



НОВІ ЗАСОБИ КІБЕРНЕТИКИ, ІНФОРМАТИКИ, ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА СИСТЕМНОГО АНАЛІЗУ

О.В. ПАЛАГІН, М.В. СЕМОТЮК, С.В. УСТЕНКО

УДК 004

ХАОТИЧНІ АРХІТЕКТУРИ — НОВИЙ НАПРЯМОК РОЗВИТКУ ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Анотація. Проведено аналіз інформаційних технологій та виявлено їхні складові: технології віртуалізації, кількісні технології, технології даних і технології знань. На основі аналізу визначено, що формування хаотичних архітектур обчислювальних систем є новим напрямком розвитку цих систем.

Ключові слова: інформаційні технології, рівні технологій, технології віртуалізації, технології даних, технології знань, хаос, хаотичні архітектури, розумні системи, комп'ютери, парадигма програмування, машинна алгебра.

ВСТУП

Нині ми вживаємо такі слова і поняття, як «інформація», «інформаційні технології», «інформаційні дані» тощо, вже не замислюючись про тисячолітню історію становлення і розвитку інформаційних технологій (ІТ). Саме завдяки їм вдалося зберегти історію людства та його культурні надбання, що накопичувалися в результаті змін поколінь, які постійно поповнювали їх новим інформаційним вмістом. Найчастіше ІТ асоціюють саме з комп'ютерними технологіями і це не випадково: поява комп'ютерів вивела ці технології на новий сучасний рівень. Оскільки інформаційна взаємодія — це комплекс симетричних взаємодій, то виникає питання: яким є вплив сучасних ІТ на розвиток обчислювальної техніки?

РІВНІ СУЧАСНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Системотехнічні засади створення засобів обчислювальної техніки визначають найістотніші риси та ознаки цього класу техніки, зумовлені сьогоденними можливостями ІТ. Ці риси та ознаки обумовлюють функціональний рівень сучасних технологій. Будь-яка технологія у широкому сенсі — це обсяг знань та засобів, які можна використовувати для виробництва товарів і послуг як економічні ресурси. Звісно, у галузі обчислювальної техніки вирішальне значення мають інформаційні технології. На рис. 1 наведено складові ІТ, а саме технології чотирьох рівнів: технології знань (ТЗ), технології даних (ТД), кількісні технології (КТ) та технології віртуалізації (ТВ).

Процес формування інформаційних технологій склався історично і почався з технологій віртуалізації, коли тій чи іншій групі об'єктів присвоювали значення кількісних показників. При цьому оцінювали не об'єкти як такі, а їхні віртуальні образи. Таким чином, технології віртуалізації — це технології, що надають змогу сприймати об'єкти навколишнього середовища за допомогою кількісних характеристик, зазвичай у знеособленому вигляді, та мають вирішальне значення для інформаційних технологій у цілому.

© О.В. Палагін, М.В. Семотюк, С.В. Устенко, 2020



Рис. 1. Рівні сучасних інформаційних технологій



Рис. 2. Структура верхніх рівнів ІТ

Другий рівень ІТ представляють кількісні технології. Вони надають змогу здійснювати оброблення відомостей про ті чи інші об'єкти у знеособленому вигляді за допомогою різних представлень, наприклад, різних систем числення — римської, арабської, десяткової, двійкової тощо. Тут можуть використовуватися різні перетворення на кшталт теоретико-числових чи спектральних.

Наступним рівнем ІТ є технології даних, що надають змогу накопичувати інформацію про об'єкти, створюючи статистичні характеристики у вигляді таблиць, зведених відомостей тощо. Іншими словами, це клас технологій, які забезпечують можливість створення баз даних.

Найвищий рівень ІТ представляють технології знань. Вони надають змогу здобувати знання та створювати бази знань та знання-орієнтовані системи. Порівняльну структуру двох останніх технологій [1] наведено на рис. 2, з якого видно як їхню схожість, так і відмінність. Взаємодію цих технологій в цілому буде розглянуто далі.

КЛАСИФІКАЦІЯ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

З урахуванням наведеної вище структури інформаційних технологій можна перейти до класифікації обчислювальних засобів та систем з огляду на їхні функціональні особливості (рис. 3). З цієї класифікації видно, що обчислювальні системи можна поділити на два великі класи: клас детермінованих систем та клас хаотичних систем [2, 3] або, інакше кажучи, систем з функціональною архітектурою.

