P\}S\{Q \wedge W\} \wedge \{U\}S\{Q \wedge W\} \Leftrightarrow \{P\}S\{Q\} \wedge \{U\}S\{W\} \wedge \{P\}S\{W\} \wedge \{U\}S\{Q\}$.

Q.E.D.

Theorem 11. $\{P\}S\{Q\} \vee \{P\}S\{W\} \Rightarrow \{P\}S\{Q \vee W\}$.

Proof. If we substitute R with P in the Theorem 7 we obtain:
$\{P\}S\{Q\} \vee \{P\}S\{W\} \Rightarrow \{P \wedge P\}S\{Q \vee W\}$
and by the Theorem ($\mathbf{T_{10}}$), we obtain:
$\{P\}S\{Q\} \vee \{P\}S\{W\} \Rightarrow \{P\}S\{Q \vee W\}$.

Q.E.D.

3.6 General Law of the Excluded Miracle

Theorem 12. $\{P\}S\{\varphi\} \Leftrightarrow (P \Leftrightarrow \varphi)$, i.e. $\{P\}S\{\varphi\} \Leftrightarrow \neg P$.

Proof. The left side of the equivalence can be written as:
$\forall x[P(x) \Rightarrow (\exists y S(x,y) \wedge \forall z(S(x,z) \Rightarrow \varphi(z)))]$.
Since the S-formula $\forall x \forall z(S(x,z) \Rightarrow \varphi(z))$ is valid if $\forall x \forall z \neg S(x,z)$ is valid, we obtain:
$\forall x[P(x) \Rightarrow (\exists y S(x,y) \wedge \forall z \neg S(x,z))]$
$\equiv \forall x[P(x) \Rightarrow \varphi(x)]$
and by the Theorem ($\mathbf{T_9}$), we obtain:
$\forall x \neg P(x)$
i.e.
$\neg P$.

Q.E.D.

3.7 Laws of Negation

Theorem 13. $\{P\}S\{Q\} \wedge \{R\}S\{\neg Q\} \Rightarrow \neg(P \wedge R)$.

Proof. If we substitute W with $\neg Q$ in the Theorem 6, we obtain:
$\{P\}S\{Q\} \wedge \{R\}S\{\neg Q\} \Rightarrow \{P \wedge R\}S\{Q \wedge \neg Q\}$
$\equiv \{P\}S\{Q\} \wedge \{R\}S\{\neg Q\} \Rightarrow \{P \wedge R\}S\{\varphi\}$
and by the Theorem 12, we obtain:
$\{P\}S\{Q\} \wedge \{R\}S\{\neg Q\} \Rightarrow \neg(P \wedge R)$.

Q.E.D.

Theorem 14. $\{P\}S\{Q\} \wedge \{P\}S\{\neg Q\} \Leftrightarrow \forall x \neg P(x)$.

Proof. If we substitute R with $\neg Q$ in the Theorem 9, we obtain:
$\{P\}S\{Q\} \wedge \{P\}S\{\neg Q\} \Leftrightarrow \{P\}S\{Q \wedge \neg Q\}$
$\equiv \{P\}S\{Q\} \wedge \{P\}S\{\neg Q\} \Leftrightarrow \{P\}S\{\varphi\}$
and by the Theorem 12, we obtain:
$\{P\}S\{Q\} \wedge \{P\}S\{\neg Q\} \Leftrightarrow \neg P$,
i.e.
$\{P\}S\{Q\} \wedge \{P\}S\{\neg Q\} \Leftrightarrow \forall x \neg P(x)$.

Q.E.D.

Theorem 15. $[\{P\}S\{\neg Q\} \Rightarrow \neg\{P\}S\{Q\}] \Leftrightarrow \exists x P(x)$.

