УДК 004.42, 510.649

Криволап А.В. [1], аспірант

Система виводу з T- та F-обмеженнями для монотонної логіки Флойда-Хоара

[1] Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Глушкова 4д,
e-mail: krivolapa@gmail.com

A.V. Kryvolap[1], PhD student.

Inference system with T- and F-constrains for monotone Floyd-Hoare logic

[1] Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d,
e-mail: krivolapa@gmail.com

В роботі розглядається розширення логіки Флойда-Хоара на випадок часткових предикатів. Для цього семантика трійки Флойда-Хоара перевизначена за допомогою спеціальної монотонної композиції. Щоб підкреслити цей факт, в подальшому побудоване розширення іменується монотонною логікою Флойда-Хоара.

Основна увага приділяється системам виводу для мови WHILE, як прикладу простої мови програмування в наведеному розширенні логіки Флойда-Хоара. Розглядаються модифікації класичної системи виводу, які будуть коректними і у випадку часткових предикатів. Одна з модифікацій ґрунтується на додаткових обмеженнях для правил, а інша – на властивостях трійок Флойда-Хоара, що зберігаються правилами системи. Проте ці системи мають недоліки, пов'язані з повнотою та складністю додаткових обмежень. Тому пропонується система виводу з T- та F-обмеженнями, які є більш простими. Для побудованої системи доводиться повнота та коректність.

Ключові слова: програмні логіки, системи виводу, часткові предикати.

Extension of Floyd-Hoare logic on partial predicates is considered in this work. For this purpose semantics of the Floyd-Hoare assertion is redefined using special monotone composition. To underline this fact, presented extension is being further referred to as monotone Floyd-Hoare logic.

Main consideration is given to the inference systems for the WHILE language as an example of the simple imperative programing language. References to examples showing that it is impossible to use classical inference system for monotone Floyd-Hoare logic are given. Modifications of the classical inference system that are sound in the case of partial predicates are studied.One of those is based upon additional constraints for the rules and the other upon properties of the Floyd-Hoare assertions that rules of the inference system preserve. Both of the abovementioned inference systems have disadvantages related to completeness and complexity of the added constraints. Thus new inference system with T- and F-constraints is suggested. It has more simple constraints for the rules. Soundness and completeness is proved for the presented inference system.

Key Words: program logic, inference system, partial predicates.

Статтю представив д. ф.-м. н., проф. Буй Д.Б.

Розширення логіки Флойда-Хоара на випадок часткових предикатів

Логіка Флойда-Хоара [1, 2] є однією з найбільш досліджених та розповсюджених програмних логік. Вона широко використовується для специфікації та верифікації програм. Основою є так звані трійки або ассерції Флойда-Хоара. Вони складаються з передумови, програми і післяумови. Трійка вважається істиною, якщо виконання програми на даних, на яких істина передумова, завершується, то на результуючих даних післяумова теж істинна. Такий простий і інтуїтивний підхід і дозволив логіці Флойда-Хоара набути такого широкого розповсюдження.

З часом все потужніші засоби ставали доступні при створенні програм, серед них вказівники, динамічна пам'ять, об'єкти. Саме тому виникли численні розширення логіки Флойда-Хоара, які б дозволяли доводити

властивості програм, які використовували ці засоби. Серед них Динамічна логіка (Dynamic logic) [3], Логіка розділення (Separation Logic) [4]. Проте в зазначених вище розширеннях, як і в класичному формулюванні, предикати вважаються тотальними. Але це справедливо не для всіх задач, тому у випадку, коли предикати можуть бути частковими, доводиться виконувати додаткові кроки, щоб звести їх до тотальних. Це зумовлює актуальність створення розширення логіки Флойда-Хоара на випадок часткових предикатів. Для цієї мети ми застосуємо композиційно-номінативний підхід. Він полягає в тому, що дані вважаються номінативними та складні предикати і функції утворюються з більш простих за допомогою композицій. За такого підходу поняття ассерції Флойда-Хоара можна задати за допомогою спеціальної композиції, яку ми будемо називати композицією Флойда-Хоара. Вона, приймаючи на вхід програму та два предикати, що задають передумову та післяумову, має результатом предикат, що задає істинність відповідної трійки.

Якщо при визначенні композиції Флойда-Хоара використати класичне означення, то отримана композиція не буде монотонною [5]. Монотонність розуміється як монотонність за включеннями графіків відображень. Вона є однією з найважливіших особливостей композиції Флойда-Хоара, так як дозволяє доводити властивості програм, базуючись на відповідних властивостях апроксимацій програм, що розглядаються. Монотонність гарантує, що при переході від менш визначених даних до більш визначених результат обчислення функцій та значення предикатів не зміняться. Саме тому було визначено спеціальну монотонну композицію Флойда-Хоара, а отримане розширення має назву монотонна логіка Флойда-Хоара.

При визначенні розширення логіки Флойда-Хоара використовується семантико-синтаксичний підхід, який полягає в тому, що спочатку визначається семантика, з якої природно випливає синтаксис як терми відповідної алгебри. Виключення складають лише формули для позначення трійок Флойда-Хоара, для яких використовується усталена нотація. А саме: $\{p\}pr\{q\}$, де p – формула, що відповідає передумові, pr – терм, що відповідає програмі, q – формула, що відповідає післяумові. Для решти випадків в подальшому не проводиться явного розрізнення між формулами та предикатами, які вони позначають, якщо це несуттєво.