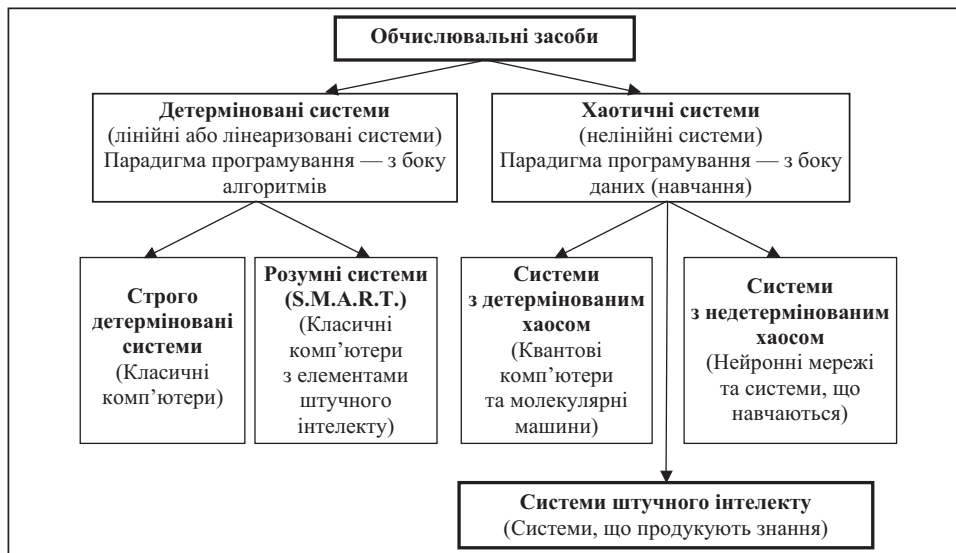


Рис. 3. Класифікація обчислювальних систем на основі ІТ

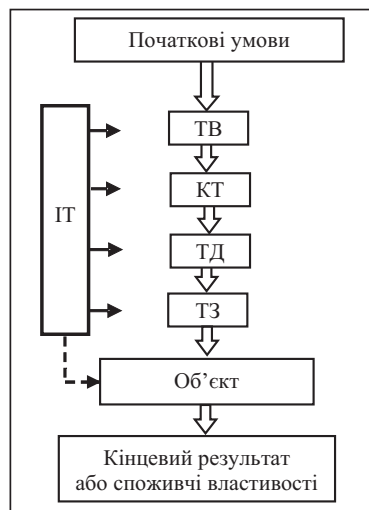


Рис. 4. Схема взаємодії технологій у випадку створення строго детермінованих систем (для старт-систем додатково показано пунктиром)

Теорія проектування таких систем ґрунтується на принципах машинної алгебри [5] згідно з такими постулатами.

1. Машинна алгебра

$$U_M = \left\langle \bigcup_{i=0}^N u_i \right\rangle \quad (1)$$

де u_i — алгебраїчні системи, для яких можна побудувати фізично реалізовані моделі, а U_M — машинна алгебра.

2. Система команд — це будь-яка сигнатура машинної алгебраїчної системи U_M

$$\Omega_M = \langle \mathbf{R}_M, \mathbf{F}_M, \mu_M \rangle, \quad (2)$$

де $\mathbf{R}_M, \mathbf{F}_M$ — множини символів машинних відношень та операцій, за допо-

Детерміновані системи можуть бути строго детермінованими системами та розумними (S.M.A.R.T.) системами. Класу строго детермінованих систем належать звичайні автомати, верстати з числовим програмним керуванням, комп'ютери з традиційною архітектурою тощо. У створенні детермінованих систем широко використовуються як ТЗ, так і ТД. Однак їхнє застосування обмежується тим, що вони є лише інструментом розробника на етапі проектування. Схему використання технологій для створення строго детермінованих систем показано на рис. 4. Розумні системи — це ті самі строго детерміновані системи, але з елементами штучного інтелекту.

