

МЕХАНІЗМИ ВКЛЮЧЕННЯ ЗАСОБІВ ФОРМАЛЬНИХ СПЕЦИФІКАЦІЙ ДО БАЗ ЗНАНЬ КОМП'ЮТЕРНИХ МОВ

Розв'язання задач підвищення рівня автоматизації сучасних систем програмування і верифікації розроблених програмних засобів вимагає збереження специфікацій типів, проблемних галузей, програмних об'єктів та критеріїв ефективності реалізацій програм. Для реалізації задач автоматизованого визначення типів та формального синтезу, верифікації і формування виконавчих кодів пропонується відображення різних потрібних елементів специфікацій комп'ютерних мов в різних форматах графів подання мов та сучасних засобів мультимедіа. Використання таких представлень забезпечує гранично ефективне звертання до баз знань попередніх рішень при доказовій побудови нових і вдосконалених програм і моделей.

Rising of level for the complex task decision in hardware/software co-design systems require storage of types, restriction, relations and affectivity criterion specifications. For implementation of automated type definition task during formal verification and code, there are proposed using of different element of computer and multimedia languages or specifications. It is shown, that application of such representations permit to organize effective processing of CAD knowledge bases and their using for proved synthesis of advanced programs and models.

Вступ

Системи автоматизованого програмування і моделювання (САПМ) [1] для комп'ютерного розв'язання задач при визначенні типів даних спираються на табличну організацію вбудованих структур даних, їх примірників та методів обробки. Таблиці САПМ зберігають характеристики типів аргументів та результатів програмних і апаратних ресурсів розв'язання задач (РРЗ) в сучасних засобах їх оптимізації, та формального синтезу і верифікації [3, 8, 9]. Базовим підходом до доказової автоматизації проектування ефективних РРЗ стало використання баз знань (БЗ) семантичного супроводження [3, 7] моделей з використанням реляційних сховищ даних в складі САПМ РРЗ.

Фактично мови програмування і моделювання мають явні структуровані специфікації типів користувача та даних та неявні, визначені стандартами для семантичної частини мови специфікації вбудованих типів. Більш потужні засобами формальної специфікації РРЗ включено до таких мов, як Z і VDM [10], що базуються на поширених математичних позначеннях і часто визначають задачі в таких спеціальних галузях математики, як теорії множин, чисел та типів, реляційна алгебра, тощо. Найближча перспектива використання описів формальних специфікацій і процесів їх перетворення на коди програм полягає у автоматизованому розв'язанні ряду задач:

- **верифікації програм і моделей** нормальними методами перевірки коректності шля-

хом взаємних перетворень специфікацій та кодів;

- **вибірки оптимальних програм і моделей алгоритмів** за критеріями, заданими спеціальними конструкціями;
- **частково автоматизованого синтезу моделей і програм** перетвореннями зв'язків і обмежень формальних специфікацій;
- **пошуку реалізацій програм і моделей алгоритмів** в базах БЗ САПМ за образами специфікацій різних рівнів;
- **формування і накопичення показників складності та ефективності** в критеріях оптимального розв'язання задач;
- **впорядковано зберігання примірників реалізації РРЗ** за ключовими елементами специфікацій;
- **автоматизованої ревізії та формування протоколів доведення** відповідності програм формальним специфікаціям та доказового синтезу програм.

Головну проблему використання цих засобів складає обмеженість реалізованих рівнів комп'ютерної обробки цих специфікацій, які звичайно обмежуються в автоматичному режимі лише їх перевіркою на суперечливість і дозволяють ручні адекватні перетворення для розв'язання суміжних задач [8]. Іншу проблему складає необхідність практично необмеженого розширення можливих зв'язків і описів специфікацій і кодів при розв'язанні задач в розширюваних предметних галузях досліджень (ПГД). Для цього треба забезпечити специфікацію окремих структурних одиниць фрагмента про-

грами, як змінних або об'єктів (підпрограми або міні-модуля), так і всієї предметної області у вигляді батьківського макромодуля.

В специфікаціях, включених до модулів на комп'ютерних мовах, треба розділити визначення типів, що відповідають математичним координатам, від специфікацій зв'язків, цілей, обмежень та обчислювальних дій.