Для того щоб визначити семантику, дамо спочатку деякі додаткові означення. Всі поняття, що не визначені в даній роботі, будемо розглядати в смислі [5].

Нехай V – множина імен, A – множина базових значень. Тоді клас номінативних множин ${}^{V}A$ визначається як клас всіх часткових відображень з множини імен V в множину базових значень A. Тобто ${}^{V}A = V \to A$.

В деяких випадках більш природно використовувати позначення, аналогічні позначенню множин. Записом $[v_i \mapsto a_i \mid i \in I]$ позначатимемо номінативну множину, в якій імена v_i мають значення a_i. А $v_i \mapsto a_i \in_n d$ позначатимемо той факт, що в даному d значенням імені v_i є a_i.

Основною функцією для номінативних множин є бінарна функція накладання. Інтуїтивно результатом застосування функції накладання до двох номінативних множин є номінативна множина, яка містить всі пари ім'я-значення з другої множини і ті пари ім'я-значення з першої, для яких в другій множині не існує пари ім'я-значення з відповідним іменем.

$$d_1 \nabla d_2 = [v \mapsto a \mid v \mapsto a \in_n d_2 \vee \\ \vee (v \mapsto a \in_n d_1 \wedge \neg \exists a'(v \mapsto a' \in_n d_2))].$$

Додатково для довільного відображення f та даних d і d' будемо використовувати наступні позначення: $f(d)\downarrow = d'$ – f визначено на d і має значення d', або ж в скороченій формі $f(d)\downarrow$ – f визначено на d; $f(d)\uparrow$ – f невизначено на d; $f[S] = \{f(d) \mid f(d)\downarrow, d \in S\}$ будемо позначати образ S відносно f; $f^{-1}[S'] = \{d \mid f(d)\downarrow, f(d) \in S'\}$ будемо позначати повний прообраз S' відносно f.

Далі дамо визначення класів відображень, що є основами в алгебрі що задає семантику розширення логіки Флойда-Хоара.

$Pr^{V,A} = {}^{V}A \to \{T, F\}$ – множина квазіарних предикатів над ${}^{V}A$, вони, зокрема, задають семантику умов в програмі. $Fn^{V,A} = {}^{V}A \to A$ – множина квазіарних ординарних функцій над ${}^{V}A$, що задають семантику виразів в програмі, $FPrg^{V,A} = {}^{V}A \to {}^{V}A$ – множина біквазіарних функцій над ${}^{V}A$, як варіант подання семантики програм.

Тоді програмні алгебри для довільних V та A, що використовується для подання семантики

монотонної логіки Флойда-Хоара, мають наступний вигляд:

$$PA(V,A) = < Pr^{V,A}, Fn^{V,A}, FPrg^{V,A}; \vee, \neg, \exists x, S_P^{\bar{x}}, S_F^{\bar{x}}, '{}x, AS^x, id, \bullet, IF, WH, FH, PC > .$$

Тепер дамо формальне визначення композиціям для довільних $x, x_1, \ldots, x_n \in V$, $d \in {}^V A$, $p, q, r \in Pr^{V,A}$, $f, g_1, \ldots, g_n \in Fn^{V,A}$, $pr, pr_1, pr_2 \in FPrg^{V,A}$.

Бінарна композиція диз'юнкції $\vee : Pr^{V,A} \times Pr^{V,A} \to Pr^{V,A}$:

$$(p \vee q)(d) = \begin{cases} T, \text{ якщо } p(d)\downarrow = T \text{ або } q(d)\downarrow = T, \\ F, \text{ якщо } p(d)\downarrow = F \text{ та } q(d)\downarrow = F, \\ \text{невизначено інакше.} \end{cases}$$

Унарна композиція заперечення $\neg : Pr^{V,A} \to Pr^{V,A}$:

$$(\neg p)(d) = \begin{cases} F, \text{ якщо } p(d)\downarrow = T, \\ T, \text{ якщо } p(d)\downarrow = F, \\ \text{невизначено інакше.} \end{cases}$$

Унарна композиція квантифікації з параметром $x \in V$, $\exists x : Pr^{V,A} \to Pr^{V,A}$:

$$(\exists x\, p)(d) = \begin{cases} T, \text{ якщо для деякого } a \in A, \\ \quad p(d\nabla[x \mapsto a])\downarrow = T, \\ F, \text{ якщо } p(d\nabla[x \mapsto a])\downarrow = F \\ \quad \text{для всіх } a \in A, \\ \text{невизначено інакше.} \end{cases}$$

n+1-арна параметрична композиція суперпозиції для предикатів з параметрами $x_1, \ldots, x_n \in V$, $S_P^{\bar{x}} : Pr^{V,A} \times \underbrace{Fn^{V,A} \times \ldots \times Fn^{V,A}}_{n} \to Pr^{V,A}$:

$$S_P^{\bar{x}}(p, g_1, \ldots, g_n)(d) \simeq \simeq p(d\nabla[x_1 \mapsto g_1(d), \ldots, x_n \mapsto g_n(d)]),$$

n+1-арна параметрична композиція суперпозиції для функцій з параметрами $x_1, \ldots, x_n \in V$, $S_F^{\bar{x}} : Fn^{V,A} \times \underbrace{Fn^{V,A} \times \ldots \times Fn^{V,A}}_{n} \to Fn^{V,A}$:

$$S_F^{\bar{x}}(f, g_1, \ldots, g_n)(d) \simeq \simeq f(d\nabla[x_1 \mapsto g_1(d), \ldots, x_n \mapsto g_n(d)]),$$