Розумні системи (S.M.A.R.T.) [4] проектують у такий самий спосіб, однак до складу самої системи включають як компонент такі ІТ, як розпізнавання ситуацій, образів, мовних команд тощо. Іншими словами, зазначені ІТ стають невід'ємною частиною самих систем.

Таблиця 1. Алгебраїчна класифікація обчислювальних засобів (R — предикатна формула, F — функціональна формула, \Rightarrow — семантичне слідування)

Тип обчислювального засобу	Вигляд ланцюжка формул	Алгебра ланцюжка формул	Алгебра сигнатури ланцюжка формул
Машина фон Неймана	$F \Rightarrow, R \Rightarrow$	Алгебра і модель — окремі команди. Алгебра функціональної формули: адитивна група, мультиплікативна група, напівгрупа	Адитивна група, напівгрупа
Сучасні ПЕОМ	$F \Rightarrow, R \Rightarrow$	Алгебра і модель — окремі команди. Алгебра функціональної формули: елементи кільця, адитивна група, мультиплікативна група, напівгрупа	Адитивна група, напівгрупа
Сигнальні процесори 1–4-го покоління (гарвардська архітектура)	$F \Rightarrow, R \Rightarrow$	Алгебра і модель — окремі команди. Алгебра функціональної формули: елементи кільця, адитивна група, мультиплікативна група, напівгрупа	Адитивна група, напівгрупа
Сигнальні процесори 5-го покоління (супергарвардська архітектура)	$F \Rightarrow$ $R \Rightarrow$ $R, F \Rightarrow$	Алгебраїчна система. Алгебра функціональної формули: елементи кільця, адитивна група, мультиплікативна група, напівгрупа	Адитивна група, напівгрупа
Сигнальні процесори (супернеймановська архітектура)	$F, R \Rightarrow$	Алгебраїчна система. Алгебра функціональної формули: кільце, поле, адитивна група, мультиплікативна група, напівгрупа	Алгебраїчна система. Кільце, адитивна та мультиплікативна група або напівгрупа та предикат

могою яких можна побудувати теорію T_M (іншими словами, спосіб організації обчислювального процесу), μ_M — їхнє машинне відображення.

3. Абстрактна архітектура — це відображення машинної алгебраїчної системи на фізичне середовище, яке може моделювати операції системи та спосіб організації обчислювального процесу

$$T_M = f^{-1}(A, u_M) \text{ і } A = f(u_M, T_M), \quad (3)$$

де f — функція інтерпретації T_M у A , f_{-1} — обернена функція інтерпретації A в T_M .

Згідно з викладеним вище можна побудувати відповідно й алгебраїчну класифікацію детермінованих систем, наведену в табл. 1.

Другим великим класом засобів обчислювальної техніки є хаотичні системи або системи з функціональною архітектурою. Їх у свою чергу поділяють на системи з детермінованим і недетермінованим хаосом.

В основу систем з детермінованим хаосом покладено те, що деякі фізичні процеси у той чи інший спосіб за заданих початкових умов збігаються з великою ймовірністю до деякого сталого результату. Але зазвичай такі процеси не можуть застосовувати усталені технології даних чи знань у зв'язку з відсутністю можливості з тих чи інших причин описати достовірно цей фізичний процес. Однак людство навчилося використовувати такі процеси, не знаючи до кінця їхньої фізичної суті. Проте це лише видимий аспект проблеми. Насправді головним є те, що ці обчислювальні процеси мають нелінійний характер. Традиційний підхід до розв'язування нелінійних задач передбачає їхню лінеаризацію, після чого можна створити алгоритми розв'язування цих задач. Оскільки з тих чи

