

Археологія і кібернетика

II. Застосування великих ЕОМ в археологічному дослідженні *

Застосування великих електронно-обчислювальних машин (ЕОМ) у сфері наукового дослідження має принципові відмінності від застосування ручних перфокарт і малих електронно-обчислювальних пристроїв (табулятори, сортувальні машини тощо). Різниця полягає у тому, що ручні перфокарти можуть лише технічно полегшити процес дослідження, визначений і здійснюваний людиною, тоді як великі ЕОМ здатні моделювати розумові процеси і, отже, якоюсь мірою замінити дослідника. В майбутньому автомати зможуть взяти на себе весь процес дослідження, залишивши людині евристичну його частину.

Щоб реально оцінити можливості сучасних ЕОМ, доцільно насамперед коротко спинитися на принципах їх дії. Тепер існує багато різних моделей автоматів, які досить істотно відрізняються один від одного своєю конструкцією та параметрами. Але є спільна основа, що дозволяє розглядати їх як варіанти єдиного цілого.

Як відомо, кожна велика ЕОМ, незалежно від конструкції і матеріального забезпечення, складається з трьох основних блоків: керуючого пристрою (КП), арифметичного (АП) та запам'ятовуючого (ЗП). Назва блоків визначає їх функції. Арифметичний пристрій здійснює переробку інформації; запам'ятовуючий зберігає інформацію — як вихідну (ту, що дослідник вводить до ЕОМ перед початком її роботи), так і проміжну (здобуту в процесі переробки вихідних відомостей); керуючий пристрій забезпечує роботу машини відповідно до заданої програми.

Вихідна інформація, введена до ЗП, у кожному випадку повинна складатися з двох основних частин: а) матеріалу, що є предметом дослідження і б) програми, яка визначає зміст і послідовність необхідних для цього операцій. Програма в свою чергу має дві частини: суму команд, які машина повинна виконати, і набір констант — наперед визначених чисел, що допомагають реалізації тієї чи іншої команди.

Слід мати на увазі, що ЕОМ — автомати дискретної дії і, отже, вся закодована інформація мусить бути подана дискретно, у вигляді скінченних чисел. Це ставить перед археологом ряд серйозних складних проблем.

Електронний принцип дії передбачає застосування двоїчної системи числення. Вона оперує лише двома знаками для відтворення чисел: 0 та 1. Наявність електричного імпульсу означає 1 («так», «плюс»); відсутність — 0 («ні», «мінус»). Всі машини мають спеціальні пристрої для переведення чисел з загальнозвжivanoї десяткової системи числення у двоїчну і навпаки. Але під певним кутом зору зручніше всю вихідну інформацію готувати у восьмеричній системі ($8=2^3$), бо в ній кожний розряд відповідає трьом розрядам двоїчної:

0—000,	4—100,	10—001 000,
1—001,	5—101,	11—001 001,
2—010,	6—110,	12—001 010
3—011,	7—111,	і т. д.

Це дає можливість легко і просто переводити числа з однієї системи у другу за допомогою так званих триад **. Так, восьмеричне число 512 у

* Першу статтю з цього циклу див.: Археологія, т. XXI, К., 1968.

** Триада — частина числа, що складається з трьох цифр (розрядів.)

двоїчному численні буде 101 001 010, де тріади 101, 001, та 010 відповідатимуть восьмеричним цифрам 5, 1 та 2. Двоїчне 100 110 101 (з тріадами 100, 110 та 101) у восьмеричному численні матиме вигляд 465. Правда, десятичне число (яким би воно не було) легко переводиться у двоїчну систему за допомогою тетрад*:

1—0001 3—0011
2—0010, і т. д.

Але в цьому випадку невикористаними будуть тетради 1011, 1100, 1101, 1110, 1111, що, враховуючи поки що низькі параметри наявних машин, і зокрема обмеженість оперативної пам'яті, є дуже небажаним.

Арифметичні дії для двоїчних чисел такі самі, як і для звичайних. Автомат записує їх у так звані комірки — пристрої, що складаються з певної (заздалегідь визначеної) кількості елементів, здатних тримати електричний імпульс (пропускати або не пропускати струм). Кожний з цих елементів відповідає одному розряду двоїчного числа, а тому і має назву «розряд комірки». Значення його може бути 0 або 1 — залежно від наявності або відсутності імпульсу.

