Київський національний університет імені Тараса Шевченка

На правах рукопису

Криволап Андрій Володимирович

УДК 004.42

ПОБУДОВА ТА ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЛОГІК

01.05.01 – теоретичні основи інформатики та кібернетики

Дисертація на здобуття наукового ступеня
кандидата фізико-математичних наук

Науковий керівник:
Нікітченко Микола Степанович
доктор фіз.-мат. наук, професор

Київ – 2016

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

$\xrightarrow{t}$	тотальне відображення
$\xrightarrow{p}$	часткове відображення
$\xrightarrow{n}$	відображення зі скінченним графіком
$f(d)\uparrow$	відображення f не визначено на даному d
$f(d)\downarrow$	відображення f визначено на даному d
$f(d)\downarrow= a$	відображення f визначено на даному d та рівне a
$f[A]$	повний образ множини A за відображенням f
$f^{-1}[A]$	повний прообраз множини A за відображенням f
;	узагальнена рівність
T	булеве значення істини
F	булеве значення хиби
∇	операція накладання
$\in_n$	відношення належності номінативній множині
$\varnothing$	порожня множина
p^T	область істинності предиката p
p^F	область хибності предиката p
$p^{\perp}$	область невизначеності предиката p
$\bigcup_i f_i$	границя ланцюга f_i
$\vdash$	відношення вивідності
$\models$	відношення істинності
$\models_T$	відношення логічного наслідку за істиною
$\models_F$	відношення логічного наслідку за хибою

ВСТУП

Актуальність теми дослідження. Проблема специфікації та автоматизації доведення коректності програмних систем є складною, але важливою для побудови надійного програмного забезпечення. Для підвищення надійності таких систем були створені різні методи їх розробки, зокрема, і формальні методи.

Більшість існуючих формальних методів (B-метод, Z, RAISE, VDM, TLA, та інші) надають засоби підтримки доведення властивостей програмних систем. Такі засоби в своїй основі мають програмні логіки.

Розробку програмних логік в своїх роботах розпочав T. Hoare, а програмологія як напрямок інформатики бере свій початок в дослідженнях В.Н. Редька. Дослідженнями програмних логік та формальних методів в цілому активно займаються як вітчизняні вчені (С.В. Єршов, Г.М. Жолткевич, С.Л. Кривий, О.А. Летичевський, А.М. Чеботарьов та інші) так і зарубіжні вчені (L. Lamport, J. Reynolds, D. Harel, E. Clarke, D. Sannella, A. Tarlecki, В.А. Непомнящий та інші).

Сучасні програмні логіки в тій чи іншій мірі базуються на логіці Флойда-Хоара. В таких логіках розглядаються предикати та функції скінченної арності, причому предикати мають бути всюди визначеними. Існуючі розширення логіки Флойда-Хоара також базуються на припущенні, що розглядаються лише всюди визначені предикати. Проте при доведенні властивостей конкретних програмних систем не завжди доводиться мати справу з тотальними предикатами, іноді властивості можуть бути невизначеними на окремих даних. В такому випадку потрібно використовувати спеціальні методи для того, щоб звести задачу до випадку тотальних предикатів.

Програмні логіки з частковими предикатами повинні базуватись на адекватних моделях програм. В дисертаційній роботі використовуються композиційні-номінативні моделі програм, які розробляються на кафедрі теорії та технології програмування факультету кібернетики Київського національного університету імені Тараса Шевченка. У цих моделях стани програм подаються номінативними (іменними) даними, семантика програм подається відображеннями над номінативними даними, зокрема квазіарними відображеннями, а засоби конструювання програм – композиціями відображень. Композиційно-номінативні моделі програм дозволяють адекватним чином формалізувати різні мови специфікацій та програмування, а також виступають формальною основою відповідних програмних логік.

Вказані обставини обґрунтовують актуальність побудови програмних логік квазіарних часткових предикатів та функцій над різними класами номінативних даних, дослідження аксіоматичних систем побудованих логік та їх властивостей.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота є складовою частиною наукових робіт, які ведуться на кафедрі теорії та технології програмування Київського національного університету імені Тараса Шевченка при виконанні фундаментальних та прикладних тем: “Розробка конструктивних математичних формалізмів для інтелектуальних систем прийняття рішень, обробки знань, еталонування мов сучасних СУБД та CASE-засобів” (№ 0106U005856, 2006-2010 рр.), “Формальні специфікації та методи розробки надійних програмних систем” (№ 0111U007052, 2011-2015 рр.), “Розробка логіко-алгоритмічних методів дослідження формальних моделей природних мов” (№ 0116U004780, 2016-2018 рр.).

Мета і завдання дисертаційного дослідження. *Метою* дисертаційної роботи є розробка програмних логік композиційно-номінативного типу для використання отриманих формальних систем в задачах специфікації та верифікації програмних систем.

Із огляду на мету в роботі ставляться такі задачі:

- побудова програмної алгебри композиційно-номінативного типу для простих номінативних даних;
- встановлення властивостей композицій програмного рівня побудованої програмної алгебри;
- визначення композиційно-номінативної програмної логіки часткових функцій та предикатів;

- розгляд існуючих аксіоматичних систем логіки Флойда-Хоара для побудованої програмної логіки, дослідження коректності та повноти;
- побудова коректних та повних аксіоматичних систем з додатковими обмеженнями;
- застосування отриманих результатів для дослідження програмних логік та аксіоматичних систем для складних ієрархічних номінативних даних та типізованих номінативних даних.

Об’єктом дисертаційного дослідження є програмні логіки часткових предикатів та функції. Предметом дослідження є аксіоматичні системи для програмних логік, питання їх повноти та коректності.

У роботі використовуються наступні *методи*: теоретико-множинні, теорії програмування, математичної логіки, композиційно-номінативного підходу.

Наукова новизна одержаних результатів. Введено нові програмні алгебри з узагальненими композиціями Флойда-Хоара та досліджено їх властивості. Доведено монотонність та неперервність композицій запропонованих програмних алгебр.

На основі розробленої програмної алгебри вперше побудовано програмні логіки композиційно-номінативного типу для простих номінативних даних, складних ієрархічних номінативних даних та типізованих номінативних даних.

Доведено що класичні аксіоматичні системи для побудованих програмних логік часткових предикатів не є коректними. Вперше розроблено спеціальні аксіоматичні системи з простими додатковими обмеженнями, аксіоматичні системи з обмеженнями, що базуються на властивостях тверджень, які виводяться. Доведено теореми про повноту та коректність побудованих аксіоматичних систем для випадків простих та складних номінативних даних.

Теоретичне та практичне значення одержаних результатів. Дисертація має теоретичну спрямованість. Теоретичне значення роботи полягає в дослідженні програмного рівня композиційно-номінативних логік для різних типів номінативних даних, визначенні основних властивостей композицій програмного рівня.

Отримані результати були впроваджені у навчальний процес за спеціальністю “Інформатика” на факультеті кібернетики Київського національного університету імені Тараса Шевченка (нормативні курси “Теорія програмування”, “Формальні методи розробки програмних систем”).

Практичне значення роботи полягає в застосуванні побудованих коректних аксіоматичних систем для доведення властивостей програмних систем, та дослідженні повноти відповідних аксіоматичних систем.

Особистий внесок здобувача. Всі результати дисертаційної роботи отримані автором самостійно, сформульовані у вигляді теорем та строго доведені з використанням допоміжних лем та тверджень, обґрунтовані з посиланнями на використані джерела. Роботи [66, 67, 69, 71] написані здобувачем особисто і всі результати, що містяться в них є оригінальними. Роботи [30, 31, 43, 44, 68, 70, 75, 77, 78] написані у співавторстві з науковим керівником Нікітченком М.С. У них керівникові належать теоретичні ідеї і постановки задач, а дисертанту – практична реалізація та доведення сформульованих теорем.

Апробація результатів дослідження. Основні положення та висновки дисертаційного дослідження обговорювались на наукових семінарах кафедри теорії та технології програмування Київського національного університету імені Тараса Шевченка.

Результати дисертаційного дослідження оприлюднені у доповідях і повідомленнях на Міжнародних та Всеукраїнських наукових конференціях, семінарах: XVIII Міжнародній конференції “Knowledge–Dialogue–Solution” (KDS 2012, Київ, Україна, 2012 р.), XVI Міжнародній конференції “Dynamical System Modelling and Stability Investigation” (DSMSI 2013, Київ, Україна, 2013 р.), X Міжнародній конференції “Theoretical and Applied Aspects of Program Systems Development” (TAAPSD’2013, Ялта, Україна, 2013 р.), II Міжнародній конференції “Mathematics of Informational Modeling ” (MIM 2013, Варна, Болгарія, 2013 р.), IX Міжнародній конференції “Information and Communications Technology in Education, Research and Industrial Applications” (ICTERI 2013, Херсон, Україна, 2013 р.), 12-th International Conference on Informatics

(INFORMATICS’2013, Спішська Нова Вес, Словаччина, 2013 р.), XI Міжнародній конференції “Theoretical and Applied Aspects of Program Systems Development” (TAAPSD’2014, Київ, Україна, 2014 р.), Всеукраїнській науково-практичній конференції “В.М. Глушков – Піонер Кібернетики” (Київський національний технічний університет “КПІ”, Київ, Україна, 2014 р.)[30, 31, 43, 66, 67, 69, 70, 77].

Публікації. Основні результати дисертаційного дослідження опубліковано у 13 працях. Серед них – 5 статей у наукових журналах і збірниках наукових праць, з них 3 статті опубліковано у фахових виданнях, затверджених ВАК України [71,75,78], 2 статті опубліковано у наукових фахових іноземних журналах [44, 68]; 8 тез та праць конференції.

Структура та обсяг дисертації. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел (80 найменувань). Загальний обсяг дисертації становить 130 с., основний зміст викладено на 118 с.

РОЗДІЛ 1
РОЗШИРЕННЯ ЛОГІКИ ФЛОЙДА-ХОАРА

Складність програмних систем, як і кількість технологій, що використовуються при їх розробці з кожним роком лише зростає, це зумовлює потребу в дослідженні методів розробки коректного програмного забезпечення. Велика кількість різноманітних формальних методів[10, 63] є тому підтвердженням. Найбільш розробленими та широко використовуваними є B-метод[1], Z[62], RAISE[6], VDM[25], TLA[32]. Всі зазначені методи мають засоби для специфікації програмних систем та доведенні властивостей систем, що розглядаються. Тому з кожним з методів пов'язана своя мова специфікації. Важливою частиною формальних методів також є можливість отримання коду програми за її специфікацією. Слід окремо виділити TLA як метод, що направлено в першу чергу на розробку саме розподілених програмних систем.

Одним з основних підходів до подання специфікацій програмних систем та їх верифікація є використання систем автоматичного доведення теорем, базуючись на програмних логіках, та їх аксіоматичних системах. Переважна частина сучасних програмних логік містить в своїй основі поняття трійки Хоара, вперше запропоноване в [23], спираючись на ідеї Флойда запропоновані в [17]. Основна ідея полягає в тому, що розглядаються твердження вигляду $\{p\}pr\{q\}$, де p – передумова, q – післяумова та pr – програма. Для таких тверджень вводиться поняття тотальної та часткової коректності. Трійка називається частково коректною, якщо на всіх даних для яких передумова вірна, програма закінчує виконання та видає дані, на яких вірна післяумова, або ж програма не завершується. Тотальна коректність вводиться аналогічно, лише вимагається, щоб програма завершувалась на даних, де передумова виконується. Слід зазначити, що умови розглядались лише як тотальні, в той час як питання завершення програми виносилось окремо і програма могла бути невизначеною на деяких вхідних даних.

Було також запропоновано логіку Флойда-Хоара, в основі якої лежить поняття трійки Хоара, та побудовано коректну та відносно повну аксіоматичну систему.

Ідея використання трійок, що складаються з передумови, програми та післяумови для представлення специфікацій програмних систем дозволяє адекватно розглядати властивості програм та доводити їх коректність. Це зумовило розробку подальших розширень логіки Флойда-Хоара[2, 11, 12], в яких більш детально розглядаються такі аспекти програмних систем, як паралельні обчислення. Одним з таких розширень є запропонована Гріес та Овікі[52] логіка для розподілених система. Для цього було введено новий оператор, що дозволяє задавати програму, що складається з окремих програм, які виконуються паралельно. Також аксіоматична система була поповнена правилом, що дозволяє довівши твердження відносно паралельних складових, за умови, що вони не впливають на значення змінних одна одної, вивести твердження для програми в цілому.

В своїй роботі[13] Дейсктра запропонував семантику перетворювачів предикатів, новий підхід до подання семантики логіки Флойда-Хоара. Хоча аксіоматична система для програмної логіки вже була побудована, проте саме використання перетворювачів предикатів дозволило сформулювати алгоритм побудови виводу в аксіоматичній системі.

Основні перетворювачі предикатів введені Дейкстрою – це найслабкіша передумова та найсильніша післяумова. Найслабкіша передумова для зафіксованих програми та предикату, що задає післяумову, $wp(pr,q)$ визначається як предикат p, такий, що трійка $\{p\}pr\{q\}$ є всюди істинною та він є логічним наслідком довільної іншої передумови для якої трійка Флойда-Хоара коректна. Аналогічно, якщо задано програму та предикат, найсильніша післяумова $sp(pr,p)$ визначається як предикат q, такий, що трійка $\{p\}pr\{q\}$ є всюди істинною та будь-який предикат, що є коректною післяумовою для відповідної трійки, логічним наслідком найсильнішої післяумови. Також було запропоновано перетворювачі предикатів, що відповідають частковій коректності трійок. Якщо зафіксувати програму, то перетворювачі предикатів найслабкішої передумови є відображенням з післяумов в передумови.

Дейкстрою в своїй роботі також наведено формули, що дозволяють за будовою програми для зафіксованої мови програмування, обчислювати передумови за післяумовами. Це дає змогу проблему істинності трійки Флойда-Хоара зводити до проблеми істинності формули першого порядку використовуючи результат, що трійка $\{p\}pr\{q\}$ істинна, якщо істинна формула $p \rightarrow wp(pr,q)$.

Використовуючи ідеї Дейкстри, Лампортом[33] було запропоновано розширити ідею перетворювачів предикатів на пошук інваріантів для програм. Введено перетворювачі предикатів найсильніший інваріант та найслабкіший інваріант. Ці перетворювачі базуються на перетворювачі найсильнішої передумови та найслабкішої післяумови відповідно. Також продемонстровано використання запропонованих перетворювачів предикатів для доведення властивостей розподілених систем.

Окрім розподілених обчислень, іншим важливим аспектом програмних систем, що потребував подальших уточнень логіки Флойда-Хоара є використання динамічної пам’яті, вказівників. Розгляд проблем пов’язаних з роботою із динамічною пам’яттю призвів до виникнення розширення логіки Флойда-Хоара, логіки розділення[48,57]. В логіці розділення основна увага приділяється стану програми в конкретний момент обчислення. Під станом розуміється сукупність стеку змінних та області динамічної пам’яті. Динамічна пам'ять розглядається як часткове функціональне відображення з натуральних чисел в натуральні числа. Всі умови розглядаються як твердження відносно стану стеку змінних та динамічної пам’яті.

Для того, щоб оперувати з вимогами, та по аналогії з розподіленими системами, мати змогу задавати умови того, що складові програми не будуть впливати на частини динамічної пам’яті, що використовуються іншими частинами, було введено додаткові логічні зв'язки: розділяюча кон'юнкція, розділяюча імплікація. Розділяюча кон'юнкція визначається наступним чином: якщо динамічну пам'ять можна розділити на дві частини, що не перетинаються та кожне з тверджень, що є аргументами розділяючої кон'юнкції виконується на своїй частині динамічної пам'яті, то і твердження, що задається кон'юнкцією виконується. Розділяюча імплікація вірна, якщо можна до області пам'яті на якій виконується засновок додати область динамічної пам'яті, що не перетинається з нею та на отриманій області пам'яті буде вірним засновок.

Окрім введених логічних зв'язок також запропоновано правило рамки для аксіоматичної системи, що дозволяє виводити твердження в яких застосовуються розділяюча кон'юнкція. Зміст даного правила виводу в тому, що якщо певна трійка вірна, то до передумови та післяумови можна додати використовуючи розділяючу кон'юнкцію однакові предикати за умови, що програма не змінює значення вільних змінних для доданого предикату.

Підхід використаний при визначені логіки розділення, а саме введення станів програми має спільні риси з визначенням номінативних множин, проте основною відмінністю є те, що виділяються окремо дві структури – стек змінних та динамічна пам'ять, кожна з яких має своє відображення, та розглядаються проблеми не іменування, а взаємодії областей пам'яті між собою. Проте, розглядаються лише тотальні предикати.

Логіка розділення заснована на логіці згрупованих імплікацій[47, 51]. Логіка згрупованих імплікацій надає засоби для розгляду тверджень відносно ресурсів системи, їх складу. Виводи в такій логіці представлені у вигляді деревовидних структур. Дана логіка є різновидом інтуїціоністської логіки, поєднуючи два підходи до визначення імплікацій та їх властивостей.

Логіка розділення використовується для доведенні коректності алгоритмів пов'язаних зі складними структурами даних та мовами програмування високого рівня. За допомогою логіки розділення можуть бути формалізовані такі складні поняття як множинне успадкування в об'єктно-орієнтованих мовах програмування[35].

Логіка розділення також може бути використана для доведення асерцій Флойда-Хоара для символьної динамічної пам'яті у випадку програм без циклів[50]. Такий результат дозволяє використовувати логіку розділення в перевірці на моделях.

Розгляд ділянок динамічної пам'яті, що не перетинаються дозволяє природнім чином використовувати логіку розділення для доведення властивостей розподілених систем. Для цього необхідно ввести правило, що дозволяє довести окремо необхідні твердження для окремих процесів, що виконуються незалежно, за умови, що ділянки динамічної пам'яті, з якими вони працюють, не перетинаються. Сформулювати відповідне твердження у вигляді правила дозволяє використання розділеної кон'юнкції, яка може бути застосована до передумов та післяумов асерцій, що стосуються окремих процесів. Відповідна логіка отримала назву розподіленої логіки розділення[49]. Слід також зазначити, що така логіка найбільш адекватно може бути застосована до випадку паралельних процесів, що не дуже тісно пов'язані. Семантика розподіленої логіки розділення була сформульована в [8], формалізовано поняття, що один з процесів володіє певним ресурсом.

Бібліотеки з реалізацією логіки розділення запропоновані для таких систем підтримки доведення теорем як Coq[64] та HOL.

Іншим підходом до переосмислення логіки Флойда-Хоара є Динамічна логіка[18, 20, 27]. Динамічна логіка є модальною логікою, в якій кожному оператору pr відповідає модальний оператор $[pr]$. Тоді твердження $[pr]p$ позначає, що після виконання оператора pr має обов'язково виконуватись p. Також вводиться дуальний модальний оператор $<pr>$, такий, що $<pr>p$ позначає, що після виконання оператора pr можливо виконуватиметься p. Перші ідеї, що лягли в основу динамічної логіки були закладені Праттом[55], та полягали в тому, щоб задавати семантику операторів як набори пар вхідний-вихідний стан.

Для побудови складних програм з більш простих в динамічній логіці розглядаються регулярні оператори, запропоновані Кліні. Основними операторами в такому випадку є оператор послідовного виконання, не детермінованого вибору та оператор ітерації. Для даних операторів показано співвідношення з умовним оператором та оператором циклу для імперативних мов програмування. Будуються на кожному з рівнів аксіоми та правила виводу. Семантика динамічної логіки представлена за допомогою семантики можливих світів Кріпке[29].

Слід окремо зазначити пропозиційну динамічну логіку[16, 22, 26, 54], для якої було досліджено проблему розв’язності та виконуваності, та побудовано повну аксіоматику. Запропоновано алгоритм перевірки виконуваності формул. Даний алгоритм базується на властивості малої моделі, що полягає в тому, що якщо формула виконувана, то вона виконувана на моделі з обмеженою кількістю станів, що залежить від довжини формули. Пропозиційна динамічна логіка також є повною.

В динамічній логіці першого порядку[19] стани розглядаються як відображення змінних в їх значення. Семантика програм в такому випадку задається за допомогою пар вхідний стан та вихідний стан.

Одним з основних результатів для динамічної логіки є той факт[25], що істинність трійки Флойда-Хоара $\{p\}pr\{q\}$ еквівалентна істинності твердження $p \to [pr]q$. Таким чином одержується також зв'язок динамічної логіки з перетворювачами предикатів, запропонованими Дейкстрою. Зручність заміни асерції Флойда-Хоара модальними операторами призвела до розробки різноманітних розширень динамічної логіки, серед яких розширення, що дозволяють працювати з розподіленими системами[53] та різноманітні ймовірнісні[14, 15, 36] варіанти динамічної логіки.

Так як базові оператори динамічної логіки є не детермінованими[21], то окремо розглядають детерміновану динамічну логіку[3], в якій допустимими є лише детерміновані програми.

Альтернативою логіці розділення та динамічній логіці є логіка співставлення[58-60]. Логіка співставлення базується на ідеях семантики логіки переписування. В даній семантиці мови програмування задається множина допустимих конфігурацій програм, як терми певної алгебри, а кроки обчислення подаються у вигляді правил переписування термів.

Специфікації в логіці співставлення задаються спеціальними формулами першого порядку, що задають алгебраїчну структуру з обмеженнями, вони називаються шаблонами. Вважається, що конфігурація задовольняє шаблону, якщо їх алгебраїчна структура співпадає та конфігурація задовольняє обмеженням шаблону.

Для логіки співставлення показано еквівалентність логіці Флойда-Хоара для випадку простої імперативної мови програмування. Доведено, що якщо для довільної асерції може бути отримано вивід в логіці Флойда-Хоара, то аналогічне доведення в термінах логіки співставлення також можливе.

Використання шаблонів дозволяє перейти від розгляду простої імперативної мови програмування до мов програмування з динамічною пам’яттю, або складними структурами даних без необхідності розширення логіки першого порядку, що лежить в основі. Для цього необхідно лише змінити вигляд шаблонів. Так, для простої імперативної мови програмування використовується шаблон, що складається з двох частих – програми та стану змінних. Випадок використання динамічної пам’яті потребує лише зміни шаблону на шаблон з трьох компонентів, дві з яких повторюють шаблон для простого випадку, а третя задає динамічну пам'ять.

Правила виводу задаються в логіці співставлення у вигляді правил переписування термів, що одній конфігурації ставлять у відповідність іншу.

Слід відмітити також розширення логіки Флойда-Хоара запропоноване в роботі [4]. В даному розширенні розглядається проблема застосування логіки Флойда-Хоара для доведення властивостей програм, написаних мовою програмування високого рівня, тобто такою, що підтримує виклик функцій, аргументи функцій, посилання та інші.

Значна кількість розширень логіки Флойда-Хоара для адекватного представлення різних аспектів сучасних мов програмування доводить актуальність проблеми пошуку необхідних умов для того, щоб розширення логіки Флойда-Хоара мало коректну та відносно повну аксіоматичну систему. Роботи [37, 38] присвячені даному питанню. Для вирішення даної проблеми використовується теорія категорій та визначаються відповідно категорія Хоара та функтор верифікації.

Розглянувши основні розширення логіки Флойда-Хоара можна прийти до висновку, що випадок часткових умов досліджений слабко. Проте робота з частковими функціями та предикатами є важливою проблему для алгебраїчного підходу до специфікації та верифікації програмних систем [5, 7, 9, 24, 28, 34, 39, 40, 56, 61]. Тому важливою є побудова композиційно-номінативних програмних логік часткових предикатів та функції, що мали б в основі припущення, що предикати можуть бути частковими, та аксіоматичні системи для такої логіки могли б бути побудовані відповідно.

Одним з перших алгебраїчний підхід до представлення семантики програм застосував Глушков в своїх системах алгоритмічних алгебр[65]. Алгоритмічна алгебра має дві основи – множину операторів та множину умов. До сигнатури алгебри входять основні операції запропоновані Дейкстрою, операція послідовного виконання, умовна операція, операція присвоєння та операція циклу, разом з операцією передбачення. Операція передбачення ставила у відповідність умові та оператору нову умову, області істинності та хиби якої були б повними образами області істинності та хиби заданої умови за оператором. В певному розумінні даний оператор є аналогом модальності запропонованої в динамічній логіці.

Слід відмітити, що при побудові алгебри, що представляє семантичну частину програмної логіки, основною задачею є монотонність та неперервність композицій, тобто операторів алгебри. Монотонність розуміється, як монотонність за визначеністю. Це дозволяє використовувати апроксимації при доведення властивостей програмних систем. Якщо ж визначати композицію

Флойда-Хоара на основі визначення часткової коректності трійки Флойда-Хоара, отримаємо композицію, що не буде монотонно, а отже і не буде неперервною. Це не важко перевірити, розглянувши трійку, програма в якій буде нескінченним циклом, передумова всюди істинним предикатом, і післяумова, всюди хибним. Так як програма ніколи не завершує роботу, то трійка буде істинною на довільних даних. Тоді, розглянемо тотожну програму, тобто програму, що не змінює стан змінних. Така програма включає в себе нескінченний цикл. Включення трактується як включення графіків. Трійка, що складається з всюди істинної передумови, тотожної програми та всюди хибної післяумови буде хибною на всіх даних, таким чином не буде включатись в попередню трійку, а отже композиція не буде монотонною. Це доводить необхідність пошуку нового визначення істинності асерції Флойда-Хоара, щоб отримана композиція була монотонною.

Якщо розглядати визначення тотальної коректності асерцій Флойда-Хоара, тобто для істинності асерції вимагати, щоб програма закінчувала свою роботу, неважко бачити, що і в такому випадку, композиція Флойда-Хоара не буде монотонною.

Висновки

Логіка Флойда-Хоара є потужними засобом для специфікації та доведенні властивостей програмних систем. Асерції Флойда-Хоара, що лежать в її основі є простим і ефективними формалізмом. Розвиток засобів та технологій, що використовуються під час розробки програмного забезпечення, зумовив проблему дослідження розширень логіки Флойда-Хоара, що дозволяли б адекватно працювати з можливостями, які надають сучасні мови програмування.

Одним з найперших розширень, що виникли є динамічна логіка. Причиною стало переосмислення Праттом подання семантики логіки Флойда-Хоара і визначенні семантики операторів через пари вхід-вихід. Динамічна логіка є модальною логікою. Найважливішим результатом для динамічної логіки є розв’язання проблеми виконуваності формул та розробка відповідного алгоритму.

Проблема доведення коректності паралельних програм призвела, також, до розробки паралельної динамічної логіки, що є розширенням динамічної логіки.

Іншим розширенням логіки Флойда-Хоара є логіка розділення, що базується на логіці згрупованих імплікацій, що дозволяє оперувати поняттями ресурсів програмних систем, їх складу. Причиною появи такої логіки є широке використання вказівників та динамічної пам’яті в мовах програмування. Важливо також, що логіка розділення також має розподілений варіант.

Одним з головних недоліків логіки розширення є те, що в ній використовується не класична логіка першого порядку, а логіка згрупованих імплікацій, що є більш складною та потребує нових засобів для перевірки розв’язності формул. Це призвело до появи логіки співставлення, що дозволяла розглядати складні структури та додаткові можливості сучасних мов програмування, не ускладнюючи логіки першого порядку, що є основою програмної логіки.

Використання поняття шаблону, конфігурації та подання обчислень як переписування відповідних термів, дозволило природнім чином задавати такі засоби мов програмування як динамічна пам'ять, складні структури даних. Важливо відмітити, що для простих імперативних мов програмування доведення за допомогою логіки співставлення співпадають з доведеннями в логіці Флойда-Хоара.

Використання теорії категорії дозволяє провести абстракцію логіки Флойда-Хоара, та визначити умови, за яких розширення логіки на певний клас проблем коректним та відносно повним.

Хоча більшість розглянути розширень Флойда-Хоара використовує певний аналог стану змінних, що має багато спільного з іменними множинами, проте у всіх випадках розглядаються лише тотальні предикати. В конкретних задачах часто виникає потреба в роботі з умовами, що задаються частковими предикатами.

У випадку часткових відображень важливою властивістю є монотонність, вона дозволяє розглядати в доведеннях апроксимації та гарантувати, що твердження доведені для апроксимацій не стануть хибними під час наступних уточнень.

В розділі показано, що при використанні визначення часткової коректності, композиція Флойда-Хоара не буде монотонною. Це зумовлює потребу в визначенні нової, монотонної композиції Флойда-Хоара. Така композиція має бути певною комбінацією тотальної та часткової коректності.

РОЗДІЛ 2
КОМПОЗИЦІЙНО-НОМІНАТИВНІ МОДЕЛІ ПРОГРАМ

При побудові програмних логік будемо застосовувати семантико-синтаксичний підхід. Спочатку розглянемо семантичний аспект, що подається за допомогою алгебр функцій та предикатів. В такому разі синтаксис програмних логік може бути заданий у вигляді термів відповідної алгебри. Також будемо притримуватись принципів композиційності, номінативності[42, 72-74, 76, 79, 80] та максимальної визначеності.

Принцип композиційності полягає в тому, що складніші програми, функції та предикати розглядаються як композиція більш простих. Принцип номінативності позначає той факт, що розглядаються номінативні дані, тобто дані представляються у вигляді відображення з множини імен змінних на множину значень відповідних змінних. Принцип максимальної визначеності говорить про те, що композиції визначаються таким чином, щоб область визначення отриманих відображень завжди була найбільшою.

2.1 Номінативні множини

Нехай зафіксовано наступні множини: V – множина імен змінних, A – множина можливих значень змінних. Тоді простою номінативною(іменною) множиною будемо називати часткове відображення $V \xrightarrow{p} A$. Відповідно класом простих номінативних множин будемо називати клас часткових відображень ${}^{V}A = V \xrightarrow{p} A.$

Для номінативних множин будемо використовувати позначення аналогічні до позначень в теорії множин: $d = [v_i \text{ а } a_i \mid i \in I]$, що означає $d(v_i)\downarrow = a_i$ для $i \in I$. Та $d(v)\uparrow$ для $v \neq v_i, i \in I$. Також, якщо пара ім'я-значення $v \text{ а } a$ належить номінативній множині d будемо використовувати позначення: $v \text{ а } a \in_n d$, що означає $d(v)\downarrow = a$.

Над номінативними множинами вводиться бінарна операція накладання. $d_1 \nabla d_2 = [v \mapsto a \mid d_2(v)\downarrow = a \vee (d_1(v)\downarrow = a \wedge d_2(v)\uparrow)]$. Таким чином результуюча номінативна множина буде визначена на іменах змінних, на яких визначені аргументи, та значення задаються відповідно, причому у випадку, коли обидві номінативні множини одночасно визначені на імені, значення визначається за другим аргументом.

2.2 Програмні алгебри

Для подання семантики умов в програмах будемо використовувати часткові предикати над простими номінативними множинами, $Pr^{V,A} : {}^{V}A \xrightarrow{p} \{T, F\}$, які в подальшому будемо називати частковими квазіарними предикатами.

Семантику виразів в програмах будемо подавати за допомогою часткових функцій з номінативних множин в множину базових значень, $Fn^{V,A} : {}^{V}A \xrightarrow{p} A$, які в подальшому будемо називати частковими квазіарними ординарними функціями.

В свою чергу семантику програм та їх операторів будемо подавати використовуючи часткові функції над простими номінативними множинами, $FPrg^{V,A} : {}^{V}A \xrightarrow{p} {}^{V}A$, які в подальшому будемо називати частковими біквазіарними функціями.

Семантику програм будемо задавати за допомогою програмних алгебр з трьома основами: множиною часткових квазіарних предикатів $Pr^{V,A}$, множиною часткових ординарних функцій $Fn^{V,A}$ та множиною біквазіарних часткових функцій $FPrg^{V,A}$. Операції таких алгебр будемо називати композиціями. Можна виділити три рівні абстракції, на яких будемо розглядати композиції: предикатний рівень, предикатно-функціональний рівень та програмний рівень.

На предикатному рівні розглядаються лише композиції над множиною часткових квазіарних предикатів $Pr^{V,A}$. Основними композиціями на даному рівні є композиція диз’юнкції $\vee$, композиція заперечення $\neg$ та параметрична композиція екзистенціальної квантифікації $\exists x\, p$.

Бінарна композиція диз'юнкції $\vee$ має тип $Pr^{V,A} \times Pr^{V,A} \xrightarrow{t} Pr^{V,A}$, та визначається наступним чином, де $p \in Pr^{V,A}, q \in Pr^{V,A}, d \in {}^{V}A$:

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

Унарна композиція заперечення $\neg$ має тип $Pr^{V,A} \xrightarrow{t} Pr^{V,A}$, та визначається наступним чином, де $p \in Pr^{V,A}, d \in {}^{V}A$:

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

Унарна параметрична композиція екзистенціальної квантифікації $\exists x$ з параметром $x \in V$ має тип $Pr^{V,A} \xrightarrow{t} Pr^{V,A}$, та визначається наступним чином, де $p \in Pr^{V,A}, q \in Pr^{V,A}, d \in {}^{V}A$:

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

На предикатно-функціональному рівні до композицій, що розглядались на предикатному рівні додаються композиції над множиною часткових квазіарних предикатів $Pr^{V,A}$ та множиною часткових ординарних квазіарних функцій $Fn^{V,A}$. Основними композиціями на даному рівні є параметричні композиції суперпозиції для предикатів $S_P^{x_1,x_2,\ldots x_n}$ та функцій $S_F^{x_1,x_2,\ldots x_n}$, параметрична композиція розіменування $'x$.

Композиція суперпозиції для функцій $S_F^{x_1,x_2,\ldots x_n}$ є $n+1$-арною параметричною композицією з параметрами $x_1, x_2, \ldots, x_n \in V$,має тип $(Fn^{V,A})^{n+1} \xrightarrow{t} Fn^{V,A}$, та визначається наступним чином, де $f, g_1, g_2, \ldots, g_n \in Fn^{V,A}, d \in {}^{V}A$:

$$S_F^{x_1,x_2,\ldots,x_n}(f, g_1, g_2, \ldots, g_n)(d) = f(d\nabla[x_1 \mapsto g_1(d), x_2 \mapsto g_2(d), \ldots, x_n \mapsto g_n(d)]).$$

Композиція суперпозиції для предикатів $S_P^{x_1,x_2,\ldots x_n}$ є $n+1$-арною параметричною композицією з параметрами $x_1,x_2,\ldots,x_n \in V$,має тип $Pr^{V,A} \times (Fn^{V,A})^n \xrightarrow{t} Pr^{V,A}$, та визначається наступним чином, де $p \in Pr^{V,A}, g_1, g_2, \ldots, g_n \in Fn^{V,A}, d \in {}^V A$:

$$S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)(d) ; \ p(d\nabla[x_1 \mapsto g_1(d), x_2 \mapsto g_2(d),\ldots,x_n \mapsto g_n(d)]).$$

Нуль-арна параметрична композиція розіменування $'x$ з параметром $x \in V$, $'x \in Fn^{V,A}$, якщо $d \in {}^V A$ то значення функції що задається композицією визначається як значення відповідної змінної для даного d, якщо вона визначена:

$$'x(d) ; \ d(x).$$

Програмний рівень визначають композиції, що дозволяють задавати побудову програм з більш простих. На даному рівні розглядаються всі основи, а саме: множина квазіарних часткових предикатів $Pr^{V,A}$, множина часткових квазіарних ординарних функцій $Fn^{V,A}$ та множина біквазіарних часткових функцій $FPrg^{V,A}$. Частина композицій відповідає за побудову нових програм, це композиції присвоєння AS^x, композиція послідовного виконання $\bullet$, умовна композиція IF, циклічна композиція WH та композиція тотожної програми id. Решта композицій дозволяють задавати семантику властивостей програм, це композиція Флойда-Хоара FH, композиція побудови умови за прообразом PC.

Унарна параметрична композиція присвоєння AS^x з параметром $x \in V$ має тип $Fn^{V,A} \xrightarrow{t} FPrg^{V,A}$, та визначається наступним чином, де $f \in Fn^{V,A}, d \in {}^V A$:

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

Бінарна композиція послідовного виконання $\bullet$ має тип $FPrg^{V,A} \times FPrg^{V,A} \xrightarrow{t} FPrg^{V,A}$, та визначається наступним чином, де $pr_1, pr_2 \in FPrg^{V,A}, d \in {}^V A$:

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

Тернарна умовна композиція IF має тип $Pr^{V,A} \times FPrg^{V,A} \times FPrg^{V,A} \xrightarrow{t} FPrg^{V,A}$, та визначається наступним чином, де $r \in Pr^{V,A}, pr_1, pr_2 \in FPrg^{V,A}, d \in {}^{V}A$:

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

Бінарна композиція циклу WH має тип $Pr^{V,A} \times FPrg^{V,A} \xrightarrow{t} FPrg^{V,A}$, та визначається наступним чином, де $r \in Pr^{V,A}, pr \in FPrg^{V,A}, d \in {}^{V}A$:

$WH(r, pr)(d) = d'$, якщо існує послідовність номінативних множин $d_0, d_1, \ldots, d_n \in {}^{V}A$ таких, що $d_0 = d, d_n = d'$ та $d_1 = pr(d_0), d_2 = pr(d_1), \ldots, d_n = pr(d_{n-1})$ разом з $r(d_0) = r(d_1) = \ldots = r(d_{n-1}) = T,\ r(d_n) = F$. Інакше, $WH(r, pr)(d)\uparrow$.