Нуль-арна параметрична композиція деномінації з параметром $x \in V$, $`x : Fn^{V,A}$:

$$`x(d) \simeq d(x) .$$

Унарна параметрична композиція присвоєння з параметром $x \in V$, $AS^x : Fn^{V,A} \to FPrg^{V,A}$:

$$AS^x(f)(d) \simeq d\nabla[x \mapsto f(d)] .$$

Нуль-арна композиція тотожної програми $id : FPrg^{V,A}$:

$$id(d) = d .$$

Бінарна композиція послідовного виконання $\bullet : FPrg^{V,A} \times FPrg^{V,A} \to FPrg^{V,A}$:

$$pr_1 \bullet pr_2(d) \simeq pr_2(pr_1(d)) .$$

Тернарна композиція розгалуження $IF : Pr^{V,A} \times FPrg^{V,A} \times FPrg^{V,A} \to FPrg^{V,A}$:

$$IF(r, pr_1, pr_2)(d) = \begin{cases} pr_1(d), \text{ якщо } r(d)\downarrow = T \\ \quad \text{та } pr_1(d)\downarrow, \\ pr_2(d), \text{ якщо } r(d)\downarrow = F \\ \quad \text{та } pr_2(d)\downarrow, \\ \text{невизначено інакше.} \end{cases}$$

Бінарна циклічна композиція $WH : Pr^{V,A} \times FPrg^{V,A} \to FPrg^{V,A}$:

$WH(r, pr)(d) = d_n$, якщо існують такі $d_1, d_2, \ldots, d_n \in {}^V A$, що $r(d)\downarrow = T$, $pr(d)\downarrow = d_1$, $pr(d_1)\downarrow = d_2, \ldots, pr(d_{n-1})\downarrow = d_n$, $r(d_1)\downarrow = T$, $r(d_n)\downarrow = F$, інакше $WH(r, pr)(d)\uparrow$.

Тернарна монотонна композиція Флойда-Хоара $FH : Pr^{V,A} \times FPrg^{V,A} \times Pr^{V,A} \to Pr^{V,A}$:

$$FH(p, pr, q)(d) = \begin{cases} T, \text{ якщо } p(d)\downarrow = F \\ \quad \text{або } q(pr(d))\downarrow = T, \\ F, \text{ якщо } p(d)\downarrow = T \\ \quad \text{та } q(pr(d))\downarrow = F, \\ \text{невизначено інакше.} \end{cases}$$

Бінарна композиція умови за прообразом $PC : FPrg^{V,A} \times Pr^{V,A} \to Pr^{V,A}$:

$$PC(pr,q)(d)=\begin{cases}T, \text{ якщо } q(pr(d))\downarrow=T,\\ F, \text{ якщо } q(pr(d))\downarrow=F,\\ \text{невизначено інакше.}\end{cases}$$

Композиція Флойда-Хоара є не лише монотонною, але й неперервною [5]. Також монотонними є решта композицій, зокрема, композиція умови за прообразом. Формально умова монотонності композиції Флойда-Хоара може бути записана наступним чином:

$$p\subseteq p', pr\subseteq pr', q\subseteq q' \Rightarrow \\ \Rightarrow FH(p,pr,q)\subseteq FH(p',pr',q').$$

Композиція умови за прообразом є в певною мірою аналогом перетворювача предикатів під назвою найслабкішої передумови, запропонованого Дейкстрою. Основною складністю при перенесенні означення Дейкстри [6] для монотонної логіки Флойда-Хоара є існування декількох можливих визначень логічного наслідку для часткових предикатів. Композиція умови за прообразом використовується для подання додаткових обмежень на правила систем виводу.

Слід зазначити, що композиція присвоєння AS^X відповідає оператору присвоєння := мови WHILE; композиція *id* відповідає оператору *skip*; умовна композиція *IF* відповідає оператору *if_then_else*; циклічна композиція *WH* відповідає оператору *while_do*.

Під істинністю предиката будемо розуміти неспростовність. Те, що предикат всюди істинний, позначатимемо $\models p$. Для того, щоб дати формальне визначення істинності, введемо позначення для областей істинності, хибності та невизначеності довільного предиката $p\in Pr^{V,A}$:

$$p^T=\{d \mid p(d)\downarrow=T\},$$

$$p^F=\{d \mid p(d)\downarrow=F\},$$

$$p^{\perp}=\{d \mid p(d)\uparrow\}.$$

Тоді за визначенням $\models p \Leftrightarrow p_J^F=\varnothing$ для довільної інтерпретації *J*. Також будемо використовувати $p\models q$ для позначення $\models p\to q$. Також визначимо додаткові відношення логічного наслідку для часткових предикатів. Вони будуть використані в наступних розділах в додаткових обмеженнях для правил.