Кількістю розрядів визначена місткість комірки, тобто довжина чисел, якими може оперувати автомат. Вона у різних ЕОМ різна, але для кожної моделі точно фіксована. Машина «Урал», наприклад, має 18-розрядні комірки, але конструкцією передбачена можливість з'єднувати їх по дві (отже, тут максимальна довжина числа — 36 розрядів), а машина «Стріла» має 42 розряди і т. д. Більш-менш нормальна довжина комірки, найчастіше вживана, — 36—48 розрядів.

Перший розряд (здебільшого він вважається останнім, бо нумерація йде справа наліво) призначається для запису знака числа (+ або —), оскільки число може бути позитивне і негативне. Це треба мати на увазі під час програмування, бо цілий ряд команд виконується по-різному, залежно від знака. Таким чином, максимальна місткість комірки в машині «Урал» становить 35 розрядів і відповідає 11 розрядам десятичних чисел; цим визначена максимальна місткість машини, її інформаційний параметр: вона може оперувати 34 млрд. чисел (або, відповідно, понять).

Запам'ятовуючий пристрій ЕОМ складається з певної кількості комірок. Він поділяється на дві основні частини — зовнішню і внутрішню (або оперативну) пам'ять. Безпосередній переробці підлягає лише та інформація, яку записано у другій частині; щоб оперувати з інформацією, записаною до зовнішньої пам'яті, її необхідно спочатку переписати в комірки ОП. Це найскладніша і найдешкульніша ланка сучасних машин.

Розмір оперативної пам'яті завжди дуже обмежений. Взагалі він дорівнює 2048 або 4096 комірок**, хоч є й більш місткі моделі. Оперативна пам'ять і швидкість дії машини — взаємопов'язані величини, причому їх взаємозалежність має зворотний характер: при визначеній конструкції чим більша пам'ять, тим повільніша дія машини і навпаки.

Натомість зовнішня пам'ять має практично необмежені параметри, але процес переписування інформації до оперативної пам'яті (невеликими порціями, бо тут знову-таки дає про себе знати обмеженість ОП) забирає багато часу і, крім того, створює додаткові труднощі в процесі дослідження. Практично операцію треба повторювати повністю для кожного переписаного масиву інформації спочатку.

Головним діючим блоком електронно-обчислювальної машини є арифметичний пристрій. Основну його частину становить суматор—комірка, в якій відбуваються арифметичні та логічні операції. Він має стільки розрядів, як і стандартна комірка пам'яті. Після виконання операції в суматорі фіксується її наслідок, вихідні числа залишаються тільки в ЗП.

* Тетрада — частина числа, що складається з чотирьох цифр.

** Не зовсім звичайний вигляд чисел пояснюється тим, що при двоїчній системі зчислення всі числа становлять 2 в ступені (в даному випадку 2¹¹ та 2¹²).

Кожній операції ЕОМ повинна відповідати одна команда, а для розв'язання завдання в цілому необхідний список команд, тобто програма.

Програмування зводиться до складання списку тих дій, які повинна виконати машина. Але цей список слід чітко відрізнити від алгоритму, тобто від логічної послідовності операцій. Остання схема є більш узагальненою, один елемент (або блок) алгоритму потребуватиме кількох, а іноді багатьох команд. Отже, машинну операцію (якій відповідає один такт роботи автомата і яка потребує тільки однієї команди) в жодному разі не можна плутати з логічною або арифметичною операцією. Обмежимося елементарним прикладом.

Скажімо, нам потрібно визначити процентне співвідношення між усіма черепками, знайденими на поселенні n , і уламками гончарного посуду (перше число позначимо a , друге — b). З точки зору алгоритму, це одна операція, але вона розпадеться на кілька етапів, кожному з яких буде відповідати окрема команда програми, а саме:

$$x = \frac{b \cdot 100}{a}.$$

Звідси випливає така послідовність в роботі автомата: 1-а команда: b береться в суматору; 2-а команда: b множиться на 100; 3-я команда: результат попередньої дії ділиться на a ; 4-а команда: остаточний результат видається дослідникові або записується в потрібну комірку пам'яті.

Цей елементарний приклад поданий за одноадресною схемою, але існують дво- і триадресні автомати, що виконують відповідно дві або три команди за один робочий такт (докладніше про це скажемо нижче).