Нуль-арна композиція тотожної програми $id \in FPrg^{V,A}$, визначається наступним чином, де $d \in {}^{V}A$:

$id(d) = d$.

Тернарна композиція Флойда-Хоара FH має тип $Pr^{V,A} \times FPrg^{V,A} \times Pr^{V,A} \xrightarrow{t} Pr^{V,A}$, та визначається наступним чином, де $p, q \in Pr^{V,A}, pr \in FPrg^{V,A}, d \in {}^{V}A$:

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

Бінарна композиція побудови умови за прообразом PC має тип $FPrg^{V,A} \times Pr^{V,A} \xrightarrow{t} Pr^{V,A}$, та визначається наступним чином, де $q \in Pr^{V,A}, pr \in FPrg^{V,A}, d \in {}^{V}A$:

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

Областю істинності предикату $p \in Pr^{V,A}$, позначатимемо p^T, будемо називати наступну множину: $p^T = \{d \mid p(d) \downarrow = T\}$.

Областю хибності предикату $p \in Pr^{V,A}$, позначатимемо p^F, будемо називати наступну множину: $p^F = \{d \mid p(d) \downarrow = F\}$.

Областю невизначеності предикату $p \in Pr^{V,A}$, позначатимемо $p^{\perp}$, будемо називати наступну множину: $p^{\perp} = \{d \mid p(d) \uparrow\}$.

Також введемо додатково позначення для образів та прообразів відповідних областей відносно біквазіарної функції $pr \in FPrg^{V,A}$:

$p^{T,pr} = pr[p^T]$, $p^{F,pr} = pr[p^F]$, $p^{\perp,pr} = pr[p^{\perp}]$, $p^{-T,pr} = pr^{-1}[p^T]$, $p^{-F,pr} = pr^{-1}[p^F]$, $p^{-\perp,pr} = pr^{-1}[p^{\perp}]$.

Таким чином з визначення композиції Флойда-Хоара для довільних $p,q \in Pr^{V,A}, pr \in FPrg^{V,A}$ випливають наступні рівності:

$$FH(p,pr,q)^T = p^F \cup q^{-T,pr},$$

$$FH(p,pr,q)^F = p^T \cap q^{-F,pr}.$$

2.3 Монотонність та неперервність композицій

Композицію $C:(Pr^{V,A})^k \times (Fn^{V,A})^l \times (FPrg^{V,A})^m \xrightarrow{t} Pr^{V,A}$ будемо називати монотонною за першим аргументом, якщо для довільних $p_1,\ldots,p_k,p_1' \in Pr^{V,A}$, $f_1,f_2,\ldots,f_l \in Fn^{V,A}$ та $t_1,t_2,\ldots,t_m \in FPrg^{V,A}$ виконується наступне: $p_1 \subseteq p_1' \Rightarrow C(p_1,\ldots,p_k,f_1,\ldots,f_l,t_1,\ldots,t_m) \subseteq C(p_1',p_2,\ldots,p_k,f_1,\ldots,f_l,t_1,\ldots,t_m)$

Тут під включенням відображень розуміється включення їх графіків.

Аналогічно визначається монотонність композиції за іншими аргументами. Композиція називається монотонною, якщо вона є монотонною за кожним аргументом.

Твердження 2.1. Композиція Флойда-Хоара є монотонною.

Доведення. Доведемо окремо монотонність за кожним аргументом.

Включення для предикатів можна розглядати як включення областей істинності та хибності.

Розглянемо перший аргумент. Потрібно довести, що для довільних $p, p', q \in Pr^{V,A}, pr \in FPrg^{V,A}$ має місце $p \subseteq p' \Rightarrow FH(p, pr, q) \subseteq FH(p', pr, q)$.

Враховуючи попередні твердження, визначення та монотонність операцій маємо:

$p \subseteq p' \Rightarrow p^T \subseteq p'^T \Rightarrow p^T \cap q^{-F,pr} \subseteq p'^T \cap q^{-F,pr} \Rightarrow FH(p, pr, q)^F \subseteq FH(p', pr, q)^F$, аналогічно $p \subseteq p' \Rightarrow p^F \subseteq p'^F \Rightarrow p^F \cup q^{-T,pr} \subseteq p'^F \cup q^{-T,pr}$ $\Rightarrow \Rightarrow FH(p, pr, q)^T \subseteq FH(p', pr, q)^T$.

Показано включення області істинності та хибності, а отже маємо $FH(p, pr, q) \subseteq FH(p', pr, q)$.

Розглянемо другий аргумент. Монотонність за другим аргументом означає, що для довільних $p, q \in Pr^{V,A}, pr, pr' \in FPrg^{V,A}$ має місце $pr \subseteq pr' \Rightarrow FH(p, pr, q) \subseteq FH(p, pr', q)$.

За монотонністю операції повного прообразу маємо $pr \subseteq pr' \Rightarrow q^{-F,pr} \subseteq q^{-F,pr'}$ та $pr \subseteq pr' \Rightarrow q^{-T,pr} \subseteq q^{-T,pr'}$.

Отже: $pr \subseteq pr' \Rightarrow p^T \cap q^{-F,pr} \subseteq p^T \cap q^{-F,pr'} \Rightarrow FH(p, pr, q)^F \subseteq FH(p, pr', q)^F$, та $pr \subseteq pr' \Rightarrow p^F \cup q^{-T,pr} \subseteq p^T \cup q^{-T,pr'} \Rightarrow FH(p, pr, q)^T \subseteq FH(p, pr', q)^T$.

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

Розглянемо третій аргумент. Аналогічно випадку з першим аргументом, потрібно довести, що для довільних $p, q, q' \in Pr^{V,A}, pr \in FPrg^{V,A}$, виконується $q \subseteq q' \Rightarrow FH(p, pr, q) \subseteq FH(p, pr, q')$.

З монотонності операцій та визначень можна отримати, що $q \subseteq q' \Rightarrow q^T \subseteq q'^T \Rightarrow p^F \cup q^{-T,pr} \subseteq p^F \cup q'^{-T,pr} \Rightarrow FH(p, pr, q)^T \subseteq FH(p, pr, q')^T$, та $q^F \subseteq q'^F \Rightarrow p^T \cap q^{-F,pr} \subseteq p^T \cap q'^{-F,pr} \Rightarrow FH(p, pr, q)^F \subseteq FH(p, pr, q')^F$.

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

Доведено монотонність за кожним аргументом, отже композиція Флойда-Хоара є монотонною.

Твердження 2.2. Композиція побудови умови за прообразом монотонна.

Доведення. Монотонність композиції будемо доводити як монотонність за кожним аргументом. З визначення композиції побудови умови за прообразом можна отримати, що $PC(pr,q)^T = q^{-T,pr}$ та $PC(pr,q)^F = q^{-F,pr}$. Потрібно показати, що для довільних $q,q' \in Pr^{V,A}, pr, pr' \in FPrg^{V,A}$ виконується $q \subseteq q' \Rightarrow PC(pr,q) \subseteq PC(pr,q')$ та також $pr \subseteq pr' \Rightarrow PC(pr,q) \subseteq PC(pr',q)$. Включення предикатів можна розглядати як включення областей істинності та хибності. Для першого твердження маємо з визначення композиції $PC(pr,q)^T \subseteq PC(pr,q')^T \Leftrightarrow q^{-T,pr} \subseteq q'^{-T,pr}$,та аналогічно для області хибності $PC(pr,q)^F \subseteq PC(pr,q')^F \Leftrightarrow q^{-F,pr} \subseteq q'^{-F,pr}$,що отримуємо з $q \subseteq q'$ враховуючи монотонність операції повного прообразу. Друге твердження доводиться аналогічно.

Для того, щоб дати визначення неперервності композиції, розглянемо наступні визначення.

Визначення. Ланцюгом функцій називається нескінченна злічення множина індексованих функцій $\{f_i \mid i \in \omega\}$, таких, що $f_i \subseteq f_{i+1}, i \in \omega$.

Аналогічно визначається ланцюг предикатів.

Визначення. Границею ланцюга будемо називати супремум відповідної множини. Позначатимемо $\bigcup_i f_i = \sup\{f_i \mid i \in \omega\}$.

Композицію $C:(Pr^{V,A})^k \times (Fn^{V,A})^l \times (FPrg^{V,A})^m \xrightarrow{t} Pr^{V,A}$ будемо називати неперервною за першим аргументом, якщо для довільного ланцюга $\{p'_i \mid i \in \omega, p'_i \in Pr^{V,A}\}$, довільних предикатів та функцій $p_2,\ldots, p_k \in Pr^{V,A}$, $f_1, f_2,\ldots, f_l \in Fn^{V,A}$ та $t_1,t_2,\ldots,t_m \in FPrg^{V,A}$, якщо множина $\{C(p_i', p_2, \ldots, p_k, f_1, \ldots, f_l, t_1, \ldots, t_m) \mid i \in \omega\}$ є ланцюгом та виконується наступне: $C(\bigcup_i p_i', p_2, \ldots, p_k, f_1, \ldots, f_l, t_1, \ldots, t_m) = \bigcup_i C(p_i', p_2, \ldots, p_k, f_1, \ldots, f_l, t_1, \ldots, t_m)$

Аналогічно визначається неперервність композиції за іншими аргументами. Композиція називається неперервною, якщо вона є неперервною за кожним аргументом.

Твердження 2.3. Композиція Флойда-Хоара неперервна.

Доведення. Доведемо неперервність за кожним аргументом.

Розглянемо перший аргумент. Необхідно довести, що для довільного ланцюга $\{p_i \mid i \in \omega, p_i \in Pr^{V,A}\}$, та довільних $q \in Pr^{V,A}, pr \in FPrg^{V,A}$ виконується $FH(\bigcup_i p_i, pr, q) ; \bigcup_i FH(p_i, pr, q)$. Так як композиція Флойда-Хоара монотонна, то множина $\{FH(p_i, pr, q) \mid i \in \omega\}$ також буде ланцюгом. Зафіксуємо довільне дане $d \in {}^V A$ та доведемо, що має місце наступне: $FH(\bigcup_i p_i, pr, q)(d) ; \bigcup_i FH(p_i, pr, q)(d)$.

Розглянемо декілька випадків відносно визначеності $\bigcup_i p_i(d)$.

Нехай $\bigcup_i p_i(d)\uparrow$. Тоді з визначення ланцюга та границі ланцюга, маємо $p_i(d)\uparrow, \forall i \in \omega$. Таким чином якщо $q(pr(d))\downarrow= T$, то $FH(\bigcup_i p_i, pr, q)(d)\downarrow= T$ та $FH(p_i, pr, q)(d)\downarrow= T, \forall i \in \omega$, звідки $\bigcup_i FH(p_i, pr, q)(d)\downarrow= T$. Інакше, $FH(\bigcup_i p_i, pr, q)(d)\uparrow$ та $\bigcup_i FH(p_i, pr, q)(d)\uparrow$. Отже, у випадку $\bigcup_i p_i(d)\uparrow$ отримуємо $FH(\bigcup_i p_i, pr, q)(d) ; \bigcup_i FH(p_i, pr, q)(d)$.

Нехай $\bigcup_i p_i(d)\downarrow$. Тоді з того, що $p_i \subseteq p_{i+1}, \forall i \in \omega$, маємо $\exists k \forall i (i \ge k \to p_i(d)\downarrow= \bigcup_i p_i(d))$. Таким чином $\forall i \ge k$ маємо

$FH(\bigsqcup_i p_i, pr, q)(d) \sqsubseteq FH(p_i, pr, q)(d)$, а отже за визначенням границі ланцюга отримуємо $FH(\bigsqcup_i p_i, pr, q)(d) \sqsubseteq \bigsqcup_i FH(p_i, pr, q)(d)$.

Дане було обрано довільним чином, звідки отримуємо, що виконується також $FH(\bigsqcup_i p_i, pr, q) \sqsubseteq \bigsqcup_i FH(p_i, pr, q)$. Отже, було доведено неперервність за першим аргументом.

Доведення для решти аргументів проводиться аналогічно, розглядаючи рівність відповідних предикатів на кожному даному, розглядаючи декілька випадків відповідно до того чи визначена границя ланцюга на даному. Потрібно зважати на той факт, що якщо границя ланцюга невизначена на певних даних, то і кожен елемент ланцюга невизначений на відповідних даних. У випадку ж якщо границя ланцюга визначена, то знайдеться номер, починаючи з якого всі елементи ланцюга будуть визначеними та рівними значенню границі на відповідному даному.

Твердження 2.4. Композиція побудови умови за прообразом неперервна.

Доведення. Для доведення неперервності композицій потрібно довести що вона неперервна за кожним аргументом.

Спочатку розглянемо перший аргумент. Неперервність за першим аргументом означає, що для довільного ланцюга біквазіарних функцій $\{pr_i \mid i \in \omega, pr_i \in FPrg^{V,A}\}$, та довільної післяумови $q \in Pr^{V,A}$, виконується наступне: $PC(\bigsqcup_i pr_i, q) \sqsubseteq \bigsqcup_i PC(pr_i, q)$. Той факт, що $\{PC(pr_i, q) \mid i \in \omega\}$ є ланцюгом випливає з монотонності композиції. Візьмемо деяке $d \in {}^V A$ та доведемо, що для нього виконується $PC(\bigsqcup_i pr_i, q)(d) \sqsubseteq \bigsqcup_i PC(pr_i, q)(d)$.

Як і у випадку з композицією Флойда-Хоара, потрібно розглянути декілька випадків відносно того факту, чи визначено $\bigsqcup_i pr_i(d)$, чи ні.

У випадку, коли $\bigsqcup\limits_i pr_i(d)\uparrow$, за визначення границі ланцюга, отримуємо $pr_i(d)\uparrow, \forall i \in \omega$. Звідки з визначення композиції $PC(pr_i,q)(d)\uparrow, \forall i \in \omega$. Отже, також і $\bigsqcup\limits_i PC(pr_i,q)(d)\uparrow$. Звідки для $\bigsqcup\limits_i pr_i(d)\uparrow$ маємо $PC(\bigsqcup\limits_i pr_i,q)(d)\ ;\ \bigsqcup\limits_i PC(pr_i,q)(d)$.

Нехай має місце протилежне твердження – $\bigsqcup\limits_i pr_i(d)\downarrow$. Тоді з властивостей границі ланцюга, маємо $\exists k \forall i(i \geq k \to pr_i(d)\downarrow = \bigsqcup\limits_i pr_i(d))$. Що означає $\forall i \geq k$ виконується $PC(\bigsqcup\limits_i pr_i,q)(d)\ ;\ PC(pr_i,q)(d)$. Разом з тим за визначенням границі ланцюга отримуємо, що $\forall i \geq k$ виконується $\bigsqcup\limits_i PC(pr_i,q)(d)\ ;\ PC(pr_i,q)(d)$, а отже $PC(\bigsqcup\limits_i pr_i,q)(d)\ ;\ \bigsqcup\limits_i PC(pr_i,q)(d)$.

Так як в доведенні не використовувалось жодних припущень відносно d, то має місце $PC(\bigsqcup\limits_i pr_i,q)\ ;\ \bigsqcup\limits_i PC(pr_i,q)$. Таким чином, композиція побудови умови за прообразом неперервна за першим аргументом.

Доведення неперервності за другим аргументом проведемо аналогічно. Потрібно показати, що для довільної біквазіарної функції $pr \in FPrg^{V,A}$, та довільного ланцюга предикатів $\{q_i \mid i \in \omega, q_i \in Pr^{V,A}\}$, має місце наступне твердження: $PC(pr,\bigsqcup\limits_i q_i)\ ;\ \bigsqcup\limits_i PC(pr,q_i)$. Монотонність композиції гарантує, що $\{PC(pr,q_i) \mid i \in \omega\}$ також буде ланцюгом. Лишається довести, що для довільного $d \in {}^V A$ виконується $PC(pr,\bigsqcup\limits_i q_i)(d)\ ;\ \bigsqcup\limits_i PC(pr,q_i)(d)$. Розглянемо як і при доведенні неперервності за першим аргументом, два випадки. В першому випадку $\bigsqcup\limits_i q_i(d)\uparrow$, а в другому – $\bigsqcup\limits_i q_i(d)\downarrow$.

Якщо $\bigcup_i q_i(d)\uparrow$, то $\forall i\in\omega$ маємо $q_i(d)\uparrow$. А отже, $\forall i\in\omega\ \ PC(pr,q_i)(d)\uparrow$. З визначення границі ланцюга $\bigcup_i PC(pr,q_i)(d)\uparrow$. Також з визначення композиції $PC(pr,\bigcup_i q_i)(d)\uparrow$. Отже, $PC(pr,\bigcup_i q_i)(d)\ ;\ \bigcup_i PC(pr,q_i)(d)$.

У випадку, коли $\bigcup_i q_i(d)\downarrow$, можна знайти номер, починаючи з якого всі елементи ланцюга визначені та рівні границі для даного *d.* Формально, $\exists k\forall i(i\geq k\rightarrow q_i(d)\downarrow=\bigcup_i q_i(d))$. Підставляючи в композицію отримуємо $\forall i\geq k$ виконується $PC(pr,\bigcup_i q_i)(d)\ ;\ PC(pr,q_i)(d)$. В свою чергу, з властивостей границі ланцюга $\forall i\geq k$ виконується $\bigcup_i PC(pr,q_i)(d)\ ;\ PC(pr,q_i)(d)$, звідки $PC(pr,\bigcup_i q_i)(d)\ ;\ \bigcup_i PC(pr,q_i)(d)$.

Зважаючи на те, що d було обрано довільним чином, виконується більш загальне твердження, $PC(pr,\bigcup_i q_i)\ ;\ \bigcup_i PC(pr,q_i)$. Отримали доведення того, що композиція неперервна за другим аргументом, а отже і неперервна в цілому.

2.4 Властивості композиції побудови умови за прообразом

Твердження 2.5. Для композиції побудови умови за прообразом справедливі наступні рівності:

$PC(id,q)=q$,

$PC(AS^x(f),q)=S_P^x(q,f)$,

$PC(pr_1\bullet pr_2,q)=PC(pr_1,PC(pr_2,q))$,

$PC(IF(r,pr_1,pr_2),q)=(r\rightarrow PC(pr_1,q))\wedge(\neg r\rightarrow PC(pr_2,q))\wedge(r\rightarrow r)$.

Доведення. Доведемо твердження показавши, що області істинності та хибності відповідних предикатів рівні.

Розглянемо $PC(id,q)=q$. З визначення композиції побудови умови за прообразом маємо $PC(id,q)^T = q^{-T,id} = id^{-1}[q^T]$. Зважаючи на те, що композиція тотожної програми задається як $id(d)=d$ для довільних $d \in {}^V A$, отримуємо $id^{-1}[q^T]=q^T$. Отже, $PC(id,q)^T = q^T$. Аналогічно отримуємо рівність для області хибності: $PC(id,q)^F = q^{-F,id} = id^{-1}[q^F]=q^F$. Отже, виконується $PC(id,q)=q$.

Розглянемо $PC(AS^x(f),q)=S_P^x(q,f)$. Доведемо спочатку для області істинності. Отримуємо твердження $PC(AS^x(f),q)^T = S_P^x(q,f)^T$. За визначенням композиції побудови умови за прообразом $PC(AS^x(f),q)^T = AS^x(f)^{-1}[q^T]$. Розкриваючи за визначенням повного прообразу $AS^x(f)^{-1}[q^T]=\{d \mid AS^x(f)(d)\in q^T\}$, звідки за визначення композиції присвоєння $\{d \mid AS^x(f)(d)\in q^T\}=\{d \mid d\nabla[x\ \mathrm{a}\ \ f(d)]\in q^T\}$. Також $\{d \mid d\nabla[x\ \mathrm{a}\ \ f(d)]\in q^T\}=\{d \mid q(d\nabla[x\ \mathrm{a}\ \ f(d)])=T\}$. Умову можна згорнути за визначенням суперпозиції $\{d \mid q(d\nabla[x\ \mathrm{a}\ \ f(d)])=T\}=\{d \mid S_P^x(q,f)(d)=T\}$. Звідки остаточно отримуємо $\{d \mid S_P^x(q,f)(d)=T\}=S_P^x(q,f)^T$ і враховуючи всі рівності $PC(AS^x(f),q)^T = S_P^x(q,f)^T$.

Аналогічно для області хибності маємо $PC(AS^x(f),q)^F$ = $AS^x(f)^{-1}[q^F]$ = $\{d \mid AS^x(f)(d) \in q^F\}$ = $\{d \mid d\nabla[x\ \mathrm{a}\ \ f(d)]\in q^F\}$ = $\{d \mid q(d\nabla[x\ \mathrm{a}\ \ f(d)])=F\}$ = $\{d \mid S_P^x(q,f)(d) =F\}$ = $S_P^x(q,f)^F$.

Доведено рівність областей істинності та хибності, отже і предикати рівні. Розглянемо $PC(pr_1 \bullet pr_2,q)=PC(pr_1,PC(pr_2,q))$. Покажемо для початку, що $(pr_1 \bullet pr_2)^{-1}[A]=pr_1^{-1}[pr_2^{-1}[A]]$. Ліву частину рівності можна розписати, як $(pr_1 \bullet pr_2)^{-1}[A]=\{d \mid pr_1 \bullet pr_2(d)\in A\}$. А за визначення композиції послідовного виконання $\{d \mid pr_1 \bullet pr_2(d)\in A\}=\{d \mid pr_2(pr_1(d))\in A\}$. Праву ж частину можна розписати, як $pr_1^{-1}[pr_2^{-1}[A]]=\{d \mid pr_1(d)\in pr_2^{-1}[A]\}$. Умова еквівалентна $pr_2(pr_1(d))\in A$, а отже права та ліва частина рівні і $(pr_1 \bullet pr_2)^{-1}[A]=pr_1^{-1}[pr_2^{-1}[A]]$.

Доведемо рівність областей істинності відповідних предикатів – $PC(pr_1 \bullet pr_2, q)^T = PC(pr_1, PC(pr_2, q))^T$. Розглядаючи ліву частину рівності отримуємо наступне: $PC(pr_1 \bullet pr_2, q)^T = (pr_1 \bullet pr_2)^{-1}[q^T]$. За показаним раніше $(pr_1 \bullet pr_2)^{-1}[q^T] = pr_1^{-1}[pr_2^{-1}[q^T]]$. Лишається тільки згорнути праву частину за визначенням композиції побудови умови за прообразом $pr_1^{-1}[pr_2^{-1}[q^T]] = pr_1^{-1}[PC(pr_2, q)]^T = PC(pr_1, PC(pr_2, q))^T$. Таким чином, маємо $PC(pr_1 \bullet pr_2, q)^T = PC(pr_1, PC(pr_2, q))^T$ і для області істинності доведено рівність.

Аналогічно для області хибності $PC(pr_1 \bullet pr_2, q)^F$ = $(pr_1 \bullet pr_2)^{-1}[q^F]$ = $pr_1^{-1}[pr_2^{-1}[q^F]]$ = $pr_1^{-1}[PC(pr_2, q)]^F = PC(pr_1, PC(pr_2, q))^F$. Отже, для області хибності також доведено, і маємо $PC(pr_1 \bullet pr_2, q) = PC(pr_1, PC(pr_2, q))$.

Розглянемо останню рівність, що стосується умовної композиції $PC(IF(r, pr_1, pr_2), q) = (r \to PC(pr_1, q)) \wedge (\neg r \to PC(pr_2, q)) \wedge (r \to r)$. Доведемо спочатку рівність областей істинності відповідних предикатів. За визначенням $PC(IF(r, pr_1, pr_2), q)^T = IF(r, pr_1, pr_2)^{-1}[q^T]$. З визначення умовної композиції та повного прообразу, зважаючи, що якщо умова r істина результат обчислюється як pr_1, а якщо хибна, то як pr_2, маємо $IF(r, pr_1, pr_2)^{-1}[q^T] = (r^T \cap pr_1^{-1}[q^T]) \cup (r^F \cap pr_2^{-1}[q^T])$. В свою чергу область істинності правої частини можна записати після перетворень композицій предикатного рівня та використання визначення композиції побудови умови за прообразом, зокрема $(p \wedge q)^T = p^T \cap q^T$, $(p \to q)^T = p^F \cup q^T$ та $(\neg p)^T = p^F$, як $(r^F \cup pr_1^{-1}[q^T]) \cap (r^T \cup pr_2^{-1}[q^T]) \cap (r^T \cup r^F)$. Якщо розкрити дужки при операції перетину та спростити враховуючи, що $r^T \cap r^F = \varnothing$, отримаємо: $((r^F \cap pr_2^{-1}[q^T]) \cup (r^T \cap pr_1^{-1}[q^T]) \cup (pr_1^{-1}[q^T] \cap pr_2^{-1}[q^T])) \cap (r^T \cup r^F)$. Далі розкриємо дужки при операції перетину, та будемо враховувати, що $r^F \cap (r^T \cup r^F) = r^F$ та $r^T \cap (r^T \cup r^F) = r^T$. Отримуємо наступний вираз: $(r^F \cap pr_2^{-1}[q^T])$ $\cup$ $(r^T \cap pr_1^{-1}[q^T])$ $\cup$ $(pr_1^{-1}[q^T] \cap pr_2^{-1}[q^T] \cap r^T)$ $\cup$

$(pr_1^{-1}[q^T] \cap pr_2^{-1}[q^T] \cap r^F)$. Зважаючи на те, що виконуються наступні включення: $(pr_1^{-1}[q^T] \cap pr_2^{-1}[q^T] \cap r^F) \subseteq (r^F \cap pr_2^{-1}[q^T])$ разом з $(pr_1^{-1}[q^T] \cap pr_2^{-1}[q^T] \cap r^T) \subseteq (r^T \cap pr_1^{-1}[q^T])$ приходимо до остаточного результату – $(r^F \cap pr_2^{-1}[q^T]) \cup (r^T \cap pr_1^{-1}[q^T])$, що співпадає з виразом для лівої частини. Це означає, що рівність для областей істинності доведено.

Для області хибності застосовуючи аналогічні міркування, $PC(IF(r, pr_1, pr_2), q)^F = IF(r, pr_1, pr_2)^{-1}[q^F] = (r^T \cap pr_1^{-1}[q^F]) \cup (r^F \cap pr_2^{-1}[q^F])$.

В свою чергу права частина рівна з врахування всіх спрощень $(r^T \cap pr_1^{-1}[q^F]) \cup (r^F \cap pr_2^{-1}[q^F])$, що співпадає з лівою частиною рівності, отже для області хибності також доведено, а отже і в цілому маємо $PC(IF(r, pr_1, pr_2), q) = (r \to PC(pr_1, q)) \wedge (\neg r \to PC(pr_2, q)) \wedge (r \to r)$. Розглянуто всі рівності, а отже твердження доведено.

2.5 Мова програмних логік для простих номінативних даних

Семантика програмних логік задається за допомогою квазіарної програмної алгебри наступного вигляду:

$$QPA(V, A) = < Pr^{V,A}, Fn^{V,A}, FPrg^{V,A}; \vee, \neg, \{S_P^{\bar{v}}\}_{\bar{v} \in U(V)}, \{S_F^{\bar{v}}\}_{\bar{v} \in U(V)}, \{`x\}_{x \in V}, \{\exists x\}_{x \in V}, id,$$
$$\{AS^x\}_{x \in V}, \bullet, IF, WH, FH, PC >$$

Тут $U(V)$ позначаємо множину всіх скінчених послідовностей з V.

Згідно з семантико-синтаксичним підходом синтаксис програмної логіки природно випливає з семантики, а саме, буде задаватись у вигляді термів програмної алгебри, що дозволяє просто визначати відображення, що задають інтерпретацію формул логіки.

Нехай задано множини предикатних символів Ps, множини функціональних символів Fs, множини програмних символів Prs та множину імен змінних V. Відповідно до запропонованих програмних алгебр надамо індуктивні визначення для термів $Tr(Ps, Fs, Prs, V)$, формул $Fr(Ps, Fs, Prs, V)$, текстів програм

$Pt(Ps,Fs,Prs,V)$, та асерцій Флойда-Хоара $FHFr(Ps,Fs,Prs,V)$. Символи композицій тут використовуються в синтаксичному сенсі.

Терми $Tr(Ps,Fs,Prs,V)$ будемо визначати наступним чином:

– якщо $f \in Fs$, то $f \in Tr(Ps,Fs,Prs,V)$;

– якщо $x \in V$, то $`x \in Tr(Ps,Fs,Prs,V)$;

– якщо $f \in Fs$, $t_1,t_2,\ldots,t_n \in Tr(Ps,Fs,Prs,V)$ та $x_1,x_2,\ldots,x_n \in V$, де $x_1,x_2,\ldots,x_n$ – різні імена змінних, та $n>0$, то $S_F^{x_1,x_2,\ldots,x_n}(f,t_1,t_2,\ldots,t_n) \in Tr(Ps,Fs,Prs,V)$.

Формули $Fr(Ps,Fs,Prs,V)$ будемо визначати наступним чином:

– якщо $p \in Ps$, то $p \in Fr(Ps,Fs,Prs,V)$;

– якщо $\Phi \in Fr(Ps,Fs,Prs,V)$, то $\neg\Phi \in Fr(Ps,Fs,Prs,V)$;

– якщо $\Phi \in Fr(Ps,Fs,Prs,V)$ та $x \in V$, то $\exists x\Phi \in Fr(Ps,Fs,Prs,V)$;

– якщо $\Phi,\Psi \in Fr(Ps,Fs,Prs,V)$, то $\Phi \vee \Psi \in Fr(Ps,Fs,Prs,V)$;

– якщо $p \in Ps$, $t_1,t_2,\ldots,t_n \in Tr(Ps,Fs,Prs,V)$ та $x_1,x_2,\ldots,x_n \in V$, де $x_1,x_2,\ldots,x_n$ – різні імена змінних, та $n>0$, то $S_P^{x_1,x_2,\ldots,x_n}(p,t_1,t_2,\ldots,t_n) \in Fr(Ps,Fs,Prs,V)$;

– якщо $q \in Fr(Ps,Fs,Prs,V)$ разом з $pr \in Pt(Ps,Fs,Prs,V)$, то відповідно $PC(pr,q) \in Fr(Ps,Fs,Prs,V)$.

Тексти програм $Pt(Ps,Fs,Prs,V)$ будемо визначати наступним чином:

– якщо $pr \in Prs$, то $pr \in Pt(Ps,Fs,Prs,V)$;

– $id \in Pt(Ps,Fs,Prs,V)$;

– якщо $f \in Tr(Ps,Fs,Prs,V)$ та $x \in V$, то $AS^x(f) \in Pt(Ps,Fs,Prs,V)$;

– якщо $pr_1, pr_2 \in Pt(Ps,Fs,Prs,V)$, то $pr_1 \bullet pr_2 \in Pt(Ps,Fs,Prs,V)$;

– якщо $pr_1, pr_2 \in Pt(Ps,Fs,Prs,V)$ та $r \in Fr(Ps,Fs,Prs,V)$, то $IF(r, pr_1, pr_2) \in Pt(Ps,Fs,Prs,V)$;

– якщо $pr \in Pt(Ps,Fs,Prs,V)$ та $r \in Fr(Ps,Fs,Prs,V)$, то $WH(r, pr) \in Pt(Ps,Fs,Prs,V)$.

Асерції Флойда-Хоара $FHFr(Ps,Fs,Prs,V)$ будемо визначати наступним чином:

– якщо $pr \in Pt(Ps,Fs,Prs,V)$ та $p,q \in Fr(Ps,Fs,Prs,V)$, то $\{p\}pr\{q\} \in FHFr(Ps,Fs,Prs,V)$.

Синтаксис та семантика програмної логіки композиційно-номінативного типу задано, залишається лише задати інтерпретації, або відображення, що термам, формулам та текстам програм будуть відповідно ставити у відповідність предикати та функції програмної алгебри. Дані відображення будуються природнім чином, якщо задані відображення, інтерпретації для предикатних, функціональних та програмних символів.

Будемо вважати, що задано тотальні відображення $I_{Ps}: Ps \xrightarrow{t} Pr^{V,A}$ – інтерпретації для предикатних символів, $I_{Fs}: Fs \xrightarrow{t} Fn^{V,A}$ – інтерпретації для функціональних символів, $I_{Prs}: Prs \xrightarrow{t} FPrg^{V,A}$ – інтерпретації для програмних символів. Задамо розширення даних відображень на формули $J_{Fr}: Fr(Ps,Fs,Prs,V) \xrightarrow{t} Pr^{V,A}$, терми $J_{Tr}: Tr(Ps,Fs,Prs,V) \xrightarrow{t} Fn^{V,A}$, тексти програм $J_{Pt}: Pt(Ps,Fs,Prs,V) \xrightarrow{t} FPrg^{V,A}$ та асерції Флойда-Хоара $J_{FHFr}: FHFr(Ps,Fs,Prs,V) \xrightarrow{t} Pr^{V,A}$ індуктивно.

Відображення $J_{Tr}: Tr(Ps,Fs,Prs,V) \xrightarrow{t} Fn^{V,A}$:

– $J_{Tr}(f) = I_{Fs}(f)$, якщо $f \in Fs$;

– $J_{Tr}(`x) = `x$;

– $J_{Tr}(S_F^{x_1,x_2,\ldots,x_n}(f,t_1,t_2,\ldots,t_n)) = S_F^{x_1,x_2,\ldots,x_n}(J_{Tr}(f),J_{Tr}(t_1),J_{Tr}(t_2),\ldots,J_{Tr}(t_n))$.

Відображення $J_{Fr}: Fr(Ps,Fs,Prs,V) \xrightarrow{t} Pr^{V,A}$:

– $J_{Fr}(p) = I_{Ps}(p)$, якщо $p \in Ps$;

– $J_{Fr}(\neg\Phi) = \neg J_{Fr}(\Phi)$;

– $J_{Fr}(\exists x\Phi) = \exists x J_{Fr}(\Phi)$;

– $J_{Fr}(\Phi \vee \Psi) = J_{Fr}(\Phi) \vee J_{Fr}(\Psi)$;

– $J_{Fr}(S_P^{x_1,x_2,\ldots,x_n}(p,t_1,t_2,\ldots,t_n)) = S_P^{x_1,x_2,\ldots,x_n}(J_{Fr}(f), J_{Tr}(t_1), J_{Tr}(t_2),\ldots, J_{Tr}(t_n))$;

– $J_{Fr}(PC(pr,q)) = PC(J_{Pt}(pr), J_{Fr}(q))$.

Відображення $J_{Pt} : Pt(Ps,Fs,Prs,V) \xrightarrow{t} FPrg^{V,A}$:

– $J_{Pt}(pr) = I_{Prs}(pr)$, якщо $pr \in Prs$;

– $J_{Pt}(id) = id$;

– $J_{Pt}(AS^x(f)) = AS^x(J_{Tr}(f))$;

– $J_{Pt}(pr_1 \bullet pr_2) = J_{Pt}(pr_1) \bullet J_{Pt}(pr_2)$;

– $J_{Pt}(IF(r, pr_1, pr_2)) = IF(J_{Fr}(r), J_{Pt}(pr_1), J_{Pt}(pr_2)))$;

– $J_{Pt}(WH(r, pr)) = WH(J_{Fr}(r), J_{Pt}(pr))$.

Відображення $J_{FHFr} : FHFr(Ps,Fs,Prs,V) \xrightarrow{t} Pr^{V,A}$:

– $J_{FHFr}(\{p\}pr\{q\}) = FH(J_{Fr}(p), J_{Pt}(pr), J_{Fr}(q))$.

Таким чином композиції програмної алгебри $QPA(V,A)$ та відображення I_{Ps}, I_{Fs}, та I_{Prs} дозволяють однозначно задати інтерпретацію J, відображення, що кожному елементу мови програмної логіки композиційно-номінативного типу для простих даних співставляє відображення програмної алгебри. Визначення. Будемо казати, що асерція Флойда-Хоара $\{p\}pr\{q\}$ істинна на інтерпретації J, позначатимемо $J \models \{p\}pr\{q\}$, якщо відповідний предикат $J_{FHFr}(\{p\}pr\{q\})$ неспростовний, тобто $J_{FHFr}(\{p\}pr\{q\})^F = \varnothing$.

Визначення. Будемо казати, що асерція Флойда-Хоара $\{p\}pr\{q\}$ всюди істинна, позначатимемо $\models \{p\}pr\{q\}$, якщо вона істинна на довільній інтерпретації J.

2.6 Програмна логіка композиційно-номінативного типу для складних ієрархічних даних

Було визначено прості номінативні дані, відповідні їм програмні алгебри та програмні логіки. Проте, не завжди дані задаються як змінні, що мають певні значення, часто потрібно мати засоби для розгляду більш складних структур даних, зокрема списків, масивів, дерев. Це зумовлює потребу в розгляді складних ієрархічних номінативних даних та побудованих на них програмних алгебр та логік.

При переході від простих номінативних даних до складних можливо декілька підходів. Найпростіший полягає в розгляді складних імен, але простих значень змінних. В цьому випадку складні імена змінних можна розглядати як скінчені послідовності елементів множини імен, а саме $v_1v_2...v_n$,де $v_1,v_2,...,v_n \in V$. Іншим підходом є розгляд простих імен, але складних значень. В цьому випадку імена розглядаються як елементи з множини імен, але значеннями можуть бути також інші номінативні множини, не просто елементи множини базових значень. Можливо також поєднати два підходи, та допускати як складні імена, так і складні значення. Саме такий підхід буде розглянуто, так як він включає в себе решту.