Логічний наслідок за істиною $p\models_T q \Leftrightarrow p_J^T\subseteq q_J^T$ для кожної інтерпретації *J*.

Логічний наслідок за хибою $p\models_F q \Leftrightarrow q_J^F\subseteq p_J^F$ для кожної інтерпретації *J*.

Системи виводу монотонної логіки Флойда-Хоара

Для того, щоб використовувати програмну логіку не лише для специфікації програм, але і для верифікації, необхідно задати систему виводу.

Перед тим, як розглянути приклади систем виводу, потрібно дати додаткові означення.

Будемо позначати $\vdash_{IC} p$, якщо формула *p* може бути виведена в системі виводу *IC*. Або $\vdash p$, якщо система виводу задана неявно.

Система виводу називається коректною, коли для довільної формули $p \vdash p \Rightarrow \models p$. Іншими словами, якщо кожна формула, що може бути виведена, є істинною.

Система виводу називається повною, коли для довільної формули $p \models p \Rightarrow \vdash p$. Іншими словами, якщо кожна істинна формула може бути виведена.

Повнота може розумітись як екстенсіональна або інтенсіональна. За екстенсіональної повноти передумови та післяумови можуть бути довільними предикатами. За інтенсіональної повноти передумови та післяумови мають бути задані формулами деякої заданої мови. В роботі розглядається лише екстенсіональна повнота.

Система виводу *CI* мови WHILE в класичній логіці Флойда-Хоара, подана в [7], є повною та коректною. Проте для випадку часткових предикатів вона перестає бути коректною. Записана для монотонної логіки Флойда-Хоара, система CI має вигляд:

$$\{S_P^x(p,h)\}\, AS^x(h)\, \{p\},$$

$$\{p\}\, id\, \{p\},$$

$$\frac{\{p\}\, pr_1\, \{q\}, \{q\}\, pr_2\, \{r\}}{\{p\}\, pr_1 \bullet pr_2\, \{r\}},$$

$$\frac{\{r\wedge p\}\, pr_1\, \{q\}, \{\neg r\wedge p\}\, pr_2\, \{q\}}{\{p\}\, IF(r, pr_1, pr_2)\, \{q\}},$$

$$\frac{\{r\wedge p\}\, pr\, \{p\}}{\{p\}\, WH(r, pr)\, \{\neg r\wedge p\}},$$

$$\frac{\{p'\}\ pr\ \{q'\}}{\{p\}\ pr\ \{q\}}, p \to p', q' \to q\,.$$

Правила позначатимемо відповідно *R_AS, R_SKIP, R_SEQ, R_IF, R_WH, R_CONS.*

Для часткових предикатів правила *R_SEQ, R_WH* та *R_CONS* не гарантують, що з істинних засновків будуть виведені істинні висновки [5]. Тому система перестає бути коректною при переході до монотонної логіки Флойда-Хоара. Одним з розв'язків цієї проблеми є використання додаткових обмежень для відповідних правил.

Система виводу з додатковими обмеженнями *AC* має наступний вигляд:

$$\{S_P^x(p,h)\}\ AS^x(h)\ \{p\}\,,$$

$$\{p\}\ id\ \{p\}\,,$$

$$\frac{\{p\}\ pr_1\ \{q\},\{q\}\ pr_2\ \{r\}}{\{p\}\ pr_1 \bullet pr_2\ \{r\}}, p \models PC(pr_1 \bullet pr_2, r)\,,$$

$$\frac{\{r \wedge p\}\ pr_1\ \{q\},\{\neg r \wedge p\}\ pr_2\ \{q\}}{\{p\}\ IF(r, pr_1, pr_2)\ \{q\}}\,,$$

$$\frac{\{r\wedge p\}pr\{p\}}{\{p\}WH(r,pr)\{\neg r\wedge p\}}, p \models PC(WH(r,pr),\neg r\wedge p),$$

$$\frac{\{p'\}\ pr\ \{q'\}}{\{p\}\ pr\ \{q\}}, p \models_T p', q' \models_F q\,.$$

Правила позначатимемо відповідно *R_AS, R_SKIP, R_SEQ', R_IF, R_WH', R_CONS'.*

Представлена система виводу з додатковими обмеженнями є повною та коректною[5]. Проте, обмеження для правил дуже складні. Це зумовлює потребу в пошуку систем виводу з спрощеними обмеженнями або іншого підходу до побудови коректної та повної системи.

Розглянемо правила системи виводу *CI*. Всі вони окрім *R_CONS* зберігають наступні властивості ассерцій Флойда-Хоара. Якщо для трійки Флойда-Хоара $\{p\}pr\{q\}$ виконується $p \models_T PC(pr,q)$, будемо називати її *T*-зростаючою. Якщо для трійки Флойда-Хоара $\{p\}pr\{q\}$ виконується $p \models_F PC(pr,q)$, будемо називати її *F*-спадаючою. Неважко переконатись, що *T*-зростаючі та *F*-спадаючі ассерції є істинними.