Пошук відповідності між цільовими специфікаціями та реалізаціями програм повинен

спиратися на ключі, на яких залежно від задачі проектування (синтезу, верифікації або оптимізації) використовують різні відношення порядку. Технології автоматизованого проектування та оптимізації програм передбачають збереження специфікацій програм та їх шаблонів в БЗ [3, 8] і доступ до них за ключами, типи яких наведені в табл. 1.

Табл. 1. Типи узагальнених ключових елементів управління в БЗ САПМ

<i>Мови</i> <i>Мовні об'єкти</i>	<i>Математичні мови специфікацій моделей</i>	<i>Проектування програм та обладнання</i>	<i>Моделювання за послідовностями подій</i>	<i>Визначення режимів проектування</i>	<i>Комп'ютерні мови специфікацій задач</i>	<i>Текстові описи моделей в документах</i>
<i>Вхідні дані та аргументи</i>	Списки типізованих імен в структурах	Списки типізованих імен в структурах	Списки імен даних управління	Параметри джерел даних і оптимізації	Списки імен типізованих структур	Текстові описи змісту вхідів
<i>Проміжні дані</i>	Імена типізованих примірників і агрегатів	Імена примірників і агрегатів	Імена примірників і агрегатів	Параметри примірників і агрегатів	Імена примірників і агрегатів	Текстові описи примірників і агрегатів
<i>Результати</i>	Списки типізованих імен	Списки типізованих імен	Списки імен впливів	Імена приймачів даних	Списки типізованих імен	Текстові описи виходів
<i>Операції</i>	Вирази прототипів зв'язків та обмежень	Вирази прототипів заголовків	Прототипи заголовків і умов запуску	Прототипи заголовків та шаблонів	Прототипи заголовків та шаблонів	Текстові описи позначень і аргументів
<i>Функції</i>	Вирази прототипів заголовків і зв'язків	Вирази прототипів заголовків	-	Прототипи заголовків та шаблонів	Прототипи заголовків та шаблонів	Текстові описи імен і аргументів
<i>Підпрограми</i>	Прототипи заголовків, зв'язків і обмежень	Прототипи заголовків і зв'язків	Прототипи заголовків і умов запуску	Прототипи заголовків та шаблонів	Прототипи заголовків та шаблонів	Текстові описи імен і аргументів
<i>Оператори мови</i>	Прототипи заголовків, зв'язків і обмежень	Передумови Післяумови	Прототипи заголовків або шаблонів	Прототипи заголовків або шаблонів	Прототипи заголовків або шаблонів	-

Проблема реалізації змісту комп'ютерних мов полягає в уніфікації дій семантичної обробки комп'ютерних і природних мов для даних основних типів, включаючи безпосередні обчислення, пояснення та адекватні взаємні перетворення. Така уніфікація семантики або змісту дозволяє визначити практично всі комп'ютерні мови і проторувати шлях до створення загальної реалізації обробки семантичної частин всіх комп'ютерних мов.

Ще одна проблема САПМ пов'язана з аналізом і відтворенням образів синтаксису всіх розділів кодів і специфікацій в текстовій, символічній (ієрогліфічній) або структуровано-графічній формі [5]. Вона пов'язана з необхідністю відображення інформації про модулі перед користувачами САПМ різних рівнів і розв'я-

зується на основі використання форм уніфікованого семантичного кодування елементів програм і моделей [6].

Формалізація постановки задачі

Неформальною метою роботи є визначення механізмів доступу до БЗ САПМ при проектуванні РРЗ різних рівнів організації і складності. Для цього потрібно визначити механізми побудови ключів доступу до змістовних та виконавчих характеристик РРЗ через вбудовані відношення порядку ключових елементів. Найпростіші відношення порядку утворюються узагальненими синтаксичними або морфологічними позначеннями назв задач і типів зі значеннями їх аргументів і результатів. Синтакси-

чні ознаки представлень РРЗ на різних рівнях моделей задаються стандартними позначеннями. До них відносяться стандарти символічних імен конструкцій та стандартних підпрограм і функцій РРЗ та їх аргументи у стандартному порядку. Звичайно ці позначення відповідають стандартним математичним позначенням, що дозволяє прискорювати пошук користувачем потрібних РРЗ в БЗ.