Для того, щоб визначити клас номінативних множин зі складними іменами та складними значенням, спочатку дамо індуктивне визначення класу номінативних множин зі складними значеннями, позначатимемо $ND(V,A)$.

Визначимо множину всіх номінативних даних зі складними значеннями наступними чином: $ND(V,A) = \bigcup_{i \geq 0} ND_i(V,A)$, де $ND_0(V,A) = A \cup \{\varnothing\}$ та $ND_{i+1}(V,A) = ND_i(V,A) \cup (V \xrightarrow{n} ND_i(V,A))$.

Тоді множиною всіх номінативних даних з складними іменами та складними значеннями будемо називати множину $NDVC(V,A)$, що складається з усіх номінативних множин $d \in ND(V^+,A)$, таких, що для довільних двох послідовностей $x_1,x_2,\ldots,x_n \in V^+$ та $y_1,y_2,\ldots,y_m \in V^+$, жодна з яких не є початком іншої та $(\ldots((d(x_1))(x_2))\ldots(x_n))\downarrow$ як і $(\ldots((d(y_1))(y_2))\ldots(y_m))\downarrow$ виконується, що жодне зі слів $x_1x_2\ldots x_n$ та $y_1y_2\ldots y_m$ не є префіксом іншого. Остання умова необхідна для того, щоб можна було однозначно задавати операцію розіменування. Визначимо далі операції розіменування та накладання. При визначені операцій слід зважати на ієрархічну будову складних номінативних даних.

Розіменуванням будемо називати унарну операцію $v \Rightarrow$ з параметром $v \in V^{+}$, що має тип $NDVC(V,A) \xrightarrow{p} NDVC(V,A)$, яка задається індукцією за довжиною параметру наступним чином:

- якщо $|v|=1$, то у випадку, коли $d(v)\downarrow$, то $v \Rightarrow (d) = d(v)$. Інакше, якщо $d/\mathrm{v} \neq \varnothing$, то $v \Rightarrow (d) = d/\mathrm{v}$, а якщо $d/\mathrm{v} = \varnothing$, то $v \Rightarrow (d)\uparrow$. Тут $d/\mathrm{v} = [w\,\mathrm{a}\ \ a \mid d(vw)\downarrow = a]$.
- якщо $|v|>1$, то можна знайти такі $x \in V$ та $v' \in V^{+}$, що $v = xv'$, тоді $v \Rightarrow (d) = v' \Rightarrow (x \Rightarrow (d))$.

Бінарна операція накладання $d_1 \nabla d_2$, що має тип $NDVC(V,A) \times NDVC(V,A) \xrightarrow{p} NDVC(V,A)$ повинна бути визначена таким чином, щоб зберігалась умова на структуру номінативної множини. Задамо операцію індукцією за складністю першого аргументу. Позначимо $NDVC_i(V,A) = NDVC(V,A) \cap ND_i(V^{+},A)$. Найбільший такий i, що $d_1 \in NDVC_i(V,A)$ будемо називати рангом d_1. Визначимо операцію накладання індукцією за рангом d_1.

Якщо ранг d_1 рівний 0. В такому випадку, якщо $d_1 \in A$, або $d_2 \in A$, то $d_1 \nabla d_2 \uparrow$, інакше $d_1 \nabla d_2 = d_2$.

Якщо значення $d_1 \nabla d_2$ задано для всіх $d_1 \in NDVC_k(V,A)$, покажемо, як задати значення для таких $d_1 \in NDVC_{k+1}(V,A)$, що $d_1 \notin NDVC_k(V,A)$. Зробимо це задавши значення $d_1 \nabla d_2 = d$ для довільного $v \in V^{+}$.

- якщо $d_2(v)\downarrow$, та в d_1 немає жодної послідовності імен, що були б префіксом v, то $d(v) = d_2(v)$;

– якщо $d_1(v)\downarrow$, та v не є префіксом жодного елементу з області визначення d_2 та d_2 не визначена на жодному префіксі v, то $d(v)=d_1(v)$;
– якщо $d_1(v)\downarrow$ та $d_1(v)\in A$, та d_2 визначено на імені, що містить v як префікс, то $d(v)=d_2/v$;
– якщо $d_1(v)\downarrow$ та $d_1(v)\notin A$, та d_2 визначено на імені, що містить v як префікс, то $d(v)=d_1\nabla(d_2/v)$;
– інакше $d(v)\uparrow$.

Семантику програмних логік з ієрархічними номінативними даними будемо задавати за допомогою програмних алгебр часткових квазіарних предикатів та часткових біквазіарних функції.

Частковими квазіарними предикатами над ієрархічними номінативними даними будемо називати відображення $Pr_{CC}^{V,A}:NDVC(V,A)\xrightarrow{p}\{T,F\}$. За допомогою них будемо задавати семантику умов в програмах.

Частковими біквазіарними функціями над ієрархічними номінативними даними будемо називати відображення типу $FPrg_{CC}^{V,A}:NDVC(V,A)\xrightarrow{p}NDVC(V,A)$. За допомогою яких будемо задавати семантику операторів та виразів. На відміну від простих номінативних даних семантика як операторів так і арифметичних виразів задається за допомогою біквазіарних функцій.

Семантика програмних задається за допомогою квазіарної програмної алгебри на ієрархічними номінативними даними з двома основами $Pr_{CC}^{V,A},FPrg_{CC}^{V,A}$ наступного вигляду:

$$IQPA(V,A)=<Pr_{CC}^{V,A},FPrg_{CC}^{V,A};\vee,\neg,\{S_P^{\bar v}\}_{\bar v\in\bar U(V^+)},\{S_F^{\bar v}\}_{\bar v\in\bar U(V^+)},\{`x\}_{x\in V^+},\{\exists x\}_{x\in V^+},id,$$
$$\{AS^x\}_{x\in V^+},\bullet,IF,WH,FH,PC>$$

Де $\bar U(V^+)$ позначаємо множину всіх скінчених послідовностей з V^+, в яких жоден елемент не є префіксом іншого елементу.

Дамо визначення всіх композицій. Як і у випадку простих номінативних даних, спочатку розглянемо композиції предикатного рівня, потім композицій предикатно-функціонального рівня і програмного рівня.

Бінарна композиція диз'юнкції $\vee$ має тип $Pr_{CC}^{V,A} \times Pr_{CC}^{V,A} \xrightarrow{t} Pr_{CC}^{V,A}$, та визначається наступним чином, де $p \in Pr_{CC}^{V,A}, q \in Pr_{CC}^{V,A}, d \in NDVC(V,A)$:

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

Унарна композиція заперечення $\neg$ має тип $Pr_{CC}^{V,A} \xrightarrow{t} Pr_{CC}^{V,A}$, та визначається наступним чином, де $p \in Pr_{CC}^{V,A}, d \in NDVC(V,A)$:

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

Унарна параметрична композиція екзистенціальної квантифікації $\exists x$ з параметром $x \in V^{+}$ має тип $Pr_{CC}^{V,A} \xrightarrow{t} Pr_{CC}^{V,A}$, та визначається наступним чином, де $p \in Pr_{CC}^{V,A}, q \in Pr_{CC}^{V,A}, d \in NDVC(V,A)$:

$$(\exists x\, p)(d) = \begin{cases} T, \text{ якщо } p(d\nabla[x \mapsto d'])\downarrow= T \text{ для деякого } d' \in NDVC(V,A), \\ F, \text{ якщо } p(d\nabla[x \mapsto d'])\downarrow= F \text{ для всіх } d' \in NDVC(V,A), \\ \text{невизначено інакше.} \end{cases}$$

Композиція суперпозиції для функцій $S_F^{x_1,x_2,\ldots x_n}$ є $n+1$-арною параметричною композицією з параметрами $x_1, x_2, \ldots, x_n \in V^{+}$ (причому жоден з елементів послідовності не є префіксом іншого) ,має тип $(FPrg_{CC}^{V,A})^{n+1} \xrightarrow{t} FPrg_{CC}^{V,A}$, та визначається наступним чином, де $f, g_1, g_2, \ldots, g_n \in FPrg_{CC}^{V,A}, d \in NDVC(V,A)$:

$$S_F^{x_1,x_2,\ldots,x_n}(f, g_1, g_2, \ldots, g_n)(d) \; ; \; f(d\nabla[x_1 \mapsto g_1(d), x_2 \mapsto g_2(d), \ldots, x_n \mapsto g_n(d)]).$$

Композиція суперпозиції для предикатів $S_P^{x_1,x_2,\ldots x_n}$ є $n+1$-арною параметричною композицією з параметрами $x_1, x_2, \ldots, x_n \in V^{+}$ (причому жоден з елементів послідовності не є префіксом іншого) ,має тип

$Pr_{CC}^{V,A} \times (FPrg_{CC}^{V,A})^n \xrightarrow{t} Pr_{CC}^{V,A}$, та визначається по аналогії до композиції суперпозиції для функцій наступним чином, де $p \in Pr_{CC}^{V,A}, g_1, g_2, \ldots, g_n \in FPrg_{CC}^{V,A}, d \in NDVC(V,A)$:

$$S_P^{x_1,x_2,\ldots,x_n}(p, g_1, g_2, \ldots, g_n)(d) ; \; p(d \nabla [x_1 \mapsto g_1(d), x_2 \mapsto g_2(d), \ldots, x_n \mapsto g_n(d)]).$$

Нуль-арна параметрична композиція розіменування $'x$ з параметром $x \in V^+$, $'x \in FPrg_{CC}^{V,A}$, якщо $d \in NDVC(V,A)$ то значення функції визначається за допомогою оператора розіменування:

$$'x(d) ; \; x \Rightarrow (d).$$

Унарна параметрична композиція присвоєння AS^x з параметром $x \in V^+$ має тип $FPrg_{CC}^{V,A} \xrightarrow{t} FPrg_{CC}^{V,A}$, та визначається наступним чином, де $f \in FPrg_{CC}^{V,A}, d \in NDVC(V,A)$:

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

Бінарна композиція послідовного виконання $\bullet$ має тип $FPrg_{CC}^{V,A} \times FPrg_{CC}^{V,A} \xrightarrow{t} FPrg_{CC}^{V,A}$, та визначається наступним чином, де $pr_1, pr_2 \in FPrg_{CC}^{V,A}, d \in NDVC(V,A)$:

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

Тернарна умовна композиція IF має тип $Pr_{CC}^{V,A} \times FPrg_{CC}^{V,A} \times FPrg_{CC}^{V,A} \xrightarrow{t} FPrg_{CC}^{V,A}$, та визначається наступним чином, де $r \in Pr_{CC}^{V,A}, pr_1, pr_2 \in FPrg_{CC}^{V,A}, d \in {}^V A$:

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

Бінарна композиція циклу WH має тип $Pr_{CC}^{V,A} \times FPrg_{CC}^{V,A} \xrightarrow{t} FPrg_{CC}^{V,A}$, та визначається наступним чином, де $r \in Pr_{CC}^{V,A}, pr \in FPrg_{CC}^{V,A}, d \in NDVC(V,A)$:

$WH(r,pr)(d)=d'$, якщо існує послідовність номінативних множин $d_0,d_1,\ldots,d_n \in {}^V A$ таких, що $d_0=d, d_n=d'$ та $d_1=pr(d_0), d_2=pr(d_1),\ldots,d_n=pr(d_{n-1})$ разом з $r(d_0)=r(d_1)=\ldots=r(d_{n-1})=T,\ r(d_n)=F$. Інакше, $WH(r,pr)(d)\uparrow$.

Нуль-арна композиція тотожної програми $id \in FPrg_{CC}^{V,A}$, визначається наступним чином, де $d \in NDVC(V,A)$:

$id(d)=d$.

Тернарна композиція Флойда-Хоара FH має тип $Pr_{CC}^{V,A} \times FPrg_{CC}^{V,A} \times Pr_{CC}^{V,A} \xrightarrow{t} Pr_{CC}^{V,A}$, та визначається наступним чином, де $p,q \in Pr_{CC}^{V,A}, pr \in FPrg_{CC}^{V,A}, d \in NDVC(V,A)$:

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

Бінарна композиція побудови умови за прообразом PC має тип $FPrg_{CC}^{V,A} \times Pr_{CC}^{V,A} \xrightarrow{t} Pr_{CC}^{V,A}$, та визначається наступним чином, де $q \in Pr_{CC}^{V,A}, pr \in FPrg_{CC}^{V,A}, d \in NDVC(V,A)$:

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

Твердження 2.6. Композиція Флойда-Хоара монотонна та неперервна .

Доведення твердження проводиться аналогічно до доведення твердження для програмних алгебр над простими номінативними даними.

Твердження 2.7. Композиція побудови умови за прообразом монотонна та неперервна.

Доведення твердження проводиться аналогічно до доведення твердження для програмних алгебр над простими номінативними даними.

Згідно з семантико-синтаксичним підходом синтаксис програмної логіки будемо задавати як множину термів програмної алгебри. Нехай задано множини предикатних символів Ps, множини програмних та функціональних символів Prs

та множину імен змінних V. Визначимо індуктивно множини формул $Fr(Ps,Prs,V)$, текстів програм $Pt(Ps,Prs,V)$, та асерцій Флойда-Хоара $FHFr(Ps,Prs,V)$.

Формули $Fr(Ps,Prs,V)$ будемо визначати наступним чином:

– якщо $p \in Ps$, то $p \in Fr(Ps,Prs,V)$;

– якщо $\Phi \in Fr(Ps,Prs,V)$, то $\neg\Phi \in Fr(Ps,Prs,V)$;

– якщо $\Phi \in Fr(Ps,Prs,V)$ та $x \in V^{+}$, то $\exists x\Phi \in Fr(Ps,Prs,V)$;

– якщо $\Phi,\Psi \in Fr(Ps,Prs,V)$, то $\Phi \vee \Psi \in Fr(Ps,Prs,V)$;

– якщо $p \in Ps$, $t_1,t_2,\ldots,t_n \in Pt(Ps,Prs,V)$ та $x_1,x_2,\ldots,x_n \in V^{+}$, де $x_1,x_2,\ldots,x_n$ – різні імена змінних, причому жодне з імен не є префіксом іншого та $n>0$, то $S_P^{x_1,x_2,\ldots,x_n}(p,t_1,t_2,\ldots,t_n) \in Fr(Ps,Prs,V)$;

– якщо $q \in Fr(Ps,Prs,V)$ разом з $pr \in Pt(Ps,Prs,V)$, то відповідно $PC(pr,q) \in Fr(Ps,Prs,V)$.

Тексти програм $Pt(Ps,Fs,Prs,V)$ будемо визначати наступним чином:

– якщо $pr \in Prs$, то $pr \in Pt(Ps,Prs,V)$;

– якщо $x \in V^{+}$, то $'x \in Pt(Ps,Prs,V)$;

– якщо $f \in Prs$, $t_1,t_2,\ldots,t_n \in Pt(Ps,Prs,V)$ та $x_1,x_2,\ldots,x_n \in V$, де $x_1,x_2,\ldots,x_n$ - різні імена змінних, , причому жодне з імен не є префіксом іншого та $n>0$, то $S_F^{x_1,x_2,\ldots,x_n}(f,t_1,t_2,\ldots,t_n) \in Pt(Ps,Prs,V)$.

– $id \in Pt(Ps,Prs,V)$;

– якщо $f \in Pt(Ps,Prs,V)$ та $x \in V^{+}$, то $AS^{x}(f) \in Pt(Ps,Prs,V)$;

– якщо $pr_1, pr_2 \in Pt(Ps,Prs,V)$, то $pr_1 \bullet pr_2 \in Pt(Ps,Prs,V)$;

– якщо $pr_1, pr_2 \in Pt(Ps,Prs,V)$ та $r \in Fr(Ps,Prs,V)$, то $IF(r, pr_1, pr_2) \in Pt(Ps,Prs,V)$;

– якщо $pr \in Pt(Ps,Prs,V)$ та $r \in Fr(Ps,Prs,V)$, то $WH(r, pr) \in Pt(Ps,Prs,V)$.

Асерції Флойда-Хоара $FHFr(Ps,Prs,V)$ будемо визначати наступним чином:

– якщо $pr \in Pt(Ps, Prs, V)$ та $p, q \in Fr(Ps, Prs, V)$, то $\{p\} pr \{q\} \in FHFr(Ps, Prs, V)$.

Для того, що задати зв'язок між синтаксисом та семантикою програмної логіки композиційно-номінативного типу над ієрархічними даними, потрібно задати інтерпретацію, тобто відображення, що спів ставляє формулам та текстам програм предикати та функції програмної алгебри, що їм відповідають. Для цього мають бути задані відображення для предикатних та програмних символів.

Нехай задано тотальні відображення $I_{Ps} : Ps \xrightarrow{t} Pr_{CC}^{V,A}$ – інтерпретації для предикатних символів та $I_{Prs} : Prs \xrightarrow{t} FPrg_{CC}^{V,A}$ – інтерпретації для програмних символів. Потрібно побудувати розширення даних відображень на формули $J_{Fr} : Fr(Ps, Prs, V) \xrightarrow{t} Pr_{CC}^{V,A}$, тексти програм $J_{Pt} : Pt(Ps, Prs, V) \xrightarrow{t} FPrg_{CC}^{V,A}$ та асерції Флойда-Хоара $J_{FHFr} : FHFr(Ps, Prs, V) \xrightarrow{t} Pr_{CC}^{V,A}$ індуктивно.

Відображення $J_{Fr} : Fr(Ps, Prs, V) \xrightarrow{t} Pr_{CC}^{V,A}$:

– $J_{Fr}(p) = I_{Ps}(p)$, якщо $p \in Ps$;

– $J_{Fr}(\neg\Phi) = \neg J_{Fr}(\Phi)$;

– $J_{Fr}(\exists x\Phi) = \exists x J_{Fr}(\Phi)$;

– $J_{Fr}(\Phi \vee \Psi) = J_{Fr}(\Phi) \vee J_{Fr}(\Psi)$;

– $J_{Fr}(S_P^{x_1, x_2, \ldots, x_n}(p, t_1, t_2, \ldots, t_n)) = S_P^{x_1, x_2, \ldots, x_n}(J_{Fr}(f), J_{Pt}(t_1), J_{Pt}(t_2), \ldots, J_{Pt}(t_n))$;

– $J_{Fr}(PC(pr, q)) = PC(J_{Pt}(pr), J_{Fr}(q))$.

Відображення $J_{Pt} : Pt(Ps, Prs, V) \xrightarrow{t} FPrg_{CC}^{V,A}$:

– $J_{Pt}(pr) = I_{Prs}(pr)$, якщо $pr \in Prs$;

– $J_{Pt}({}`x) = {}`x$;

– $J_{Pt}(S_F^{x_1, x_2, \ldots, x_n}(f, t_1, t_2, \ldots, t_n)) = S_F^{x_1, x_2, \ldots, x_n}(J_{Pt}(f), J_{Pt}(t_1), J_{Pt}(t_2), \ldots, J_{Pt}(t_n))$;

– $J_{Pt}(id) = id$;

– $J_{Pt}(AS^x(f)) = AS^x(J_{Pt}(f))$;

– $J_{Pt}(pr_1 \bullet pr_2) = J_{Pt}(pr_1) \bullet J_{Pt}(pr_2)$;

– $J_{Pt}(IF(r, pr_1, pr_2)) = IF(J_{Fr}(r), J_{Pt}(pr_1), J_{Pt}(pr_2)))$;

– $J_{Pt}(WH(r, pr)) = WH(J_{Fr}(r), J_{Pt}(pr))$.

Відображення $J_{FHFr} : FHFr(Ps, Prs, V) \xrightarrow{t} Pr_{CC}^{V,A}$:

– $J_{FHFr}(\{p\}pr\{q\}) = FH(J_{Fr}(p), J_{Pt}(pr), J_{Fr}(q))$.

За допомогою визначень композиції програмної алгебри $IQPA(V, A)$ та відображень для базових символів I_{Ps} та I_{Prs} можна однозначно задати інтерпретацію J, яка є відображення, що кожному елементу мови програмної логіки композиційно-номінативного типу для ієрархічних даних ставить у відповідність відображення програмної алгебри $IQPA(V, A)$.

Висновки

Програмні алгебри часткових квазіарних предикатів та функцій дозволяють адекватно задавати семантичний аспект програмних логік. В попередньому розділі було показано, що композиція програмного рівня, що може бути отримана при використанні класичного визначення часткової істинності асерції Флойда-Хоара не є монотонною для випадку часткових предикатів, що зумовлює необхідність в побудові монотонної композиції Флойда-Хоара використовуючи принцип максимальної визначеності. Представлена композиція є не лише монотонною, а й неперервною.

Вводиться композиція побудови умови за прообразом, яка є аналогом перетворювача предикатів найслабкіша передумова та оператора передбачення систем алгоритмічних алгебр Глушкова. Така композиція також є монотонною та неперервною. Для даної композиції доведено рівності, що є модифікованими аналогами перетворень для найслабкішої передумови.

Семантико-синтаксичний підхід означає, що спочатку визначається семантичний аспект, за яким будується синтаксичний. Таким чином, маючи програмні алгебри часткових квазіарних предикатів та функцій, можна задати мову логіки, як множину термів відповідної алгебри. В такому випадку інтерпретаційний аспект природно випливає з визначень композицій.

На найвищому рівні абстракції можна виділити такі випадки складних ієрархічних номінативних даних: номінативні дані зі складними іменами та простими значеннями, номінативні дані з простими іменами та складними значеннями та номінативні дані зі складними іменами та складними значеннями. В роботі розглядається випадок складних імен та складних значень. Будуються програмні алгебри часткових квазіарних предикатів та функцій над ієрархічними номінативними даними. Слід відмітити, що в таких алгебрах не розрізняються ординарні квазіарні функції та біквазіарні функції. Запропоновані композиції Флойда-Хоара та побудови умови за прообразом також є монотонними та неперервними. Як і у випадку простих номінативних даних, семантико-синтаксичний підхід дозволяє просто визначити мову програмної логіки, задавши відповідну програмну алгебру.

РОЗДІЛ 3
АКСІОМАТИЧНІ СИСТЕМИ ФЛОЙДА-ХОАРІВСЬКОГО ТИПУ ДЛЯ ЧАСТКОВИХ ПРЕДИКАТІВ

Для того, щоб програмну логіку композиційно-номінативного типу можна було застосовувати для доведення коректності програм, необхідно побудувати аксіоматичну систему. Така аксіоматична система повинна бути коректною та повною. Той факт, що формула Φ виводиться в аксіоматичній системі I будемо позначати $\vdash_I \Phi$, або якщо відомо, яка аксіоматична система мається на увазі, $\vdash \Phi$. Також якщо задано інтерпретацію J, то будемо позначати предикат, що відповідає формулі Φ в інтерпретації J відповідно Φ_J.

Визначення. Аксіоматична система I називається коректною, якщо для довільної формули Φ виконується $\vdash_I \Phi \Rightarrow \models \Phi$.

Визначення. Аксіоматична система I називається повною, якщо для довільної формули Φ виконується $\models \Phi \Rightarrow \vdash_I \Phi$.

Існує декілька підходів до розгляду повноти аксіоматичної системи – екстенсіональний та інтенсіональний. За екстенсіональним підходом, передумови та післяумови можуть представляти довільний предикат. Проте за інтесіональним підходом, передумови та післяумови повинні бути представлені формулами заданої мови.

3.1 Аксіоматична система логіки Флойда-Хоара для тотальних предикатів

Спочатку розглянемо систему правил виводу в класичній логіці Флойда-Хоара для тотальних предикатів задану для мови WHILE[41]. Запишемо її в термінах програмної логіки над простими номінативними даними, позначимо таку аксіоматичну систему *CI*. Потрібно перевірити, чи буде така аксіоматична система повною та коректною.

$$R\_AS \quad \overline{\{S_P^x(p,f)\}AS^x(f)\{p\}},$$

$$R\_SKIP \quad \overline{\{p\}id\{p\}},$$

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

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

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

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

Твердження 3.1. Аксіоматична система CI не є коректною.

Доведення. Розглянемо формули p, q та r разом з інтерпретацією J такі, що $p_J(d)\downarrow= T$, $r_J(d)\downarrow= F$ та $q_J(d)\uparrow$ для довільного $d \in {}^V A$. Тоді за правилом R_SKIP маємо $\vdash_{CI} \{p\}id\{p\}$, звідки так як $p \to q$, за правилом R_CONS отримуємо $\vdash_{CI} \{p\}id\{q\}$. Аналогічно так як $q \to r$, та $\vdash_{CI} \{q\}id\{r\}$ за правилом R_CONS виводиться $\vdash_{CI} \{q\}id\{r\}$. Застосувавши до виведених асерцій правило R_SEQ маємо $\vdash_{CI} \{p\}id \bullet id\{r\}$. Але для довільного $d \in {}^V A$ $\{p\}id \bullet id\{r\}_J(d)\downarrow= F$ так як $p_J(d)\downarrow= T$, а за визначенням $id \bullet id(d) = id(id(d)) = d$ та $r_J(d)\downarrow= F$. Отже не виконується, що $\models \{p\}id \bullet id\{r\}$, звідки аксіоматична система не є коректною.

Варто зазначити, що можна навести аналогічні приклади також і для застосування правила R_WH. Одним зі шляхів розв'язання даної проблеми є додавання умов до правил виводу, що призводять до виведення хибних формул.

Нехай на множині формул програмної логіки $Fr(Ps,Fs,Prs,V)$ додатково задано бінарні відношення LC, TC та FC.

Визначення. Формула q є логічним наслідком формули p, позначатимемо $p \models q$, якщо $\models p \to q$.

Визначення. Формула q є логічним наслідком формули p за істинною, позначатимемо $p \models_T q$, якщо $p_J^T \subseteq q_J^T$ для довільної інтерпретації J.

Визначення. Формула q є логічним наслідком формули p за хибою, позначатимемо $p \models_F q$, якщо $q_J^F \subseteq p_J^F$ для довільної інтерпретації J. В подальшому, якщо з контексту зрозуміло, що мається на увазі, формула логіки, чи предикат, що відповідає їй в заданій інтерпретації, не будемо явно їх розрізняти.

3.2 Аксіоматична система з доданими обмеженнями

Розглянемо наступну аксіоматичну систему з доданими обмеженнями AC:

$$R\_AS \quad \frac{}{\{S_P^x(p, f)\}AS^x(f)\{p\}},$$

$$R\_SKIP \quad \frac{}{\{p\}id\{p\}},$$

$$R\_SEQ' \quad \frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1 \bullet pr_2\{r\}}, LC(p, PC(pr_1 \bullet pr_2, r)),$$

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

$$R\_WH' \quad \frac{\{r \wedge p\}pr\{p\}}{\{p\}WH(r, pr)\{\neg r \wedge p\}}, LC(p, PC(WH(r, pr), \neg r \wedge p)),$$

$$R\_CONS' \quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}}, TC(p, p'), FC(q', q).$$

Теорема 3.1. *Якщо відношення LC співпадає з відношенням логічного наслідку, відношення TC співпадає з відношенням логічного наслідку за істиною та відношення FC співпадає з відношенням логічного наслідку за хибою, аксіоматична система AC коректна.*

Доведення. Доведемо теорему індукцією за довжиною виводу.

База індукції. За один крок можна вивести асерції лише використовуючи правила R_AS та R_SKIP.

Розглянемо правило *R_AS*. Маємо $\vdash \{S_P^x(p,f)\}AS^x(f)\{p\}$. Отже, потрібно показати, що $\models \{S_P^x(p,f)\}AS^x(f)\{p\}$. Іншими словами, що в будь-якій інтерпретації не існує такого $d \in {}^V A$, для якого $FH(S_P^x(p,f), AS^x(f), p)(d) \downarrow = F$. За визначенням композиції Флойда-Хоара, це можливо якщо $S_P^x(p,f)(d) \downarrow = T$ та $p((AS^x(f))(d)) \downarrow = F$. Звідки з визначень відповідних композицій отримуємо $p(d \nabla [x \mapsto f(d)]) \downarrow = T$ та $p(d \nabla [x \mapsto f(d)]) \downarrow = F$. Прийшли до протиріччя, а отже $\models \{S_P^x(p,f)\}AS^x(f)\{p\}$.

Розглянемо правило *R_SKIP*. Так як $\vdash \{p\}id\{p\}$, потрібно показати, що $\models \{p\}id\{p\}$. За визначенням це означає, що в будь-якій інтерпретації не існує такого $d \in {}^V A$, для якого $FH(p, id, p)(d) \downarrow = F$. Що можливо лише якщо $p(d) \downarrow = T$ та $p(id(d)) = p(d) \downarrow = F$. Маємо протиріччя, таким чином $\models \{p\}id\{p\}$.

Крок індукції. Припустимо, що всі асерції, що виводяться менше ніж за *k* кроків істинні, доведемо, що і для *k* також виконується дане твердження. Для цього розглянемо різні випадки в залежності від того, яке правило було застосовано на останньому кроці.

Розглянемо правило *R_SEQ'*. Маємо $\vdash \{p\}pr_1 \bullet pr_2\{r\}$, що було виведено з $\vdash \{p\}pr_1\{q\}$ та $\vdash \{q\}pr_2\{r\}$ за умови, що $LC(p, PC(pr_1 \bullet pr_2, r))$. Так як $\vdash \{p\}pr_1\{q\}$ та $\vdash \{q\}pr_2\{r\}$ було отримано менше ніж за *k* кроків, то за припущенням індукції $\models \{p\}pr_1\{q\}$ та $\models \{q\}pr_2\{r\}$. Також з того, що відношення LC співпадає з відношенням логічного наслідку отримуємо $p \models PC(pr_1 \bullet pr_2, r)$. З визначення логічного наслідку маємо $\models p \to PC(pr_1 \bullet pr_2, r)$, що означає, що для довільної інтерпретації $(p \to PC(pr_1 \bullet pr_2, r))^F = \varnothing$, звідки $p^T \cap PC(pr_1 \bullet pr_2, r)^F = \varnothing$. Проте, за означенням композиції побудови умови за прообразом $PC(pr_1 \bullet pr_2, r)^F = (pr_1 \bullet pr_2)^{-1}[r^F] = r^{-F, pr_1 \bullet pr_2}$. Підставляючи в попередню рівність маємо $p^T \cap r^{-F, pr_1 \bullet pr_2} = \varnothing$. Так як $FH(p, pr_1 \bullet pr_2, r)^F = p^T \cap r^{-F, pr_1 \bullet pr_2}$, отже відповідна асерція істинна на довільній інтерпретації, а отже $\models \{p\}pr_1 \bullet pr_2\{r\}$.

Розглянемо правило *R_IF*. В такому випадку отримуємо $|-\{p\}IF(r, pr_1, pr_2)\{q\}$, що виводиться згідно правила з $|-\{r \wedge p\}pr_1\{q\}$ та $|-\{\neg r \wedge p\}pr_2\{q\}$. З того, що $|-\{r \wedge p\}pr_1\{q\}$ та $|-\{\neg r \wedge p\}pr_2\{q\}$ виводиться менше ніж за k кроків за припущенням індукції маємо $|=\{r \wedge p\}pr_1\{q\}$ та $|=\{\neg r \wedge p\}pr_2\{q\}$. Зафіксуємо деяку довільну інтерпретацію та покажемо, що в такому випадку $FH(p, IF(r, pr_1, pr_2), q)^F = \varnothing$. В такому випадку, так як інтерпретація була обрана довільним чином, то твердження виконується для будь-якої інтерпретації, а отже $|=\{p\}IF(r, pr_1, pr_2)\{q\}$.

Для обраної інтерпретації розглянемо довільне $d \in {}^V A$, таке, що $p(d)\downarrow = T$. За визначенням лише такі дані можуть належати $FH(p, IF(r, pr_1, pr_2), q)^F$. Тоді розглянемо різні випадки для значень $r(d)$.

Нехай $r(d)\downarrow = T$. В такому випадку $IF(r, pr_1, pr_2)(d) \cong pr_1(d)$ і якщо $pr_1(d)\uparrow$, то також $IF(r, pr_1, pr_2)(d)\uparrow$, а отже $d \notin q^{-F, IF(r, pr_1, pr_2)}$ і значить $d \notin FH(p, IF(r, pr_1, pr_2), q)^F$. У випадку, якщо $pr_1(d)\downarrow$, то виконується $IF(r, pr_1, pr_2)(d) = pr_1(d)$, а отже $d \notin q^{-F, pr_1} \Leftrightarrow d \notin q^{-F, IF(r, pr_1, pr_2)}$. Зважаючи на те, що $|=\{r \wedge p\}pr_1\{q\}$, а отже $FH(r \wedge p, pr_1, q)^F = \varnothing$, а також $(r \wedge p)(d)\downarrow = T$ маємо $d \notin q^{-F, pr_1}$. Звідки $d \notin q^{-F, IF(r, pr_1, pr_2)}$, а отже $d \notin FH(p, IF(r, pr_1, pr_2), q)^F$. Таким чином при $r(d)\downarrow = T$ виконується $d \notin FH(p, IF(r, pr_1, pr_2), q)^F$.

Нехай виконується $r(d)\downarrow = F$. Тоді $IF(r, pr_1, pr_2)(d) \cong pr_2(d)$. В такому випадку якщо $pr_2(d)\uparrow$, то і $IF(r, pr_1, pr_2)(d)\uparrow$, звідки отримуємо $d \notin q^{-F, IF(r, pr_1, pr_2)}$ що рівносильне $d \notin FH(p, IF(r, pr_1, pr_2), q)^F$. Якщо ж має місце $pr_2(d)\downarrow$, то з узагальненої рівності маємо $IF(r, pr_1, pr_2)(d) = pr_2(d)$, а отже як і в попередньому випадку $d \notin q^{-F, pr_2} \Leftrightarrow d \notin q^{-F, IF(r, pr_1, pr_2)}$. Так як за припущенням $|=\{\neg r \wedge p\}pr_2\{q\}$, або ж для зафіксованої інтерпретації $FH(\neg r \wedge p, pr_2, q)^F = \varnothing$, що разом з $(\neg r \wedge p)(d)\downarrow = T$ дає $d \notin q^{-F, pr_2}$. Це еквівалентно $d \notin q^{-F, IF(r, pr_1, pr_2)}$, звідки

$d \notin FH(p, IF(r, pr_1, pr_2), q)^F$. Доведено, що у випадку $r(d)\downarrow = F$ також виконується $d \notin FH(p, IF(r, pr_1, pr_2), q)^F$.

Розглянемо останній випадок, коли $r(d)\uparrow$. В такому випадку $IF(r, pr_1, pr_2)(d)\uparrow$, а отже, $d \notin q^{-F, IF(r, pr_1, pr_2)}$ і $d \notin FH(p, IF(r, pr_1, pr_2), q)^F$.

Всі випадки було розглянуто, отримуємо $FH(p, IF(r, pr_1, pr_2), q)^F = \varnothing$, звідки $\models \{p\} IF(r, pr_1, pr_2) \{q\}$.

Розглянемо правило *R_WH'*. З $\vdash \{r \wedge p\} pr \{p\}$ виводимо відповідно до правила $\vdash \{p\} WH(r, pr) \{\neg r \wedge p\}$ за умови, що $LC(p, PC(WH(r, pr), \neg r \wedge p))$. Зважаючи на те, що $\vdash \{r \wedge p\} pr \{p\}$ було виведено менше ніж за k кроків, то за припущенням індукції $\models \{r \wedge p\} pr \{p\}$. Разом з тим, враховуючи, що відношення LC співпадає з відношенням логічного наслідку отримуємо $p \models PC(WH(r, pr), \neg r \wedge p)$. Звідки отримуємо $\models p \to PC(WH(r, pr), \neg r \wedge p)$, за визначення це означає, що для довільної інтерпретації $(p \to PC(WH(r, pr), \neg r \wedge p))^F = \varnothing$, що в свою чергу рівносильне $p^T \cap PC(WH(r, pr), \neg r \wedge p)^F = \varnothing$. Розгортаючи за визначенням композиції побудови умови за прообразом та враховуючи позначення, отримуємо $PC(WH(r, pr), \neg r \wedge p)^F = (WH(r, pr))^{-1}[\neg r \wedge p^F] = \neg r \wedge p^{-F, WH(r, pr)}$. Разом з попередньою рівністю маємо $p^T \cap \neg r \wedge p^{-F, WH(r, pr)} = \varnothing$. Що означає $FH(p, WH(r, pr), \neg r \wedge p)^F = \varnothing$, отже відповідна асерція істинна на довільній інтерпретації, а отже $\models \{p\} WH(r, pr) \{\neg r \wedge p\}$.

Розглянемо правило *R_CONS'*. Виводиться $\vdash \{p\} pr \{q\}$ з $\vdash \{p'\} pr \{q'\}$ для яких виконуються умови, $TC(p, p')$ та $FC(q', q)$. Також з того, що $\vdash \{p'\} pr \{q'\}$ виводиться менше ніж за k кроків, за припущенням індукції маємо $\models \{p'\} pr \{q'\}$. Зафіксуємо довільну інтерпретацію і доведемо, що для неї істинна $\{p\} pr \{q\}$. З того, що $\models \{p'\} pr \{q'\}$, випливає $FH(p', pr, q')^F = \varnothing$, що рівносильне $p'^T \cap q'^{-F, pr} = \varnothing$. Так як відношення TC співпадає з відношенням логічного наслідку за істиною, то $p^T \subseteq p'^T$. Аналогічно з того, що відношення FC співпадає з відношенням логічного наслідку за хибою, маємо $q'^F \subseteq q^F$. З останнього за монотонністю операції повного прообразу $q'^{-F,pr} \subseteq q^{-F,pr}$. Об'єднуємо всі твердження, враховуючи властивості перетину, $p^T \cap q^{-F,pr} \subseteq p'^T \cap q'^{-F,pr} = \varnothing$, отже $FH(p, pr, q)^F = p^T \cap q^{-F,pr} = \varnothing$.