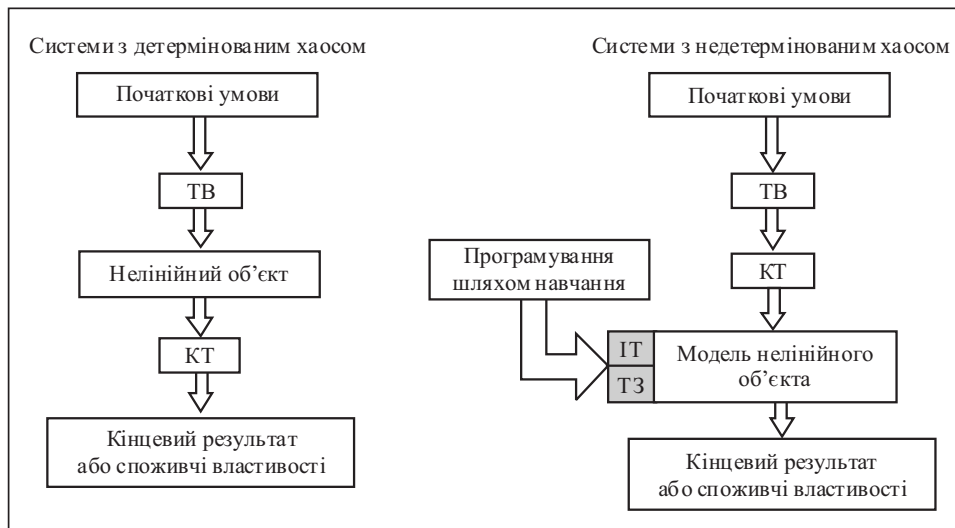


Рис. 5. Схема взаємодії технологій у випадку створення хаотичних систем

інших причин здійснення лінеаризації не завжди є можливим, тобто технології даних та технологія знань є недосяжними, використовують «хаос» як спосіб опису та розв'язування цих нелінійних задач. Звідси й походить назва «хаотичні системи». Схему взаємодії технологій у цьому випадку показано на рис. 5. З неї видно, що технології знань та даних знаходяться у самому нелінійному об'єкті, який ми використовуємо для розв'язування таких задач.

Слід зауважити, що на практиці цей спосіб розв'язування нелінійних задач не є прийнятним, оскільки важко, а іноді й неможливо підібрати об'єкти з потрібними характеристиками. Простіше замінити нелінійні об'єкти на математичні моделі. Як моделі найчастіше застосовують нейронні мережі або перцептронні чи звичайні комп'ютери з програмами, що використовують хаотичні алгоритми.

Перейдемо до поняття систем з недетермінованим хаосом. Схему використання технологій у цьому випадку представлено на рис. 5, де нелінійний об'єкт замінено на його математичну модель. Зауважимо, що застосування нейронних мереж у розвитку засад створення обчислювальної техніки на сучасному етапі є вельми успішним. Проте цей успіх зумовлений не стільки їхнім використанням для розв'язування задач штучного інтелекту, скільки тим, що вони забезпечують можливість розв'язування саме нелінійних задач. При цьому виникає питання, в який спосіб потрібно задати нелінійний об'єкт, якщо його аналітичне представлення є невідомим? Відповідь дає жива природа — цим способом є навчання, тобто підбір параметрів шляхом проб та помилок. Для технічних систем маємо зміну парадигми програмування: з програмування з боку алгоритмів на програмування з боку даних.

ПАРАДИГМА ПРОГРАМУВАННЯ ТА ЦИФРОВІ АВТОМАТИ, ЩО МОЖУТЬ НАВЧАТИСЯ

Спершу слід уточнити поняття алгоритму розв'язування задач. Будь-яка редукція тієї чи іншої формули або аналітичної залежності є алгоритмом тоді, коли вона записана у вигляді бінарних операцій та бінарних відношень.

Розглянемо деякі властивості генетичних алгоритмів, які на сучасному етапі широко застосовуються на практиці. Ці властивості одержано в результаті низки досліджень, яким раніше не приділяли достатньої уваги. Вони є такими:

- використання багатоінваріантних представлень одних і тих самих даних (наприклад, ознака ділення на будь-яке число);
- інваріантність до зсуву та перестановки, використання комбінаторики (реакція кліків $2^n - 1$ [6]);
- інваріантність відносно як функціональних перетворень, так і предикатних залежностей (наприклад, в арифметиці — ділення);
- зменшення розмірності простору представлень (використання симетрії, голограм і фракталів);
- віртуалізація представлень (зображення — слово);
- зміна парадигми програмування (програмування з боку даних);
- явище реплікації як спосіб побудови високонадійних систем.