Однак впорядкування за синтаксисом лексикографічних або мультимедійних символів не дозволяє систематизувати та згрупувати разом РРЗ однієї ПГД для локального співставлення споріднених рішень. Більш ґрунтовну система-

тизацію РРЗ в БЗ, що сприяє доведенню їх відповідності специфікаціям, можна організувати на основі використання змістовних характеристик елементів ПГД як ключів в сховищах зберігання. Змістовні або семантичні характеристики РРЗ в назвах значень та їх типів у прив'язці до ПГД, де вони визначені. За аналогією з семантичним кодуванням текстів [6] семантичні характеристики іменованих або ідентифікованих позначень включають кодові поля ідентифікації відповідного рівня ієрархічних ПГД, показані в табл. 2, та коди деталізації полів ролі та стану відповідного елемента семантики іменування даних методів та об'єктів.

Табл. 2. Рівні представлення семантичних ознак в ключових іменах БЗ САПМ

<i>Кодування Ключові елементи</i>	<i>Уніфіковані представлен- ня моделей і програм</i>	<i>Уніфіковані представ- лення мов специфікації</i>	<i>Уніфіковані представ- лення ма- тематики</i>	<i>Описи мо- делей на природних мовах</i>	<i>Мультиме- дійні пред- ставлення моделей</i>	<i>Узагальнене семантичне кодування моделей</i>
<i>Назви ПГД та її еле- ментів</i>	Набори імен типів в описах ПГД	Набори імен ПГД типів даних і об'єктів	Набори імен ПГД та її координат	Текстовий опис назви і змісту ПГД	Мультимедійні позначення ПГД	Ієрархія кодів словників і змісту ПГД
<i>Назви коор- динат</i>	Числових і абстрактних типів	Числових типів і об'єктів	Типів математичних даних	Текстові назви типів	Вразки відтворення типів	Коди назв і змісту типів
<i>Назви зв'яз- ків коорди- нат</i>	Назви блоків контролю обмежень	Назви виразів обмежень і зв'язків	Назви формул обмежень і зв'язків	Текстові назви зв'язків і обмежень	Мультимедійні символи зв'язків	Семантичні коди назв зв'язків
<i>Назви типі- зованих РРЗ</i>	Примірників і агрегатів РРЗ з аргументами	Примірників і агрегатів РРЗ з аргументами	Специфікацій примірників і агрегатів РРЗ	Текстові описи примірників і агрегатів	Мультимедійні символи примірників	Семантичні коди назв і агрегатів РРЗ
<i>Назви ар- гументів і змінних</i>	Типізованих аргументів і змінних	Типізованих аргументів і змінних	Типізованих аргументів і змінних	Текстові описи аргументів і їх типів	Мультимедійні символи аргументів	Семантичні коди аргументів і змінних
<i>Назви зв'язків і обмежень</i>	Контрольних виразів зв'язків і обмежень	Контрольних виразів зв'язків і обмежень	Контрольних виразів зв'язків і обмежень	Текстові описи зв'язків і обмежень	Мультимедійні позначення зв'язків	Семантичні коди виразів зв'язків
<i>Назви кри- теріїв складності</i>	Об'ємних, часових, комбінованих	Об'ємних, часових, комбінованих	Об'ємних, часових, комбінованих	Текстові описи критеріїв складності	Мультимедійні символи складності	Семантичні коди виразів складності
<i>Назви кри- теріїв ко- ректності</i>	Умов збіжності та точності	Умов збіжності та точності	Умов збіжності і точності	Текстові описи умов збіжності	Мультимедійні символи точності	Семантичні коди виразів точності

Якщо назви або ідентифікатори побудовані за так званою угорською схемою іменування, то елементи імені послідовно показують належність до об'єктів або їх груп за іменами, діями та, за необхідності, режимами використання. Імена окремих полів можуть формуватися на основі слів природних мов, відзначених номерами однотипних об'єктів, а для можливості реконфігурації імен доцільно використовувати поля зміщення відносно канонічної позиції при

відтворенні [8]. Імена спочатку прив'язуються до типа об'єкта, а потім включають номери та змістовні коди елементів об'єкта. При визначенні ПГД для проектування слід розрізняти їх фізичний зміст та їх математичний зміст з абстрагуванням від фізики значень та одиниць їх виміру. Для розв'язання задач достатньо мати *повну* математичну модель і систему пояснень та інтерпретації операцій та операторів і відтворення картин ПГД і результатів.