Так як жодні припущення щодо властивостей інтерпретації не використовувались в доведенні, то твердження виконується для довільної інтерпретації, звідки $\models \{p\} pr \{q\}$.

Розглянуто всі правила, а отже за індукцією аксіоматична система коректна.

Теорема 3.2. Якщо відношення LC співпадає з відношенням логічного наслідку, відношення TC співпадає з відношенням логічного наслідку за істиною та відношення FC співпадає з відношенням логічного наслідку за хибою, аксіоматична система AC екстенсіонально повна.

Доведення. Доведемо спочатку, що для довільного програмного тексту pr та формули q, виконується $\vdash \{PC(pr, q)\} pr \{q\}$. Доведення будемо проводити індукцією за структурою pr.

База індукції. Найпростіший випадок, коли програмний текст це id або $AS^x(f)$.

Розглянемо випадок $AS^x(f)$. За правилом R_AS отримуємо $\vdash \{S_P^x(q, f)\} AS^x(f) \{q\}$. Згідно доведених раніше властивостей композиції побудови умови за прообразом для довільної інтерпретації виконується $PC(AS^x(f), q) = S_P^x(q, f)$, а отже і $PC(AS^x(f), q) \models_T S_P^x(q, f)$. Так як відношення TC співпадає з відношенням логічного наслідку за істиною, то $TC(PC(AS^x(f), q), S_P^x(q, f))$. Тоді використовуючи правило R_CONS' з $\vdash \{S_P^x(q, f)\} AS^x(f) \{q\}$ отримуємо $\vdash \{PC(AS^x(f), q)\} AS^x(f) \{q\}$.

Розглянемо випадок id. За правилом R_SKIP можна отримати $\vdash \{q\} id \{q\}$. Було доведено, що не залежно від інтерпретації має місце $PC(id, q) = q$, звідки

$PC(id,q) \models_T q$. Так як відношення TC співпадає з відношенням логічного наслідку за істиною, то $TC(PC(id,q),id)$. Додаткові умови виконуються, тому застосуємо правило *R_CONS'* до $|-\{q\}id\{q\}$, та отримаємо $|-\{PC(id,q)\}id\{q\}$.

Крок індукції. Потрібно розглянути випадки, коли програмний текст утворено з простіших за допомогою послідовного виконання, розгалуження та циклу.

Розглянемо випадок $pr_1 \bullet pr_2$. Потрібно довести, що $|-\{PC(pr_1 \bullet pr_2,q)\}pr_1 \bullet pr_2\{q\}$. За припущенням індукції для pr_1 та pr_2 маємо $|-\{PC(pr_2,q)\}pr_2\{q\}$ та $|-\{PC(pr_1,PC(pr_2,q))\}pr_1\{PC(pr_2,q)\}$. Якщо довести, що $LC(PC(pr_1,PC(pr_2,q)),PC(pr_1 \bullet pr_2,q))$, то можна застосувати правило *R_SEQ*' та отримати $|-\{PC(pr_1,PC(pr_2,q))\}pr_1 \bullet pr_2\{q\}$. Якщо виконується $TC(PC(pr_1 \bullet pr_2,q),PC(pr_1,PC(pr_2,q)))$, то використовуючи правило *R_CONS'* для $|-\{PC(pr_1,PC(pr_2,q))\}pr_1 \bullet pr_2\{q\}$ можна вивести $|-\{PC(pr_1 \bullet pr_2,q)\}pr_1 \bullet pr_2\{q\}$. Залишається лише показати, що відповідні умови виконуються.

Доведемо $LC(PC(pr_1,PC(pr_2,q)),PC(pr_1 \bullet pr_2,q))$. Так як LC співпадає з відношенням логічного наслідку, це означає, що потрібно довести $PC(pr_1,PC(pr_2,q)) \models PC(pr_1 \bullet pr_2,q)$. З визначення логічного наслідку це означає, що потрібно довести $\models PC(pr_1,PC(pr_2,q)) \to PC(pr_1 \bullet pr_2,q)$. Припустимо, що це не виконується, а отже, існує інтерпретація, така, що для неї $(PC(pr_1,PC(pr_2,q)) \to PC(pr_1 \bullet pr_2,q))^F \neq \varnothing$. Тоді існує таке $d \in {}^V A$, що $(PC(pr_1,PC(pr_2,q)) \to PC(pr_1 \bullet pr_2,q))(d) \downarrow= F$. Це рівносильне тому, що $PC(pr_1,PC(pr_2,q))(d) \downarrow= T$ та $PC(pr_1 \bullet pr_2,q)(d) \downarrow= F$. Послідовно перетворюючи за визначенням композиції побудови умови за прообразом кожне з тверджень отримуємо $(PC(pr_2,q))(pr_1(d)) \downarrow= T$ та $q(pr_2(pr_1(d))) \downarrow= T$ разом з $q(pr_1 \bullet pr_2(d)) \downarrow= F$. Звідки з визначення композиції послідовного виконання $q(pr_2(pr_1(d))) \downarrow= F$. Прийшли до протиріччя, отже умова виконується.

Доведемо $TC(PC(pr_1 \bullet pr_2, q), PC(pr_1, PC(pr_2, q)))$. Так як TC співпадає з відношенням логічного наслідку за істинною, це означає, що потрібно довести $PC(pr_1 \bullet pr_2, q) \models_T PC(pr_1, PC(pr_2, q))$. Розглянемо довільну інтерпретацію, та покажемо, що для будь-якого $d \in {}^V A$, якщо $d \in PC(pr_1 \bullet pr_2, q)^T$, то $d \in PC(pr_1, PC(pr_2, q))^T$. Використовуючи визначення відповідних композицій, $d \in PC(pr_1 \bullet pr_2, q)^T$ означає $q(pr_1 \bullet pr_2(d)) \downarrow = T$, тобто $q(pr_2(pr_1(d))) \downarrow = T$. Аналогічно $d \in PC(pr_1, PC(pr_2, q))^T$ еквівалентно $(PC(pr_2, q))(pr_1(d)) \downarrow = T$ та $q(pr_2(pr_1(d))) \downarrow = T$. Отримали однакові вирази, а отже $d \in PC(pr_1 \bullet pr_2, q)^T \Leftrightarrow d \in PC(pr_1, PC(pr_2, q))^T$. Так як розглядалась довільна інтерпретація, то отримуємо $PC(pr_1 \bullet pr_2, q) \models_T PC(pr_1, PC(pr_2, q))$.

Обидві умови доведено, отже $\vdash \{PC(pr_1 \bullet pr_2, q)\} pr_1 \bullet pr_2 \{q\}$.

Розглянемо випадок $IF(r, pr_1, pr_2)$. Потрібно довести, що $\vdash \{PC(IF(r, pr_1, pr_2), q)\} IF(r, pr_1, pr_2) \{q\}$. За припущенням індукції для pr_1 та pr_2 маємо $\vdash \{PC(pr_1, q)\} pr_1 \{q\}$ та $\vdash \{PC(pr_2, q)\} pr_2 \{q\}$. Якщо довести, що $TC(r \wedge PC(IF(r, pr_1, pr_2), q), PC(pr_1, q))$, можна застосувати правило *R_CONS'* до $\vdash \{PC(pr_1, q)\} pr_1 \{q\}$ та отримати $\vdash \{r \wedge PC(IF(r, pr_1, pr_2), q)\} pr_1 \{q\}$. Аналогічно, маючи $TC(\neg r \wedge PC(IF(r, pr_1, pr_2), q), PC(pr_2, q))$ виводиться $\vdash \{\neg r \wedge PC(IF(r, pr_1, pr_2), q)\} pr_2 \{q\}$. Тоді до двох виведених асерцій можна застосувати правило *R_IF* та вивести $\vdash \{PC(IF(r, pr_1, pr_2), q)\} IF(r, pr_1, pr_2) \{q\}$. Доведемо, що зазначені вище умови виконуються. Спочатку розглянемо $TC(r \wedge PC(IF(r, pr_1, pr_2), q), PC(pr_1, q))$. Так як TC співпадає з відношенням логічного наслідку за істинною, це означає, що потрібно довести $r \wedge PC(IF(r, pr_1, pr_2), q) \models_T PC(pr_1, q)$. Розглянемо довільну інтерпретацію та покажемо, що для довільного $d \in {}^V A$, якщо $(r \wedge PC(IF(r, pr_1, pr_2), q))(d) \downarrow = T$, то $(PC(pr_1, q))(d) \downarrow = T$. Перше твердження означає, що $r(d) \downarrow = T$ та $(PC(IF(r, pr_1, pr_2), q))(d) \downarrow = T$. З визначення композиції побудови умови за прообразом отримуємо $q((IF(r, pr_1, pr_2))(d))\downarrow= T$. Так як $r(d)\downarrow= T$, то умовну композицію можна розписати як $(IF(r, pr_1, pr_2))(d)\downarrow= pr_1(d)$, тому виконується $q(pr_1(d))\downarrow= T$, звідки в свою чергу $(PC(pr_1,q))(d)\downarrow= T$, що і треба було довести. З довільного вибору інтерпретації отримуємо $r \wedge PC(IF(r, pr_1, pr_2),q) \models_T PC(pr_1,q)$.

За аналогією до попередньої умови доведемо $TC(\neg r \wedge PC(IF(r, pr_1, pr_2),q), PC(pr_2,q))$. Так як TC співпадає з відношенням логічного наслідку за істинною, умова еквівалентна $\neg r \wedge PC(IF(r, pr_1, pr_2),q) \models_T PC(pr_2,q)$. Для доведення даного твердження візьмемо довільну інтерпретацію та покажемо включення областей істинності. Покажемо, що для довільного $d \in {}^V A$, якщо $(\neg r \wedge PC(IF(r, pr_1, pr_2),q))(d)\downarrow= T$, то $(PC(pr_2,q))(d)\downarrow= T$. З визначення кон'юнкції отримуємо $r(d)\downarrow= F$ та $(PC(IF(r, pr_1, pr_2),q))(d)\downarrow= T$. З визначення умовної композиції при $r(d)\downarrow= F$ виконується $(IF(r, pr_1, pr_2))(d)\downarrow= pr_2(d)$. За визначеннями композицій тоді $(PC(IF(r, pr_1, pr_2),q))(d)\downarrow= T$ еквівалентно $q((IF(r, pr_1, pr_2))(d))\downarrow= T$, що в свою чергу еквівалентно $q(pr_2(d))\downarrow= T$. З отриманого твердження слідує $(PC(pr_2,q))(d)\downarrow= T$, що і треба було довести. Так як інтерпретація була обрана довільним чином, доведено $\neg r \wedge PC(IF(r, pr_1, pr_2),q) \models_T PC(pr_2,q)$.

Всі необхідні умови розглянуто, звідки виконується шукане твердження $\vdash \{PC(IF(r, pr_1, pr_2),q)\} IF(r, pr_1, pr_2) \{q\}$.

Розглянемо випадок циклу, $WH(r, pr)$. В такому разі, потрібно довести, що $\vdash \{PC(WH(r, pr),q)\} WH(r, pr) \{q\}$. Для спрощення викладок позначимо $p = PC(WH(r, pr),q)$. Так як розглядається екстенсіональна повнота, вважатимемо, що існує формула p' така, що для довільної інтерпретації для довільного $d \in {}^V A$ якщо $p(d)\downarrow= T$, то $p'(d)\downarrow= T$, інакше $p'(d)\downarrow= F$. За припущенням індукції для pr маємо $\vdash \{PC(pr, p')\} pr \{p'\}$. Якщо довести, що

$TC(r \wedge p', PC(pr, p'))$, можна застосувати правило *R_CONS'* до $\vdash \{PC(pr, p')\} pr \{p'\}$ та отримати $\vdash \{r \wedge p'\} pr \{p'\}$. Тоді, якщо виконується $LC(p', PC(WH(r, pr), \neg r \wedge p'))$ можна застосувати правило *R_WH'* для того, щоб вивести $\vdash \{p'\} WH(r, pr) \{\neg r \wedge p'\}$. Якщо $FC(\neg r \wedge p', q)$ та $TC(p, p')$, можна з $\vdash \{p'\} WH(r, pr) \{\neg r \wedge p'\}$ за правилом *R_CONS'* отримати $\vdash \{p\} WH(r, pr) \{q\}$, тобто $\vdash \{PC(WH(r, pr), q)\} WH(r, pr) \{q\}$.

Доведемо, що виконуються необхідні умови. Розглянемо спочатку $TC(r \wedge p', PC(pr, p'))$. Так як TC співпадає з відношенням логічного наслідку за істинною, умова еквівалентна $r \wedge p' \models_T PC(pr, p')$. Доведемо, що включення областей істинності виконується для будь-якої інтерпретації. Якщо для деякого $d \in {}^V A$ має місце $(r \wedge p')(d) \downarrow = T$, то $p'(d) \downarrow = T$ та $r(d) \downarrow = T$. Якщо $p'(d) \downarrow = T$, то $p(d) \downarrow = T$, отже $(PC(WH(r, pr), q))(d) \downarrow = T$. З визначення композиції $q((WH(r, pr))(d)) \downarrow = T$. За визначенням циклічної композиції, так як $r(d) \downarrow = T$, то можна спочатку обчислити першу ітерацію циклу, а потім продовжити цикл. Іншими словами виконується $(W\ H(\ r,\ p\ )r\downarrow) = (d\ ) \quad W(\ H\ (r\ \downarrow_I$, отже $p(pr(d)) \downarrow = T$, звідки $p'(pr(d)) \downarrow = T$. Це еквівалентно $(PC(pr, p'))(d) \downarrow = T$. Так як розглядалась довільна інтерпретація маємо $r \wedge p' \models_T PC(pr, p')$.

Розглянемо умову $LC(p', PC(WH(r, pr), \neg r \wedge p'))$. Так як LC співпадає з відношенням логічного наслідку, це означає, що потрібно довести $p' \models PC(WH(r, pr), \neg r \wedge p')$. З визначення логічного наслідку це означає, що потрібно довести $\models p' \to PC(WH(r, pr), \neg r \wedge p')$. Припустимо, що це не виконується, а отже, існує інтерпретація, така, що для неї існує $d \in {}^V A$, що $(p' \to PC(WH(r, pr), \neg r \wedge p'))(d) \downarrow = F$. Це рівносильне тому, що $p'(d) \downarrow = T$ та $(PC(WH(r, pr), \neg r \wedge p'))(d) \downarrow = F$. За визначенням композиції побудови умови за прообразом отримуємо $(\neg r \wedge p')((WH(r, pr))(d)) \downarrow = F$. Так як $(WH(r, pr))(d) \downarrow$, позначимо $d' = (WH(r, pr))(d)$, то за визначенням циклічної композиції

$(\neg r)(d')\downarrow = T$, тоді лишається тільки випадок $p'(d')\downarrow = F$. Якщо $p'(d)\downarrow = T$, то за побудовою $p(d)\downarrow = T$, тобто $(PC(WH(r,pr),q))(d)\downarrow = T$. Це еквівалентно $q((WH(r,pr))(d))\downarrow = T$. За визначенням циклічної композиції якщо $r(d')\downarrow = F$, то $(WH(r,pr))(d') = d'$, підставимо в попереднє твердження та отримаємо $q((WH(r,pr))(d'))\downarrow = T$. Згортаючи за визначеннями отримаємо $p(d')\downarrow = T$, а отже і $p'(d')\downarrow = T$. Прийшли до протиріччя, отже умова виконується.

Умова $TC(p,p')$ виконується так як за припущенням для кожної інтерпретації області істинності відповідних предикатів співпадають. Залишається лише довести, що виконується $FC(\neg r \wedge p', q)$. Так як FC співпадає з відношенням логічного наслідку за хибою, це означає, що потрібно довести $\neg r \wedge p' \models_F q$. Візьмемо довільну інтерпретацію та покажемо включення областей хибності. Якщо для деякого $d \in {}^V A$ має місце $q(d)\downarrow = F$, потрібно показати, що $(\neg r \wedge p')(d)\downarrow = F$. Розглянемо різні випадки для r. Якщо $r(d)\downarrow = T$, то все виконується. Якщо $r(d)\downarrow = F$, то $WH(r,pr)(d)\downarrow = d$, тоді $p(d)\downarrow = q(d) = F$. За припущенням $p'(d)\downarrow = F$, звідки $(\neg r \wedge p')(d)\downarrow = F$. Якщо $r(d)\uparrow$, то $WH(r,pr)(d)\uparrow$, а отже $p(d)\uparrow$ та за припущенням $p'(d)\downarrow = F$ і твердження виконується. Розглянуто всі випадки, отже виконується включення областей хибності для довільної інтерпретації і $\neg r \wedge p' \models_F q$.

Всі умови було доведено, отже $\vdash \{PC(WH(r,pr),q)\} WH(r,pr)\{q\}$.

Розглянуто всі можливі випадки і за індукцією отримуємо, що для довільного програмного тексту pr та формули q, виконується $\vdash \{PC(pr,q)\} pr\{q\}$.

Якщо ми маємо істинну асерцію $\models \{p\} pr\{q\}$, то для довільної інтерпретації існує предикат q', такий, що $q'(d)\downarrow = T$, якщо існує $d' \in {}^V A$, такий що $p(d')\downarrow = T$ та $pr(d')\downarrow = d$; $q'(d)\downarrow = F$, якщо $q(d)\downarrow = F$ та $q'(d)\uparrow$ інакше. Так як розглядається екстенсіональна повнота, то формулою q' будемо позначати такий предикат. Так як $\models \{p\} pr\{q\}$, то такий предикат існує та визначається однозначно.

Також розглянемо формулу p', якій в кожній інтерпретації відповідає предикат p', що $p'(d)\downarrow = T$, якщо $p(d)\downarrow = T$ та $pr(d)\downarrow$; $p'(d)\downarrow = F$, якщо $p(d)\downarrow = F$ та невизначений інакше. Тоді за правилом *R_SKIP* $\vdash \{p\}id\{p\}$. За визначенням p' виконується $p \models_F p'$, тобто $FC(p, p')$, отже використовуючи правило *R_CONS'* $\vdash \{p\}id\{p'\}$. За визначенням q' виконується $q' \models_F q$, або $FC(q', q)$. Покажемо, що $p' \models_T PC(pr, q')$. Якщо на деякій інтерпретації $p'(d)\downarrow = T$, це означає $p(d)\downarrow = T$ та $pr(d)\downarrow$, тоді за визначенням q' виконується $q'(pr(d))\downarrow = T$, що доводить $p' \models_T PC(pr, q')$, або $TC(p', PC(pr, q'))$. За доведеним раніше $\vdash \{PC(pr, q')\}pr\{q'\}$, застосовуємо правило *R_CONS'* та отримуємо $\vdash \{p'\}pr\{q\}$. До $\vdash \{p\}id\{p'\}$ та $\vdash \{p'\}pr\{q\}$ можна застосувати правило *R_SEQ'* отримаємо $\vdash \{p\}id \bullet pr\{q\}$, скорочуючи за властивостями відповідних композицій отримуємо $\vdash \{p\}pr\{q\}$. Умова правила виконується так як $\models \{p\}pr\{q\}$,звідки легко вивести, що $p \models PC(pr, q)$, що еквівалентно $p \models PC(id \bullet pr, q)$.

3.3 Аксіоматичні системи з *T*-зростаючими та *F*-спадними асерціями

Хоча аксіоматична система *AC* коректна та екстенсіонально повна, обмеження додані для правил складні та надлишкові, тому виникає потреба в спрощенні додаткових обмежень. Зробити це допомагає той факт, що в аксіоматичній системі *CI* для всіх правил окрім *R_CONS* виконується наступне, якщо в засновках правил для всіх асерцій $\{p\}pr\{q\}$ мали для довільної інтерпретації J властивість $p_J^{T,pr} \subseteq q_J^T$, то і висновки правил зберігали дану властивість. Аналогічно зберігається також властивість $p \models_F PC(pr, q)$. Такі спостереження допомагають побудувати коректну аксіоматичну систему змінивши умову лише для правила *R_CONS*. Проте в такому разі очевидно, що системи не можуть бути повними, а лише повними на певному класі асерцій.

Побудована за таким принципом аксіоматична система для відношення логічного наслідку за істинною, назвемо її *T*-зростаючою, або *TI* буде мати наступний вигляд:

$$R\_AS \quad \frac{}{\{S_P^x(p,f)\}AS^x(f)\{p\}},$$

$$R\_SKIP \quad \frac{}{\{p\}id\{p\}},$$

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

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

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

$$R\_CONS'' \quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}},TC(p,p'),TC(q',q).$$

Дуальна аксіоматична система, побудована взявши за основу логічний наслідок за хибою, яку будемо називати F-спадною, або FD буде мати наступний вигляд:

$$R\_AS \quad \frac{}{\{S_P^x(p,f)\}AS^x(f)\{p\}},$$

$$R\_SKIP \quad \frac{}{\{p\}id\{p\}},$$

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

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

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

$$R\_CONS''' \quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}},FC(p,p'),FC(q',q).$$

Теорема 3.3. Якщо відношення TC співпадає з відношенням логічного наслідку за істиною, аксіоматична система TI коректна.

Доведення. Доведемо спочатку, що кожне правило зберігає для асерцій $\{p\}pr\{q\}$ властивість $p^{T,pr} \subseteq q^{T}$ яка виконується для довільної інтерпретації.

Розглянемо правило *R_AS*. Потрібно показати, що $S_P^x(p,f)^{T,AS^x(f)} \subseteq p^T$ для довільних інтерпретацій. Тоді, якщо для деякого $d \in {}^V A$, виконується $S_P^x(p,f)(d)\downarrow = T$. За визначенням композиції суперпозиції це еквівалентно $p(d\nabla[x \mapsto f(d)])\downarrow = T$. Це означає, що $f(d)\downarrow$, а отже також $AS^x(f)(d)\downarrow$. В такому випадку за визначенням композиції присвоєння маємо $p((AS^x(f))(d))\downarrow = p(d\nabla[x \mapsto f(d)]) = T$. Це означає, що якщо виконується $AS^x(f)(d) \in S_P^x(p,f)^{T,AS^x(f)}$, то і $AS^x(f)(d) \in\subseteq p^T$. Таким чином, доведено $S_P^x(p,f)^{T,AS^x(f)} \subseteq p^T$. свою чергу $PC(AS^x(f),p)\downarrow = T$ за визначенням виконується якщо $p((AS^x(f))(d))\downarrow = T$. Якщо розкрити за визначенням композиції присвоєння, маємо $p(d\nabla[x \mapsto f(d)])\downarrow = T$. Таким чином, якщо в довільній інтерпретації якщо для деякого $d \in {}^V A$ має місце $S_P^x(p,f)(d)\downarrow = T$, то для такого d також $PC(AS^x(f),p)\downarrow = T$, а отже $S_P^x(p,f) \models_T PC(AS^x(f),p)$.

Розглянемо правило *R_SKIP*. Потрібно показати, що виконується $p^{T,id} \subseteq p^T$. Зважаючи на визначення композиції тотожної програми та операції повного образу, можна сказати, що повним образом множини відносно композиції тотожної програми буде ця ж множина. Звідки очевидно випливає, що $p^{T,id} \subseteq p^T$ має місце для довільної інтерпретації.

Розглянемо правило *R_SEQ*. Потрібно довести, що якщо виконується $p^{T,pr_1} \subseteq q^T$ та $q^{T,pr_2} \subseteq r^T$, то виконується $p^{T,pr_1 \bullet pr_2} \subseteq r^T$. Розглянемо довільну інтерпретацію. Нехай для деякого $d'' \in {}^V A$ виконується $d'' \in p^{T,pr_1 \bullet pr_2}$, тоді існує такий $d \in {}^V A$, що $p(d)\downarrow = T$ та $pr_1 \bullet pr_2(d)\downarrow = d''$. Це в свою чергу означає, що $pr_1(d)\downarrow = d'$ разом з $pr_2(d')\downarrow = d''$ для деякого $d' \in {}^V A$. Тоді $d' \in p^{T,pr_1}$ та з $p^{T,pr_1} \subseteq q^T$ випливає, що $d' \in q^T$. Це означає, що $d'' \in q^{T,pr_2}$. Зважаючи на $q^{T,pr_2} \subseteq r^T$ отримуємо $d'' \in r^T$. Так як d'' було обрано довільним чином, то $p^{T,pr_1 \bullet pr_2} \subseteq r^T$.

Розглянемо правило *R_IF*. Потрібно довести, що якщо для довільної інтерпретації виконується $(r \wedge p)^{T,pr_1} \subseteq q^T$ та $(\neg r \wedge p)^{T,pr_2} \subseteq q^T$, то $p^{T,IF(r,pr_1,pr_2)} \subseteq q^T$. Візьмемо довільне $d \in {}^V A$, таке, що $d \in p^{T,IF(r,pr_1,pr_2)}$. За визначенням повного образу це означає, що існує таке $d' \in {}^V A$, що $p(d') \downarrow = T$ та $IF(r, pr_1, pr_2)(d') \downarrow = d$. За визначенням умовної композиції можливо два випадки для значення умови циклу на даному d': $r(d') \downarrow = T$ та $r(d') \downarrow = F$.

Нехай $r(d') \downarrow = T$. В такому випадку також має бути $pr_1(d') \downarrow$ та $IF(r, pr_1, pr_2)(d') = pr_1(d')$, звідки $pr_1(d') \downarrow = d$. Так як $p(d') \downarrow = T$, то $(r \wedge p)(d') = T$, а отже $d \in (r \wedge p)^{T,pr_1}$. З врахуванням того, що $(r \wedge p)^{T,pr_1} \subseteq q^T$ отримуємо $d \in q^T$. Таким чином $r(d) \downarrow = T$ виконується $p^{T,IF(r,pr_1,pr_2)} \subseteq q^T$.

Нехай виконується $r(d') \downarrow = F$. Тоді $pr_2(d') \downarrow$ та $IF(r, pr_1, pr_2)(d') = pr_2(d')$, звідки $pr_2(d') \downarrow = d$. З того, що $r(d') \downarrow = F$ маємо $(\neg r \wedge p)(d') = T$, звідки $d \in (\neg r \wedge p)^{T,pr_2}$. В такому випадку з $(\neg r \wedge p)^{T,pr_2} \subseteq q^T$ отримуємо $d \in q^T$. Доведено, що у випадку $r(d) \downarrow = F$ також виконується $p^{T,IF(r,pr_1,pr_2)} \subseteq q^T$.

Отже, для всіх випадків доведено що для довільної інтерпретації $p^{T,IF(r,pr_1,pr_2)} \subseteq q^T$.

Розглянемо правило *R_WH*. Потрібно довести, що якщо для довільної інтерпретації виконується $(r \wedge p)^{T,pr} \subseteq p^T$, то для довільної інтерпретації $p^{T,WH(r,pr)} \subseteq (\neg r \wedge p)^T$. Зафіксуємо довільну інтерпретацію для якої і будемо розглядати всі предикати. Візьмемо довільне $d \in {}^V A$, таке, що $d \in p^{T,WH(r,pr)}$. За визначенням повного образу це означає, що існує таке $d' \in {}^V A$, що $p(d') \downarrow = T$ та $WH(r, pr)(d') \downarrow = d$. В такому випадку з визначенням циклічної композиції існують такі $d_0, d_1, \ldots, d_n \in {}^V A$, що $d_0 = d'$ та $d_1 = pr(d_0)$, $d_2 = pr(d_1)$, …, $d_n = pr(d_{n-1})$ та $d_n = d$ разом з $r(d_0) \downarrow = T$, $r(d_1) \downarrow = T, \ldots, r(d_{n-1}) \downarrow = T$ та $r(d_n) \downarrow = F$. Тоді так як $(r \wedge p)(d_0) \downarrow = T$, то $d_1 \in (r \wedge p)^{T,pr}$, а отже з $(r \wedge p)^{T,pr} \subseteq p^T$ маємо $d_1 \in p^T$. Таким чином $p(d_1) \downarrow = T$. Аналогічно $(r \wedge p)(d_1) \downarrow = T$.

Застосовуючи такі міркування послідовно до всіх $d_0, d_1, \ldots, d_{n-1}$ отримуємо $p(d_n)\downarrow = T$, звідки $(\neg r \wedge p)(d_n)\downarrow = T$ і $d \in (\neg r \wedge p)^T$. Так як d було обрано довільним чином, то $p^{T,WH(r,pr)} \subseteq (\neg r \wedge p)^T$

Розглянемо правило *R_CONS''*. Для даного правила вважаємо, що виконується для довільної інтерпретації $p'^{T,pr} \subseteq q'^T$ та виконуються умови, тобто $TC(p,p')$ та $TC(q',q)$. В такому випадку потрібно довести, що $p^{T,pr} \subseteq q^T$ для довільної інтерпретації. Будемо вважати, що зафіксована деяка інтерпретація відносно якої і розглядаються формули. Так як відношення TC співпадає з відношенням логічного наслідку за істиною, то $p^T \subseteq p'^T$ та $q'^T \subseteq q^T$. З властивостей повного образу маємо $p^{T,pr} \subseteq p'^{T,pr}$. Об'єднуючи всі включення отримуємо $p^{T,pr} \subseteq p'^{T,pr} \subseteq q'^T \subseteq q^T$, а отже виконується $p^{T,pr} \subseteq q^T$.

Всі правила було розглянуто, а отже, було доведено, що якщо $\vdash \{p\}pr\{q\}$, то для довільної інтерпретації виконується $p^{T,pr} \subseteq q^T$. Покажемо, що в такому випадку $\models \{p\}pr\{q\}$. Нехай дане твердження не виконується, тоді існує інтерпретація J та дане $d \in {}^V\!A$ такі, що $FH(p,pr,q)_J(d)\downarrow = F$. Тоді $p_J(d)\downarrow = T$, та $q_J(pr_J(d))\downarrow = F$. Але в такому випадку $pr_J(d) \in p^{T,pr}$ та $pr_J(d) \in q^F$, а отже, $pr_J(d) \notin q^T$. Прийшли до протиріччя, звідки $\models \{p\}pr\{q\}$.

Теорема 3.4. Якщо відношення TC співпадає з відношенням логічного наслідку за істиною, аксіоматична система TI екстенсіонально повна на класі асерцій $\{p\}pr\{q\}$ для яких виконується $p \models_T PC(pr,q)$.

Доведення. Доведемо спочатку, що для довільного програмного тексту pr та формули q, виконується $\vdash \{PC(pr,q)\}pr\{q\}$. Доведемо це індукцією за структурою pr.

База індукції. Згідно індуктивного визначення множини програмних текстів до бази індукції належать id та $AS^x(f)$.

Розглянемо випадок $AS^x(f)$. За правилом *R_AS* отримуємо $|-\{S_P^x(q,f)\}AS^x(f)\{q\}$. Попередньо було доведено, що для довільної інтерпретації $PC(AS^x(f),q)=S_P^x(q,f)$, звідки $PC(AS^x(f),q)\models_T S_P^x(q,f)$. Що за припущенням те ж, що й $TC(PC(AS^x(f),q),S_P^x(q,f))$. Застосовуємо до $|-\{S_P^x(q,f)\}AS^x(f)\{q\}$ правило *R_CONS''* та отримуємо $|-\{PC(AS^x(f),q)\}AS^x(f)\{q\}$.

Розглянемо випадок id. Використовуючи правило *R_SKIP*, виводимо $|-\{q\}id\{q\}$. Аналогічно попередньому має місце $PC(id,q)=q$, що доводить $PC(id,q)\models_T q$. Так як TC за припущення це відношення логічного наслідку за істиною, то $TC(PC(id,q),id)$. Отже, можна застосувати правило *R_CONS''* до $|-\{q\}id\{q\}$, щоб вивести $|-\{PC(id,q)\}id\{q\}$.

Крок індукції. Згідно визначення основними конструкціями для побудови програмних текстів є послідовне виконання, галуження та цикл. Припускаємо, що для простіших програмних текстів з яких за допомогою відповідних засобів було отримано програмний текст, що розглядається, виконується твердження, що доводиться.

Розглянемо випадок послідовного виконання, $pr_1 \bullet pr_2$. Потрібно довести, що $|-\{PC(pr_1 \bullet pr_2,q)\}pr_1 \bullet pr_2\{q\}$. За припущенням індукції для pr_1 та pr_2 твердження має місце, а отже виконуються наступне: $|-\{PC(pr_2,q)\}pr_2\{q\}$ та $|-\{PC(pr_1,PC(pr_2,q))\}pr_1\{PC(pr_2,q)\}$. Застосувавши правило *R_SEQ'*, отримуємо $|-\{PC(pr_1,PC(pr_2,q))\}pr_1 \bullet pr_2\{q\}$. В такому випадку, якщо довести $TC(PC(pr_1 \bullet pr_2,q),PC(pr_1,PC(pr_2,q)))$, то можна використати правило *R_CONS''* для $|-\{PC(pr_1,PC(pr_2,q))\}pr_1 \bullet pr_2\{q\}$ та отримати $|-\{PC(pr_1 \bullet pr_2,q)\}pr_1 \bullet pr_2\{q\}$.

Покажемо, що потрібна умова виконується. Доведемо $TC(PC(pr_1 \bullet pr_2,q),PC(pr_1,PC(pr_2,q)))$. Так як TC співпадає з відношенням логічного наслідку за істинною, це означає, що потрібно довести $PC(pr_1 \bullet pr_2,q)\models_T PC(pr_1,PC(pr_2,q))$. Розглянемо довільну інтерпретацію, та покажемо, що для будь-якого $d \in {}^V A$, якщо $d \in PC(pr_1 \bullet pr_2, q)^T$, то $d \in PC(pr_1, PC(pr_2, q))^T$. Використовуючи визначення відповідних композицій, $d \in PC(pr_1 \bullet pr_2, q)^T$ означає $q(pr_1 \bullet pr_2(d)) \downarrow= T$, тобто $q(pr_2(pr_1(d))) \downarrow= T$. Аналогічно $d \in PC(pr_1, PC(pr_2, q))^T$ еквівалентно $(PC(pr_2, q))(pr_1(d)) \downarrow= T$ та $q(pr_2(pr_1(d))) \downarrow= T$. Отримали однакові вирази, а отже $d \in PC(pr_1 \bullet pr_2, q)^T \Leftrightarrow d \in PC(pr_1, PC(pr_2, q))^T$. Так як розглядалась довільна інтерпретація, то отримуємо $PC(pr_1 \bullet pr_2, q) \models_T PC(pr_1, PC(pr_2, q))$.

Обидві умови доведено, отже $|-\{PC(pr_1 \bullet pr_2, q)\} pr_1 \bullet pr_2 \{q\}$.

Розглянемо випадок галуження, $IF(r, pr_1, pr_2)$. Потрібно довести, що $|-\{PC(IF(r, pr_1, pr_2), q)\} IF(r, pr_1, pr_2)\{q\}$. За припущенням для pr_1 та pr_2 виконується $|-\{PC(pr_1, q)\} pr_1 \{q\}$ та $|-\{PC(pr_2, q)\} pr_2 \{q\}$. В такому випадку потрібно знайти спосіб вивести засновки правила *R_IF*, щоб застосувати його та отримати $|-\{PC(IF(r, pr_1, pr_2), q)\} IF(r, pr_1, pr_2)\{q\}$. Для цього потрібно довести, що $TC(r \wedge PC(IF(r, pr_1, pr_2), q), PC(pr_1, q))$ і можна застосувати правило *R_CONS''* до $|-\{PC(pr_1, q)\} pr_1 \{q\}$, щоб вивести перший засновок правила, $|-\{r \wedge PC(IF(r, pr_1, pr_2), q)\} pr_1 \{q\}$. Аналогічно для другого засновку, довівши $TC(\neg r \wedge PC(IF(r, pr_1, pr_2), q), PC(pr_2, q))$ можна також застосувати *R_CONS''* і отримати $|-\{\neg r \wedge PC(IF(r, pr_1, pr_2), q)\} pr_2 \{q\}$.

Доведемо, що потрібні вище умови мають місце. Так як TC співпадає з відношенням логічного наслідку за істинною, то перша умова еквівалентна $r \wedge PC(IF(r, pr_1, pr_2), q) \models_T PC(pr_1, q)$. Розглянемо довільну інтерпретацію та покажемо, що для довільного $d \in {}^V A$, якщо $(r \wedge PC(IF(r, pr_1, pr_2), q))(d) \downarrow= T$, то $(PC(pr_1, q))(d) \downarrow= T$. В такому випадку виконується $r(d) \downarrow= T$ та $(PC(IF(r, pr_1, pr_2), q))(d) \downarrow= T$. З визначення композиції побудови умови за прообразом отримуємо $q((IF(r, pr_1, pr_2))(d)) \downarrow= T$. Зважаючи на те, що $r(d) \downarrow= T$, з визначення умовної композиції $(IF(r, pr_1, pr_2))(d) \downarrow= pr_1(d)$, тому виконується

$q(pr_1(d))\downarrow=T$, звідки в свою чергу $(PC(pr_1,q))(d)\downarrow=T$, що і треба було довести. З довільного вибору інтерпретації отримуємо $r\wedge PC(IF(r,pr_1,pr_2),q)\models_T PC(pr_1,q)$.