Кожна ЕОМ має свою систему команд і свою манеру їх зображення. Шифр кожної з них складається з двох частин: адреси і коду операції. Адреса — це номер комірки (чи комірок) ЗП, де зберігається потрібна інформація, тобто число (або числа), яке бере участь в інформації, або номер комірки, куди треба записати одержаний результат. Код операції визначає зміст дії автомата. Це умовне число, точно фіксоване для кожної машини.

Таким чином, звичайна команда повинна мати вигляд: ab , де a — код операції, b — адреса. Наприклад, для машини «Урал» команда додавання матиме такий вигляд: 01 0001. Це значить: «число, що зберігається в комірці 0001, додати до числа, записаного в суматорі». Команда «послати на пам'ять» матиме вигляд: 16 0002, тобто означатиме: «число, що значиться в суматорі, записати в комірку 0002» і т. д.

Переважає більшість реально існуючих моделей належить до одно- і триадресних. Одноадресна команда, як це видно з самої назви, має на увазі лише одну комірку, тобто одне число (крім записаного в суматорі). Наведені вище приклади команд для машини «Урал» належать до цієї категорії.

Триадресні команди мають на увазі три комірки — у двох з них зберігається інформація, яка підлягає переробці (числа, що беруть участь в операції), а в третій має бути записаний результат. Наприклад, у машині «Стріла» команда 0001 0002 3777 01 означає: «число, що зберігається в комірці 0001, додати до числа, записаного в комірці 0002 і наслідок записати в комірку 3777». У цьому прикладі на додавання видно, що дія, яка від триадресної машини потребує лише однієї команди, в одноадресній вимагатиме аж три. Отже, на машині «Урал» подібна операція виглядатиме таким чином:

1-а команда: 02 0001 (в суматор записується число, що зберігається у комірці 0001).

2-а команда: 01 0002 (додається число, зафіксоване в комірці 0002).

3-я команда: 16 3777 (сума записується в комірку 3777).

Однак думати, що одноадресна машина потребує програми, втричі довшої від триадресної, неправильно. Є багато операцій (особливо логічних), де одній триадресній команді відповідають дві одноадресні чи навіть одна. Тому для певного класу завдань одноадресні системи навіть вигідніші. Вони дають економію часу, бо виконання однієї такої команди здійснюється швидше, ніж триадресної, не кажучи вже про економію пам'яті. Прийнято вважати, що в середньому програма в згаданих системах приблизно вдвічі довшою від такої самої програми, розрахованої на триадресну конструкцію.

Реальна робота автомата (а тим самим і програмування) зумовлюється списком команд, які він здатний виконати. Всяка модель ЕОМ має точно визначену систему команд. Нормальна кількість їх — в межах від 30 до 50. Кожна команда має свій кодівий номер. При всій відмінності у конструкції автоматів зміст команд більш-менш однаковий. Вони поділяються на три основні частини: а) команди арифметичних дій; б) команди логічних дій та в) команди управління. Перші дві не потребують спеціальних роз'яснень (додавання, віднімання, множення, ділення, віднімання модуля, формування знака і т. д.).

Команди управління повинні забезпечити нормальне функціонування автомата відповідно до поставленого завдання і виробленої програми. Це введення інформації до пам'яті, видача результату на друк, посилення числа в суматор, переведення інформації із зовнішньої пам'яті до оперативної і навпаки; засилання числа в індексний реєстр, зупинка машини тощо.

Для нас найбільш цікаві команди логічних дій; саме за їх допомогою автомат може моделювати розумові процеси. До цієї групи, зокрема, належать такі дії:

Виділення частини числа. Ця команда виконується за допомогою константи — числа, яке має одиниці в розрядах, що підлягають виділенню, а в інших розрядах — нулі. Число, записане в суматорі (S), порозрядно множиться на константу; в результаті ті розряди, що мають в ній одиниці, зберігають своє значення, в решту записуються нулі (адже $x \times 0 = 0$, а $x \times 1 = x$). Наприклад, у числі 312 652 127 потрібно виділити середню частину (розряди 4—6). Для цього використаємо константу 000 111 000. Внаслідок порозрядного множення матимемо числа 000 652 000.