Опис математичної моделі ПГД включає у своєму заголовку назву ПГД, списки визначень координат $\mathbf{X} = \{X_1, \dots, X_k, \dots, X_{k_{\max}}\}$, де k – номери координат або їх однорідних агрегатів X^r_k , де r – мірність агрегатів координат, які утворюють підпростори. Кожний опис координати X_k включає визначення множини або домену припустимих значень $\text{dom } X_k$, а при конкретизації ПГД розширяється специфікаціями фізичних одиниць виміру $u(X_k)$.

Наявність математичних відношень і зв'язків координат $R(X_k)$ скорочує обсяг простору значень і множину можливих варіантів рішень, створюючи обмежені простори і фрактали як простори однорідних координат X^r_k з дробовою мірністю r . Таким чином, при визначенні координат конкретної ПГД важливо забезпечити їх несуперечність, тобто наявність непорожніх наборів сумісних значень координат, що дозволяють визначати комплекси дійсних (не уявних) об'єктів ПГД.

Більшість задач в рамках будь-якої ПГД розв'язується відносно окремих примірників об'єктів ПГД, а також картин і сцен, що формуються і відбуваються над комплексами об'єктів. Картини і сцени ПГД утворюються базовими примірниками об'єктів та їх агрегатами елементів. Для доведення коректності списки внутрішніх зв'язків, відношень та обмежень базових об'єктів і агрегатів доцільно розмістити в спеціальних розділах описів. Списки заголовків РРЗ базовими математичними методами або списки заголовків правил доведення повинні скласти основу ключів пошуку в БЗ потрібних реалізацій РРЗ та шляхів доведення їх відповідності специфікаціям типів, примірників і зв'язків.

Спираючись на гіпотезу, що будь-яка задача над об'єктами будь-якої складності може розв'язуватися у межах координатних просторів, що включають геометричний та/або топологічний підпростори, підпростори розвитку, які покривають координати динамічного континууму. При повному описі їх доцільно деталізувати через підпростори фізичних координат з визначеними відношеннями порядку та одиницями вимірів. Таким чином, головною задачею цієї роботи стало **створення набору конструкцій комп'ютерної мови специфікацій C_j та структур їх подання як вузлів дерева T** , що включає семантично значимі вузли $V_0, \dots, V_{li}, \dots, V_{l_{\max}}$, зв'язані в граф, названий в [8] графом алгоритму, який відображує підлеглість операторів і операцій [4], узагальнений для

конструкції опису і специфікації даних. Таке подання повинно однозначно представляти типи, що визначають координати і примірники даних з додержанням відношень семантичного порядку принаймні для канонічних форм представлення. Змістовна обробка операторів опису C_j повинна надавати повну інформацію про характеристики типів даних та їх примірників для розв'язання задач аналізу, синтезу та генерації виконавчих кодів в межах обраної ПГД.

Основні конструкції і структури розділів специфікацій комп'ютерних мов

Конструкції комп'ютерних мов з суворою типізацією звичайно спираються на базовий стиль декларації об'єктів мови. Наприклад, мова C/C++ використовує ім'я типа як попередника до описів його примірника, а мова Pascal – як відповідного наступника. В той самий час конструкції описів типів мають деякі різні деталі (роздільники або ключові слова). Такі самі стилі можна зберегти і при описі ПГД і в додаткових розділах описів класів і методів об'єктів.