За аналогією до попередньої умови потрібно довести $\neg r\wedge PC(IF(r,pr_1,pr_2),q)\models_T PC(pr_2,q)$. Для доведення даного твердження також для довільної інтерпретації покажемо включення областей істинності. Розглянемо довільне $d\in {}^V A$, та покажемо, що якщо $(\neg r\wedge PC(IF(r,pr_1,pr_2),q))(d)\downarrow=T$, то $(PC(pr_2,q))(d)\downarrow=T$. З визначення кон'юнкції отримуємо $r(d)\downarrow=F$ та $(PC(IF(r,pr_1,pr_2),q))(d)\downarrow=T$. З визначення умовної композиції при $r(d)\downarrow=F$ виконується $(IF(r,pr_1,pr_2))(d)\downarrow=pr_2(d)$. За визначеннями композицій тоді $(PC(IF(r,pr_1,pr_2),q))(d)\downarrow=T$ еквівалентно $q((IF(r,pr_1,pr_2))(d))\downarrow=T$, що в свою чергу еквівалентно $q(pr_2(d))\downarrow=T$. З отриманого твердження слідує $(PC(pr_2,q))(d)\downarrow=T$, що і треба було довести. Так як інтерпретація була обрана довільним чином, доведено $\neg r\wedge PC(IF(r,pr_1,pr_2),q)\models_T PC(pr_2,q)$.

Всі необхідні умови розглянуто, звідки виконується шукане твердження $\vdash\{PC(IF(r,pr_1,pr_2),q)\}IF(r,pr_1,pr_2)\{q\}$.

Розглянемо останній випадок, цикл, або $WH(r,pr)$. Доведемо, що $\vdash\{PC(WH(r,pr),q)\}WH(r,pr)\{q\}$. Для спрощення викладок позначимо $p=PC(WH(r,pr),q)$. За припущенням індукції для pr маємо $\vdash\{PC(pr,p)\}pr\{p\}$. Якщо довести, що $TC(r\wedge p,PC(pr,p))$, можна застосувати правило *R_CONS''* до $\vdash\{PC(pr,p)\}pr\{p\}$ та отримати $\vdash\{r\wedge p\}pr\{p\}$. Тоді, застосувавши правило *R_WH*, виведемо $\vdash\{p\}WH(r,pr)\{\neg r\wedge p\}$. Вивести бажане твердження можна за допомого правила *R_CONS''*, для цього потрібно довести, що виконується умова, тобто $TC(\neg r\wedge p,q)$. Тоді з $\vdash\{p\}WH(r,pr)\{\neg r\wedge p\}$ виводимо $\vdash\{p\}WH(r,pr)\{q\}$, або зважаючи на позначення $\vdash\{PC(WH(r,pr),q)\}WH(r,pr)\{q\}$.

Доведемо, що виконуються необхідні умови. Розглянемо спочатку $TC(r \wedge p, PC(pr, p))$. Так як TC співпадає з відношенням логічного наслідку за істинною, умова еквівалентна $r \wedge p \models_T PC(pr, p)$. Доведемо, що включення областей істинності виконується для будь-якої інтерпретації. Якщо для деякого $d \in {}^V A$ має місце $(r \wedge p)(d) \downarrow = T$, то $p(d) \downarrow = T$ та $r(d) \downarrow = T$. Отже $(PC(WH(r, pr), q))(d) \downarrow = T$. З визначення композиції $q((WH(r, pr))(d)) \downarrow = T$. Легко перевірити, що за визначенням циклічної композиції, у випадку, коли $r(d) \downarrow = T$ та $(WH(r, pr))(d) \downarrow$, то виконується $(WH(r, pr))(d) \downarrow = (WH(r, pr))(pr(d)) \downarrow$, отже $p(pr(d)) \downarrow = T$. Це еквівалентно $(PC(pr, p))(d) \downarrow = T$. З довільності вибору інтерпретації отримуємо $r \wedge p \models_T PC(pr, p)$.

Залишається лише довести, що виконується $TC(\neg r \wedge p, q)$. Так як TC співпадає з відношенням логічного наслідку за істинною, це означає, що потрібно довести $\neg r \wedge p \models_T q$. Візьмемо довільну інтерпретацію та покажемо включення областей істинності. Якщо для деякого $d \in {}^V A$ має місце $(\neg r \wedge p)(d) \downarrow = T$, то це означає, що $r(d) \downarrow = F$ та $p(d) \downarrow = T$. Так як $r(d) \downarrow = F$, то $WH(r, pr)(d) \downarrow = d$, тоді $q(d) \downarrow = p(d) = T$. Отже виконується включення областей істинності для довільної інтерпретації і $\neg r \wedge p \models_T q$.

Всі умови було доведено, отже $\vdash \{PC(WH(r, pr), q)\} WH(r, pr) \{q\}$.

Розглянуто всі можливі випадки і за індукцією отримуємо, що для довільного програмного тексту pr та формули q, виконується $\vdash \{PC(pr, q)\} pr \{q\}$.

Якщо ми маємо істинну асерцію, $\models \{p\} pr \{q\}$ для якої має місце $p \models_T PC(pr, q)$, що значить за припущенням, що $TC(p, PC(pr, q))$. Було доведено, що $\vdash \{PC(pr, q)\} pr \{q\}$, тоді можна застосувати правило *R_CONS''*, додаткові умови виконуються, а отже отримуємо $\vdash \{p\} pr \{q\}$.

Теорема 3.5. Якщо відношення FC співпадає з відношенням логічного наслідку за хибою, аксіоматична система FD коректна.

Доведення. Для того, щоб довести коректність аксіоматичної системи, доведемо, для кожного правила, що воно зберігає для асерцій $\{p\}pr\{q\}$ властивість $p \models_F PC(pr,q)$.

Розглянемо правило *R_AS*. Потрібно довести, що виконується $S_P^x(p,f) \models_F PC(AS^x(f),p)$. Зафіксуємо довільну інтерпретацію та доведемо включення областей хибності. Розглянемо деяке $d \in {}^V A$, таке, що $PC(AS^x(f),p)(d) \downarrow= F$. За визначенням композиції побудови умови за прообразом це означає, що $(AS^x(f))(d)\downarrow$ та $p(AS^x(f)(d)) \downarrow= F$. Звідки за визначенням композиції присвоєння отримуємо $p(d\nabla[x \mapsto f(d)]) \downarrow= F$. Разом з тим $S_P^x(p,f)(d) \downarrow= F$ з визначення композиції суперпозиції еквівалентно $p(d\nabla[x \mapsto f(d)]) \downarrow= F$. Таким чином якщо має місце $PC(AS^x(f),p)(d) \downarrow= F$, то також виконується і $S_P^x(p,f)(d) \downarrow= F$ для довільного $d \in {}^V A$. Звідки маємо включення областей хибності, враховуючи, що інтерпретація була обрана довільним чином, отримуємо $S_P^x(p,f) \models_F PC(AS^x(f),p)$.

Розглянемо правило *R_SKIP*. Для даного правила потрібно довести, що виконується $p \models_F PC(id,p)$. Враховуючи властивості композиції побудови умови за прообразом $PC(id,p)$; p для довільної інтерпретації. Узагальнена рівність гарантує включення відповідних областей хибності, отже виконується $p \models_F PC(id,p)$.

Розглянемо правило *R_SEQ*. Потрібно довести, що якщо виконується $p \models_F PC(pr_1,q)$ та $q \models_F PC(pr_2,r)$, то виконується $p \models_F PC(pr_1 \bullet pr_2,r)$. Розглянемо довільну інтерпретацію. Нехай для деякого $d \in {}^V A$ виконується $(PC(pr_1 \bullet pr_2,r))(d) \downarrow= F$, тоді з визначення композиції побудови умови за прообразом $r(pr_1 \bullet pr_2(d)) \downarrow= F$, а за визначенням композиції послідовного виконання $r(pr_2((pr_1(d))) \downarrow= F$. Згорнемо за визначенням композиції побудови умови за прообразом і отримаємо $(PC(pr_2,r))(pr_1(d)) \downarrow= F$.

Так як $q \models_F PC(pr_2, r)$, то $q(pr_1(d)) \downarrow = F$. Знову згортаючи за визначенням композиції побудови умови за прообразом $PC(pr_1, q)(d) \downarrow = F$, і за допомогою $p \models_F PC(pr_1, q)$ маємо $p(d) \downarrow = F$. Так як дані та інтерпретація були обрані довільним чином це доводить $p \models_F PC(pr_1 \bullet pr_2, r)$.

Розглянемо правило *R_IF*. Потрібно довести, що якщо виконується $(r \wedge p) \models_F PC(pr_1, q)$ та $(\neg r \wedge p) \models_F PC(pr_2, q)$, то $p \models_F PC(IF(r, pr_1, pr_2), q)$. Візьмемо довільну інтерпретацію. Розглянемо довільне $d \in {}^V A$, таке, що $PC(IF(r, pr_1, pr_2), q)(d) \downarrow = F$. За визначенням композиції побудови умови за прообразом це означає, що $IF(r, pr_1, pr_2)(d) \downarrow$ та $q(IF(r, pr_1, pr_2)(d)) \downarrow = F$. Так як $IF(r, pr_1, pr_2)(d) \downarrow$, можливо лише два випадки для предикату *r*: $r(d) \downarrow = T$ та $r(d) \downarrow = F$.

Якщо $r(d) \downarrow = T$, то $pr_1(d) \downarrow$ та $IF(r, pr_1, pr_2)(d) = pr_1(d)$. Звідки отримуємо $q(pr_1(d)) \downarrow = F$. Тому $PC(pr_1, q)(d) \downarrow = F$, що разом з $(r \wedge p) \models_F PC(pr_1, q)$ дає $(r \wedge p)(d) \downarrow = F$. Так як $r(d) \downarrow = T$, то лишається тільки випадок $p(d) \downarrow = F$. Отже, у випадку $r(d) \downarrow = T$ має місце $PC(IF(r, pr_1, pr_2), q)^F \subseteq p^F$ для обраної інтерпретації.

В іншому випадку, якщо $r(d) \downarrow = F$, то повинно виконуватись $pr_2(d) \downarrow$ та $IF(r, pr_1, pr_2)(d) = pr_2(d)$. Звідки $q(pr_2(d)) \downarrow = F$. Це еквівалентно $PC(pr_2, q)(d) \downarrow = F$, в такому випадку з $(\neg r \wedge p) \models_F PC(pr_2, q)$ отримуємо $(\neg r \wedge p)(d) \downarrow = F$. Тоді враховуючи $r(d) \downarrow = F$, маємо $p(d) \downarrow = F$. Таким чином для $r(d) \downarrow = F$ також виконується $PC(IF(r, pr_1, pr_2), q)^F \subseteq p^F$ для обраної інтерпретації.

Було розглянуто всі випадки, тому $p \models_F PC(IF(r, pr_1, pr_2), q)$.

Розглянемо правило *R_WH*. Потрібно довести, що якщо виконується $(r \wedge p) \models_F PC(pr, p)$, то $p \models_F PC(WH(r, pr), \neg r \wedge p)$. Візьмемо довільну інтерпретацію і покажемо, що для неї виконується включення відповідних областей хибності. Розглянемо довільне $d \in {}^V A$, для якого $PC(WH(r, pr), \neg r \wedge p)(d) \downarrow = F$. Тоді з визначення композиції побудови умови за прообразом $(\neg r \wedge p)((WH(r, pr))(d)) \downarrow = F$. Таким чином існує $d' \in {}^V A$ таке, що $(WH(r, pr))(d) \downarrow = d'$. В такому випадку з визначенням циклічної композиції існують такі $d_0, d_1, \ldots, d_n \in {}^V A$, що $d_0 = d$ та $d_1 = pr(d_0), d_2 = pr(d_1), \ldots,$ $d_n = pr(d_{n-1})$ та $d_n = d'$ разом з $r(d_0) \downarrow = T,$ $r(d_1) \downarrow = T, \ldots, r(d_{n-1}) \downarrow = T$ та $r(d_n) \downarrow = F$. Враховуючи, що $(\neg r \wedge p)(d') \downarrow = F$ та $r(d_n) \downarrow = F$ маємо $p(d_n) \downarrow = F$, тобто $r(pr(d_{n-1})) \downarrow = F$, або $(PC(pr, r))(d_{n-1}) \downarrow = F$. Тоді з $(r \wedge p) \models_F PC(pr, p)$ отримуємо $(r \wedge p)(d_{n-1}) \downarrow = F$. Але $r(d_{n-1}) \downarrow = T$ і тому $p(d_{n-1}) \downarrow = F$. Повторюючи такі міркування для всіх ітерацій циклу в зворотному напрямку маємо $p(d_0) \downarrow = F$, що рівносильне $p(d) \downarrow = F$. Довели включення областей хибності для довільної інтерпретації, звідки $p \models_F PC(WH(r, pr), \neg r \wedge p)$.

Розглянемо правило *R_CONS'''*. Для даного правила вважаємо, що виконується $p' \models_F PC(pr, q')$ разом з $FC(p, p')$ та $FC(q', q)$. Доведемо, що має місце $p \models_F PC(pr, q)$. Розглянемо довільну інтерпретацію. Відношення FC за умовою теореми співпадає з відношенням логічного наслідку за хибою, тому виконується $p'^F \subseteq p^F$ та $q^F \subseteq q'^F$. А $p' \models_F PC(pr, q')$ можна розписати як $PC(pr, q')^F \subseteq p'^F$. Зважаючи на визначення композиції побудови умови за прообразом з $q^F \subseteq q'^F$ отримуємо $PC(pr, q)^F \subseteq PC(pr, q')^F$. Звівши всі включення множин, маємо наступне: $PC(pr, q)^F \subseteq PC(pr, q')^F \subseteq p'^F \subseteq p^F$, звідки $PC(pr, q)^F \subseteq p^F$. Доведено що дане включення виконується для довільної інтерпретації, а отже $p \models_F PC(pr, q)$.

Для всіх правил було доведено потрібне твердження, звідки отримуємо, що якщо $\vdash \{p\} pr \{q\}$, то виконується $p \models_F PC(pr, q)$. Потрібно показати, що тоді $\models \{p\} pr \{q\}$. Нехай асерція не є істинною, отже існує інтерпретація J та дане $d \in {}^V A$ такі, що $FH(p, pr, q)_J(d) \downarrow = F$. Тоді $p_J(d) \downarrow = T$, та $q_J(pr_J(d)) \downarrow = F$.

Проте з доведеної властивості $p \models_F PC(pr,q)$ маємо $PC(pr,q)_J^F \subseteq p_J^F$. З $q_J(pr_J(d)) \downarrow = F$ виводимо, що $d \in PC(pr,q)_J^F$. В такому випадку також $d \in p_J^F$, тобто $p_J(d) \downarrow = F$. Прийшли до протиріччя, отже, $\models \{p\}pr\{q\}$.

Теорема 3.6. Якщо відношення FC співпадає з відношенням логічного наслідку за хибою, аксіоматична система FD не є екстенсіонально повною на класі асерцій $\{p\}pr\{q\}$ для яких виконується $p \models_F PC(pr,q)$.

Доведення. При доведенні коректності аксіоматичної системи було показано, що якщо $\vdash \{p\}pr\{q\}$, то $p \models_F PC(pr,q)$. Таким чином, розглядати питання повноти, включаючи асерції, для яких не виконується $p \models_F PC(pr,q)$ недоцільно.

Покажемо, що не для всіх зазначених асерцій можна побудувати вивід в аксіоматичній системі *FD.* Так як розглядається екстенсіональна повнота, вважаємо, що умови можуть задаватись довільними предикатами. Проблеми з виводом виникають, коли текст програми це цикл, або галуження, в яких умови задаються предикатами. Так як предикати можуть бути частковими, у випадку, коли умова невизначена, результат роботи програми також невизначений.

Нехай формули p, q та r такі що для довільної інтерпретації J, виконується $p_J(d)\uparrow$, $q_J(d) \downarrow = F$ та $r_J(d)\uparrow$ для довільного $d \in {}^V A$. Легко показати, що в такому випадку $p \models_F PC(IF(r,id,id),q)$. Так як r відповідає в довільній інтерпретації всюди невизначений предикат, то $IF(r,id,id)$ в свою чергу відповідає біквазіарна функція, що не визначена на жодному даному, а отже за визначенням композиції побудови умови за прообразом $PC(IF(r,id,id),q)$ також всюди невизначений предикат. Проте за означенням композиції Флойда-Хоара для таких p, q та r виконується $\models \{p\}IF(r,id,id)\{q\}$. Отже, якщо аксіоматична система екстенсіонально повна на класі асерцій $\{p\}pr\{q\}$ для яких виконується $p \models_F PC(pr,q)$, то $\vdash \{p\}IF(r,id,id)\{q\}$. Покажемо, що це неможливо. $\vdash \{p\}IF(r,id,id)\{q\}$ можна отримати використовуючи або правило *R_IF,* або правило *R_CONS'''.*

Якщо було використано правило *R_IF*, в такому випадку засновки правила мають виводитись в аксіоматичній системі, що означає $|-\{r \wedge p\}id\{q\}$ та $|-\{\neg r \wedge p\}id\{q\}$. В такому випадку для довільної інтерпретації $r \wedge p$ та $\neg r \wedge p$ відповідають всюди невизначені предикати, а $PC(id,q)$ предикат, що видає хибу на всіх даних. В такому випадку не виконується ні $r \wedge p |=_F PC(id,q)$, ні $\neg r \wedge p |=_F PC(id,q)$. В такому випадку за доведеними властивостями аксіоматичної системи не можливо, щоб виконувалось $|-\{r \wedge p\}id\{q\}$ та $|-\{\neg r \wedge p\}id\{q\}$, а отже, неможливо вивести $|-\{p\}IF(r,id,id)\{q\}$ використовуючи правило *R_IF*.

Якщо було використано правило *R_CONS'''*, в такому випадку для деяких p' та q' в аксіоматичній системі $|-\{p'\}IF(r,id,id)\{q'\}$ та виконуються додаткові умови, а саме $FC(p,p')$ та $FC(q',q)$, що означає $p |=_F p'$ та $q' |=_F q$. Якщо $q' |=_F q$, то для q', на довільній інтерпретації відповідний предикат на всіх даних видає хибу. В свою чергу $p |=_F p'$ означає, що область хибності предиката, що відповідає p' на довільній інтерпретації теж порожня множина. В такому випадку для $\{p'\}IF(r,id,id)\{q'\}$ мають місце всі твердження, що і для $\{p\}IF(r,id,id)\{q\}$ і застосовуючи відповідні міркування $\{p'\}IF(r,id,id)\{q'\}$ також неможливо вивести в аксіоматичній системі. Розглянули всі можливі випадки, отже хоча й $|=\{p\}IF(r,id,id)\{q\}$ та має місце $p |=_F PC(IF(r,id,id),q)$, але не виконується $|-\{p\}IF(r,id,id)\{q\}$.

3.4 Аксіоматичні системи з *T*- та *F*-обмеженнями

Розглянуті аксіоматичні системи *TI* та *FD* коректні, для них вводяться лише додаткові умови для правила *R_CONS*. Проте, не завжди для вирішення проблеми достатньо повноти аксіоматичної системи лише на класі трійок. Тому актуальним залишається питання побудови аксіоматичних систем, що мали б відносно прості додаткові умови для правил та були б екстенсіонально повні. Коректність аксіоматичної системи *AC* показала, що введення додаткових умов потребують лише правила *R_SEQ*, *R_WH,* та *R_CONS*. Застосовуючи ідеї, використані при побудові аксіоматичних систем *TI* та *FD* можна спростити додаткові умови в *AC*. Так як обмеження отриманої аксіоматичної системи залежать лише від відношень TC та FC, будемо називати її системо з *T*- та *F*-обмеженнями, або аксіоматичною системою TF. Правила мають наступний вигляд:

$$R\_AS \quad \frac{}{\{S_P^x(p,f)\}AS^x(f)\{p\}},$$

$$R\_SKIP \quad \frac{}{\{p\}id\{p\}},$$

$$R\_SEQ'' \quad \frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1 \bullet pr_2\{r\}}, TC(p, PC(pr_1, q)) \vee FC(q, PC(pr_2, r)),$$

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

$$R\_WH'' \quad \frac{\{r \wedge p\}pr\{p\}}{\{p\}WH(r, pr)\{\neg r \wedge p\}}, TC(r \wedge p, PC(pr, p)) \vee FC(r \wedge p, PC(pr, p)),$$

$$R\_CONS' \quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}}, TC(p, p'), FC(q', q).$$

Так як в додаткових умовах для правил R_SEQ'' та WH'' повинна виконуватись хоча б одна частина, розділимо для зручності кожне з даних правил на два правила, що будуть відрізнятись лише додатковою умовою.

$$R\_SEQ\_T \quad \frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1 \bullet pr_2\{r\}}, TC(p, PC(pr_1, q)),$$

$$R\_SEQ\_F \quad \frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1 \bullet pr_2\{r\}}, FC(q, PC(pr_2, r)),$$

$$R\_WH\_T \quad \frac{\{r \wedge p\}pr\{p\}}{\{p\}WH(r, pr)\{\neg r \wedge p\}}, TC(r \wedge p, PC(pr, p)),$$

$$R\_WH\_F \quad \frac{\{r \wedge p\}pr\{p\}}{\{p\}WH(r, pr)\{\neg r \wedge p\}}, FC(r \wedge p, PC(pr, p)).$$

Теорема 3.7. Якщо відношення *TC* співпадає з відношенням логічного наслідку за істиною та відношення *FC* співпадає з відношенням логічного наслідку за хибою, аксіоматична система *TF* коректна.

Доведення. Доведемо теорему індукцією за довжиною виводу.

База індукції. Якщо розглядати виводи за один крок, то можуть бути використані лише правила що не мають засновків, які фактично є аксіомами. Це правила *R_AS* та *R_SKIP*.

Розглянемо правило *R_AS*. Так як правило співпадає з відповідним правилом аксіоматичної системи *AC*, потрібно також довести, що $\models \{S_P^x(p,f)\}AS^x(f)\{p\}$. Повторюючи всі викладки доводимо, що в довільній інтерпретації не існує даного $d \in {}^V A$, для якого відповідна композиція Флойда-Хоара хибна, або $FH(S_P^x(p,f), AS^x(f), p)(d) \downarrow = F$. Звідки $S_P^x(p,f)(d) \downarrow = T$ та $p((AS^x(f))(d)) \downarrow = F$. Що рівносильне $p(d \nabla [x \mapsto f(d)]) \downarrow = T$ та $p(d \nabla [x \mapsto f(d)]) \downarrow = F$. Маємо протиріччя, отже для жодної інтерпретації не можна знайти такої номінативної множини, щоб асерція була хибною, звідки $\models \{S_P^x(p,f)\}AS^x(f)\{p\}$.

Розглянемо правило *R_SKIP*. Як і правило *R_AS* це правило співпадає з правилом аксіоматичної системи *AC*. Тому для доведення $\models \{p\}id\{p\}$ повторюємо всі кроки і покажемо, що в довільній інтерпретації не існує номінативної множини $d \in {}^V A$, такої, що $FH(p, id, p)(d) \downarrow = F$. Дане твердження виконується лише за умов $p(d) \downarrow = T$ та $p(id(d)) = p(d) \downarrow = F$. Прийшли до протиріччя, отже не можна знайти інтерпретацію на якій би асерція була хибною, таким чином $\models \{p\}id\{p\}$.

Крок індукції. Припустимо, що всі асерції, що виводяться менше ніж за k кроків істинні, доведемо, що і для k також виконується дане твердження. Для цього розглянемо різні випадки в залежності від того, яке правило було застосовано на останньому кроці.

Розглянемо правило *R_SEQ_T*. Маємо $\vdash \{p\}pr_1 \bullet pr_2\{r\}$, що було виведено з $\vdash \{p\}pr_1\{q\}$ та $\vdash \{q\}pr_2\{r\}$ за умови, що $TC(p, PC(pr_1, q))$. Так як $\vdash \{p\}pr_1\{q\}$ та

$\vdash \{q\} pr_2 \{r\}$ було отримано менше ніж за k кроків, то за припущенням індукції $\models \{p\} pr_1 \{q\}$ та $\models \{q\} pr_2 \{r\}$. Також з того, що відношення TC співпадає з відношенням логічного наслідку за істинною, отримуємо $p \models_T PC(pr_1, q)$. Потрібно довести, що $\models \{p\} pr_1 \bullet pr_2 \{r\}$. Покажемо, що для жодної інтерпретації не існує такого $d \in {}^V A$, що $(FH(p, pr_1 \bullet pr_2, r))(d) \downarrow= F$. Це можливо лише якщо $p(d) \downarrow= T$ та $r(pr_1 \bullet pr_2(d)) \downarrow= F$. Так як $p \models_T PC(pr_1, q)$, то за визначенням композиції побудови умови за прообразом та логічного наслідку за істиною виконується $PC(pr_1, q)(d) \downarrow= T$, звідки $q(pr_1(d)) \downarrow= T$. За визначенням композиції послідовного виконання $r(pr_1 \bullet pr_2(d)) \downarrow= F$ еквівалентно $r(pr_2(pr_1(d))) \downarrow= F$. Проте з умови $\models \{q\} pr_2 \{r\}$ не можливо, щоб одночасно для деякого $d' \in {}^V A$ мало місце $q(d') \downarrow= T$ та $r(pr_2(d')) \downarrow= F$. Прийшли до протиріччя, а отже $\models \{p\} pr_1 \bullet pr_2 \{r\}$.

Розглянемо правило *R_SEQ_F*. Аналогічно до правила *R_SEQ_T* з $\vdash \{p\} pr_1 \{q\}$ та $\vdash \{q\} pr_2 \{r\}$ але за умови, що $FC(q, PC(pr_2, r))$ виводиться $\vdash \{p\} pr_1 \bullet pr_2 \{r\}$. Так як $\vdash \{p\} pr_1 \{q\}$ та $\vdash \{q\} pr_2 \{r\}$ було отримано менше ніж за k кроків, то за припущенням індукції $\models \{p\} pr_1 \{q\}$ та $\models \{q\} pr_2 \{r\}$. Також з того, що відношення FC співпадає з відношенням логічного наслідку за хибою, отримуємо $q \models_F PC(pr_2, r)$. Доведемо, що $\models \{p\} pr_1 \bullet pr_2 \{r\}$. Для цього потрібно показати, що не існує інтерпретації та такого $d \in {}^V A$, що $(FH(p, pr_1 \bullet pr_2, r))(d) \downarrow= F$. Нехай це не так, тоді для такої інтерпретації та номінативної множини мають місце $p(d) \downarrow= T$ та $r(pr_1 \bullet pr_2(d)) \downarrow= F$. За визначенням композиції послідовного виконання $r(pr_1 \bullet pr_2(d)) \downarrow= F$ еквівалентно $r(pr_2(pr_1(d))) \downarrow= F$. Звідки $pr_1(d) \in PC(pr_2, r)^F$, враховуючи, що $q \models_F PC(pr_2, r)$ виводимо $q(pr_1(d)) \downarrow= F$. Проте $p(d) \downarrow= T$ та $q(pr_1(d)) \downarrow= F$ суперечить з тим, що $\models \{q\} pr_2 \{r\}$. Прийшли до протиріччя, а отже $\models \{p\} pr_1 \bullet pr_2 \{r\}$.

Розглянемо правило *R_IF*. Дане правило співпадає з відповідним правилом аксіоматичної системи *AC*. За правилом з $|-\{r \wedge p\}pr_1\{q\}$ та $|-\{\neg r \wedge p\}pr_2\{q\}$ виводиться $|-\{p\}IF(r, pr_1, pr_2)\{q\}$. Так як $|-\{r \wedge p\}pr_1\{q\}$ та $|-\{\neg r \wedge p\}pr_2\{q\}$ виводяться менше ніж за k кроків, то за припущенням індукції виконується $|=\{r \wedge p\}pr_1\{q\}$ та $|=\{\neg r \wedge p\}pr_2\{q\}$. Для того, щоб довести $|=\{p\}IF(r, pr_1, pr_2)\{q\}$, покажемо, що для кожної інтерпретації виконується $FH(p, IF(r, pr_1, pr_2), q)^F = \varnothing$.

Розглянемо довільну інтерпретацію і покажемо, що для будь-якого $d \in {}^V A$, такого, що $p(d)\downarrow = T$, не виконується $FH(p, IF(r, pr_1, pr_2), q)(d)\downarrow = F$. Для того, щоб розгорнути умовну композицію, розглянемо різні випадки для значень $r(d)$.

Якщо $r(d)\downarrow = T$, то такому випадку $IF(r, pr_1, pr_2)(d)$; $pr_1(d)$. Тоді, якщо $pr_1(d)\uparrow$, то і $IF(r, pr_1, pr_2)(d)\uparrow$, а отже $q(IF(r, pr_1, pr_2)(d))\uparrow$ і значить $FH(p, IF(r, pr_1, pr_2), q)(d)\uparrow$. Інакше, якщо $pr_1(d)\downarrow$, то виконується $IF(r, pr_1, pr_2)(d) = pr_1(d)$, а отже $q(pr_1(d))\downarrow = F \Leftrightarrow q(IF(r, pr_1, pr_2)(d))\downarrow = F$. Зважаючи на те, що $|=\{r \wedge p\}pr_1\{q\}$, а отже $FH(r \wedge p, pr_1, q)^F = \varnothing$, а також $(r \wedge p)(d)\downarrow = T$ то не вірно, що $q(pr_1(d))\downarrow = F$. Звідки не виконується $q(IF(r, pr_1, pr_2)(d))\downarrow = F$, отже і не вірно, що $FH(p, IF(r, pr_1, pr_2), q)(d)\downarrow = F$. Таким чином при $r(d)\downarrow = T$ не виконується $FH(p, IF(r, pr_1, pr_2), q)(d)\downarrow = F$.

Якщо ж $r(d)\downarrow = F$. Тоді $IF(r, pr_1, pr_2)(d)$; $pr_2(d)$. В такому випадку якщо $pr_2(d)\uparrow$, то і $IF(r, pr_1, pr_2)(d)\uparrow$, звідки отримуємо $q(IF(r, pr_1, pr_2)(d))\uparrow$ що рівносильне $FH(p, IF(r, pr_1, pr_2), q)(d)\uparrow$. Якщо ж має місце $pr_2(d)\downarrow$, то з узагальненої рівності маємо $IF(r, pr_1, pr_2)(d) = pr_2(d)$, а отже як і в попередньому випадку $q(pr_2(d))\downarrow = F \Leftrightarrow q(IF(r, pr_1, pr_2)(d))\downarrow = F$. Так як за припущенням $|=\{\neg r \wedge p\}pr_2\{q\}$, тобто для інтерпретації, що розглядається, $FH(\neg r \wedge p, pr_2, q)^F = \varnothing$. Разом з $(\neg r \wedge p)(d)\downarrow = T$ отримуємо, що неможливо

$q(pr_2(d))\downarrow= F$. Отже неможливо і $q(IF(r, pr_1, pr_2)(d))\downarrow= F$. Доведено, що у випадку $r(d)\downarrow= F$ також не виконується $FH(p, IF(r, pr_1, pr_2), q)(d)\downarrow= F$.

Розглянемо останній випадок, коли $r(d)\uparrow$. В такому випадку $IF(r, pr_1, pr_2)(d)\uparrow$, а отже, $q(IF(r, pr_1, pr_2)(d))\uparrow$ і $FH(p, IF(r, pr_1, pr_2), q)(d)\uparrow$.

Розглянуто всі випадки, доведено, що не можливо $FH(p, IF(r, pr_1, pr_2), q)(d)\downarrow= F$, звідки $\models \{p\}IF(r, pr_1, pr_2)\{q\}$.

Розглянемо правило *R_WH_T*. З $\vdash \{r \wedge p\}pr\{p\}$ виводимо відповідно до правила $\vdash \{p\}WH(r, pr)\{\neg r \wedge p\}$ за умови, що $TC(r \wedge p, PC(pr, p))$. Зважаючи на те, що $\vdash \{r \wedge p\}pr\{p\}$ було виведено менше ніж за k кроків, то за припущенням індукції $\models \{r \wedge p\}pr\{p\}$. Разом з тим, враховуючи, що відношення TC співпадає з відношенням логічного наслідку за істиною, отримуємо $r \wedge p \models_T PC(pr, p)$. Потрібно довести $\models \{p\}WH(r, pr)\{\neg r \wedge p\}$, для цього покажемо, що для довільної інтерпретації не існує $d \in {}^V A$, такого, що $FH(p, WH(r, pr), \neg r \wedge p)(d)\downarrow= F$. Нехай це не так і таке номінативне дане існує. Це можливо лише якщо $p(d)\downarrow= T$ та $(\neg r \wedge p)(WH(r, pr)(d))\downarrow= F$. Таким чином існує $d' \in {}^V A$ таке, що $(WH(r, pr))(d)\downarrow= d'$. В такому випадку з визначенням циклічної композиції існують такі $d_0, d_1, \ldots, d_n \in {}^V A$, що $d_0 = d$ та $d_1 = pr(d_0)$, $d_2 = pr(d_1)$, …, $d_n = pr(d_{n-1})$ та $d_n = d'$ разом з $r(d_0)\downarrow= T$, $r(d_1)\downarrow= T, \ldots, r(d_{n-1})\downarrow= T$ та $r(d_n)\downarrow= F$. Враховуючи, що $p(d)\downarrow= T$ та $r(d_0)\downarrow= T$, то з $r \wedge p \models_T PC(pr, p)$ маємо $PC(pr, p)(d_0)\downarrow= T$, тобто $p(pr(d_0))\downarrow= T$ або $p(d_1)\downarrow= T$. Повторюючи послідовно дані міркування, маємо $p(d_n)\downarrow= T$ разом з $r(d_n)\downarrow= F$ маємо $(\neg r \wedge p)(d_n)\downarrow= T$, що рівносильне $(\neg r \wedge p)(WH(r, pr)(d))\downarrow= T$. Прийшли до протиріччя, отже $\models \{p\}WH(r, pr)\{\neg r \wedge p\}$.

Розглянемо правило *R_WH_F*, що відрізняється від попереднього лише умовою. З $\vdash \{r \wedge p\}pr\{p\}$ виводимо відповідно до правила $\vdash \{p\}WH(r, pr)\{\neg r \wedge p\}$ за умови, що $FC(r \wedge p, PC(pr, p))$. Зважаючи на те, що

$\vdash \{r \wedge p\} pr \{p\}$ було виведено менше ніж за k кроків, то за припущенням індукції $\models \{r \wedge p\} pr \{p\}$. Разом з тим, враховуючи, що відношення FC співпадає з відношенням логічного наслідку за істиною, отримуємо $r \wedge p \models_F PC(pr, p)$. Доведемо $\models \{p\} WH(r, pr) \{\neg r \wedge p\}$, показавши, що для довільної інтерпретації не існує номінативного даного $d \in {}^V A$, такого, що $FH(p, WH(r, pr), \neg r \wedge p)(d) \downarrow = F$. Нехай таке твердження не виконується і відповідне номінативне дане існує. Це можливо лише якщо $p(d) \downarrow = T$ та $(\neg r \wedge p)(WH(r, pr)(d)) \downarrow = F$. Таким чином результат обчислення циклу визначений та існує $d' \in {}^V A$ таке, що $(WH(r, pr))(d) \downarrow = d'$. В такому випадку з визначенням циклічної композиції існують такі $d_0, d_1, \ldots, d_n \in {}^V A$, що $d_0 = d$ та $d_1 = pr(d_0), d_2 = pr(d_1), \ldots$, $d_n = pr(d_{n-1})$ та $d_n = d'$ разом з $r(d_0) \downarrow = T$, $r(d_1) \downarrow = T, \ldots, r(d_{n-1}) \downarrow = T$ та $r(d_n) \downarrow = F$. Враховуючи, що $(\neg r \wedge p)(d') \downarrow = F$ та $r(d_n) \downarrow = F$ маємо $p(d_n) \downarrow = F$, тобто $r(pr(d_{n-1})) \downarrow = F$, або $(PC(pr, r))(d_{n-1}) \downarrow = F$. Тоді з $(r \wedge p) \models_F PC(pr, p)$ отримуємо $(r \wedge p)(d_{n-1}) \downarrow = F$. Але $r(d_{n-1}) \downarrow = T$ і тому $p(d_{n-1}) \downarrow = F$. Продовжуючи дані викладки, приходимо до $p(d_0) \downarrow = F$, тобто $p(d) \downarrow = F$. Маємо протиріччя, отже припущення не виконується і $\models \{p\} WH(r, pr) \{\neg r \wedge p\}$.