Аналіз цих властивостей свідчить про те, що, хоча кількість алгоритмів оброблення інформації, наявна в ДНК нейрона, є великою, вона усе-таки є обмеженою, тобто скінченною. Недостача алгоритмів ДНК, якщо вона виникає, компенсується багатоінваріантністю представлень одних і тих самих даних. Це означає, що обчислювальну потужність кожного алгоритму можна обчислити як

$$N = \sum_i^l r_i \cdot k, \quad (4)$$

де r_i — кількість редукцій цього алгоритму, k — кількість можливих інваріантів представлень даних для цього алгоритму. Фактично величина N може бути необмеженою, тобто пошук алгоритму може виявитися нереалізованим протягом прийнятного часу через велику кількість варіантів. Тому природа чи еволюція обрала інший шлях — зміну парадигми програмування з боку алгоритмів на програмування з боку даних, тобто навчання. Одним з видів програмування з боку даних є складання таблиць. Якщо таблиця складається у довільний спосіб (тобто є неупорядкованою, наприклад, у разі одержання експериментальних даних) для великої кількості даних, то неможливо виявити алгоритм складання цієї таблиці. Тоді надалі користування таблицею можна вважати процесом навчання. Цим принципом навчання або, точніше, псевдо-навчання можна скористатися для побудови звичайних автоматів, що навчаються. На рис. 6 наведено комбінаційну логічну схему, що здатна програмуватися з боку даних, тобто навчатися, оскільки її створюють не аналітичним шляхом або за допомогою таблиці істинності, а шляхом подання тих чи інших значень у довільному порядку. Ця схема працює у такий спосіб. В основу елемента покладено мультиплексор і два регістри, один з яких є



Рис. 6. Логічний елемент, здатний до навчання

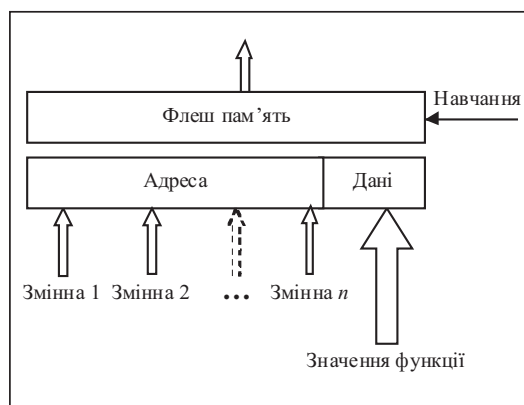


Рис. 7. Функціональний елемент, здатний до навчання

регістром зсуву, а інший — набором тригерів для запам'ятовування значень логічної функції. Для навчання цієї схеми на адресний вхід мультиплексора подають код, якому треба навчити схему. Далі запускають регістр зсуву, яким рухається «одиниця». У разі збігу адреси мультиплексора з його інформаційним входом на виході мультиплексора теж з'являється одиниця, яка і запам'ятовується у відповідному тригері значень логічної функції. Для великих масивів даних таке запам'ятовування збігається з навчанням, тому що немає потреби шукати алгоритм одержання цієї логічної функції. Очевидно, що для навчання будь-якої функції крім логічної можна використовувати ансамбль таких логічних елементів, який зрештою перетворюється на флеш-пам'ять або програмовану логічну інтегральну схему (ПЛІС) (рис. 7). Отже, проведений у цьому розділі аналіз та модулярна модель нейрона [7] надають змогу побудувати аналоги нейронних мереж на основі вже досліджених автоматів і схем обчислювальної техніки та змінити парадигму програмування з програмування з боку алгоритму на програмування з боку даних.