Більшість стандартних конструкцій і прийомів об'єктно-орієнтованих мов програмування зручно використати при побудові розширень для представлення специфікацій. Так для опису просторів ПГД зручно використовувати механізм абстрактних класів з успадкуванням базових класів або типів, що описують координати ПГД. До опису будь-якого класу (координати і цілої ПГД) можна додати спеціальний розділ обмежень **restriction**, що представляє у вигляді логічних зв'язки, які притаманні координатам або полям класу і обмежують домени значень, та обмеження, характерні для окремих груп задач в ПГД.

Опис ПГД в формі специфікації включає її іменування та описи координат або типів, а також заголовки або шаблони функцій, тобто цей опис визначається як спеціальний клас об'єкта ПГД з супроводженням типів, які визначають координати елементів ПГД. Таким чином, визначення ПГД не потребує включення кардинально нових синтаксичних конструкцій. Описи сцен та дій в ПГД додатково включають специфікації примірників об'єктів і зв'язки обмежень їх координат, тощо. Таким чином, їх можна побудувати як класи об'єктів, успадкованих від класу ПГД з наступною деталізацією або додаванням нових об'єктів.

Вибір деревоподібної структури для узагальненого внутрішнього подання комп'ютерних мов

Спочатку приділимо увагу внутрішньому поданню виконавчої частини комп'ютерної мови. Найбільш поширеними деревоподібними структурами в трансляторах є дерева синтаксичного розбору, що формуються синтаксичним аналізатором і двійкові спрямовані ациклічні графи (САГ), що визначають піддерева підлеглих виразів комп'ютерних мов [1, 9]. Для розв'язання задач семантичної обробки ряд авторів [8] використовує так звані графи алгоритмів, які можна розглядати як проміжні форми між раніше описаними видами дерев та графів і цільовою формою, що має мінімальне різноманіття вузлів операцій та частин операторів.

Для раціональної організації семантичної обробки комп'ютерної мови слід сформулювати набір вимог, які визначатимуть порядок обробки графів. Ці вимоги стосуватимуться механізмів формування та обробки типів даних у виконавчих та декларативних конструкціях мов:

- у вузлі, що відповідає імені або константі, важливо зберігати код типа даних іменованого або константного об'єкта мови та проміжного результату;
- вузли графа повинні дозволяти подання елементів будь-яких виконавчих та специфікаційних конструкцій;
- важливо використовувати кодування типів, що дозволяє зберігати в таблицях однозначні коди типів та узагальнені коди для споріднених типів;
- бажано мати однакові структури для пошуку ключових елементів вузлів дерева в БЗ, щоб мінімізувати розміри таблиць відповідності для семантичної обробки.

Через дві останні вимоги для семантичної обробки більш зручні двійкові графи. Щоб якомога краще задовольнити перелічені вимоги як основу внутрішнього подання використаємо різновид графа алгоритму, який будемо називати **графом підлеглості** операцій та елементів операторів. Для математичних виразів комп'ютерних мов він повністю співпадає з базовою формою САГ [4] і потребує узагальнення для операторів специфікацій і структурного програмування. Основні правила побудови та організації цього графа можна сформулювати наступним чином:

- Всі знаки операції та базові ключові слова операторів відображаються в графах внутрішнього подання бінарними зв'язками.
- Відображення унарних операцій включає порожній зв'язок на місці відсутнього аргументу.
- Тернарні операції та оператори структурованого програмування також розбиваються на підграфи відповідно порядку їх виконання;
- Індукування елементів масиву розглядається за аналогією з двомісною операцією "[]" мови C++, лівий аргумент якої визначає масив, а правий – номер його елемента. Так само описи та виклики функцій і розділення заголовка і тіла блоку розглядаються як узагальнені операції "()" і "{}".
- Послідовність обробки однорідних підграфів вузлів визначення елементів списків можна розглядати як узагальнення обробки допоміжної операції кома "," і порожнього оператора ";" в мові C/C++.
- Для вузлів передачі управління операторів **break**, **continue**, **return** та **goto** передбачаються зв'язки управління на вузли наступної обробки.
- Будь-який підграф має єдиний кореневий вузол, який звичайно переглядається або обробляється першим або останнім в блоці структурного програмування (вхід до внутрішніх вузлів блоку повз кореневий вузол вважається некоректним).