Теорема. Правила *R_SEQ, R_IF, R_WH* не виводять ассерції з класу *T*-зростаючих (*F*-спадаючих) .

Доведення може бути знайдено в [5] і тому не приводиться.

Наведені вище факти показують, що якщо модифікувати правило *R_CONS* таким чином, щоб воно зберігало або властивість ассерції бути *T*-зростаючою, або *F*-спадаючою, можна отримати коректні системи виводу.

Система виводу *TI* для *T*-зростаючих ассерцій виглядає наступним чином:

$$\{S_P^x(p,h)\}\ AS^x(h)\ \{p\}\,,$$

$$\{p\}\ id\ \{p\}\,,$$

$$\frac{\{p\}\ pr_1\ \{q\},\{q\}\ pr_2\ \{r\}}{\{p\}\ pr_1 \bullet pr_2\ \{r\}}\,,$$

$$\frac{\{r \wedge p\}\ pr_1\ \{q\},\{\neg r \wedge p\}\ pr_2\ \{q\}}{\{p\}\ IF(r, pr_1, pr_2)\ \{q\}}\,,$$

$$\frac{\{r \wedge p\}\ pr\ \{p\}}{\{p\}\ WH(r, pr)\ \{\neg r \wedge p\}}\,,$$

$$\frac{\{p'\}\ pr\ \{q'\}}{\{p\}\ pr\ \{q\}}, p \models_T p', q' \models_T q\,.$$

Правила позначатимемо відповідно *R_AS, R_SKIP, R_SEQ, R_IF, R_WH, R_CONS''.*

Аналогічна система виводу може бути побудована для *F*-спадаючих ассерцій. Так як існують істинні ассерції, що не належать жодному з вищезазначених класів, очевидним є те, що жодна з побудованих систем виводу не є повною. Проте вони є повними для відповідних класів ассерцій.

Таким чином, хоча наведені системи виводу є коректними, але мають суттєві недоліки. Складні додаткові обмеження та повнота лише для певного класу ассерцій ускладнюють використання відповідних систем виводу. Тому знову постає проблема пошуку коректної та повної системи виводу з достатньо простими обмеженнями на правила.

Системи виводу з T- та F-обмеженнями

Розглянемо систему виводу *AC*, основним її недоліком є складність обмежень в правилах *R_SEQ'* та *R_WH'*. Щоб спростити ці обмеження, варто взяти до уваги властивості ассерцій, що були використані при побудові системи *TI*.

Система виводу *TF* з *T*- та *F*-обмеженнями виглядає наступним чином:

$$\{S_P^x(p,h)\}\ AS^x(h)\ \{p\}\,,$$

$$\{p\}\, id\, \{p\},$$

$$\frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1 \bullet pr_2\{r\}}, \quad \begin{matrix} p \models_T PC(pr_1,q) \vee \\ \vee q \models_F PC(pr_2,r) \end{matrix},$$

$$\frac{\{r \wedge p\}\ pr_1\ \{q\},\{\neg r \wedge p\}\ pr_2\ \{q\}}{\{p\}\ IF(r, pr_1, pr_2)\ \{q\}},$$

$$\frac{\{r \wedge p\} pr \{p\}}{\{p\} WH(r, pr) \{\neg r \wedge p\}}, \quad \begin{matrix} r \wedge p \models_T PC(pr,p) \vee \\ \vee r \wedge p \models_F PC(pr,p) \end{matrix},$$

$$\frac{\{p'\}\ pr\ \{q'\}}{\{p\}\ pr\ \{q\}}, p \models_T p', q' \models_F q.$$

Правила позначатимемо відповідно *R_AS, R_SKIP, R_SEQ'', R_IF, R_WH'', R_CONS'.* Для зручності правила *R_SEQ''* та *R_WH''* можна розділити на два правила кожне:

$$\frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1 \bullet pr_2\{r\}}, p \models_T PC(pr_1,q),$$

$$\frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1 \bullet pr_2\{r\}}, q \models_F PC(pr_2,r),$$

$$\frac{\{r \wedge p\} pr \{p\}}{\{p\} WH(r, pr) \{\neg r \wedge p\}}, r \wedge p \models_T PC(pr,p),$$

$$\frac{\{r \wedge p\} pr \{p\}}{\{p\} WH(r, pr) \{\neg r \wedge p\}}, r \wedge p \models_F PC(pr,p).$$

Для позначення розділених правил будемо використовувати відповідно *R_SEQ_T, R_SEQ_F, R_WH_T, R_WH_F.*

Теорема. Система виводу *TF* коректна.

Доведемо індукцією за довжиною виводу.

База індукції.

Для правила *R_AS* потрібно довести $\models \{S_P^x(p,h)\}\, AS^x(h)\, \{p\}$. Іншими словами, потрібно показати, що не існує такого $d \in {}^V A$, що $S_P^x(p,h)(d)\downarrow = T$ та $p(AS^x(h)(d))\downarrow = F$. Проте за визначенням $S_P^x(p,h)(d) = p(d\nabla[x \mapsto h(d)])$ та $p(AS^x(h)(d)) = p(d\nabla[x \mapsto h(d)])$. Що і доводить неможливість існування такого $d \in {}^V A$.