Розглянемо правило *R_CONS'*. Як і у випадку з аксіоматичною системою *AC* з $\vdash \{p'\} pr \{q'\}$ для яких виконуються умови, $TC(p, p')$ та $FC(q', q)$, виводиться $\vdash \{p\} pr \{q\}$. Так як $\vdash \{p'\} pr \{q'\}$ виводиться менше ніж за k кроків, за припущенням індукції маємо $\models \{p'\} pr \{q'\}$. Так як відношення TC співпадає з відношенням логічного наслідку за істиною, то для довільної інтерпретації $p^T \subseteq p'^T$. Аналогічно з того, що відношення FC співпадає з відношенням логічного наслідку за хибою, маємо для довільної інтерпретації $q'^F \subseteq q^F$. Щоб довести $\models \{p\} pr \{q\}$, для довільної інтерпретації покажемо, що $FH(p, pr, q)^F = \varnothing$. Так як $\models \{p'\} pr \{q'\}$, то для відповідної інтерпретації $FH(p', pr, q')^F = \varnothing$, що рівносильне $p'^T \cap q'^{-F, pr} = \varnothing$. Звідки за монотонністю операції повного прообразу, отримуємо $q'^{-F,pr} \subseteq q^{-F,pr}$. Об'єднуємо всі твердження та з монотонності операції перетину двох множин, $p^T \cap q^{-F,pr} \subseteq p'^T \cap q'^{-F,pr} = \varnothing$, отже $FH(p,pr,q)^F = p^T \cap q^{-F,pr} = \varnothing$. З того, що розглядалась довільна інтерпретація слідує, що $\models \{p\}pr\{q\}$.

Всі правила було розглянуто, таким чином, за індукцією аксіоматична система коректна.

Теорема 3.8. Якщо відношення TC співпадає з відношенням логічного наслідку за істиною та відношення FC співпадає з відношенням логічного наслідку за хибою, аксіоматична система TF екстенсіонально повна.

Доведення. Для того, щоб довести повноту аксіоматичної системи, доведемо, що виконується $\vdash \{PC(pr,q)\}pr\{q\}$ для довільного програмного тексту pr та формули q. Так як правила відповідають конструкціям мови програмування, то доводити будемо індукцією за структурою pr.

База індукції. До програмних тексти, що не утворені з простіших включають в себе лише id та $AS^x(f)$.

Розглянемо випадок присвоєння $AS^x(f)$. В такому випадку за допомогою правила R_AS можна вивести $\vdash \{S_P^x(q,f)\}AS^x(f)\{q\}$. Якщо показати, що $TC(PC(AS^x(f),q),S_P^x(q,f))$, то застосувавши правило R_CONS' з $\vdash \{S_P^x(q,f)\}AS^x(f)\{q\}$ отримуємо $\vdash \{PC(AS^x(f),q)\}AS^x(f)\{q\}$. Зважаючи на те, що відношення TC співпадає з відношенням логічного наслідку за істиною, умова еквівалентна $PC(AS^x(f),q) \models_T S_P^x(q,f)$. За попередньо доведеними властивостями $PC(AS^x(f),q) = S_P^x(q,f)$, що і доводить потрібну умову.

Розглянемо випадок тотожної програми, id. Аналогічно до попереднього випадку застосовуючи правило R_SKIP можна отримати $\vdash \{q\}id\{q\}$. Якщо виконується $TC(PC(id,q),id)$, то можна застосувати правило R_CONS' до $\vdash \{q\}id\{q\}$, та отримати $\vdash \{PC(id,q)\}id\{q\}$. Так як відношення TC співпадає з відношенням логічного наслідку за істиною, то умову можна записати, як

$PC(id,q) \models_T q$. В свою чергу для довільної інтерпретації $PC(id,q) = q$, отже, умова виконується.

Крок індукції. Складні програмні тексти можуть бути утворені з простіших лише за допомогою послідовного виконання, розгалуження та циклу. Розглянемо окремо дані випадки.

Розглянемо випадок послідовного виконання, нехай програмний текст має вигляд $pr_1 \bullet pr_2$. Тоді потрібно довести, що $|-\{PC(pr_1 \bullet pr_2, q)\} pr_1 \bullet pr_2 \{q\}$. Використовуючи припущення індукції маємо, що для pr_1 та pr_2 твердження виконується, отже $|-\{PC(pr_2,q)\} pr_2 \{q\}$ та $|-\{PC(pr_1, PC(pr_2,q))\} pr_1 \{PC(pr_2,q)\}$. В такому випадку для того, щоб використати правило *R_SEQ_F* треба довести, що $FC(PC(pr_2,q), PC(pr_2,q))$. Застосувавши відповідне правило, отримаємо в свою чергу $|-\{PC(pr_1, PC(pr_2,q))\} pr_1 \bullet pr_2 \{q\}$. Лишається тільки показати, що має місце умова $TC(PC(pr_1 \bullet pr_2, q), PC(pr_1, PC(pr_2,q)))$ і можна використати правило *R_CONS'* до твердження $|-\{PC(pr_1, PC(pr_2,q))\} pr_1 \bullet pr_2 \{q\}$ та вивести $|-\{PC(pr_1 \bullet pr_2, q)\} pr_1 \bullet pr_2 \{q\}$. Залишається лише показати, що відповідні умови виконуються.

Доведемо $FC(PC(pr_2,q), PC(pr_2,q))$. Так як FC співпадає з відношенням логічного наслідку за хибою, то потрібно довести $PC(pr_2,q) \models_F PC(pr_2,q)$, що виконується з визначення логічного наслідку за хибою

Доведемо умову, що залишилась, $TC(PC(pr_1 \bullet pr_2, q), PC(pr_1, PC(pr_2,q)))$. Так як TC співпадає з відношенням логічного наслідку за істинною, то потрібно довести $PC(pr_1 \bullet pr_2, q) \models_T PC(pr_1, PC(pr_2,q))$. Покажемо, що для довільної інтерпретації виконується включення відповідних множин. Нехай для деякого $d \in {}^V A$ має місце $d \in PC(pr_1 \bullet pr_2, q)^T$. Використовуючи визначення відповідних композицій, $d \in PC(pr_1 \bullet pr_2, q)^T$ означає $q(pr_1 \bullet pr_2(d)) \downarrow = T$, тобто $q(pr_2(pr_1(d))) \downarrow = T$. Це еквівалентно за визначенням $(PC(pr_2,q))(pr_1(d)) \downarrow = T$, звідки $(PC(pr_1, PC(pr_2,q)))(d) \downarrow = T$. Отже $d \in PC(pr_1, PC(pr_2,q))^T$, таким чином включення областей істинності доведено, а з довільності вибору інтерпретації маємо $PC(pr_1 \bullet pr_2, q) \models_T PC(pr_1, PC(pr_2, q))$.

Всі умови доведено, тому $|-\{PC(pr_1 \bullet pr_2, q)\} pr_1 \bullet pr_2 \{q\}$.

Розглянемо випадок галуження, $IF(r, pr_1, pr_2)$. В такому випадку вважаючи, що для pr_1 та pr_2 твердження виконується, потрібно довести, що $|-\{PC(IF(r, pr_1, pr_2), q)\} IF(r, pr_1, pr_2)\{q\}$. В такому разі з припущення індукції маємо, що $|-\{PC(pr_1, q)\} pr_1\{q\}$ та $|-\{PC(pr_2, q)\} pr_2\{q\}$. Для того, щоб застосувати правило *R_IF*, з даних тверджень потрібно за допомогою правила *R_CONS'* вивести засновки для правила у відповідному вигляді. Для того, щоб застосувати правило *R_CONS'* до $|-\{PC(pr_1, q)\} pr_1\{q\}$ та отримати $|-\{r \wedge PC(IF(r, pr_1, pr_2), q)\} pr_1\{q\}$ потрібно показати, що виконується умова, тобто $TC(r \wedge PC(IF(r, pr_1, pr_2), q), PC(pr_1, q))$. Аналогічно, щоб з $|-\{PC(pr_2, q)\} pr_2\{q\}$ отримати, треба довести додаткову умову у вигляді $TC(\neg r \wedge PC(IF(r, pr_1, pr_2), q), PC(pr_2, q))$. Тоді можна застосувати правило *R_IF* та вивести $|-\{PC(IF(r, pr_1, pr_2), q)\} IF(r, pr_1, pr_2)\{q\}$.

Доведемо спочатку $TC(r \wedge PC(IF(r, pr_1, pr_2), q), PC(pr_1, q))$. Так як *TC* співпадає з відношенням логічного наслідку за істинною, потрібно показати $r \wedge PC(IF(r, pr_1, pr_2), q) \models_T PC(pr_1, q)$. Для цього на довільній інтерпретації покажемо включення областей істинності візьмемо довільне $d \in {}^V A$, та доведемо, що якщо $(r \wedge PC(IF(r, pr_1, pr_2), q))(d) \downarrow= T$, то $(PC(pr_1, q))(d) \downarrow= T$. Отже маємо з припущення $r(d) \downarrow= T$ та $(PC(IF(r, pr_1, pr_2), q))(d) \downarrow= T$. З визначення композиції побудови умови за прообразом отримуємо $q((IF(r, pr_1, pr_2))(d)) \downarrow= T$. Так як $r(d) \downarrow= T$, то для умовної композиції має місце $(IF(r, pr_1, pr_2))(d) \downarrow= pr_1(d)$, тому виконується $q(pr_1(d)) \downarrow= T$, звідки в свою чергу $(PC(pr_1, q))(d) \downarrow= T$, що і треба було довести. Отже доведено включення і так як розглядалась довільна інтерпретація, доведено $r \wedge PC(IF(r, pr_1, pr_2), q) \models_T PC(pr_1, q)$.

Розглянемо $TC(\neg r \wedge PC(IF(r, pr_1, pr_2), q), PC(pr_2, q))$. Виконання даної умови доведемо за аналогією до попередньої. Так як TC співпадає з відношенням логічного наслідку за істинною, умова еквівалентна $\neg r \wedge PC(IF(r, pr_1, pr_2), q) \models_T PC(pr_2, q)$. Для доведення даного твердження візьмемо довільну інтерпретацію та покажемо включення областей істинності. Покажемо, що для довільного $d \in {}^V A$, якщо $(\neg r \wedge PC(IF(r, pr_1, pr_2), q))(d) \downarrow= T$, то $(PC(pr_2, q))(d) \downarrow= T$. З визначення кон'юнкції отримуємо $r(d) \downarrow= F$ та $(PC(IF(r, pr_1, pr_2), q))(d) \downarrow= T$. З визначення умовної композиції при $r(d) \downarrow= F$ виконується $(IF(r, pr_1, pr_2))(d) \downarrow= pr_2(d)$. За визначеннями композицій тоді $(PC(IF(r, pr_1, pr_2), q))(d) \downarrow= T$ еквівалентно $q((IF(r, pr_1, pr_2))(d)) \downarrow= T$, що в свою чергу еквівалентно $q(pr_2(d)) \downarrow= T$. З отриманого твердження слідує $(PC(pr_2, q))(d) \downarrow= T$, що і треба було довести. Так як інтерпретація була обрана довільним чином, доведено $\neg r \wedge PC(IF(r, pr_1, pr_2), q) \models_T PC(pr_2, q)$.

Всі необхідні умови розглянуто, отже правила можна застосувати і має місце $\vdash \{PC(IF(r, pr_1, pr_2), q)\} IF(r, pr_1, pr_2) \{q\}$.

Розглянемо випадок циклу, $WH(r, pr)$. В такому разі, потрібно довести, що $\vdash \{PC(WH(r, pr), q)\} WH(r, pr) \{q\}$, якщо для pr виконується твердження. Введемо позначення $p = PC(WH(r, pr), q)$. Так як розглядається екстенсіональна повнота, вважатимемо, що існує формула p' така, що для довільної інтерпретації для довільного $d \in {}^V A$ якщо $p(d) \downarrow= T$, то $p'(d) \downarrow= T$, інакше $p'(d) \downarrow= F$. Таку формулу і будемо розглядати разом з pr та за припущенням індукції $\vdash \{PC(pr, p')\} pr \{p'\}$. Для того, щоб використати правило *R_WH_T* або *R_WH_F* потрібно дану асерцію привести до відповідного вигляду, це можна зробити, якщо виконується додаткова умова, $TC(r \wedge p', PC(pr, p'))$ і, застосувавши правило *R_CONS'* до $\vdash \{PC(pr, p')\} pr \{p'\}$, отримати $\vdash \{r \wedge p'\} pr \{p'\}$. До даного твердження можна застосувати одне з правил для циклу, оберемо правило *R_WH_T*. Додаткова умова в такому випадку має вигляд $TC(r \wedge p', PC(pr, p'))$.

Тоді за правилом *R_WH_T* виводиться $\vdash \{p'\}WH(r, pr)\{\neg r \wedge p'\}$. Шукане $\vdash \{PC(WH(r, pr), q)\}WH(r, pr)\{q\}$, або ж відповідно $\vdash \{p\}WH(r, pr)\{q\}$ можна вивести використовуючи правило *R_CONS'* з засновком $\vdash \{p'\}WH(r, pr)\{\neg r \wedge p'\}$. Додаткові умови в такому випадку мають вигляд $FC(\neg r \wedge p', q)$ та $TC(p, p')$.

Доведемо, що виконуються необхідні умови. Умова $TC(p, p')$ найпростіша, та виконується, бо з визначення p' для кожної інтерпретації області істинності предикатів що відповідають p та p' співпадають.

Умова для правил *R_WH_T* та *R_CONS'* співпадає, розглянемо її. Потрібно довести, що має місце $TC(r \wedge p', PC(pr, p'))$. Так як TC співпадає з відношенням логічного наслідку за істинною, умова еквівалентна $r \wedge p' \models_T PC(pr, p')$. Зафіксуємо довільну інтерпретацію та доведемо включення відповідних областей істинності. Потрібно показати, що якщо для деякого $d \in {}^V A$ має місце $(r \wedge p')(d) \downarrow = T$, то $(PC(pr, p'))(d) \downarrow = T$. З істинності кон'юнкції $p'(d) \downarrow = T$ та $r(d) \downarrow = T$. Якщо $p'(d) \downarrow = T$, то $p(d) \downarrow = T$, що в свою чергу позначає $(PC(WH(r, pr), q))(d) \downarrow = T$. Розгорнемо за визначенням композиції побудови умови за прообразом $q((WH(r, pr))(d)) \downarrow = T$. Також за визначенням циклічної композиції, якщо $r(d) \downarrow = T$, то тіло циклу буде виконано принаймні один раз, тобто $(WH(r, pr))(d) \downarrow = (WH(r, pr))(pr(d)) \downarrow$. Отже значення циклу рівні на d та $pr(d)$, тому враховуючи $p(d) \downarrow = T$, виконується $p(pr(d)) \downarrow = T$, звідки $p'(pr(d)) \downarrow = T$. Це еквівалентно $(PC(pr, p'))(d) \downarrow = T$. Так як інтерпретація обрана довільним чином, то має місце $r \wedge p' \models_T PC(pr, p')$.

Залишається лише умова $FC(\neg r \wedge p', q)$. Так як FC співпадає з відношенням логічного наслідку за хибою, умова еквівалентна $\neg r \wedge p' \models_F q$. Потрібно для довільної інтерпретації та показати включення областей хибності заданих предикатів. Візьмемо довільне $d \in {}^V A$ таке, що $q(d) \downarrow = F$ та доведемо, що в такому випадку $(\neg r \wedge p')(d) \downarrow = F$. Розглянемо різні випадки для *r*. Якщо

$r(d)\downarrow = T$, то $(\neg r \wedge p')(d)\downarrow = F$ за визначенням кон'юнкції. Якщо $r(d)\downarrow = F$, то для циклу $WH(r,pr)(d)\downarrow = d$, і, зважаючи на те, що $p = PC(WH(r,pr),q)$, маємо $p(d)\downarrow = q(d) = F$. В такому разі $p'(d)\downarrow = F$, звідки $(\neg r \wedge p')(d)\downarrow = F$. Якщо $r(d)\uparrow$, то $WH(r,pr)(d)\uparrow$, а отже $p(d)\uparrow$,а $p'(d)\downarrow = F$ і також $(\neg r \wedge p')(d)\downarrow = F$. Розглянуто всі випадки, отже включення областей хибності виконується для довільної інтерпретації, що означає $\neg r \wedge p' \models_F q$.

Всі потрібні умови доведено, звідки $\vdash \{PC(WH(r,pr),q)\}WH(r,pr)\{q\}$.

Розглянуто всі можливі випадки і за індукцією отримуємо, що для довільного програмного тексту pr та формули q, виконується $\vdash \{PC(pr,q)\}pr\{q\}$.

Залишається довести, що якщо $\models \{p\}pr\{q\}$, то $\vdash \{p\}pr\{q\}$. Так як розглядається екстенсіональна повнота, то формулою q' будемо позначати такий предикат, що для кожної інтерпретації $q'(d)\downarrow = T$, якщо існує $d' \in {}^{V}A$, такий що $p(d')\downarrow = T$ та $pr(d')\downarrow = d$; $q'(d)\downarrow = F$, якщо $q(d)\downarrow = F$ та $q'(d)\uparrow$ інакше. Так як $\models \{p\}pr\{q\}$, то це такий предикат існує та визначено однозначно. Також розглянемо формулу p', якій в кожній інтерпретації відповідає предикат p', що $p'(d)\downarrow = T$, якщо $p(d)\downarrow = T$ та $p'(d)\downarrow = F$ інакше. Разом з ним розглянемо формулу p'', якій в кожній інтерпретації відповідає предикат p'', що $p''(d)\downarrow = T$, якщо $p'(d)\downarrow = T$ та $pr(d)\downarrow$; $p''(d)\downarrow = F$, якщо $p'(d)\downarrow = F$ та невизначений інакше. Очевидно, що $p \models_T p'$, тобто $TC(p,p')$. Тоді, якщо $\vdash \{p'\}pr\{q\}$ за правилом *R_CONS'* отримуємо $\vdash \{p\}pr\{q\}$. Використовуючи правило *R_SKIP* маємо $\vdash \{p'\}id\{p'\}$. Очевидно, що з визначення p'' виконується $p' \models_F p''$, тобто $FC(p',p'')$, отже використовуючи правило *R_CONS'* $\vdash \{p'\}id\{p''\}$. За визначенням q' виконується $q' \models_F q$, або $FC(q',q)$. Потрібно довести, що $p'' \models_T PC(pr,q')$. Якщо на деякій інтерпретації $p''(d)\downarrow = T$, це означає $p'(d)\downarrow = T$, а значить $p(d)\downarrow = T$ та $pr(d)\downarrow$, тоді за визначенням q' виконується $q'(pr(d))\downarrow = T$, що доводить $p'' \models_T PC(pr,q')$, або $TC(p'',PC(pr,q'))$. За доведеним раніше

$\vdash \{PC(pr,q')\}pr\{q'\}$, застосовуємо правило *R_CONS'* та отримуємо $\vdash \{p''\}pr\{q\}$. Якщо до $\vdash \{p'\}id\{p''\}$ та $\vdash \{p''\}pr\{q\}$ можна застосувати правило *R_SEQ_F* отримаємо $\vdash \{p'\}id \bullet pr\{q\}$, скорочуючи за властивостями відповідних композицій отримуємо $\vdash \{p'\}pr\{q\}$. Умова правила $FC(p'',PC(pr,q))$ еквівалентна $p'' \models_F PC(pr,q)$. Якщо на довільній інтерпретації для деякого $d' \in {}^V A$ виконується $PC(pr,q)(d) \downarrow = F$, звідки $q(pr(d)) \downarrow = F$. Враховуючи $\models \{p\}pr\{q\}$ не можливо, що $p(d) \downarrow = T$, отже $p'(d) \downarrow = F$ та $p''(d) \downarrow = F$. Таким чином, доведено, що $p'' \models_F PC(pr,q)$ і умова потрібна для застосування *R_SEQ_F* виконується. Підсумовуючи все, маємо $\vdash \{p\}pr\{q\}$.

3.5 Аксіоматичні системи для композиційно-номінативних програмних логік над ієрархічними даними

Для випадку ієрархічних даних в мові програмної логіки виділяються лише множина формул та множина програмних текстів. Так як терми окремо не виділяються, то правила побудови програмних текстів для програмних логік над ієрархічними даними окрім аналогів правил для програмних логік над простими номінативними даними містять також правило, що стосується композиції суперпозиції. Це зумовлює той факт, що використання аксіоматичних систем для програмних логік над простими номінативними даними без додаткових правил не можливо. В такому випадку аксіоматичні системи не будуть повними, так як не матимуть засобів для виводу тверджень для програмних текстів, що побудовані з використанням суперпозиції. Побудувати такі правила допомагає спостереження, що з визначення семантики суперпозиції, суперпозицію в тотожну програму можна порівняти з одночасним присвоєнням і тому можна побудувати відповідне правило базуючись на правилі для присвоєння. Проте потрібно мати правило для суперпозиції в довільний текст програми. Якщо представити виконання суперпозиції для тексту програм, як зміну значень відповідних імен, а потім обчислення біквазіарної функції, що відповідає тексту програми на отриманому стані, то суперпозицію в текст програми можна представити як послідовне виконання аналогічної суперпозиції в тотожну програму і заданої програми. Так отримуємо наступні правила:

$$R\_SFID_{CC} \quad \frac{}{\{S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{p\}},$$

$$R\_SF_{CC} \quad \frac{\{p\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\bullet f\{q\}}{\{p\}S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)\{q\}}.$$

Додавши дані правила до правил аксіоматичної системи *CI*, отримаємо аксіоматичну систему CI_{CC}, що відповідає системі правил виводу в класичній логіці Флойда-Хоара.

$$R\_AS_{CC} \quad \frac{}{\{S_P^x(p,f)\}AS^x(f)\{p\}},$$

$$R\_SKIP_{CC} \quad \frac{}{\{p\}id\{p\}},$$

$$R\_SFID_{CC} \quad \frac{}{\{S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{p\}},$$

$$R\_SF_{CC} \quad \frac{\{p\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\bullet f\{q\}}{\{p\}S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)\{q\}},$$

$$R\_SEQ_{CC} \quad \frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1\bullet pr_2\{r\}},$$

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

$$R\_WH_{CC} \quad \frac{\{r\wedge p\}pr\{p\}}{\{p\}WH(r,pr)\{\neg r\wedge p\}},$$

$$R\_CONS_{CC} \quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}},\, p\to p', q'\to q.$$

Як і для аксіоматичної системи *CI*, для потрібно CI_{CC} дослідити, чи буде вона коректною та повною.

Твердження 3.2. Аксіоматична система CI_{CC} не є коректною.

Доведення. Як і для випадку простих номінативних даних, можна розглянути приклад, формули p, q та r що для деякої інтерпретації J, має місце $p_J(d)\downarrow = T$,

$r_J(d)\downarrow = F$ та $q_J(d)\uparrow$ для довільного $d \in NDVC(V, A)$. Тоді застосовуючи правила R_SKIP_{CC} та R_CONS_{CC} отримуємо $\vdash_{CI_{CC}} \{p\}id\{q\}$ та $\vdash_{CI_{CC}} \{q\}id\{r\}$. Звідки за правилом R_SEQ_{CC} маємо $\vdash_{CI_{CC}} \{p\}id \bullet id\{r\}$. Не важко показати, що для довільного $d \in NDVC(V, A)$ має місце $\{p\}id \bullet id\{r\}_J(d)\downarrow = F$. Отже не виконується, що $\models \{p\}id \bullet id\{r\}$ та аксіоматична система CI_{CC} не є коректною.

Отже для ієрархічних даних класична аксіоматична система також не буде коректною і тому потрібно розглянути аксіоматичні системи отримані за допомогою введення додаткових обмежень. Розглянемо аксіоматичні системи отримані з досліджених для простих номінативних даних за допомогою додавання правил R_SFID_{CC} та R_SF_{CC} специфічних для логіки над ієрархічними даними.

Вважатимемо, що на множині формул програмної логіки $Fr(Ps, Prs, V)$ додатково задано бінарні відношення LC_{CC}, TC_{CC} та FC_{CC}.

Аксіоматична система з доданими обмеженнями AC_{CC} в такому випадку буде мати наступний вигляд:

$$R\_AS_{CC} \quad \frac{}{\{S_P^x(p, f)\}AS^x(f)\{p\}},$$

$$R\_SKIP_{CC} \quad \frac{}{\{p\}id\{p\}},$$

$$R\_SFID_{CC} \quad \frac{}{\{S_P^{x_1, x_2, \ldots, x_n}(p, g_1, g_2, \ldots, g_n)\}S_F^{x_1, x_2, \ldots, x_n}(id, g_1, g_2, \ldots, g_n)\{p\}},$$

$$R\_SF_{CC} \quad \frac{\{p\}S_F^{x_1, x_2, \ldots, x_n}(id, g_1, g_2, \ldots, g_n) \bullet f\{q\}}{\{p\}S_F^{x_1, x_2, \ldots, x_n}(f, g_1, g_2, \ldots, g_n)\{q\}},$$

$$R\_SEQ'_{CC} \quad \frac{\{p\}pr_1\{q\}, \{q\}pr_2\{r\}}{\{p\}pr_1 \bullet pr_2\{r\}}, LC_{CC}(p, PC(pr_1 \bullet pr_2, r)),$$

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

$$R\_WH'_{CC} \quad \frac{\{r \wedge p\}pr\{p\}}{\{p\}WH(r, pr)\{\neg r \wedge p\}}, LC_{CC}(p, PC(WH(r, pr), \neg r \wedge p)),$$

$$R\_CONS'_{CC}\quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}}, TC_{CC}(p,p'), FC_{CC}(q',q).$$

Теорема 3.9. Якщо відношення LC_{CC} співпадає з відношенням логічного наслідку, відношення TC_{CC} співпадає з відношенням логічного наслідку за істиною та відношення FC_{CC} співпадає з відношенням логічного наслідку за хибою, аксіоматична система AC_{CC} коректна.

Доведення. Доведемо теорему індукцією за довжиною виводу.

База індукції. За один крок можна вивести асерції лише використовуючи правила R_AS_{CC}, R_SKIP_{CC} та R_SFID_{CC}. Правила R_AS_{CC} та R_SKIP_{CC} співпадають з правилами R_AS та R_SKIP, тому для них доведення проводиться аналогічно.

Розглянемо правило R_SFID_{CC}. За даним правилом виводиться $\vdash \{S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{p\}$. Таким чином, потрібно показати, що $\models \{S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{p\}$. Іншими словами, що для довільної інтерпретації не існує такого $d \in NDVC(V,A)$, що $FH(S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n), S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n), p)(d) \downarrow= F$. Використовуючи визначення композиції Флойда-Хоара, це виконується, якщо $S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)(d) \downarrow= T$ та $p(S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n))(d)) \downarrow= F$. Звідки $p(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)]) \downarrow= T$ для першого твердження, та для другого, $p(id(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)])) \downarrow= F$. Враховуючи визначення композиції тотожної програми прийшли до протиріччя, тому бажане твердження має місце, а саме $\models \{S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{p\}$.

Крок індукції. Припустимо, що всі асерції, що виводяться менше ніж за k кроків істинні, доведемо, що і для k також виконується дане твердження. Розглянемо різні випадки в залежності від того, яке правило було застосовано на останньому кроці. Так як правила $R\_SEQ'_{CC}$, R_IF_{CC}, $R\_WH'_{CC}$ та $R\_CONS'_{CC}$ співпадають з правилами R_SEQ', R_IF, R_WH' та R_CONS', відповідні твердження для них можуть бути доведені за аналогією. Лишається розглянути тільки правило R_SF_{CC}. За правилом $\vdash\{p\}S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)\{q\}$ виходячи з того, що $\vdash\{p\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\bullet f\{q\}$. Останнє було виведено менше ніж за k кроків, то за припущенням індукції $\models\{p\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\bullet f\{q\}$. Потрібно довести, що також $\models\{p\}S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)\{q\}$. Доведемо це, показавши, що на кожній інтерпретації обом текстам програм відповідають одна біквазіарна функція. Візьмемо довільну інтерпретацію та довільне $d\in NDVC(V,A)$. Тоді за визначенням, $S_F^{x_1,x_2,\ldots,x_n}(f,g_1,\ldots,g_n)(d)\equiv f(d\nabla[x_1\mapsto g_1(d),\ldots,x_n\mapsto g_n(d)])$. З іншого боку $(S_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)\bullet f)(d)\equiv f((S_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n))(d))$. Враховуючи, що $S_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)(d)\equiv id(d\nabla[x_1\mapsto g_1(d),\ldots,x_n\mapsto g_n(d)])$, а в свою чергу $id(d\nabla[x_1\mapsto g_1(d),\ldots,x_n\mapsto g_n(d)])\equiv d\nabla[x_1\mapsto g_1(d),\ldots,x_n\mapsto g_n(d)]$, отримуємо $(S_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)\bullet f)(d)\equiv f(d\nabla[x_1\mapsto g_1(d),\ldots,x_n\mapsto g_n(d)])$, що і доводить $S_F^{x_1,x_2,\ldots,x_n}(f,g_1,\ldots,g_n)(d)\equiv(S_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)\bullet f)(d)$. Так як функції однакові для довільної інтерпретації, то якщо асерція істинна для одного тексту програми, за визначенням істинності, вона істинна і для іншого. Отже, доведено потрібне твердження і правило коректне.

Розглянуто всі правила, а отже за індукцією аксіоматична система AC_{CC} коректна.

Теорема 3.10. Якщо відношення LC_{CC} співпадає з відношенням логічного наслідку, відношення TC_{CC} співпадає з відношенням логічного наслідку за істиною та відношення FC_{CC} співпадає з відношенням логічного наслідку за хибою, аксіоматична система AC_{CC} екстенсіонально повна.

Доведення. Доведемо спочатку, що для довільного програмного тексту pr та формули q, виконується $\vdash\{PC(pr,q)\}pr\{q\}$. Доведення будемо проводити індукцією за структурою pr. Так як правила R_AS_{CC}, R_SKIP_{CC}, $R\_SEQ'_{CC}$, R_IF_{CC}, $R\_WH'_{CC}$ та $R\_CONS'_{CC}$ співпадають з правилами R_AS, R_SKIP,

R_SEQ', R_IF, R_WH' та R_CONS', випадки, коли програмний текст це id, $AS^x(f)$, $pr_1 \bullet pr_2$, $IF(r, pr_1, pr_2)$, або $WH(r, pr)$ доводяться аналогічно до аксіоматичної системи AC.

Розглянемо випадок $S_F^{x_1,x_2,\ldots,x_n}(f, g_1, g_2, \ldots, g_n)$. Потрібно довести, що $\vdash \{PC(S_F^{x_1,x_2,\ldots,x_n}(f, g_1, g_2, \ldots, g_n), q)\} S_F^{x_1,x_2,\ldots,x_n}(f, g_1, g_2, \ldots, g_n)\{q\}$. Доведемо, що $\vdash \{PC(S_F^{x_1,x_2,\ldots,x_n}(f, g_1, g_2, \ldots, g_n), q)\} S_F^{x_1,x_2,\ldots,x_n}(id, g_1, g_2, \ldots, g_n) \bullet f\{q\}$. Це дозволить, застосувавши правило R_SF_{CC} отримати потрібне твердження. За припущенням індукції $\vdash \{PC(f, q)\} f\{q\}$. Якщо показати, що додаткова умова правила $R\_SEQ'_{CC}$ виконується, а також довести, що має місце $\vdash \{PC(S_F^{x_1,x_2,\ldots,x_n}(f, g_1, g_2, \ldots, g_n), q)\} S_F^{x_1,x_2,\ldots,x_n}(id, g_1, g_2, \ldots, g_n)\{PC(f, q)\}$, можна вивести $\vdash \{PC(S_F^{x_1,x_2,\ldots,x_n}(f, g_1, g_2, \ldots, g_n), q)\} S_F^{x_1,x_2,\ldots,x_n}(id, g_1, g_2, \ldots, g_n) \bullet f\{q\}$. За правилом R_SFID_{CC} для формули $PC(f, q)$ можна вивести $\vdash \{S_P^{x_1,x_2,\ldots,x_n}(PC(f, q), g_1, g_2, \ldots, g_n)\} S_F^{x_1,x_2,\ldots,x_n}(id, g_1, g_2, \ldots, g_n)\{PC(f, q)\}$. Звідки за $R\_CONS'_{CC}$ $\vdash \{PC(S_F^{x_1,x_2,\ldots,x_n}(f, g_1, g_2, \ldots, g_n), q)\} S_F^{x_1,x_2,\ldots,x_n}(id, g_1, g_2, \ldots, g_n)\{PC(f, q)\}$, якщо $TC(PC(S_F^{x_1,x_2,\ldots,x_n}(f, g_1, g_2, \ldots, g_n), q), S_P^{x_1,x_2,\ldots,x_n}(PC(f, q), g_1, g_2, \ldots, g_n))$. Умова виконується, бо обидві формули на довільній інтерпретації для довільного ієрархічного номінативного даного $d \in NDVC(V, A)$ одночасно якщо визначені, то рівні $q(f(d\nabla[x_1 \mapsto g_1(d), \ldots, x_n \mapsto g_n(d)]))$. Додаткову умову правила $R\_SEQ'_{CC}$ легко перевірити, так як для довільної інтерпретації результат суперпозиції в біквазіарну функцію співпадає з послідовним виконанням суперпозиції в тотожну функцію та відповідної біквазіарної функції.

Розглянуто всі можливі випадки і за індукцією отримуємо, що для довільного програмного тексту pr та формули q, виконується $\vdash \{PC(pr, q)\} pr\{q\}$.

Якщо ми маємо істинну асерцію, $\models \{p\} pr\{q\}$ та для довільного програмного тексту pr та формули q, виконується $\vdash \{PC(pr, q)\} pr\{q\}$, доведення того факту, що $\vdash \{p\} pr\{q\}$ відбувається аналогічно до доведення відповідного факту для аксіоматичної системи AC. Це можливо, бо AC_{CC} також містить правила, що були використані в доведені. Отже, доведено, що аксіоматична система AC_{CC} екстенсіонально повна.

Як і у випадку програмних логік для простих номінативних даних, аксіоматична система з доданими умовами є коректною та екстенсіонально повною, проте вона також має відповідні недоліки, а саме складність та надлишковість умов. Тому розглянемо аксіоматичну систему TI_{CC} для програмних логік над складними ієрархічними даними, отриману з TI. Вона буде мати наступний вигляд:

$$R\_AS_{CC} \quad \frac{}{\{S_P^x(p,f)\}AS^x(f)\{p\}},$$

$$R\_SFID_{CC} \quad \frac{}{\{S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{p\}},$$

$$R\_SF_{CC} \quad \frac{\{p\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\bullet f\{q\}}{\{p\}S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)\{q\}},$$

$$R\_SKIP_{CC} \quad \frac{}{\{p\}id\{p\}},$$

$$R\_SEQ_{CC} \quad \frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1\bullet pr_2\{r\}},$$

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

$$R\_WH_{CC} \quad \frac{\{r\wedge p\}pr\{p\}}{\{p\}WH(r,pr)\{\neg r\wedge p\}},$$

$$R\_CONS''_{CC} \quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}},TC_{CC}(p,p'),TC_{CC}(q',q).$$

Теорема 3.11. Якщо відношення TC співпадає з відношенням логічного наслідку за істиною, аксіоматична система TI коректна.

Доведення. Доведемо спочатку, що кожне правило зберігає для асерцій $\{p\}pr\{q\}$ властивість, що для довільної інтерпретації виконується $p^{T,pr}\subseteq q^T$.

Доведення для правил R_AS_{CC}, R_SKIP_{CC}, R_SEQ_{CC}, R_IF_{CC}, R_WH_{CC} та $R\_CONS''_{CC}$ аналогічне доведенню відповідної властивості для правил R_AS, R_SKIP, R_SEQ, R_IF, R_WH та R_CONS'' аксіоматичної системи *TI*.

Розглянемо правило R_SFID_{CC}. Потрібно показати, що для довільної інтерпретації $S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)^{T,S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)} \subseteq p^T$. Якщо для деякого $d \in NDVC(V,A)$ виконується $S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)(d)\downarrow = T$. За визначенням композиції суперпозиції це еквівалентно $p(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)])\downarrow = T$. Це означає, що $g_i(d)\downarrow$, а отже також $S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)(d)\downarrow$. Отже $p(S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)(d))\downarrow = p(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)]) = T$. В такому випадку за визначенням композиції присвоєння маємо $p((AS^x(f))(d))\downarrow = p(d\nabla[x \mapsto f(d)]) = T$. Це означає, що якщо виконується $S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)(d) \in S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)^{T,S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)}$, то також має місце $S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)(d) \in p^T$. Таким чином, доведено потрібне включення.

Розглянемо правило R_SF_{CC}. Потрібно показати, що якщо виконується твердження для засновку, то воно виконується і для висновку правила. Так як засновок і висновок відрізняються лише текстом програми, покажемо, що на довільній інтерпретації їм відповідає одна біквазіарна функція. Візьмемо довільну інтерпретацію та довільне $d \in NDVC(V,A)$. Тоді за визначенням, $\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(f,g_1,\ldots,g_n)(d) \cong f(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)])$. З іншого боку $(\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)\bullet f)(d) \cong f((\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n))(d))$. Враховуючи, що $\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)(d) \cong id(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)])$, а в свою чергу $id(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)]) \cong d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)]$, отримуємо $(\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)\bullet f)(d) \cong f(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)])$, що і доводить $\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(f,g_1,\ldots,g_n)(d) \cong (\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)\bullet f)(d)$.

Всі правила було розглянуто, а отже, було доведено, що якщо $\vdash \{p\}pr\{q\}$, то для довільної інтерпретації виконується $p^{T,pr} \subseteq q^{T}$. Покажемо, що в такому випадку $\models \{p\}pr\{q\}$. Нехай дане твердження не виконується, тоді існує інтерпретація J та дане $d \in NDVC(V,A)$ такі, що $FH(p,pr,q)_J(d) \downarrow= F$. Тоді $p_J(d) \downarrow= T$, та $q_J(pr_J(d)) \downarrow= F$. Але в такому випадку $pr_J(d) \in p^{T,pr}$ та $pr_J(d) \in q^{F}$, а отже, $pr_J(d) \notin q^{T}$. Прийшли до протиріччя, звідки $\models \{p\}pr\{q\}$.