Первинний граф підлеглості доцільно будувати з дерев синтаксичного розбору через **дерева або графи підлеглості операцій та конструкцій** [4]. В таких деревах повинні повторюватися всі підграфи з однаковими діями, а в графах повторні піддерева замінюються посиланнями на їх першу появу. На рис. 1 показано приклад побудови графа підлеглості конструкцій для наступного опису даних на мові C/C++: **const struct** rec rc, *prc; ...

В подібних операторах описів та специфікацій типів і класів даних та абстрактних типів і класів необхідно застосовувати допоміжний кореневий вузол визначення типу, який посилається піддерево визначення типу і має спеціальний тип вузла `UsrDefTp`. Оператори явного визначення типів (**type** мови Pascal і **typedef** мови C/C++) використовують вузли базових ключових слів як корені для посилань на визначення застосованих в заголовках описів іменованих типів користувачів.

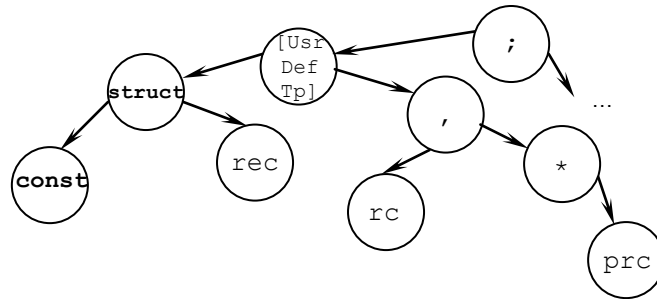


Рис. 1. Дерево оператора визначення даних мина struct
const struct rec rc, *prc;...

Описи операцій, функцій, класів та їх шаблонів в мові C/C++ розділяються на дві частини: описи заголовків та описи тіла визначення. Описи заголовків бібліотечних підпрограм повинні використовуватися як ключі при пошуку потрібних функцій. При формуванні графа мови C/C++ кореневий вузол визначається дужками тіла, ліве піддерево визначає типізований результат і список типізованих аргументів

підпрограми, а праве піддерево – оператори примірника реалізації. Дерево специфікації для початку визначення прикладу функції `int main(int argc, char*argv[]) {initHsh(thStud);...}` показано на рис. 2. Таким чином, дерева визначення підпрограм мають чітко виділений заголовок, який формує ключ пошуку в БЗ.

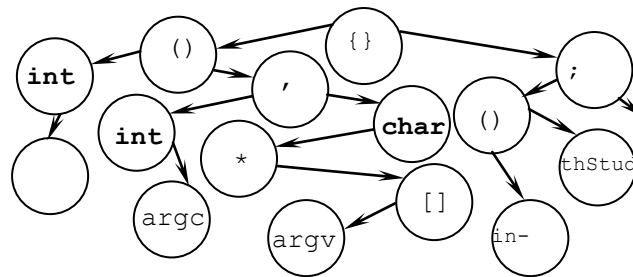


Рис. 2. Дерево оператора визначення функції
int main(int argc, char*argv[]) {initHsh(thStud);...}

Специфікації визначень класів і розширених типів користувача відображається аналогічно визначенню функцій з чітко виділеним заголовком. При цьому важливо включити до опису класу чи типу всі наявні елементи типів примірника та заголовки методів, включених до класу чи типу. Такі додаткові розділи специфікацій як розділ обмежень **restriction** також повинні входити для ключів визначення підпрограм, класів або для описів ПГД. Таким чином, частини обмежень типів, класів і підпрограм повинні бути приєднані до заголовків для полегшеного пошуку в БЗ.

ціальні вузли теоретико-множинних операцій і спеціальних позначень, зв'язків, серед яких лише деякі, що визначають тип використання підпрограми та її аргументи використовуються в ключових частинах визначень.

При комп'ютерному зберіганні Z-специфікацій необхідно насамперед визначити механізми деревоподібного відображення таких спеціальних конструкцій, як схеми, в яких спеціальні вузли типа `ScmStr` відділяють піддерево заголовку, яке використовується як ключ, від піддерева тіла схеми. Для інших спеціальних позначень Z-специфікацій треба визначити спе-

Особливості визначення структури ключових полів БЗ для пошуку специфікацій типів, даних і задач

Оскільки процес створення РРЗ пов'язаний з поданнями специфікацій, програм, моделей і процесів доведення [7] важливо використати узагальнений механізм формування семантики функціональних моделей обчислень і всіх потрібних оцінок їх складності.