ОБЧИСЛЮВАЛЬНІ СИСТЕМИ З ХАОТИЧНОЮ АРХІТЕКТУРОЮ

Розглянемо організацію обчислень у хаотичних системах. Відомо [2], що ці системи здатні до розвитку і повинні мати для цього деякий вільний обчислювальний ресурс, що слугує відповідною мотивацією. Однак обмеженням, що накладається на цей вільний ресурс, є певна система команд. Вона за своєю суттю є обмежувачем можливості розвитку, оскільки не може розширюватися (поле коду операції має фіксовану довжину в бітах). Тому розширення обчислювальних можливостей забезпечують шляхом віртуалізації за рахунок зниження продуктивності. Для розв'язування цієї задачі можна скористатися моделлю машини Тюрінга або Поста в потрібній інтерпретації. Нехай обчислювальна машина складається з масиву функціональних та логічних елементів у вигляді ПЛІС, інтерфейсу між пам'яттю та ПЛІС, власне пам'яті, компілятора для ПЛІС з інтерпретатором команд та його базою даних, керувального процесора та вузла навчання (рис. 8.) У керувальному процесорі апаратно реалізовано тільки команди пересилання, скажімо як в машині Поста (маються на увазі лише команди пересилання між пам'яттю та ПЛІС). Користувач такої машини за допомогою компілятора ПЛІС створює самостійно той чи інший пристрій, наприклад арифметичний блок або обчислювач функцій тощо. Далі він розміщує його в адресному просторі машини у такий спосіб, що інформаційні входи цих блоків мають свою окрему адресу в адресному просторі. Тоді, пересилаючи дані з пам'яті у створені блоки, можна виконувати ті чи інші операції. Створені операції заносяться в базу даних інтерпретатора. Інтерпретатор далі інтерпретує програму, написану мовою високого рівня,

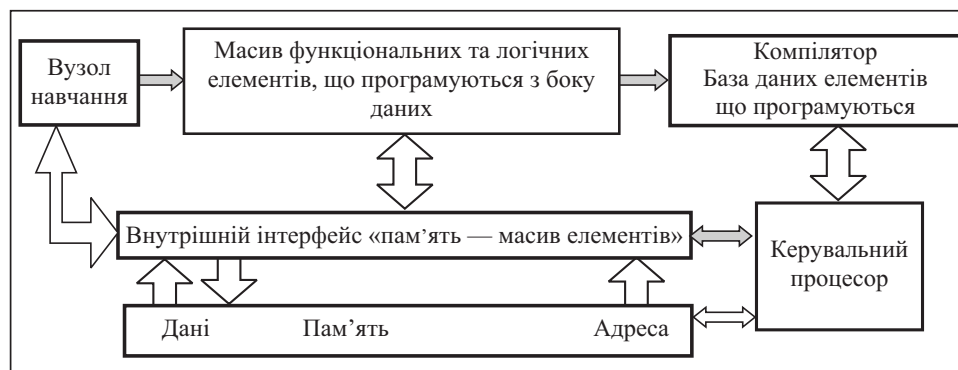


Рис. 8. Процесор з хаотичною (функціональною) архітектурою

в послідовність пересилань. Зрозуміло, що програма буде складатися лише з пересилань. Оскільки користувач створює систему команд за своїми уподобаннями, ця система команд, а також архітектура буде хаотичною, бо відрізнятиметься від системи команд, яку створить інший користувач, маючи таку саму систему. Завдяки цьому можна створити комп'ютери з мінімальною уразливістю до вірусів, тобто забезпечити кіберзахист своїх програм.

Далі виникає питання: що робити, коли користувач не має достатньо знань у галузі створення обчислювачів? Рішення є таким — включити до складу системи нейронну мережу. Шляхом навчання користувач може навчити цю мережу розв'язувати відповідні задачі. Однак після навчання нейронна мережа стає детермінованою системою. Отже, за допомогою розробленої модулярної моделі нейрона можна у формальний спосіб створити алгоритм розв'язування тієї чи іншої задачі. Далі, маючи алгоритм, можна синтезувати потрібний обчислювальний блок за допомогою компілятора ПЛІС. Оскільки ці перетворення є формальними, то процес створення алгоритму можна автоматизувати. У підсумку за такого підходу сам користувач не буде знати свою систему команд. Однак, змінюючи у потрібний спосіб свої обчислювальні блоки, користувач розвиватиме свою систему відповідно до своїх потреб.