Proof. By the Theorem ($\mathbf{T_{15}}$), the left side of the equivalence become:
$\neg\{P\}S\{\neg Q\} \vee \neg\{P\}S\{Q\}$
and subsequently, by the Theorem ($\mathbf{T_{12}}$), we obtain:
$\neg[\{P\}S\{\neg Q\} \wedge \{P\}S\{Q\}]$.
Then, by the Theorem 14, we obtain:
$\neg[\forall x \neg P(x)]$
and after that, by the Theorem ($\mathbf{T_6}$), we obtain:
$\exists x P(x)$.

Q.E.D.

Theorem 16. $\{P\}S\{Q\} \wedge \{\neg P\}S\{Q\} \Leftrightarrow \forall x \exists y S(x,y) \wedge \forall x \forall z(S(x,z) \Rightarrow Q(z))$.

Proof. If we substitute R with $\neg P$ in the Theorem 8, we obtain:
$\{P \vee \neg P\}S\{Q\}$
$\equiv \{\tau\}S\{Q\}$.
Since:
$\{\tau\}S\{Q\} \leftrightarrow \forall x[\tau(x) \Rightarrow \exists y S(x,y) \wedge \forall z(S(x,z) \Rightarrow Q(z))]$,
by the Theorem ($\mathbf{T_8}$), we obtain:
$\forall x[\exists y S(x,y) \wedge \forall z(S(x,z) \Rightarrow Q(z))]$.

Q.E.D.

Theorem 17. $\exists x \exists z S(x,z) \wedge \neg Q(z) \Rightarrow [\{\neg P\}S\{Q\} \Rightarrow \neg\{P\}S\{Q\}]$.

Proof. By the Theorem ($\mathbf{T_{15}}$), the right side of the implication can be written as:
$\neg\{\neg P\}S\{Q\} \vee \neg\{P\}S\{Q\}$
and by the Theorem ($\mathbf{T_{12}}$), we obtain:
$\neg[\{\neg P\}S\{Q\} \wedge \{P\}S\{Q\}]$.
Then, by the Theorem 16, we obtain:
$\neg\forall x[\exists y S(x,y) \wedge \forall z(S(x,z) \Rightarrow Q(z))]$
$\equiv \exists x \neg[\exists y S(x,y) \wedge \forall z(S(x,z) \Rightarrow Q(z))]$.
By the Theorem ($\mathbf{T_{13}}$), we obtain:
$\exists x[\neg\exists y S(x,y) \vee \neg\forall z(S(x,z) \Rightarrow Q(z))]$
$\equiv \exists x[\forall y \neg S(x,y) \vee \exists z \neg(S(x,z) \Rightarrow Q(z))]$
and by the Theorem ($\mathbf{T_{15}}$), we obtain:
$\exists x[\forall y \neg S(x,y) \vee \exists z \neg(\neg S(x,z) \vee Q(z))]$.
After that, by the Theorem ($\mathbf{T_{12}}$), we obtain:
$\exists x[\forall y \neg S(x,y) \vee \exists z(S(x,z) \wedge \neg Q(z))]$
$\equiv \exists x \forall y \neg S(x,y) \vee \exists x \exists z(S(x,z) \wedge \neg Q(z))$
and finally, by the Theorem ($\mathbf{T_{11}}$), we obtain:

$\exists x \exists z(S(x,z) \wedge \neg Q(z)) \Rightarrow \exists x \exists z(S(x,z) \wedge \neg Q(z)) \vee \exists x \forall y \neg S(x, y)$.

Q.E.D.