Такий механізм створюється на базі узагальненого абстрактного типу аналітичного даних [5], в якому використовуються спеціальні канонічні форми зберігання [5, 8] для полегшення зіставлень варіантів рішень програм і доведень і впорядковане кодування [6] семантики імено-

ваних даних і операцій. Вимоги до канонічних форм подання кодів програм та протоколів верифікації [5] спираються на відношення лінійного порядку \leq [4] на базі кодів типів різних термінальних і нетермінальних вузлів дерев і графів підлеглості.

Висновки

Для розв'язання задач автоматизованої формальної верифікації і синтезу в системах програмування і моделювання доцільно розширити мови програмування засобами таких мов математичної специфікації як Z, представленими в стилі лінійних конструкцій мов програмування.

1. Для підвищення рівня автоматизації пошуку РРЗ за семантичними ознаками у вбудованій БЗ систем програмування складені ключі пошуку створюють назви цих ресурсів і списки типів аргументів або вхідних та вихідних даних операторів введення-виведення у складі відповідних предметних галузей.

2. Додатковими частинами ключових елементів пошуку потрібного РРЗ є розділи обме-

жень і критерії вибору модулів програм і моделей та їх складових частин, які дозволяють стати точно розпізнавати підрозділи БЗ, що орієнтовані на задані ПГД та обмеження, притаманні окремим випадкам задач.

2. Крім того узагальнені реалізації комп'ютерних мов і мов специфікації дозволяють організувати за єдиною методикою різні варіанти співставлень, впорядкувань та перетворень в системах програмування і моделювання, а також дають можливість одержати об'єктивно підтримані доведення тверджень і відповідності умовам в різних галузях математики, що може стати істотною комп'ютерною підтримкою при розробці математичних моделей.

3. Розширення комп'ютерних мов формальними специфікаціями координат ПГД як абстрактних типів зв'язків координат і змінних, а також обмежень координат і даних, відкриває можливість створення семантичних описів на базі природних мов і мов представлення різних видів мультимедійних даних, що надає можливість організації доступу до елементів РРЗ через більш зручні елементи текстових пояснень.

Список посилань

1. Ахо А., Сети Р., Ульман Дж. Компиляторы: принципы, технологии, инструменты: Пер. с англ. – М.: Издательский дом Вильямс, 2001. – 768 с.
2. Лисков Б., Гатэг Дж. Использование абстракций и спецификаций при разработке программ. Пер. с англ. М.: Мир, 1989. 424 с.
3. Пустоваров В.И. Побудова баз знань для сумісного створення програм і обладнання та їх формальної верифікації // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка – К.: «Век+». 2008, 51, с. 41-46.
4. Пустоваров В.И., Коваленко С.Ю. Організація сховищ даних для мов опису та маніпуляцій з інформаційними сутностями // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка – К.: «Век+». 2009, 51, с. 86 – 91.
5. Пустоваров В.И. Супроводження контролю відповідності програм і моделей формальним специфікаціям задач // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка – К.: «Век+». 2007, 47, с. 269-279.
6. Пустоваров В.И. Семантически ориентированное кодирование моделей и его эффективное применение для решения задач искусственного интеллекта // Электронное моделирование – К., - 1998, 20, № 4. с.32-42.
7. Hehner E.C.R. Practical theory of programming. Springer-Verlag, New York, 1993 – 243 p.
8. Metzger R.C., Zhaofang W. Automatic algorithm recognition and replacement: a new approach to program optimization / The MIT Press, Cambridge, 2000. 219 p.
9. Muchnick S.S. Advanced compiler design and implementation – San Francisco, Morgan Kaufmann Publishers, 1997. – 856 p.
10. Woodcock J., Davies J. Using Z. Specification, Refinement, and Proof. C.A.R. Hoare Series editor, 1995 – 390 p.