Теорема 3.12. Якщо відношення TC_{CC} співпадає з відношенням логічного наслідку за істиною, аксіоматична система TI_{CC} екстенсіонально повна на класі асерцій $\{p\}pr\{q\}$ для яких виконується $p \models_T PC(pr,q)$.

Доведення. Доведемо спочатку, що для довільного програмного тексту pr та формули q, виконується $\vdash \{PC(pr,q)\}pr\{q\}$. Доведення будемо проводити індукцією за структурою pr. Так як правила R_AS_{CC}, R_SKIP_{CC}, R_SEQ_{CC}, R_IF_{CC}, R_WH_{CC} та $R\_CONS''_{CC}$ співпадають з правилами R_AS, R_SKIP, R_SEQ, R_IF, R_WH та R_CONS'', випадки, коли програмний текст це id, $AS^{x}(f)$, $pr_1 \bullet pr_2$, $IF(r,pr_1,pr_2)$, або $WH(r,pr)$ доводяться аналогічно до аксіоматичної системи TI.

Розглянемо випадок $S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)$. Як і для аксіоматичної системи AC_{CC}, з $\vdash \{PC(S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n),q)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n) \bullet f\{q\}$ за правилом R_SF_{CC} можна вивести необхідну асерцію, тобто $\vdash \{PC(S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n),q)\}S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)\{q\}$. В свою чергу, $\vdash \{PC(S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n),q)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n) \bullet f\{q\}$ виводиться за правилом R_SEQ_{CC} так як за припущенням індукції $\vdash \{PC(f,q)\}f\{q\}$ та доведемо, що $\vdash \{PC(S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n),q)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{PC(f,q)\}$. За правилом R_SFID_{CC} для формули $PC(f,q)$ можна вивести $\vdash \{S_P^{x_1,x_2,\ldots,x_n}(PC(f,q),g_1,g_2,\ldots,g_n)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{PC(f,q)\}$. Звідки за $R\_CONS''_{CC}$ $\vdash \{PC(S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n),q)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{PC(f,q)\}$, якщо $TC(PC(S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n),q),S_P^{x_1,x_2,\ldots,x_n}(PC(f,q),g_1,g_2,\ldots,g_n))$. Умова виконується, бо обидві формули на довільній інтерпретації для довільного ієрархічного номінативного даного $d \in NDVC(V,A)$ одночасно якщо визначені, то рівні $q(f(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)]))$.

Розглянуто всі можливі випадки і за індукцією отримуємо, що для довільного програмного тексту pr та формули q, виконується $\vdash \{PC(pr,q)\}pr\{q\}$.

Якщо ми маємо істинну асерцію, $\models \{p\}pr\{q\}$ для якої має місце $p \models_T PC(pr,q)$, що значить за припущенням, що $TC_{CC}(p,PC(pr,q))$. Було доведено, що $\vdash \{PC(pr,q)\}pr\{q\}$, тоді можна застосувати правило $R\_CONS''_{CC}$, додаткові умови виконуються, а отже отримуємо $\vdash \{p\}pr\{q\}$. Отже, доведено, що аксіоматична система TI_{CC} екстенсіонально повна на класі асерцій таких, що $p \models_T PC(pr,q)$.

Було доведено, що аксіоматична система для програмної логіки над складними ієрархічними даними побудована з *T*-зростаючої аксіоматичної системи також коректна та екстенсіонально повна на класі *T*-зростаючих асерцій.

Розглянемо аксіоматичну систему FD_{CC}, що побудована на основі аксіоматичної системи *FD*, дуальної до аксіоматичної системи TI . Вона буде мати наступний вигляд:

$$R\_AS_{CC} \quad \frac{}{\{S_P^x(p,f)\}AS^x(f)\{p\}},$$

$$R\_SKIP_{CC} \quad \frac{}{\{p\}id\{p\}},$$

$$R\_SFID_{CC} \quad \frac{}{\{S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{p\}},$$

$$R\_SF_{CC} \quad \frac{\{p\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\bullet f\{q\}}{\{p\}S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)\{q\}},$$

$$R\_SEQ_{CC} \quad \frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1 \bullet pr_2\{r\}},$$

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

$$R\_WH_{CC} \quad \frac{\{r \wedge p\}pr\{p\}}{\{p\}WH(r, pr)\{\neg r \wedge p\}},$$

$$R\_CONS'''_{CC} \quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}}, FC_{CC}(p, p'), FC_{CC}(q', q).$$

Теорема 3.13. Якщо відношення FC_{CC} співпадає з відношенням логічного наслідку за хибою, аксіоматична система FD_{CC} коректна.

Доведення. Для того, щоб довести коректність аксіоматичної системи, доведемо, для кожного правила, що воно зберігає для асерцій $\{p\}pr\{q\}$ властивість $p \models_F PC(pr, q)$.

Доведення для правил R_AS_{CC}, R_SKIP_{CC}, R_SEQ_{CC}, R_IF_{CC}, R_WH_{CC} та $R\_CONS'''_{CC}$ аналогічне доведенню відповідної властивості для правил R_AS, R_SKIP, R_SEQ, R_IF, R_WH та R_CONS''' аксіоматичної системи FD.

Розглянемо правило R_SFID_{CC}. Потрібно показати, що $S_P^{x_1, x_2, \ldots, x_n}(p, g_1, g_2, \ldots, g_n) \models_F PC(S_F^{x_1, x_2, \ldots, x_n}(id, g_1, g_2, \ldots, g_n), p)$. Нехай для довільної інтерпретації для деякого $d \in NDVC(V, A)$ виконується $PC(S_F^{x_1, x_2, \ldots, x_n}(id, g_1, g_2, \ldots, g_n), p)(d) \downarrow = F$. За визначенням композиції суперпозиції та композиції побудови умови за прообразом, це еквівалентно $p(id(d \nabla [x_1 \mapsto g_1(d), \ldots, x_n \mapsto g_n(d)])) \downarrow = F$. Це означає, що $p(d \nabla [x_1 \mapsto g_1(d), \ldots, x_n \mapsto g_n(d)]) \downarrow = F$ та $g_i(d) \downarrow$, а отже також. Отже за визначенням композиції суперпозиції $S_P^{x_1, x_2, \ldots, x_n}(p, g_1, g_2, \ldots, g_n)(d) \downarrow = F$. Доведено включення областей хибності, звідки так як було обрано довільну інтерпретацію, маємо $S_P^{x_1, x_2, \ldots, x_n}(p, g_1, g_2, \ldots, g_n) \models_F PC(S_F^{x_1, x_2, \ldots, x_n}(id, g_1, g_2, \ldots, g_n), p)$.

Розглянемо правило R_SF_{CC}. Потрібно показати, що якщо виконується твердження для засновку, то воно виконується і для висновку правила. Так як засновок і висновок відрізняються лише текстом програми, покажемо, що на довільній інтерпретації їм відповідає одна біквазіарна функція. Візьмемо довільну інтерпретацію та довільне $d \in NDVC(V,A)$. Тоді за визначенням, $S_F^{x_1,x_2,\ldots,x_n}(f,g_1,\ldots,g_n)(d) = f(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)])$. З іншого боку $(S_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)\bullet f)(d) = f((S_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n))(d))$. Враховуючи, що $S_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)(d) = id(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)])$, а в свою чергу $id(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)]) = d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)]$, отримуємо $(S_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)\bullet f)(d) = f(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)])$, що і доводить $S_F^{x_1,x_2,\ldots,x_n}(f,g_1,\ldots,g_n)(d) = (S_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)\bullet f)(d)$.

Для всіх правил було доведено потрібне твердження, звідки отримуємо, що якщо $\vdash \{p\}pr\{q\}$, то виконується $p \models_F PC(pr,q)$. Потрібно показати, що тоді $\models \{p\}pr\{q\}$. Нехай асерція не є істинною, отже існує інтерпретація J та дане $d \in NDVC(V,A)$ такі, що $FH(p,pr,q)_J(d)\downarrow = F$. Тоді $p_J(d)\downarrow = T$, та $q_J(pr_J(d))\downarrow = F$. Проте з доведеної властивості $p \models_F PC(pr,q)$ маємо $PC(pr,q)_J^F \subseteq p_J^F$. З $q_J(pr_J(d))\downarrow = F$ виводимо, що $d \in PC(pr,q)_J^F$. В такому випадку також $d \in p_J^F$, тобто $p_J(d)\downarrow = F$. Прийшли до протиріччя, отже, $\models \{p\}pr\{q\}$.

Теорема 3.14. Якщо відношення FC_{CC} співпадає з відношенням логічного наслідку за хибою, аксіоматична система FD_{CC} не є екстенсіонально повною на класі асерцій $\{p\}pr\{q\}$ для яких виконується $p \models_F PC(pr,q)$.

Доведення. При доведенні коректності аксіоматичної системи було показано, що якщо $\vdash \{p\}pr\{q\}$, то $p \models_F PC(pr,q)$. Таким чином, асерції, для яких не виконується $p \models_F PC(pr,q)$ не можуть бути виведені в FD_{CC}.

Покажемо, також, що існують асерції, для яких виконується $p \models_F PC(pr,q)$, але вони не можуть бути в аксіоматичній системі FD_{CC}. Розглянемо як приклад асерцію, аналогічний асерції для якої не існувало виводу в FD.

Нехай формули p, q та r такі що для довільної інтерпретації J, виконується $p_J(d)\uparrow$, $q_J(d)\downarrow = F$ та $r_J(d)\uparrow$ для довільного $d \in NDVC(V,A)$. Розглянемо асерцію $\{p\}IF(r,id,id)\{q\}$. Неважко показати, що в такому випадку виконується умова, тобто $p \models_F PC(IF(r,id,id),q)$. Разом з тим за означенням композиції Флойда-Хоара виконується $\models \{p\}IF(r,id,id)\{q\}$. Отже, якщо аксіоматична система екстенсіонально повна на класі асерцій $\{p\}pr\{q\}$ для яких виконується $p \models_F PC(pr,q)$, то $\vdash \{p\}IF(r,id,id)\{q\}$. Для набору правил без R_SF_{CC} та R_SFID_{CC} легко продемонструвати, що не виконується $\vdash \{p\}IF(r,id,id)\{q\}$. Але останнім правилом в виводі $\{p\}IF(r,id,id)\{q\}$ не може бути ні правило R_SF_{CC} , ні R_SFID_{CC}. Саме тому всі висновки, що було вірно для FD має місце і для FD_{CC}. Отже для $\{p\}IF(r,id,id)\{q\}$ не можна побудувати вивід в FD_{CC} і тому аксіоматична система не є екстенсіонально повною на класі асерцій $\{p\}pr\{q\}$ для яких виконується $p \models_F PC(pr,q)$.

Таким чином для аксіоматичних систем TI_{CC} та FD_{CC} також виконуються відповідні властивості аксіоматичних систем *TI* та *FD*. Вони також коректні, але TI_{CC} повна лише для певного класу асерцій. В той же час клас асерцій на якому FD_{CC} є повною потребує подальшого дослідження. Саме тому означені аксіоматичні системи, як і системи, що служили для них основою варто використовувати лише в спеціальних випадках.

Залишається лише розглянути аксіоматичну систему TF_{CC} для програмних логік для ієрархічних даних, побудовану на основі аксіоматичної системи з T- та F-обмеженнями для програмної логіки над простими номінативними даними. Як і для попередніх аксіоматичних систем, для кожного правила системи *TF* візьмемо його аналог в програмній логіці над ієрархічними даними та додамо правила для суперпозиції, одразу будемо розглядати окремі правила для послідовного виконання та циклу, що відрізняються лише додатковими умовами. В такому випадку правила мають наступний вигляд:

$$R\_AS_{CC} \quad \frac{}{\{S_P^x(p,f)\}AS^x(f)\{p\}},$$

$$R\_SKIP_{CC} \quad \frac{}{\{p\}id\{p\}},$$

$$R\_SFID_{CC} \quad \frac{}{\{S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{p\}},$$

$$R\_SF_{CC} \quad \frac{\{p\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\bullet f\{q\}}{\{p\}S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)\{q\}},$$

$$R\_SEQ\_T_{CC} \quad \frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1\bullet pr_2\{r\}},TC_{CC}(p,PC(pr_1,q)),$$

$$R\_SEQ\_F_{CC} \quad \frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1\bullet pr_2\{r\}},FC_{CC}(q,PC(pr_2,r)),$$

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

$$R\_WH\_T_{CC} \quad \frac{\{r\wedge p\}pr\{p\}}{\{p\}WH(r,pr)\{\neg r\wedge p\}},TC_{CC}(r\wedge p,PC(pr,p)),$$

$$R\_WH\_F_{CC} \quad \frac{\{r\wedge p\}pr\{p\}}{\{p\}WH(r,pr)\{\neg r\wedge p\}},FC_{CC}(r\wedge p,PC(pr,p)).$$

$$R\_CONS'_{CC} \quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}},TC_{CC}(p,p'),FC_{CC}(q',q).$$

Теорема 3.15. Якщо відношення TC_{CC} співпадає з відношенням логічного наслідку за істиною та відношення FC_{CC} співпадає з відношенням логічного наслідку за хибою, аксіоматична система TF_{CC} коректна.

Доведення. Доведемо теорему індукцією за довжиною виводу.

База індукції. За один крок можна вивести асерції лише використовуючи правила R_AS_{CC}, R_SKIP_{CC} та R_SFID_{CC}. Правила R_AS_{CC} та R_SKIP_{CC} співпадають з правилами R_AS та R_SKIP, тому для них доведення проводиться аналогічно.

Розглянемо правило R_SFID_{CC}. Потрібно показати, що $\models\{S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{p\}$. Нехай це не виконується, отже існує та $d\in NDVC(V,A)$, такі, що відповідний предикат на даному повертає хибу, тобто $FH(S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n),S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n),p)(d)\downarrow=F$. Використовуючи визначення композиції Флойда-Хоара, це виконується, якщо $S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)(d)\downarrow=T$ та $p(S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n))(d))\downarrow=F$. Розпишемо обидва твердження за визначеннями відповідних композицій, отримуємо $p(d\nabla[x_1\mapsto g_1(d),\ldots,x_n\mapsto g_n(d)])\downarrow=T$ для першого твердження, та $p(id(d\nabla[x_1\mapsto g_1(d),\ldots,x_n\mapsto g_n(d)]))\downarrow=F$ для другого. Враховуючи визначення композиції тотожної програми прийшли до протиріччя, тому бажане твердження має місце, а саме$\models\{S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{p\}$.

Крок індукції. Припустимо, що всі асерції, що виводяться менше ніж за k кроків істинні, доведемо, що і для k також виконується дане твердження. Розглянемо різні випадки в залежності від того, яке правило було застосовано на останньому кроці. Так як правила $R\_SEQ\_T_{CC}$, $R\_SEQ\_F_{CC}$, R_IF_{CC}, $R\_WH\_T_{CC}$, $R\_WH\_F_{CC}$ та $R\_CONS'_{CC}$ співпадають з правилами R_SEQ_T, R_SEQ_F, R_IF, R_WH_T, R_WH_F та R_CONS', відповідні твердження для них можуть бути доведені за аналогією. Лишається розглянути тільки правило R_SF_{CC}. За правилом $\vdash\{p\}S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)\{q\}$ виходячи з того, що $\vdash\{p\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\bullet f\{q\}$. Останнє було виведено менше ніж за k кроків, то за припущенням індукції $\models\{p\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\bullet f\{q\}$. Потрібно довести, що в такому випадку $\models\{p\}S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)\{q\}$. Це легко довести, що на довільній інтерпретації $S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\bullet f$ та $S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)$ відповідає одна біквазіарна функція. Для цього візьмемо довільну інтерпретацію та довільне $d\in NDVC(V,A)$ І розпишемо за визначенням, $\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(f,g_1,\ldots,g_n)(d)\equiv f(d\nabla[x_1\mapsto g_1(d),\ldots,x_n\mapsto g_n(d)])$. З іншого боку $(\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)\bullet f)(d)\equiv f((\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n))(d))$. Враховуючи, що $\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)(d)\equiv id(d\nabla[x_1\mapsto g_1(d),\ldots,x_n\mapsto g_n(d)])$, а в свою чергу $id(d\nabla[x_1\mapsto g_1(d),\ldots,x_n\mapsto g_n(d)])\equiv d\nabla[x_1\mapsto g_1(d),\ldots,x_n\mapsto g_n(d)]$, отримуємо

$(\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)\bullet f)(d)\ ;\ f(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)])$, що і доводить $\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(f,g_1,\ldots,g_n)(d)\ ;\ (\mathrm{S}_F^{x_1,x_2,\ldots,x_n}(id,g_1,\ldots,g_n)\bullet f)(d)$. Отже, правило коректне.

Всі правила було розглянуто, таким чином, за індукцією аксіоматична система коректна.

Теорема 3.16. Якщо відношення TC_{CC} співпадає з відношенням логічного наслідку за істиною та відношення FC_{CC} співпадає з відношенням логічного наслідку за хибою, аксіоматична система TF_{CC} екстенсіонально повна.

Доведення. Доведемо спочатку, що для довільного програмного тексту pr та формули q, виконується $\vdash\{PC(pr,q)\}pr\{q\}$. Доведення будемо проводити індукцією за структурою pr. Так як правила R_AS_{CC}, R_SKIP_{CC}, $R\_SEQ\_T_{CC}$, $R\_SEQ\_F_{CC}$, R_IF_{CC}, $R\_WH\_T_{CC}$, $R\_WH\_F_{CC}$ та $R\_CONS'_{CC}$ співпадають з правилами R_AS, R_SKIP, R_SEQ_T, R_SEQ_F, R_IF, R_WH_T, R_WH_F та R_CONS', випадки, коли програмний текст це id, $AS^x(f)$, $pr_1\bullet pr_2$, $IF(r,pr_1,pr_2)$, або $WH(r,pr)$ доводяться аналогічно до аксіоматичної системи TF.

Розглянемо випадок $S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)$. За припущенням індукції $\vdash\{PC(f,q)\}f\{q\}$. Також за правилом R_SFID_{CC} можна отримати $\vdash\{S_P^{x_1,x_2,\ldots,x_n}(PC(f,q),g_1,g_2,\ldots,g_n)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{PC(f,q)\}$. Якщо $TC(PC(S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n),q),S_P^{x_1,x_2,\ldots,x_n}(PC(f,q),g_1,g_2,\ldots,g_n))$, то за правилом $R\_CONS'_{CC}$ $\vdash\{PC(S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n),q)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\{PC(f,q)\}$. В такому випадку, якщо $FC_{CC}(PC(f,q),PC(f,q))$, що очевидно, то за правилом $R\_SEQ\_F_{CC}$ зважаючи на те, що $\vdash\{PC(f,q)\}f\{q\}$, отримуємо, $\vdash\{PC(S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n),q)\}S_F^{x_1,x_2,\ldots,x_n}(id,g_1,g_2,\ldots,g_n)\bullet f\{q\}$. До отриманого застосовуємо R_SF_{CC} та отримаємо шукане твердження, $\vdash\{PC(S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n),q)\}S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n)\{q\}$.

Залишається довести, що виконується умова для правила $R\_CONS'_{CC}$, тобто $TC(PC(S_F^{x_1,x_2,\ldots,x_n}(f,g_1,g_2,\ldots,g_n),q),S_P^{x_1,x_2,\ldots,x_n}(PC(f,q),g_1,g_2,\ldots,g_n))$. Легко показати, що обидві формули якщо визначені на довільній інтерпретації для довільного ієрархічного номінативного даного $d \in NDVC(V,A)$, то рівні одночасно $q(f(d\nabla[x_1 \mapsto g_1(d),\ldots,x_n \mapsto g_n(d)]))$.

Розглянуто всі можливі випадки і за індукцією отримуємо, що для довільного програмного тексту pr та формули q, виконується $\vdash \{PC(pr,q)\}pr\{q\}$.

Якщо ми маємо істинну асерцію, $\models \{p\}pr\{q\}$ та для довільного програмного тексту pr та формули q, виконується $\vdash \{PC(pr,q)\}pr\{q\}$, доведення того факту, що $\vdash \{p\}pr\{q\}$ відбувається аналогічно до доведення відповідного факту для аксіоматичної системи TF. Таким чином, доведено, що аксіоматична система TF_{CC} екстенсіонально повна.

Було розглянуто всі аксіоматичні системи, що будуються для програмної логіки над ієрархічними даними, базуючись на аксіоматичних системах для програмної логіки над простими номінативними даними. Також показано, що в такі аксіоматичні системи зберігають властивості коректності та екстенсіональної повноти.

Висновки

Аксіоматична система класичної логіки Флойда-Хоара для простої імперативної мови WHILE не є коректною при перенесенні в програмні логіки часткових квазіарних функцій та предикатів. Це зумовлює потребу в побудові коректних аксіоматичних систем. Одним з можливих підходів до розв'язку даної проблеми є введення додаткових обмежень на правила виводу, що призводять до виводу хибних асерцій. Такі обмеження в системі AC дозволяють отримати коректну та екстенсіонально повну аксіоматичну систему, проте є дуже складними та надлишковими для практичного використання.

Системи TI та FD, що базуються на властивостях асерцій, що виводяться, коректні, проте не є екстенсіонально повними. Дані аксіоматичні системи мають прості обмеження на правила, проте той факт, що вони не є повними не дозволяє широко використовувати їх в доведенні коректності програм.

Поєднання попередніх підходів дозволяє отримати аксіоматичну систему *TF*, що має відносно прості додаткові умови є коректною та екстенсіонально повною.

Так як в композиційно-номінативних програмних логіках над складними ієрархічними даними не розрізняються терми та програмні тести, то вважається, що програми можуть бути утворені також за допомогою композиції суперпозиції. Таким чином, для того, щоб бути екстенсіонально повними, аксіоматичні системи повинні мати також правила і для композиції суперпозиції. Побудовані за аналогією до аксіоматичних систем композиційно-номінативних програмних логік над простими номінативними даними аксіоматичні системи є коректними та екстенсіонально повними за винятком TI_{CC} та FD_{CC} які не є екстенсіонально повними.

РОЗДІЛ 4
БАГАТОСОРТНІ АКСІОМАТИЧНІ СИСТЕМИ

В попередніх розділах було розглянуто програмні логіки для простих номінативних даних та складних, ієрархічних даних. В обох випадках значення для імен простих змінних належать деякій множині базових значень. Хоча для ієрархічних даних значеннями змінних могли бути не тільки базові елементи, але й номінативні множини, вважається, що всі прості значення належать одному типу. Проте в мовах програмування виділяються різні прості типи даних, наприклад, цілі числа, символи, числа з плаваючою крапкою. Кожна змінна має певний тип, типи змінних та значення мають бути узгоджені. Для того, щоб мати змогу оперувати з такими поняттями, потрібно розглянути типізовані композиційно-номінативні програмні логіки. В роботах[45,46] представлено спосіб для побудови на основі композиційно-номінативної логіки функцій та предикатів, типізованої композиційно-номінативної логіки функцій та предикатів. Використаємо наведений спосіб для програмного рівня і отримаємо типізовану, або багатосортну композиційно-номінативну програмну логіку.

4.1 Багатосортна композиційно-номінативна програмна алгебра

Використаємо як і у випадках програмних логік для простих даних та ієрархічних даних семантико-синтаксичний підхід. Спочатку побудуємо програмну алгебру, за допомогою якої задаватимемо семантику програмної логіки, потім визначимо мову логіки та відношення інтерпретації.

Для того, щоб ввести багатосортну програмну алгебру, потрібно спочатку ввести поняття типізованої номінативної множини.

Нехай V – множина імен змінних, T – клас типів змінних та $\tau : V \xrightarrow{t} T$ відображення імен змінних в їх тип.

Визначення. Клас типізованих номінативних множин, $NST(V,T,\tau)$ задається як $NST(V,T,\tau) = \{d : V \xrightarrow{p} \bigcup_{A \in T} A \mid \forall v \in V(d(v)\downarrow \Rightarrow d(v) \in \tau(v))\}$.

Таким чином типізована номінативна множина це часткове відображення з множини імен в множину всіх можливих базових значень, таке що для кожного змінної значення має належати типу змінної.

Для того позначати включення елементів до типізованої номінативної множини, задавати множину будемо використовувати позначення для простих номінативних множин. Операція накладання також визначається по аналогії до операції накладання для простих номінативних множин.

Визначення. Багатосортними частковими квазіарними предикатами будемо називати часткові предикати над типізованими номінативними множинами, $Pr_T^{V,T,\tau} : NST(V,T,\tau) \xrightarrow{p} \{T,F\}$.

Визначення. Багатосортними частковими квазіарними ординарними функціями для фіксованого типу $A \in T$ будемо називати часткові функції з типізованих номінативних множин в множину значень типу A, $Fn_T^{A,V,T,\tau} : NST(V,T,\tau) \xrightarrow{p} A$.

Визначення. Багатосортними частковими біквазіарними функціями будемо називати часткові функції над типізованими номінативними множинами, $FPrg_T^{V,T,\tau} : NST(V,T,\tau) \xrightarrow{p} NST(V,T,\tau)$.

За допомогою багатосортних часткових квазіарних предикатів будемо задавати семантику умов. Семантику виразів певного типу будемо представляти як багатосортні часткові квазіарні ординарні функції відповідного типу. А семантику операторів і програм – як багатосортні часткові біквазіарні.

Багатосортну програмну алгебру в такому випадку будемо задавати як алгебру з наступними основами: множиною багатосортних часткових квазіарних предикатів $Pr_T^{V,T,\tau}$, множинами багатосортних часткових ординарних функцій $Fn_T^{A,V,T,\tau}$ для кожного $A \in T$ та множиною багатосортних біквазіарних часткових функцій $FPrg_T^{V,T,\tau}$. Операції алгебри, або композиції будемо визначати аналогічно композиціям програмної алгебри для простих номінативних даних. В такому випадку багатосортна програмна алгебра матиме наступний вигляд:

$TQPA(V,T,\tau) = < Pr_T^{V,T,\tau}, \{Fn_T^{A,V,T,\tau}\}_{A\in T}, FPrg_T^{V,T,\tau}; \vee, \neg, \{S_P^{\bar{v}}\}_{\bar{v}\in U(V)}, \{S_F^{\bar{v}}\}_{\bar{v}\in U(V)}, \{\forall x\}_{x\in V},$
$\{\exists x\}_{x\in V}, id, \{AS^x\}_{x\in V}, \bullet, IF, WH, FH, PC >$

Де $U(V)$ позначаємо множину всіх скінчених послідовностей з V.

Дамо визначення композицій.

Бінарна композиція диз'юнкції $\vee$ має тип $Pr_T^{V,T,\tau} \times Pr_T^{V,T,\tau} \xrightarrow{t} Pr_T^{V,T,\tau}$, та визначається наступним чином, де $p \in Pr_T^{V,T,\tau}, q \in Pr_T^{V,T,\tau}, d \in NST(V,T,\tau)$:

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

Унарна композиція заперечення $\neg$ має тип $Pr_T^{V,T,\tau} \xrightarrow{t} Pr_T^{V,T,\tau}$, та визначається наступним чином, де $p \in Pr_T^{V,T,\tau}, NST(V,T,\tau)$:

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

Унарна параметрична композиція екзистенціальної квантифікації $\exists x$ з параметром $x \in V$ має тип $Pr_T^{V,T,\tau} \xrightarrow{t} Pr_T^{V,T,\tau}$, та визначається наступним чином, де $p \in Pr_T^{V,T,\tau}, q \in Pr_T^{V,T,\tau}, NST(V,T,\tau)$:

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

Композиція суперпозиції для функцій $S_F^{x_1,x_2,\ldots x_n}$ є $n+1$-арною параметричною композицією з параметрами $x_1, x_2, \ldots, x_n \in V$,має тип $\bigcup_{A\in T} Fn_T^{A,V,T,\tau} \times Fn_T^{\tau(x_1),V,T,\tau} \times \ldots \times Fn_T^{\tau(x_n),V,T,\tau} \xrightarrow{t} \bigcup_{A\in T} Fn_T^{A,V,T,\tau}$, та визначається наступним чином, де $f \in \bigcup_{A\in T} Fn_T^{A,V,T,\tau}, g_1 \in Fn_T^{\tau(x_1),V,T,\tau}, \ldots, g_n \in Fn_T^{\tau(x_n),V,T,\tau}, d \in NST(V,T,\tau)$:

$$S_F^{x_1,x_2,\ldots,x_n}(f, g_1, g_2, \ldots, g_n)(d) ; \; f(d\nabla[x_1 \mapsto g_1(d), x_2 \mapsto g_2(d), \ldots, x_n \mapsto g_n(d)]).$$

Композиція суперпозиції для предикатів $S_P^{x_1,x_2,\ldots x_n}$ є $n+1$-арною параметричною композицією з параметрами $x_1,x_2,\ldots,x_n \in V$,має тип $Pr_T^{V,T,\tau} \times Fn_T^{\tau(x_1),V,T,\tau} \times\ldots\times Fn_T^{\tau(x_n),V,T,\tau} \xrightarrow{t} Pr_T^{V,T,\tau}$, та визначається наступним чином, де $p \in Pr_T^{V,T,\tau}, g_1 \in Fn_T^{\tau(x_1),V,T,\tau},\ldots, g_n \in Fn_T^{\tau(x_n),V,T,\tau}, d \in NST(V,T,\tau)$:

$$S_P^{x_1,x_2,\ldots,x_n}(p,g_1,g_2,\ldots,g_n)(d) ; \ p(d\nabla[x_1 \mapsto g_1(d), x_2 \mapsto g_2(d),\ldots,x_n \mapsto g_n(d)]).$$

Нуль-арна параметрична композиція розіменування $`x$ з параметром $x \in V$, $`x \in Fn_T^{\tau(x),V,T,\tau}$, якщо $d \in NST(V,T,\tau)$ то значення функції що задається композицією визначається як значення відповідної змінної для даного d, якщо вона визначена:

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

Унарна параметрична композиція присвоєння AS^x з параметром $x \in V$ має тип $Fn_T^{\tau(x),V,T,\tau} \xrightarrow{t} FPrg_T^{V,T,\tau}$, та визначається наступним чином, де $f \in Fn_T^{\tau(x),V,T,\tau}, d \in NST(V,T,\tau)$:

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

Бінарна композиція послідовного виконання $\bullet$ має тип $FPrg_T^{V,T,\tau} \times FPrg_T^{V,T,\tau} \xrightarrow{t} FPrg_T^{V,T,\tau}$, та визначається наступним чином, де $pr_1, pr_2 \in FPrg_T^{V,T,\tau}, d \in NST(V,T,\tau)$:

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

Тернарна умовна композиція IF має тип $Pr_T^{V,T,\tau} \times FPrg_T^{V,T,\tau} \times FPrg_T^{V,T,\tau} \xrightarrow{t} FPrg_T^{V,T,\tau}$, та визначається наступним чином, де $r \in Pr_T^{V,T,\tau}, pr_1, pr_2 \in FPrg_T^{V,T,\tau}, d \in NST(V,T,\tau)$:

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

Бінарна композиція циклу WH має тип $Pr_T^{V,T,\tau} \times FPrg_T^{V,T,\tau} \xrightarrow{t} FPrg_T^{V,T,\tau}$, та визначається наступним чином, де $r \in Pr_T^{V,T,\tau}, pr \in FPrg_T^{V,T,\tau}, d \in NST(V,T,\tau)$:

$WH(r,pr)(d)=d'$, якщо існує послідовність номінативних множин $d_0,d_1,\ldots,d_n \in NST(V,T,\tau)$ таких, що $d_0=d, d_n=d'$ та $d_1=pr(d_0), d_2=pr(d_1),\ldots,d_n=pr(d_{n-1})$ разом з $r(d_0)=r(d_1)=\ldots=r(d_{n-1})=T,\ r(d_n)=F$. Інакше, $WH(r,pr)(d)\uparrow$.

Нуль-арна композиція тотожної програми $id \in FPrg_T^{V,T,\tau}$, визначається наступним чином, де $d \in NST(V,T,\tau)$:

$id(d)=d$.

Тернарна композиція Флойда-Хоара FH має тип $Pr_T^{V,T,\tau} \times FPrg_T^{V,T,\tau} \times Pr_T^{V,T,\tau} \xrightarrow{t} Pr_T^{V,T,\tau}$, та визначається наступним чином, де $p,q \in Pr_T^{V,T,\tau}, pr \in FPrg_T^{V,T,\tau}, d \in NST(V,T,\tau)$:

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

Бінарна композиція побудови умови за прообразом PC має тип $FPrg_T^{V,T,\tau} \times Pr_T^{V,T,\tau} \xrightarrow{t} Pr_T^{V,T,\tau}$, та визначається наступним чином, де $q \in Pr_T^{V,T,\tau}, pr \in FPrg_T^{V,T,\tau}, d \in NST(V,T,\tau)$:

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

Твердження 4.1. Композиція Флойда-Хоара для багатосортних програмних алгебр монотонна та неперервна.

Доведення. Доведення проводить аналогічно доведенню для композиції Флойда-Хоара для простих номінативних даних.

Твердження 4.2. Композиція побудови умови за прообразом для багатосортних програмних алгебр монотонна та неперервна.

Доведення. Доведення проводить аналогічно доведенню для композиції побудови умови за прообразом для простих номінативних даних.

4.2 Мова багатосортної композиційно-номінативної програмної логіки

Згідно з семантико-синтаксичним підходом синтаксис програмної логіки природно випливає з семантики, а саме, буде задаватись у вигляді термів багатосортної програмної алгебри. На відміну від програмної логіки над простими даними, для кожного терму та імені змінної необхідно задавати сорт, якому в семантиці буде відповідати певний тип.

Нехай S – множина сортів, та $\xi : V \xrightarrow{t} S$ – відображення, що кожному імені змінної співставляє у відповідність сорт змінної. Тоді будемо $\Sigma^S = (V, S, \xi)$ називати сигнатурою подання сортів.

Нехай задано множини предикатних символів Ps, множини функціональних символів Fs та множини програмних символів Prs. Також для функціональних символів має бути задано сорт їх результату за допомогою відображення $\zeta : Fs \xrightarrow{t} S$. Тоді будемо $\Sigma^L = (\Sigma^S, Ps, Fs, Prs, \zeta)$ називати сигнатурою мови багатосортної програмної логіки. Надамо індуктивні визначення для термів $Tr(\Sigma^L)$, формул $Fr(\Sigma^L)$, текстів програм $Pt(\Sigma^L)$, та асерцій Флойда-Хоара $FHFr(\Sigma^L)$. Символи композицій тут використовуються в синтаксичному сенсі. Також введемо додатково відображення $\psi : Tr(\Sigma^L) \xrightarrow{t} S$, що кожному терму приписує його сорт.

Терми $Tr(\Sigma^L)$ будемо визначати наступним чином:

– якщо $f \in Fs$, то $f \in Tr(\Sigma^L)$ та $\psi(f) = \zeta(f)$;

– якщо $x \in V$, то $`x \in Tr(\Sigma^L)$ та $\psi(`x) = \xi(x)$;

– якщо $f \in Fs$, $t_1, t_2, \ldots, t_n \in Tr(\Sigma^L)$ та $x_1, x_2, \ldots, x_n \in V$, де $x_1, x_2, \ldots, x_n$ - різні імена змінних, та $n > 0$, і сорти термів співпадають з сортами відповідних змінних, тобто $\psi(t_1) = \xi(x_1), \ldots, \psi(t_n) = \xi(x_n)$, то $S_F^{x_1, x_2, \ldots, x_n}(f, t_1, t_2, \ldots, t_n) \in Tr(\Sigma^L)$ та $\psi(S_F^{x_1, x_2, \ldots, x_n}(f, t_1, t_2, \ldots, t_n)) = \zeta(f)$.

Формули $Fr(\Sigma^L)$ будемо визначати наступним чином:

- якщо $p \in Ps$, то $p \in Fr(\sum^L)$;
- якщо $\Phi \in Fr(\sum^L)$, то $\neg\Phi \in Fr(\sum^L)$;
- якщо $\Phi \in Fr(\sum^L)$ та $x \in V$, то $\exists x\Phi \in Fr(\sum^L)$;
- якщо $\Phi, \Psi \in Fr(\sum^L)$, то $\Phi \vee \Psi \in Fr(\sum^L)$;
- якщо $p \in Ps$, $t_1, t_2, \ldots, t_n \in Tr(\sum^L)$ та $x_1, x_2, \ldots, x_n \in V$, де $x_1, x_2, \ldots, x_n$ – різні імена змінних, та $n > 0$, $n > 0$, і сорти термів співпадають з сортами відповідних змінних, тобто $\psi(t_1) = \xi(x_1), \ldots, \psi(t_n) = \xi(x_n)$, то $S_P^{x_1, x_2, \ldots, x_n}(p, t_1, t_2, \ldots, t_n) \in Fr(Ps, Fs, Prs, V)$;
- якщо $q \in Fr(\sum^L)$ разом з $pr \in Pt(\sum^L)$, то відповідно $PC(pr, q) \in Fr(\sum^L)$.