Для правила *R_SKIP* потрібно довести $\models \{p\} id \{p\}$. Це очевидно, бо для довільного даного $d \in {}^V A$, $id(d) = d$, звідки $p(id(d)) = p(d)$.

Крок індукції.

Для правила *R_SEQ_T* потрібно довести, що маючи $\models \{p\}pr_1\{q\}$ та $\models \{q\}pr_2\{r\}$ разом з $p \models_T PC(pr_1,q)$, отримуємо $\models \{p\}pr_1 \bullet pr_2\{r\}$.

Розглянемо довільне $d \in {}^V A$, таке, що $p(d)\downarrow = T$. З $p \models_T PC(pr_1,q)$ отримуємо $q(pr_1(d))\downarrow = T$. А з $\models \{q\}pr_2\{r\}$ отримуємо, що неможливо, щоб $r(pr_2(pr_1(d)))\downarrow = F$. Разом з тим, що за визначенням $pr_2(pr_1(d)) = pr_1 \bullet pr_2(d)$ це доводить $\models \{p\}pr_1 \bullet pr_2\{r\}$.

Для правила *R_SEQ_F* доведення аналогічне.

Для правила *R_IF* потрібно довести, що маючи $\models \{r \wedge p\}pr_1\{q\}$ та $\models \{\neg r \wedge p\}pr_2\{q\}$, отримуємо $\models \{p\} IF(r, pr_1, pr_2)\{q\}$.

Візьмемо довільне $d \in {}^V A$, таке, що $p(d)\downarrow = T$. Розглянемо декілька випадків значень $r(d)$. Якщо $r(d)\downarrow = T$, тоді $(r \wedge p)(d)\downarrow = T$. А отже з $\models \{r \wedge p\}pr_1\{q\}$ отримаємо, що неможливо, щоб $q(pr_1(d))\downarrow = F$. Також за умови $r(d)\downarrow = T$, маємо $IF(r, pr_1, pr_2)(d) = pr_1(d)$. Таким чином при $r(d)\downarrow = T$ неможливо, щоб $q(IF(r, pr_1, pr_2)(d))\downarrow = F$. У випадку $r(d)\downarrow = F$ все аналогічно. А якщо $r(d)\uparrow$, то $IF(r, pr_1, pr_2)(d)\uparrow$. Отже, маємо $\models \{p\} IF(r, pr_1, pr_2)\{q\}$.

Для правила *R_WH_T* потрібно довести, що маючи $\models \{r \wedge p\} pr \{p\}$ та $r \wedge p \models_T PC(pr,p)$, отримуємо $\models \{p\} WH(r, pr)\{\neg r \wedge p\}$.

Розглянемо довільне $d \in {}^V A$, таке, що $p(d)\downarrow = T$. Випадок, коли $WH(r, pr)(d)\uparrow$ тривіальний. Якщо $WH(r, pr)(d)\downarrow = d_n$, то за визначенням існують елементи $d_0, d_1, \ldots, d_n$. Для них виконується $d_0 = d, d_1 = pr(d_0), \ldots, d_n = pr(d_{n-1})$ разом з $r(d_0)\downarrow = T, \ldots, r(d_{n-1})\downarrow = T, r(d_n)\downarrow = F$. Тоді за $r \wedge p \models_T PC(pr,p)$ маємо $p(d_1)\downarrow = T, \ldots, p(d_n)\downarrow = T$. Таким чином $(\neg r \wedge p)(d_n)\downarrow = (\neg r \wedge p)(WH(r, pr)(d)) = T \neq F$. Це доводить $\models \{p\} WH(r, pr)\{\neg r \wedge p\}$.

Для правила *R_WH_F* доведення аналогічне.

Для правила *R_CONS'* потрібно довести, що маючи $\models \{p'\} pr \{q'\}$ разом з $p \models_T p'$ та $q' \models_F q$, отримуємо $\models \{p\} pr \{q\}$.

Розглянемо $d \in {}^V A$, довільний такий, що $p(d)\downarrow = T$. З $p \models_T p'$ отримуємо $p'(d)\downarrow = T$. Якщо $q(pr(d))\downarrow = F$, тоді з $q' \models_F q$ маємо $q'(pr(d))\downarrow = F$. Проте це неможливо, бо

$\models\{p'\}pr\{q'\}$, отже невірно, що $q(pr(d))\downarrow=F$ і отримуємо $\models\{p\}pr\{q\}$.

Було розглянуто всі правила, отже кожна ассерція, що виводиться, є істинною і система виводу *TF* коректна. Що і треба було довести.

Теорема. Система виводу *TF* екстенсіонально повна.

Спочатку індукцією за будовою програми доведемо, що для довільних $pr\in FPrg^{V,A}$, $q\in Pr^{V,A}$ виконується $\vdash\{PC(pr,q)\}pr\{q\}$.

База індукції.

Неважко перевірити, що для композицій тотожної програми та присвоєння справедливо $PC(id,q)=q$ та $PC(AS^x(h),q)=S_p^x(q,h)$ для довільних $q\in Pr^{V,A}$, $h\in Fn^{V,A}$ та $x\in V$. Тому використовуючи правила *R_AS, R_SKIP*, маємо $\vdash\{PC(id,q)\}id\{q\}$ і $\vdash\{PC(AS^x(h),q)\}AS^x(h)\{q\}$.