ВЗАЄМОДІЯ РІВНІВ ІТ ТА СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ

Розглянемо взаємодію рівнів інформаційних технологій на простому прикладі. Нехай потрібно обчислити вагу тіла. Для цього скористаємося законом Ньютона $F = ma$. Безперечно, цей закон є складовою технології знань. Однак вагу тіла на основі цього закону визначають за формулою $P = mg$. Для обчислення ваги потрібно знати масу тіла m та прискорення вільного падіння g . Відомо, що прискорення g не є величиною сталою для всіх точок земної поверхні, тому потрібно мати його значення для вибраної точки. Крім того, слід обрати систему одиниць вимірювання як прискорення, так і маси тіла. Отже, ми маємо спуститися нижче на рівень технології даних. Далі потрібно виконати обчислення, а це означає, що треба вибрати систему числення, в якій здійснюватимуться розрахунки, тобто представлення даних. При цьому потрібно здійснити перехід ще на один рівень нижче, а саме на рівень кількісних технологій. Після обчислення ваги тіла необхідно інтерпретувати одержані результати, тобто переміститися нижче ще на один рівень — рівень технології віртуалізації, оскільки віртуалізація є одним з видів інтерпретації фактів.

Зауважимо, що процес формування знань [8] також можна здійснити у такий самий спосіб. Спершу слід сформулювати на основі тієї чи іншої аксіоматики гіпотезу як деякий збіг фактів, що має місце. Далі потрібно довести, що цей збіг фактів не є випадковим, а має місце за будь-яких обставин. Коли такий доказ є усталеним, то це означає, що гіпотеза перейшла у **продукцію**, де причина і наслідок пов'язані семантичним слідуванням, яке не підпорядковується закону контрапозиції. Іншими словами, якщо причина є істинною, то і наслідок є істинним. Нині цей підхід досить часто застосовується для створення квантових комп'ютерів, молекулярних машин тощо. Однак це не означає, що справджується обернене твердження (коли наслідок є істинним, то і причина є істинною). Його потрібно довести, тоді семантичне слідування перейде в логічне слідування. Мовою математики це означає, що доведено теорему. Далі, щоб створити теорію, необхідно вивчити всі редукції отриманої залежності та здійснити їхнє узагальнення. У такий спосіб створюються знання. З цього прикладу видно, що ми маніпулюємо тими чи іншими рівнями інформаційних технологій, спускаючись кожного разу все нижче і нижче.

Однак існує інший шлях — рух від нижніх рівнів технологій до вищих. Розглянемо і його. Для цього уточнимо поняття систем штучного інтелекту. Будемо



Рис. 9. Схема отримання знань

вважати, що система має штучний інтелект тоді, коли вона здатна продукувати технологію знань. Очевидно, що шлях одержання знань через гіпотези в цьому випадку не є прийнятним, тому скористаємося зворотним шляхом, тобто рухатимемося рівнями технологій знизу вгору. Для цього скористаємося нейронною мережею і навчимо її розв'язувати ту чи іншу просту задачу. Після навчання мережа стане детермінованим пристроєм, оскільки в результаті будуть визначені всі необхідні пороги нейронів, а також коефіцієнти передачі та зв'язки між нейронами нейронної мережі. Якщо проаналізувати ці параметри за допомогою кількісних технологій, то можна знайти алгоритм розв'язування цієї задачі.

Як метод, що задовольняє вимоги добування знань на першому етапі, використовують схему понейронного розгляду нейромережі і опису кожного нейрона за допомогою одного або декількох продукційних правил. Зрозуміло, що вигляд видобутих продукційних правил залежить передусім від типу нелінійної функції нейрона та коефіцієнтів підсилення чи послаблення сигналів (безперервних або дискретних), що проходять через нейрон. Далі ці продукційні правила піддають статистичному аналізу за допомогою програм, які являють собою модифікований процесор логічного виведення або нейроімітатор, за допомогою яких верифікується та чи інша **продукція** для того, щоб семантичне слідування перейшло в логічне слідування. Якщо це досягнуто, то перший етап виконано. Далі ці результати зводять воедино і у матричному вигляді представляють користувачам для експертизи.

Другим етапом, що завершує процес одержання знань є вилучення з отриманих залежностей представлень конкретних параметрів (кількісних характеристик) у такий спосіб, щоб вони були незалежними від цих представлень. Схему, що ілюструє цей процес в цілому, наведено на рис. 9.