Тексти програм $Pt(\sum^L)$ будемо визначати наступним чином:

- якщо $pr \in Prs$, то $pr \in Pt(\sum^L)$;
- $id \in Pt(\sum^L)$;
- якщо $f \in Tr(\sum^L)$, $x \in V$ та $\psi(f) = \xi(x)$, то $AS^x(f) \in Pt(Ps, Fs, Prs, V)$;
- якщо $pr_1, pr_2 \in Pt(\sum^L)$, то $pr_1 \bullet pr_2 \in Pt(\sum^L)$;
- якщо $pr_1, pr_2 \in Pt(\sum^L)$ та $r \in Fr(\sum^L)$, то $IF(r, pr_1, pr_2) \in Pt(\sum^L)$;
- якщо $pr \in Pt(\sum^L)$ та $r \in Fr(\sum^L)$, то $WH(r, pr) \in Pt(\sum^L)$.

Асерції Флойда-Хоара $FHFr(\sum^L)$ будемо визначати наступним чином:

- якщо $pr \in Pt(\sum^L)$ та $p, q \in Fr(\sum^L)$, то $\{p\}pr\{q\} \in FHFr(\sum^L)$.

Синтаксис та семантика програмної логіки композиційно-номінативного типу задано, залишається лише задати інтерпретації, або відображення, що термам, формулам та текстам програм будуть відповідно ставити у відповідність предикати та функції програмної алгебри. Дані відображення будуються природнім чином, якщо задані відображення, інтерпретації для предикатних, функціональних та програмних символів. Також важливою є інтерпретація для сортів.

Будемо вважати, що задано тотальні відображення $I_{Ps}:Ps \xrightarrow{t} Pr_T^{V,T,\tau}$ – інтерпретації для предикатних символів, $I_{Fs}:Fs \xrightarrow{t} \bigcup\limits_{A\in T} Fn_T^{A,V,T,\tau}$ – інтерпретації для функціональних символів, $I_{Prs}:Prs \xrightarrow{t} FPrg_T^{V,T,\tau}$ – інтерпретації для програмних символів $I_S:S \xrightarrow{t} T$ – інтерпретації для сортів. Так як задано інтерпретацію для сортів, відображення, що імені спів ставляє сорт ,можна задати відображення для типів $\tau = I_S \circ \xi$. Задамо розширення даних відображень на формули $J_{Fr}:Fr(\Sigma^L) \xrightarrow{t} Pr_T^{V,T,\tau}$, терми $J_{Tr}:Tr(\Sigma^L) \xrightarrow{t} \bigcup\limits_{A\in T} Fn_T^{A,V,T,\tau}$, тексти програм $J_{Pt}:Pt(\Sigma^L) \xrightarrow{t} FPrg_T^{V,T,\tau}$ та асерції Флойда-Хоара $J_{FHFr}:FHFr(\Sigma^L) \xrightarrow{t} Pr_T^{V,T,\tau}$ індуктивно.

Відображення $J_{Tr}:Tr(\Sigma^L) \xrightarrow{t} \bigcup\limits_{A\in T} Fn_T^{A,V,T,\tau}$:

- $J_{Tr}(f) = I_{Fs}(f)$, якщо $f \in Fs$;
- $J_{Tr}(`x) = `x$;
- $J_{Tr}(S_F^{x_1,x_2,\ldots,x_n}(f,t_1,t_2,\ldots,t_n)) = S_F^{x_1,x_2,\ldots,x_n}(J_{Tr}(f), J_{Tr}(t_1), J_{Tr}(t_2),\ldots, J_{Tr}(t_n))$.

Відображення $J_{Fr}:Fr(\Sigma^L) \xrightarrow{t} Pr_T^{V,T,\tau}$:

- $J_{Fr}(p) = I_{Ps}(p)$, якщо $p \in Ps$;
- $J_{Fr}(\neg\Phi) = \neg J_{Fr}(\Phi)$;
- $J_{Fr}(\exists x\Phi) = \exists x J_{Fr}(\Phi)$;
- $J_{Fr}(\Phi \vee \Psi) = J_{Fr}(\Phi) \vee J_{Fr}(\Psi)$;
- $J_{Fr}(S_P^{x_1,x_2,\ldots,x_n}(p,t_1,t_2,\ldots,t_n)) = S_P^{x_1,x_2,\ldots,x_n}(J_{Fr}(f), J_{Tr}(t_1), J_{Tr}(t_2),\ldots, J_{Tr}(t_n))$;
- $J_{Fr}(PC(pr,q)) = PC(J_{Pt}(pr), J_{Fr}(q))$.

Відображення $J_{Pt}:Pt(\Sigma^L) \xrightarrow{t} FPrg_T^{V,T,\tau}$:

- $J_{Pt}(pr) = I_{Prs}(pr)$, якщо $pr \in Prs$;
- $J_{Pt}(id) = id$;
- $J_{Pt}(AS^x(f)) = AS^x(J_{Tr}(f))$;

– $J_{Pt}(pr_1 \bullet pr_2) = J_{Pt}(pr_1) \bullet J_{Pt}(pr_2)$;

– $J_{Pt}(IF(r, pr_1, pr_2)) = IF(J_{Fr}(r), J_{Pt}(pr_1), J_{Pt}(pr_2)))$;

– $J_{Pt}(WH(r, pr)) = WH(J_{Fr}(r), J_{Pt}(pr))$.

Відображення $J_{FHFr} : FHFr(\sum^L) \xrightarrow{t} Pr_T^{V,T,\tau}$:

– $J_{FHFr}(\{p\}pr\{q\}) = FH(J_{Fr}(p), J_{Pt}(pr), J_{Fr}(q))$.

Таким чином композиції програмної алгебри $TQPA(V,T,\tau)$ та відображення I_{Ps}, I_{Fs}, I_S та I_{Prs} дозволяють однозначно задати інтерпретацію J, відображення, що кожному елементу мови програмної логіки композиційно-номінативного типу для простих даних зіставляє відображення програмної алгебри.

4.3 Аксіоматичні системи для багатосортної композиційно-номінативної програмної логіки

Розглянемо аксіоматичні системи для багатосортної композиційно-номінативної програмної логіки, побудовані використовуючи ідеї аксіоматичних систем для простих номінативних даних. Необхідно ввести бінарні відношення TC_T, FC_T та LC_T над формулами $Fr(\sum^L)$, це дозволить розглянути аналоги аксіоматичних систем CI, AC, TI, FD, TF, які позначимо відповідно CI_T, AC_T, TI_T, FD_T, TF_T. Для таких аксіоматичних систем потрібно дослідити питання коректності та повноти. Розглянемо спочатку аксіоматичну систему CI_T.

$$R\_AS_T \quad \frac{}{\{S_P^x(p, f)\}AS^x(f)\{p\}},$$

$$R\_SKIP_T \quad \frac{}{\{p\}id\{p\}},$$

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

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

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

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

Твердження 4.3. Аксіоматична система CI_T не є коректною.

Доведення. Правила R_SEQ_T , R_WH_T та R_CONS_T як і відповідні правила системи CI виводять хибні асерції з істинних. Це можна перевірити розглянувши для правила R_SEQ_T , наприклад, p як формулу, що задає всюди істинний предикат, q як формулу, що задає всюди істинний предикат, а r як формулу, що задає всюди хибний предикат, а як програми – тотожну програму.

Аксіоматична система AC_T отримується з CI_T заміною правил R_SEQ_T , R_WH_T та R_CONS_T на правила $R\_SEQ'_T$, $R\_WH'_T$ та $R\_CONS'_T$.

$$R\_AS_T \quad \frac{}{\{S_P^x(p, f)\}AS^x(f)\{p\}},$$

$$R\_SKIP_T \quad \frac{}{\{p\}id\{p\}},$$

$$R\_SEQ'_T \quad \frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1 \bullet pr_2\{r\}}, LC_T(p, PC(pr_1 \bullet pr_2, r)),$$

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

$$R\_WH'_T \quad \frac{\{r \wedge p\}pr\{p\}}{\{p\}WH(r, pr)\{\neg r \wedge p\}}, LC_T(p, PC(WH(r, pr), \neg r \wedge p)),$$

$$R\_CONS'_T \quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}}, TC_T(p, p'), FC_T(q', q).$$

Теорема 4.1. Якщо відношення LC_T співпадає з відношенням логічного наслідку, відношення TC_T співпадає з відношенням логічного наслідку за істиною та відношення FC_T співпадає з відношенням логічного наслідку за хибою, аксіоматична система AC_T коректна та екстенсіонально повна.

Доведення. Коректність аксіоматичної системи доводиться індукцією за довжиною виводу. Варто зазначити, що коректність правил доводиться аналогічно доведенню коректності правил аксіоматичної системи AC. Розгляд типізованих номінативних даних є суттєвим лише для правила R_AS_T. Проте, як і у випадку простих номінативних даних отримуємо $S_P^x(p,f)(d) = p(d\nabla[x \mapsto f(d)])$ та $AS^x(f)(d) = d\nabla[x \mapsto f(d)]$. Відповідність типів змінної x та результату функції f гарантується з визначення елементів мови логіки.

Повнота аксіоматичної системи випливає з того, що для довільних програмного тексту pr та формули q виконується $\vdash \{PC(pr,q)\}pr\{q\}$. Дане твердження доводиться за будовою pr аналогічно до випадку простих номінативних даних. Властивості правила R_AS_T використані при доведенні коректності показують, що доведення для даного правила може бути проведено аналогічно до правила R_AS в AC.

Як і у випадку простих номінативних даних, легко показати, що, якщо $\models \{p\}pr\{q\}$, то вивід $\vdash \{p\}pr\{q\}$ можна звести до виводу $\vdash \{PC(pr,q)\}pr\{q\}$. А отже аксіоматична система AC_T є екстенсіонально повною.

Властивості правил виводу для системи CI також мають місце і у випадку CI_T, що дозволяє побудувати аксіоматичні системи TI_T та FD_T, що можуть бути отримані з CI_T заміною правила R_CONS_T на правило R_CONS_T'', та правило R_CONS_T''' відповідно.

$$R\_AS_T \quad \frac{}{\{S_P^x(p,f)\}AS^x(f)\{p\}},$$

$$R\_SKIP_T \quad \frac{}{\{p\}id\{p\}},$$

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

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

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

$$R\_CONS''_T \quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}}, TC_T(p,p'), TC_T(q',q).$$

Та

$$R\_AS_T \quad \frac{}{\{S_P^x(p,f)\}AS^x(f)\{p\}},$$

$$R\_SKIP_T \quad \frac{}{\{p\}id\{p\}},$$

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

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

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

$$R\_CONS'''_T \quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}}, FC_T(p,p'), FC_T(q',q).$$

Теорема 4.2. Якщо відношення TC_T співпадає з відношенням логічного наслідку за істиною, аксіоматична система TI_T коректна та екстенсіонально повна на класі асерцій $\{p\}pr\{q\}$ для яких виконується $p \models_T PC(pr,q)$.

Доведення. Доведемо коректність, показавши, що якщо $\vdash \{p\}pr\{q\}$, то виконується для довільної інтерпретації J $p_J^{\ T,pr} \subseteq q_J^{\ T}$. Для цього покажемо, що дане співвідношення виконується для висновків правил R_AS_T та R_SKIP_T, а також зберігається іншими правилами. Для всіх правил, крім R_AS_T, це можна показати аналогічно до аксіоматичної системи TI.

Розглянемо правило R_AS_T. Потрібно показати, що для довільної інтерпретації J виконується $S_P^x(p,f)_J^{\ T,AS^x(f)} \subseteq p_J^{\ T}$. Нехай $d \in S_P^x(p,f)_J^{\ T,AS^x(f)}$ за визначенням це означає, що для деякого $d' \in S_P^x(p,f)_J^{\ T}$ виконується $d = AS^x(f)_J(d')$. Маємо $S_P^x(p,f)_J(d') \downarrow = T$ і з визначення композиції суперпозиції

$p_J(d'\nabla[x \mapsto f_J(d')])\downarrow = T$. В свою чергу з визначення композиції присвоєння $d = AS^x(f)_J(d') = d'\nabla[x \mapsto f_J(d')]$, а отже $p_J(d)\downarrow = T$ і $d \in {p_J}^T$, що і треба було довести. Всі правила було розглянуто, отже твердження виконується, а з визначення композиції Флойда-Хоара отримуємо, що в такому разі $\models \{p\}pr\{q\}$.

Екстенсіональна повнота аксіоматичної системи TI_T на відповідному класі асерцій доводиться аналогічно доведенню відповідного твердження для системи TI. Спочатку доводиться, що виводиться асерція $\{PC(pr,q)\}pr\{q\}$ для довільних програмного тексту pr та формули q. Звідки з $p \models_T PC(pr,q)$ маємо, що за правилом R_CONS_T'' можна вивести $\{p\}pr\{q\}$.

Теорема 4.3. Якщо відношення FC_T співпадає з відношенням логічного наслідку за хибою, аксіоматична система FD_T коректна та не є екстенсіонально повною на класі асерцій $\{p\}pr\{q\}$ для яких виконується $p \models_F PC(pr,q)$.

Доведення. Доведемо коректність, показавши, що для довільної асерції $\{p\}pr\{q\}$, якщо вона виводиться в FC_T, то виконується $p \models_F PC(pr,q)$. Як і в для попередніх аксіоматичних систем, варто показати, що потрібне твердження виконується для правила R_AS_T, решта правил розглядається аналогічно до аксіоматичної системи FD.

Розглянемо правило R_AS_T. Потрібно показати, що виконується $S_P^x(p,f) \models_F PC(AS^x(f),p)$. В свою чергу для довільної інтерпретації J $S_P^x(p,f)_J(d) = p_J(d\nabla[x \mapsto f_J(d)])$, а $PC(AS^x(f),p)_J(d)\downarrow = F$, якщо має місце $p_J(AS^x(f)_J(d))\downarrow = F$, або з визначення композиції присвоєння $p_J(d\nabla[x \mapsto f_J(d)])\downarrow = F$, так як інтерпретація була обрана довільним чином, це доводить $S_P^x(p,f) \models_F PC(AS^x(f),p)$. Всі правила було розглянуто, отже твердження виконується, а з визначення композиції Флойда-Хоара отримуємо, що в такому разі $\models \{p\}pr\{q\}$.

Для того, щоб показати, що аксіоматична система FD_T не є екстенсіонально повною на класі асерцій $\{p\}pr\{q\}$ для яких виконується $p \models_F PC(pr,q)$, потрібно розглянути асерцію $\{p\}IF(r,id,id)\{q\}$, де p та r – формули такі, що їм в довільній інтерпретації відповідає всюди невизначений предикат, а формулі q відповідає всюди хибний предикат. За визначенням така асерція буде всюди істиною та для неї виконується $p \models_F PC(IF(r,id,id,q)$, проте неважко перевірити, що вона не може бути виведена в аксіоматичній системі FD_T. Таким чином, аксіоматична система не є коректною для заданого класу асерцій.

Як і у випадку простих номінативних даних, аксіоматичні системи з простими обмеженнями, що базуються на властивостях асерцій, що виводяться є коректними, проте також не позбавлені основного недоліку аксіоматичних систем TI, FD і не є екстенсіонально повними в загальному. Це зумовлює потребу в розгляді аксіоматичних систем зі складнішими додатковими умовами, зокрема системи TF_T, яка може бути одержана з CI_T по аналогії з відповідними аксіоматичними системами для простих номінативних даних. Для цього необхідно замінити правило R_SEQ_T на правила $R\_SEQ\_T_T$ та $R\_SEQ\_F_T$; правило R_WH_T на правила $R\_WH\_T_T$ та $R\_WH\_F_T$; а правило R_CONS_T замінити на $R\_CONS'_T$.

$$R\_AS_T \quad \frac{}{\{S_P^x(p,f)\}AS^x(f)\{p\}},$$

$$R\_SKIP_T \quad \frac{}{\{p\}id\{p\}},$$

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

$$R\_CONS'_T \quad \frac{\{p'\}pr\{q'\}}{\{p\}pr\{q\}},TC_T(p,p'),FC_T(q',q).$$

$$R\_SEQ\_T_T \quad \frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1 \bullet pr_2\{r\}},TC_T(p,PC(pr_1,q)),$$

$$R\_SEQ\_F_T \quad \frac{\{p\}pr_1\{q\},\{q\}pr_2\{r\}}{\{p\}pr_1 \bullet pr_2\{r\}},FC_T(q,PC(pr_2,r)),$$

$$R\_WH\_T_T \quad \frac{\{r \wedge p\}pr\{p\}}{\{p\}WH(r, pr)\{\neg r \wedge p\}}, TC_T(r \wedge p, PC(pr, p)),$$

$$R\_WH\_F_T \quad \frac{\{r \wedge p\}pr\{p\}}{\{p\}WH(r, pr)\{\neg r \wedge p\}}, FC_T(r \wedge p, PC(pr, p)).$$

Теорема 4.4. Якщо відношення TC_T співпадає з відношенням логічного наслідку за істиною та відношення FC_T співпадає з відношенням логічного наслідку за хибою, аксіоматична система TF_T коректна та екстенсіонально повна.

Доведення. Доведення коректності та екстенсіональної повноти аксіоматична система TF_T проводиться аналогічно до доведення відповідних тверджень для аксіоматичної системи *TF*. Варто зазначити, що відмінності можуть бути лише в правилах для яких суттєва наявність типів, це правило R_AS_T, проте в доведенні Теореми 4.1. було показано, що для даного правила виконуються всі ті властивості, що і для правила R_AS.

Висновки.

Сучасні мови програмування мають вбудовану систему типів тієї чи іншої складності. Помилки, що виникають в програмних системах і пов'язані з проблемою невідповідності значень змінних і їх типу складно визначити, особливо за динамічної типізації. Тому проблема формалізації типізованих даних, понять строгої, динамічної типізації в мовах програмування є актуальною для побудови коректного програмного забезпечення.

В даному розділі розглядаються композиційно-номінативні програмні логіки над типізованими номінативними даними, або ж багатосортні програмні логіки. Приводиться визначення типізованої номінативної множини, яка отримується з простої шляхом приписування змінним їх типів, під типом розуміється множина всіх можливих значень змінної. Для типізованих номінативних множин визначаються основні операції над ними.

Побудовано багатосортні програмні алгебри. Слід відзначити, що, на відміну від програмних алгебр над простими номінативними даними, ординарні квазіарні функції мають тип і повертають значення фіксованого типу та відповідність типів гарантується визначенням операції суперпозиції, присвоєння. У випадку типізованих номінативних даних композиції Флойда-Хоара та побудови умови за прообразом також є монотонними та неперервними.

Хоча синтаксичний аспект багатосортної програмної логіки отримано природнім чином з семантичного, як терми відповідної програмної алгебри, для термів вводиться поняття сорту та відображення що зіставляє терму його сорт, показано як з відображення для функціональних символів та визначення композицій можна отримати відображення для термів.

Побудовано аксіоматичні системи CI_T, AC_T, TI_T, FD_T, TF_T, що є аналогами аксіоматичних систем CI, AC, TI, FD, TF для багатосортної програмної логіки. Для таких аксіоматичних систем доведено теореми про коректність і екстенсіональну повноту. Показано, що властивості співпадають з випадком простих номінативних даних.

ВИСНОВКИ

Основним результатом дисертації є дослідження програмних логік композиційно-номінативного типу, що розв’язує завдання побудови коректних та повних аксіоматичних систем відповідних програмних логік, і має суттєве значення для розвитку формальних методів розробки програмного забезпечення обчислювальних машин і систем. Зокрема:

1. Доведено монотонність та неперервність композицій для побудованих композиційно-номінативних програмних алгебр часткових функцій та предикатів над простими номінативними даними, наведено еквівалентні перетворення для них.
2. Визначено синтаксичний та інтерпретаційний аспект програмних логік композиційно-номінативного типу, що базуються на побудованих програмних алгебрах.
3. Доведено, що і у випадках ієрархічних номінативних даних та типізованих номінативних даних композиції програмних алгебр будуть монотонними та неперервними. Побудовано програмні логіки складних номінативних даних, семантика яких задається за допомогою відповідних програмних алгебр
4. Для програмних логік доведено, що класична аксіоматична система, запропонована для мови *WHILE*, не є коректною.
5. Розроблено аксіоматичну систему з доданими обмеженнями, що є коректною та екстенсіонально повною. Показано, що аналогічна аксіоматична система може бути побудована як для програмних логік над ієрархічними номінативними даними, так і для програмних логік над типізованими номінативними даними.
6. Побудовано аксіоматичну систему зі спрощеними обмеженнями в якій виводяться лише твердження для яких виконується умова T-зростання, також побудовано дуальну F-спадну аксіоматичну систему. Доведено їх коректність та показано, що вони не будуть екстенсіонально повними для

всіх тверджень, проте *T*-зростаюча аксіоматична система буде екстенсіонально повною для достатньо широкого класу тверджень.

7. Поєднуючи підходи, використані для побудови коректних аксіоматичних систем за допомогою додаткових обмежень, отримано коректну та екстенсіонально повну систему з додатковими обмеженнями.
8. Доведено теореми про повноту та коректність аксіоматичних систем для випадків складних номінативних даних.

Слід відмітити, що при переході від логіки Флойда-Хоара, в якій розглядались лише тотальні предикати скінченної арності, до програмних логік часткових квазіарних предикатів над номінативними даними, аксіоматична система не є коректною. Це зумовило проблему пошуку шляхів модифікації аксіоматичної системи, щоб вона була коректною. Одним з таких шляхів є визначення додаткових умов для правил. В роботі продемонстровано, що існує також багато підходів до введення додаткових умов. Таким чином було отримано різні коректні аксіоматичні системи. Проте деякі з них були надлишковими, а деякі не були повними.

Варто зазначити, що для кожної аксіоматичної системи програмної логіки композиційно-номінативного типу над простими номінативними даними, аналоги, побудовані для програмних логік над складнішими номінативними даними, зберігали властивості бути коректними та екстенсіонально повними.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Abrial J. –R. The B-book: Assigning Programs to Meanings / J. -R. Abrial // New York, USA, Cambridge University Press, 1996, P. 816.
2. Apt. K. R. Ten years of Hoare's logic: a survey—part I / K. R. Apt // ACM Trans. Programming Languages and Systems, Vol. 3, 1981, P. 431–483.
3. Ben-Ari M. Deterministic propositional dynamic logic: finite models, complexity and completeness / M. Ben-Ari, J. Y. Halpern, A. Pnueli // Journal Computer Systems Science, Vol. 25, 1982, P. 402-417.
4. Berger M. A logical analysis of aliasing in imperative higher-order functions / M. Berger, K. Honda, N. Yoshida // ICFP 2005, P. 280–293.
5. Bergmann M. An Introduction to Many-Valued and Fuzzy Logic: Semantics, Algebras, and Derivation Systems / M. Bergmann // Cambridge, UK, Cambridge University Press, 2008, P. 342.
6. Bjorner D. Formal, model-oriented software development methods: From VDM to ProCoS & from RAISE to LaCoS / D. Bjorner, A. E. Haxthausen, K. Havelund // Future Generation Computer Systems, Vol. 7, 1992, P. 111-138.
7. Blamey S. Partial Logic / S. Blamey // Handbook of Philosophical Logic / eds. D. Gabbay, F. Guenthner, Springer Netherlands, 1986, P. 1–70.
8. Brookes S. A Semantics for Concurrent Separation Logic / S. Brookes // Theoretical Computer Science Vol. 375, 2007, P. 227–270.
9. Burmeister P. A Model Theoretic Oriented Approach to Partial Algebras / P. Burmeister // Akademie-Verlag, 1986.
10. Clarke E. M. Formal methods: State of the art and future directions / E. M. Clarke, J. M. Wing // ACM Computing Surveys (CSUR), 1996, Vol. 28, No. 4, P. 626–643.
11. Clarke E. M. Programming language constructs for which it is impossible to obtain good Hoare axiom systems / E. M. Clarke // Journal of the Association for Computing Machinery, Vol. 26, 1979, P. 129–147.

12. Cook S. A. Soundness and completeness of an axiom system for program verification / S. A. Cook // SIAM Journal on Computing, Vol. 7, 1978, P. 70–90.
13. Dijkstra E.W. A Discipline of Programming / E.W. Dijkstra // Prentice-Hall, Englewwod Cliffs, New Jersey, 1976.
14. Feldman Y. A. A decidable propositional dynamic logic with explicit probabilities / Y. A. Feldman // Information and Control, Vol. 63, 1984, P. 11-38.
15. Feldman Y. A. A probabilistic dynamic logic / Y. A. Feldman, D. Harel // Journal on Computer Systems Sciences, Vol. 28, 1984, P. 193–215.
16. Fischer M. J. Propositional dynamic logic of regular programs / M. J. Fisher, R.E. Ladner // Journal Computer Systems Science, Vol. 18(2), 1979, P. 194-211.
17. Floyd R.W. Assigning Meanings to Programs / R.W. Floyd // Proceedings of the American Mathematical Society Symposia on Applied Mathematics, Vol. 19, 1967, P. 19-31.
18. Harel D. Dynamic logic, Handbook of Philosophical Logic / D. Harel, D. Kozen, J. Tiuryn // 1984 P. 497-604.
19. Harel D. First-Order Dynamic Logic / D. Harel // Lecture Notes in Computer Science, Vol. 68, Springer-Verlag, 1979.
20. Harel D. More on looping vs. repeating in dynamic logic / D. Harel, D. Peleg // Information Processing Letters, Vol. 20, 1985, P. 87–90.
21. Harel D. Nondeterminism in logics of programs / D. Harel, V.R. Pratt // In Proceeding of 5th Symposium on Principles of Programming Languages, ACM, 1978 P. 203-213.
22. Harel D. Propositional dynamic logic of flowcharts / D. Harel, R. Sherman // Information and Control, Vol. 64, 1985 P. 119-135.
23. Hoare C.A.R. An Axiomatic Basis for Computer Programming / C.A.R. Hoare // Communications of the ACM, Issue 12, 1969 P. 576-580,583.
24. Jones C. B. Reasoning About Partial Functions in the Formal Development of Programs / C. B. Jones // Electron. Notes Theor. Comput. Sci, 2006, Vol. 145, P. 3–25.

25. Jones Cliff B. Systematic Software Development Using VDM / C. B. Jones // 2nd ed, Upper Saddle River, USA, Prentice-Hall, 1990, P. 333.
26. Kozen D. An elementary proof of the completeness of PDL / D. Kozen R. Parikh //Theoretical Computer Science, Vol. 14, 1981, P. 113–118.
27. Kozen D. Logics of programs / D. Kozen, J. Tiuryn // Handbook of Theoretical Computer Science, Vol. B, North Holland, Amsterdam, 1990 P. 789-840.
28. Kreowski H.-J. Partial Algebra Flows from Algebraic Specifications / H.-J. Kreowski // 14th Int. Colloquium on Automata, Languages and Programming, Vol. 267 of Lecture Notes in Computer Science, Springer, Berlin, 1987 P. 521-530.
29. Kripke S. Semantical Considerations on Modal Logic / S. Kripke // Acta Philosophica Fennica, 1963, Vol. 16, P. 83–94.
30. Kryvolap A. Program Algebras with monotone Floyd-Hoare Composition / A. Kryvolap, M. Nikitchenko, W. Schreiner // ICTERI 2013, Proceedings of the 9th International Conference on ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer, Kherson, Ukraine, June 19-22, 2013 P. 533-549.
31. Kryvolap A. Extending Floyd-Hoare logic for partial pre- and postconditions / A. Kryvolap, M. Nikitchenko, W. Schreiner // ICTERI 2013, CCIS, Springer, Heidelberg, 2013 P. 355-378.
32. Lamport L. The Temporal Logic of Actions / L. Lamport // ACM Transactions on Programming Languages and Systems, 1993, P. 1–52.
33. Lamport L. Win and sin: Predicate Transformers for Concurrency / L. Lamport // ACM Transactions on Programming Languages and Systems Vol. 12, 1990 P. 396-428.
34. Lukasiewicz J. O logice trojwartosciowej / J. Lukasiewicz // Ruch filozoficzny, 1920, Vol. 5, P. 170–171.
35. Luo C Separation logic for multiple inheritance / C. Luo, S. Qin // Electronic notes in theoretical computer science, Vol. 212, 2008, P. 27-40.

36. Makowsky J. A. Probabilistic propositional dynamic logic / J. A. Makowsky, M. L. Tiomkin // Manuscript, 1980.
37. Martin U. Abstract Hoare logic / U. Martin, E. A. Mathiesen, P. Oliva // CSL'06, LNCS, 2006.
38. Martin U. A General Framework for Sound and Complete Floyd-Hoare Logics / R. Arthan, U. Martin, E. A. Mathiesen, P. Oliva // ACM Transactions on Computational Logic, 2010, P. 1–30.
39. Mehta F. A Practical Approach to Partiality – A Proof Based Approach / F. Mehta // Formal Methods and Software Engineering / eds. S. Liu, T. Maibaum, K. Araki, Springer Berlin Heidelberg, 2008, P. 238–257.
40. Mosses P. D. CASL Reference Manual: The Complete Documentation of the Common Algebraic Specification Language / P.D. Mosses // Lecture Notes in Computer Science, Vol. 2960, Springer, Berlin, 2004.
41. Nielson H.R. Semantics with Applications: A Formal Introduction / H.R. Nielson, F. Nielson // John Wiley & Sons Inc., 1992 P. 240.
42. Nikitchenko N.S. A Composition Nominative Approach to Program Semantics. / N. S. Nikitchenko, Technical Report IT-TR 1998-020, Technical University of Denmark, 1998, P. 103.
43. Nikitchenko M. Inference systems for Floyd-Hoare logic with partial predicates / M. Nikitchenko, A. Kryvolap // Proceedings of the Twelfth International Conference INFORMATICS' 2013, November 5th-7th, Spisska Nova Ves, Slovak Republic, 2013, P. 88-93.
44. Nikitchenko M. Properties of inference systems for Floyd-Hoare logic with partial predicates / M. Nikitchenko, A. Kryvolap // Acta Electrotechnica et Informatica, Vol. 13 No. 4, 2013, pp. 70-78.
45. Nikitchenko M. Satisfiability and Validity Problems in Many-sorted Composition-Nominative Pure Predicate Logics / M. Nikitchenko, V Tymofieiev // ICTERI 2012, CCIS, Springer, Heidelberg, 2013 P. 89-110.

46. Nikitchenko M.S. Satisfiability in Composition-Nominative Logics / M.S. Nikitchenko, V.G Tymofieiev // Central European Journal of Computer Science, Vol. 2, Issue 3, 2012 P. 194-213.
47. O'Hearn P. BI as an assertion language for mutable data structures / S. Ishtiaq, P. O'Hearn // POPL 28th, 2001, P. 14–26.
48. O'Hearn P. Local reasoning about programs that alter data structures / P. O'Hearn, J. Reynolds, H. Yang // CSL'01, LNCS/ L. Fribourg, Ed. Vol. 2142, Springer-Verlag, 2001. P. 1–19.
49. O'Hearn P.W. Resources, Concurrency and Local Reasoning / P.W. O'Hearn // Theoretical Computer Science, Vol. 375, 2007, P. 271–307.
50. O'Hearn P.W. Symbolic Execution with Separation Logic / J. Berdine, C. Calcagno, P. W. O'Hearn // APLAS 2005, LNCS 3780, 2005, P. 52-68.
51. O'Hearn P.W. The logic of bunched implications / P.W. O'Hearn, D. J. Pym // Bulletin of Symbolic Logic, Vol. 5(2), 1999 P. 215–244.
52. Owicki S.S. Verifying properties of parallel programs: An axiomatic approach / S.S. Owicki, D. Gries // Communications of the ACM, Vol. 19(5), 1976 P. 279–285.
53. Peleg D. Concurrent dynamic logic / D. Peleg // Journal Association Computer Machines, Vol. 34(2), 1987, P. 450-479.
54. Pratt V.R. A practical decision method for propositional dynamic logic / V. R. Pratt // Proceedings of 10th Symposium on Theory of Computation, ACM, 1978, P. 326–337.
55. Pratt V. R. Semantical considerations on Floyd-Hoare logic / V.R. Pratt // In Proceedings of 17th Symposium on Foundatioms of Computer Sciences, 1976, P. 109-121.
56. Reichel H. Initial Computability, Algebraic Specifications, and Partial Algebras / H. Reichel // Oxford University Press, 1987.
57. Reynolds J.C. Separation Logic: A logic for Shared Mutable Data Structures / J.C. Reynolds // LICS, 2002 P. 55-74.

58. Rosu G. Matching Logic: An Alternative to Hoare/Floyd Logic / G. Rosu, C. Ellison, W. Schulte // AMAST'10, LNCS 6486, 2010, P. 142-162.
59. Rosu G. Matching Logic: A New Program Verification Approach / G. Rosu, A. Stefanescu // ICSE'11, ACM, 2011, P. 868-871.
60. Rosu G. From Rewriting Logic Executable Semantics to Matching Logic Program Verification / G. Rosu, C. Ellison, W. Schulte // [Електронний ресурс]. – Режим доступу : http://hdl.handle.net/2142/13159 .
61. Sannella D. Foundations of Algebraic Specification and Formal Software Development / D. Sannella, A. Tarlecki // Monographs in Theoretical Computer Science, Springer, 2012.
62. Spivey J. M. The Z Notation: A Reference Manual / J. M. Spivey // 2nd ed, Upper Saddle River, USA, Prentice-Hall, 1989, P. 158.
63. Woodcock J. Formal methods: Practice and experience / J. Woodcock, P. G. Larsen, J. Bicarregui, J. Fitzgerald // ACM Computing Surveys, 2009, Vol. 41, No. 4, P. 19.
64. The Ynot Project homepage [Електронний ресурс]. – Режим доступу : http://ynot.cs.harvard.edu/.
65. Глушков В.М. Алгебра. Языки. Программирование / В.М. Глушков, Г.Е. Цейтлин, Е.Л. Ющенко // Наук. Думка, 1974, С. 331.
66. Криволап А.В. Властивості програмних алгебр часткових функцій та предикатів / А.В. Криволап // Матеріали всеукраїнської науково-практичної конференції «В.М. Глушков – піонер кібернетики», 11 грудня, 2014, Київ, С. 58-60.
67. Криволап А.В. Властивості систем виводу монотонних логік Флойда-Хоара над ієрархічними даними / А.В. Криволап // Праці одинадцятої міжнародної науково-практичної конференції з програмування TAAPSD’2014, 15 – 17 грудня 2014 року, Київ, С. 155-158.
68. Криволап А.В. Многосортная монотонная логика Флойда-Хоара / А.В. Криволап, Н.С. Никитченко // International Journal: Information Theories and Applications №4(20), 2013, С. 331-341.

69. Криволап А.В. Питання повноти систем виводу монотонних логік Флойда-Хоара / А.В. Криволап // XVI International Conference Dynamical System Modelling and Stability Investigation, Abstracts of the conference reports, Kiev, May 29-31, 2013, С.403.
70. Криволап А.В. Семантические свойства монотонных логик Флойда-Хоара / А.В. Криволап, Н.С. Никитченко // Международные конференции KDS плюс MeL, Киев,10-14 сентября 2012 года, труды конференцій, С. 32-33.
71. Криволап А.В. Система виводу з T- та F-обмеженнями для монотонної логіки Флойда-Хоара / А.В. Криволап // Вісник Київського національного університету імені Тараса Шевченка, Серія Фізико-математичні науки, випуск №4, 2014, С. 211-218.
72. Никитченко Н. С. Композиционно-номинативные аспекты адресного программирования / Н. С. Никитченко // Кибернетика и системный анализ, 2009, № 6, С. 24–35.
73. Нікітченко М. С. Композиційно-номінативні логіки над ієрархічними даними / М. С. Нікітченко, С. С. Шкільняк // Проблеми програмування, 2010, № 2-3, С. 48–57.
74. Никитченко Н.С. Композиционно-номинативный подход к уточнению понятия программы/ Н.С. Никитченко // Проблемы программирования, 1999, № 1, С. 16–31.
75. Нікітченко М.С. Композиція побудови умови за прообразом у монотонних логіках Флойда-Хоара / М.С. Нікітченко, А.В. Криволап // Вісник Київського національного університету імені Тараса Шевченка, Серія Фізико-математичні науки, випуск №3, 2014, С. 166-171.
76. Нікітченко М.С. Математична логіка та теорія алгоритмів / М.С. Нікітченко, С.С. Шкільняк // Видавництво Київського національного університету імені Тараса Шевченка, Київ, 2008, С. 528.
77. Нікітченко М.С. Проблема спростовності у монотонних логіках Флойда-Хоара / М.С. Нікітченко, А.В. Криволап // Праці дев’ятої міжнародної

науково-практичної конференції з програмування TAAPSD'2013, 25 травня – 2 червня 2013 року, Ялта, С. 97-99.

78. Нікітченко М.С. Семантичні властивості монотонних логік Флойда-Хоара / М.С. Нікітченко, А.В. Криволап // Вісник Київського національного університету імені Тараса Шевченка, Серія Фізико-математичні науки, випуск №3, 2012, С. 215-222.
79. Нікітченко М.С. Теорія програмування. Частина 1 / М.С. Нікітченко // Ніжин, Видавництво НДУ імені Миколи Гоголя, 2010, С. 129.
80. Шкільняк С.С. Спектр логік часткових предикатів, орієнтованих на композиційно-номінативні моделі програм : автореф. дис. на здобуття наук. ступеня д-ра фіз.-мат. наук : спец. 01.05.01 "Теоретичні основи інформатики та кібернетики" / Шкільняк Степан Степанович // Київ. нац. ун-т ім. Тараса Шевченка, 2010, С. 32.