Крок індукції.

Розглянемо композицію послідовного виконання. Потрібно показати, що $\vdash\{PC(pr_1\bullet pr_2,q)\}pr_1\bullet pr_2\{q\}$. За припущенням індукції $\vdash\{PC(pr_2,q)\}pr_2\{q\}$ та $\vdash\{PC(pr_1,PC(pr_2,q))\}pr_1\{PC(pr_2,q)\}$. Тоді за правилом *R_SEQ_T або R_SEQ_F* отримуємо $\vdash\{PC(pr_1,PC(pr_2,q))\}pr_1\bullet pr_2\{q\}$. Додаткові умови виконуються, так як для *R_SEQ_F* умова має вигляд $PC(pr_2,q)\models_F PC(pr_2,q)$. Якщо $PC(pr_1\bullet pr_2,q)\models_T PC(pr_1,PC(pr_2,q))$, то застосовуючи правило *R_CONS'*, отримаємо шукане твердження $\vdash\{PC(pr_1\bullet pr_2,q)\}pr_1\bullet pr_2\{q\}$.

Щоб довести потрібний логічний наслідок, розглянемо довільне $d\in{}^VA$, таке, що $PC(pr_1\bullet pr_2,q)(d)\downarrow=T$. За визначенням, маємо $q(pr_1\bullet pr_2(d))\downarrow=T$, або $q(pr_2(pr_1(d)))\downarrow=T$. Згортаючи за визначення композиції передумови за прообразом $(PC(pr_2,q))(pr_1(d))\downarrow=T$ і $PC(pr_1,PC(pr_2,q))(d)\downarrow=T$, що і доводить $PC(pr_1\bullet pr_2,q)\models_T PC(pr_1,PC(pr_2,q))$.

Розглянемо умовну композицію. Потрібно показати, що виконується наступне: $\vdash\{PC(IF(r,pr_1,pr_2),q)\}IF(r,pr_1,pr_2)\{q\}$. За припущенням індукції $\vdash\{PC(pr_1,q)\}pr_1\{q\}$ та $\vdash\{PC(pr_2,q)\}pr_2\{q\}$. Якщо довести $r\wedge PC(IF(r,pr_1,pr_2),q)\models_T PC(pr_1,q)$ та $\neg r\wedge PC(IF(r,pr_1,pr_2),q\models_T PC(pr_2,q)$, то використовуючи правила *R_CONS'* та *R_IF*, отримаємо шукане твердження.

Доведемо першу умову, друга умова доводиться аналогічно. Візьмемо довільне $d\in{}^VA$, таке, що $r\wedge PC(IF(r,pr_1,pr_2),q)(d)\downarrow=T$. Тоді маємо $r(d)\downarrow=T$ і $q(IF(r,pr_1,pr_2)(d))\downarrow=T$. Так як $r(d)\downarrow=T$, то $IF(r,pr_1,pr_2)(d)=pr_1(d)$, а отже $q(pr_1(d))\downarrow=T$, або $PC(pr_1,q)(d)\downarrow=T$, що і треба було довести.

Розглянемо циклічну композицію. Потрібно показати, що $\vdash\{PC(WH(r,pr),q)\}WH(r,pr)\{q\}$. Позначимо для зручності $p=PC(WH(r,pr),q)$. Розглянемо предикат p', такий, що $p'(d)\downarrow=T\Leftrightarrow p(d)\downarrow=T$ і $p'(d)\downarrow=F$ у всіх інших випадках. За припущенням індукції $\vdash\{PC(pr,p')\}pr\{p'\}$. Якщо довести, що $r\wedge p'\models_T PC(pr,p')$, тоді за правилом *R_CONS'* отримаємо $\vdash\{r\wedge p'\}pr\{p'\}$, звідки за правилом *R_WH_T* маємо $\vdash\{p'\}WH(r,pr)\{\neg r\wedge p'\}$. Тоді якщо $p\models_T p'$ та $\neg r\wedge p'\models_F q$ за *R_CONS'* маємо $\vdash\{PC(WH(r,pr),q)\}WH(r,pr)\{q\}$.

Твердження $p\models_T p'$ виконується за побудовою p'.

Доведемо $\neg r\wedge p'\models_F q$. Візьмемо деяке дане d, таке, що $q(d)\downarrow=F$, можливо три випадки відносно значення предикату r.

Якщо $r(d)\downarrow=T$, то $\neg r\wedge p'(d)\downarrow=F$.

Якщо $r(d)\downarrow=F$, то $WH(r,pr)(d)\downarrow=d$ і тоді $p(d)\downarrow=PC(WH(r,pr),q)(d)=q(d)=F$. Звідки $p'(d)\downarrow=p(d)=F$, а отже $\neg r\wedge p'(d)=F$.

Якщо $r(d)\uparrow$, то $WH(r,pr)(d)\uparrow$, а отже і $p(d)\uparrow$. Звідки $p'(d)\downarrow=F$, а отже $\neg r\wedge p'(d)=F$.