ВИСНОВКИ

Аналіз ІТ показав, що ці технології загалом мають такі чотири рівні ІТ: технології віртуалізації, кількісні технології, технології даних та технології знань. З цього витікає, що ці рівні розділяють засоби обчислювальної техніки на два класи, які мають істотну відмінність: клас з традиційною архітектурою та клас з хаотичною архітектурою. Основним аспектом розподілу є зміна парадигми програмування, а саме заміна програмування з боку алгоритмів на програмування з боку даних (навчання). Оскільки процес навчання не є строго детермінованим і завжди відбувається в різних умовах, архітектура таких засобів обчислювальної техніки стає хаотичною. Іншими словами, виникає новий напрямок розвитку обчислювальної техніки — хаотичні архітектури, при

цьому хаотичною є не лише архітектура власне засобів цієї техніки, а й система їхніх команд, що дає можливість створити новий клас інформаційних систем, які є захищеними від кібератак і мають інші корисні характеристики.

СПИСОК ЛІТЕРАТУРИ

1. Park S.H., Suh M.W. Data technology and e-statistics with their applications in industry. *Proc. 4th Asian Regional Section of the International Association of Statistical Computing* (December 5–7, 2002, Busan, Korea). Busan, 2002. P. 201–204.
2. Андреев Ю.В., Дмитриев А.С., Куминов Д.А. Хаотические процессоры. *Успехи современной радиоэлектроники (Зарубежная радиоэлектроника)*. 1997. №. 10. С. 50–79.
3. Doran G.T. There's a S.M.A.R.T. way to write management's goals and objectives. *Management Review (AMA FORUM)*. 1981. Vol. 70, Iss. 11. P. 35–36.
4. US Patent № 07096347. Ditto W., Sinha S., Murali K. Method and apparatus for a chaotic computing module. August 22, 2006.
5. Семотюк М.В. Заметки по машинной алгебре. Киев: Сталь, 2012. 250 с.
6. Tsien J.Z. A postulate on the brain's basic wiring logic. *Trends Neurosci.* 2015. Vol. 38, Iss. 11. P. 669–671. <https://doi.org/10.1016/j.tins.2015.09.002>.
7. Клюкин В.И., Николаенков Ю.К. Электрические и математические модели нейронов. НС прямого распространения Воронеж: Издательский центр Воронеж. гос. ун-та, 2008. 63 с.
8. Лукьянец В.Г. Автоматизация процесса получения знаний. *Высшее техническое образование: проблемы и пути развития = Engineering education: challenges and developments: Материалы VIII Междунар. науч.-метод. конф.* (17–18 ноября 2016, г. Минск). Минск: БГУИР, 2016. С. 11–13.

Надійшла до редакції 05.09.2019

А.В. Палагин, М.В. Семотюк, С.В. Устенко **ХАОТИЧЕСКИЕ АРХИТЕКТУРЫ — НОВОЕ НАПРАВЛЕНИЕ РАЗВИТИЯ** **ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ**

Аннотация. Проведен анализ информационных технологий и выявлены их составляющие: технологии виртуализации, количественные технологии, технологии данных и технологии знаний. На основе анализа определено, что формирование хаотических архитектур вычислительных систем является новым направлением развития этих систем.

Ключевые слова: информационные технологии, уровни технологий, технологии виртуализации, технологии данных, технологии знаний, хаос, хаотические архитектуры, умные системы, компьютеры, парадигма программирования, машинная алгебра.

A.V. Palagin, M.V. Semotiuk, S.V. Ustenko **CHAOTIC ARCHITECTURES: A NEW TREND IN COMPUTERS**

Abstract. Information technologies are analyzed and their components are identified as virtualization technologies, quantitative technologies, data technologies, and knowledge technologies. Based on the analysis, it is determined that chaotic architectures of computer systems are a new trend in the development of these systems.

Keywords: information technology, technology levels, virtualization technology, data technology, knowledge technology, chaos, chaotic architectures, smart systems, computers, programming paradigm, machine algebra.

Палагин Александр Васильевич,
академік НАН України, доктор техн. наук, заступник директора Інституту кібернетики ім. В.М. Глушкова НАН України, Київ, e-mail: palagin_a@ukr.net.

Семотюк Мирослав Васильевич,
кандидат техн. наук, провідний науковий співробітник Інституту кібернетики ім. В.М. Глушкова НАН України, Київ, e-mail: semo@i.ua.

Устенко Станіслав Веніамінович,
доктор екон. наук, професор, професор кафедри Київського національного економічного університету імені Вадима Гетьмана, Київ, e-mail: stasustenko@ukr.net.