The proofs of Theorems 12, 13 and 14 in `Coq` are following:

```
Variable A: Set.
Variables P : A->Prop.
Variable S: A->A->Prop.
Definition phi (x:A) := False.
Theorem t12 : (forall x:A, (P x -> ((exists
y:A, S x y)/\(forall z:A, (S x z->phi z))))
<-> (P x<->phi x)).
firstorder.
```

```
Variable A: Set.
Variables P Q R: A->Prop.
Variable S: A->A->Prop.
Theorem  t13  :  ((forall  x:A,  (P  x  ->
((exists y:A, S x y)/\(forall z:A, (S x z-
>Q z))))) /\ (forall x:A, (R x -> ((exists
y:A,  S  x  y)/\(forall  z:A,  (S  x  z->(~Q
z))))))) -> (forall x:A, (~(P x /\ R x))).
firstorder.
```

```
Variable A: Set.
Variables P Q: A->Prop.
Variable S: A->A->Prop.
Theorem  t14  :  ((forall  x:A,  (P  x  ->
((exists y:A, S x y)/\(forall z:A, (S x z-
>Q z))))) /\ (forall x:A, (P x -> ((exists
y:A,  S  x  y)/\(forall  z:A,  (S  x  z->(~Q
z))))))) <-> (forall x:A, (~(P x))).
firstorder.
```

Corollary 2. $\{P\}S\{Q\} \wedge \{P\}S\{\neg Q\} \Leftrightarrow (P \Leftrightarrow \varphi)$.

Proof. From the Theorem 14 we obtain:
$\forall x \neg P(x)$,
i.e. $P \Leftrightarrow \varphi$.

Q.E.D.

Corollary 3. $[\{P\}S\{\neg Q\} \Rightarrow \neg\{P\}S\{Q\}] \Leftrightarrow \neg(P \Leftrightarrow \varphi)$.

Proof. From Theorem 15 we obtain:
$\exists x P(x)$
$\equiv \neg(\forall x \neg P(x))$,
i.e. $\neg(P \Leftrightarrow \varphi)$.

Q.E.D.

4. Conclusion

In this paper, we have proposed a new method for formalizing the general Hoare logic rules. The approach is based on axioms and theorems of first-order predicate logic. It uses so-called *S*-formulas, which are defined on the abstract state space of a virtual machine. Proving program correctness and/or establishing new theorems conform to proving the validity of the appropriate *S*-formula, and for that, we need only the first-order predicate logic. The mathematical mechanism developed apart from being general, brings together Hoare's ideas and first-order predicate logic. It also enables automatic proofs of program correctness and/or new theorems. In addition, we have provided strictly formal proofs for the general laws of Hoare logic and some of them are proven using `Coq` automatic prover.

Our future research will be aimed towards investigating formalization of the special Hoare rules (*while*, *if-then-else* etc.) and more complex properties and relationships that exist between preconditions, postconditions and syntax units. The second line of work will be development of algorithms for automated program correctness proofs.

Acknowledgements

This work was partially supported by the Ministry of Science and Education of the Republic of Serbia within the projects: ON 174026 and III 044006.

References

[1]. Gordon, M. J. C. : *Programming Language Theory and its Implementation*. Prentice Hall International (UK) Ltd., Hertfordshire, UK, 1988.

[2]. Kupusinac, A. : Analysis of Characteristics of Dynamic Postconditions in Hoare Triples. Ph.D. thesis, *(in Serbian)*, Faculty of Technical Sciences, Novi Sad, Serbia, 2010.

[3]. Kupusinac, A. and Malbaški, D. : Analysis of Loop Semantics using *S*-formulas. TEM Journal: Vol. 1, No. 2, pp. 72-77, 2012.

[4]. Hoare, C. A. R. and Jifeng, H. : *Unifying Theories of Programming*. Prentice-Hall, London, 1998.

[5]. Bertot, Y. and Castéran, P. : Interactive Theorem Proving and Program Development, Coq'Art: the Calculus of Inductive Constructions. Texts in Theoretical Computer Science. An EATCS Series, Vol. 25, Springer-Verlag, Berlin, Heidelberg, 2004.

[6]. The Coq Development Team: The Coq Proof Assistant Reference Manual Version 8.2-bugfix. INRIA, Orsay, France, 2009.

[7]. Hoare, C. A. R. and Lauer, P. E. : Consistent and Complementary Formal Theories of the Semantics of Programming Languages. Acta Informatica, 3, 135–153, 1974.

Corresponding author: Aleksandar Kupusinac
Institution: University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia
E-mail: sasak@uns.ac.rs