Для всіх випадків отримали $\neg r\wedge p'(d)=F$, отже $\neg r\wedge p'\models_F q$.

Доведемо $r\wedge p'\models_T PC(pr,p')$. Якщо для деякого даного d виконується $(r\wedge p')(d)\downarrow=T$, то відповідно $r(d)\downarrow=T$ і $p(d)\downarrow=p'(d)=T$. За визначенням p маємо $q(WH(r,pr)(d))\downarrow=T$. Отже $WH(r,pr)(d)\downarrow$, що разом з $r(d)\downarrow=T$ дає існування стану $d'=pr(d)$ такого, що $WH(r,pr)(d')\downarrow=WH(r,pr)(d)$. Тоді за визначення композиції передумови за

прообразом отримуємо $p(d')\downarrow = T$, а отже і $p'(d') = p'(pr(d))\downarrow = T$, або $PC(pr, p')(d)\downarrow = T$.

Було розглянуто всі композиції, отже $\vdash \{PC(pr,q)\}pr\{q\}$ для довільних *pr* та *q*.

Для довільної істинної ассерції $\models \{p\}pr\{q\}$ можна визначити предикат q', наступним чином:

$$q'(d) = \begin{cases} T, \text{ якщо } p(pr^{-1}(d))\downarrow = T, \\ F, \text{ якщо } q(d)\downarrow = F, \\ \text{невизначено інакше.} \end{cases}$$

Такий предикат можливий завдяки визначення істинності, що гарантує однозначність $q'(d)$ для довільного даного. Неважко перевірити, що $p \models_T PC(pr, q')$ та $q' \models_F q$. За доведеним раніше $\vdash \{PC(pr,q')\}pr\{q'\}$. Тоді за правилом *R_CONS'* можна отримати $\vdash \{p\}pr\{q\}$, а отже система виводу повна. Що і треба було довести.

Висновки

В роботі було розглянуто монотонну логіку Флойда-Хоара. Особливу увагу було приділено системам виводу. Наведено класичну систему виводу, яка перестає бути коректною при застосуванні для часткових предикатів. Також розглянуто систему виводу з додатковими обмеженнями та систему виводу для класів *T*-зростаючих та *F*-спадаючих ассерцій. Відповідні системи коректні та екстенсіонально повні, проте мають ряд недоліків.

Побудовано систему виводу з T- та F-обмеженнями, яка була позбавлена від більшості з приведених недоліків. Для побудованої системи доведено коректність та екстенсіональну повноту. Проте, проблема побудови систем виводу з більш простими обмеженнями залишається актуальною.

Список використаних джерел

1. *Floyd R.W.* Assigning meanings to programs / Floyd R.W. // Proceedings of the American Mathematical Society Symposia on Applied Mathematics. – 1967. – Vol. **19**. – P. 19-31.
2. *Hoare C.A.R.* An axiomatic basis for computer programming / Hoare C.A.R. // Communications of ACM. – 1969. – Issue **12**. – P. 578-580,583.
3. *Harel D., Kozen D., Tiuryn J.* Dynamic logic / Harel D., Kozen D., Tiuryn J. // Handbook of Philosophical Logic. – 1984. – P. 497-604.
4. *Reynolds J.C.* Separation logic: A logic for shared mutable data structures / Reynolds J.C. // LICS'02. – 2002. – P. 55-74.
5. *Kryvolap A., Nikitchenko M., Schreiner W.* Extending Floyd-Hoare logic for partial pre- and postconditions / Kryvolap A., Nikitchenko M., Schreiner W // ICTERI 2013, CCIS.– Springer, Heidelberg. – 2013. – Vol. **412**. –P. 355-378.
6. *Edsger W. Dijkstra*. Guarded commands, nondeterminacy and formal derivation of program / Edsger W. Dijkstra // Communications of the ACM. – 1975. – №**18(8)**. – P. 453-457.
7. *Nielson H.R., Nielson F.* Semantics with applications: a formal introduction / Nielson H.R., Nielson F. – John Wiley & Sons Inc. – 1992. – 240 p.

References

1. FLOYD, R.W. (1967) Assigning meanings to programs. In *Proceedings of the American Mathematical Society Symposia on Applied Mathematics.* vol. 19 p. 19-31.
2. HOARE, C.A.R. (1969) An axiomatic basis for computer programming. In *Communications of ACM*. isssue 12 p. 578-580,583.
3. HAREL, D., KOZEN, D. and TIURYN, J. (1984) Dynamic logic. In *Handbook of Philosophical Logic.* P. 497-604.
4. REYNOLDS, J.C. (2002) Separation logic: A logic for shared mutable data structures. In LICS'02. P. 55-74.
5. KRYVOLAP, A., NIKITCHENKO, M. and SCHREINER, W. (2013) Extending Floyd-Hoare logic for partial pre- and postconditions. In ICTERI 2013, CCIS vol. 412 P. 355-378.
6. DIJKSTRA, E. (1975) Guarded commands, nondeterminacy and formal derivation of program. *Communications of the ACM*. 18(8) p. 453-457.
7. NIELSON, H.R. and NIELSON, F. (1992) *Semantics with applications: a formal introduction*. John Wiley & Sons Inc.

Надійшла до редколегії 01.07.14