Зсунення числа. Іноді в процесі логічної переробки виникає потреба перенести якусь частину числа в інші розряди. Наприклад, виділену середню частину числа 000 652 000 треба перенести з розрядів 4—6 до 7—9 (652 000 000). Цій меті служить спеціальна команда, причому зсунення може здійснюватися як вправо, так і вліво.

Формування числа. За цією командою відбувається логічне порозрядне додавання константи, записаної в комірці, що є адресою команди (a), до числа, яке міститься в суматорі (S). За її допомогою можна змінювати будь-які коди за будь-якою системою.

Порівняння чисел. Машина здійснює операцію порозрядного заперечення однозначності між числами (S) та (a). Якщо вони тотожні, то внаслідок операції вміст суматора дорівнюватиме нулю: $(S) = 0$, в решті випадків $(S) \neq 0$.

Важливими є переходи. Їх два: безумовний та умовний. Іноді ЕОМ передбачають кілька варіантів останнього.

В результаті безумовного переходу послідовна реалізація програми переривається і управління одержує команда, записана в комірці (a), що є адресою команди переходу.

Умовний перехід має різний режим роботи — залежно від наслідку попередньої операції. Конструкцією ЕОМ передбачено вироблення особливого сигналу ω (омега), значення якого може бути 0 або 1 і залежить від певної умови (здебільшого від вмісту суматора чи реєстра). При де-

яких командах, наприклад, сигнал ω спрацьовує, коли $(S) = 0$ або, навпаки, якщо $(S) \neq 0$ (скажімо, при операції порівняння чисел), або коли $(S) < 0$ і т. д. Так, якщо $\omega = 0$, команда умовного переходу передає управління наступній команді програми, а якщо $\omega = 1$, — команді, записаній в комірниці a . При постановці логічних задач ця команда є однією з найбільш уживаних.

Так само велике значення мають команди *циклічних дій*. Дуже часто певну групу операцій треба повторювати неодноразово при переробці однорідної інформації (зокрема, при так званих переборах, тобто випробуванні великої кількості кодів на ту або іншу ознаку). У таких випадках немає потреби повторювати цю групу команд в програмі стільки разів, скільки потрібно (тим більше, що кількість повторів заздалегідь відома далеко не завжди), — це привело б до неймовірного збільшення програми. Тому циклїчну групу команд записують один раз, але водночас забезпечують її повторення з модифікацією досліджуваних (чи перероблюваних) даних рівно стільки, скільки потрібно. Команди циклїчних дій базуються на різному принципі; здебільшого кожна модель ЕОМ пропонує дослідникові кілька варіантів. Вибір найдоцільнішої альтернативи залежить від характеру завдання.

Для археолога важливі й арифметичні, й логічні можливості ЕОМ. Зараз серед дослідників популярнішими є перші, хоч, поза сумнівом, другі відкривають незрівнянно ширші й далекосяжніші горизонти. Втім, незважаючи на початкову стадію в розробці проблеми, уже зараз можна говорити про два основні напрями в радянській історичній науці. Так, науковці Москви, Новосибірська та Естонії орієнтуються переважно на постановку кількісних задач і дослідження кількісних відносин, тобто в центрі уваги опиняються в основному арифметичні можливості автоматів. Натомість лєнінградська та київська група цікавляться в основному постановкою якісних завдань, для реалізації яких насамперед важливі логічні можливості.

Перший напрям на сучасному етапі розробки обіцяє більше практичних наслідків і справді може вже зараз істотно полегшити роботу археолога. Йдеться переважно про задачі статистичного класу, що з погляду методичного ніяких труднощів не становлять. По суті, методика археологічного дослідження залишається тією самою і автоматом відводиться лише роль допоміжного, технічного засобу, покликаного звільнити людину від монотонних обчислень, що потребують багату часу.

Другий напрям, незрівнянно складніший, вимагає серйозного перегляду методичних основ археологічної науки. Він передбачає автоматизацію якісних процесів дослідження, моделювання логічної основи мислення. Тут простір дії практично необмежений, але на цьому шляху дослідника чекає безліч підводних каменів, прикрих несподіванок і труднощів, які часом здаються нездоланими. Тому сьогодні навряд чи варто чекати якихось суттєвих наслідків чисто утилітарного значення — вони мають з'явитися пізніше, коли переосмислення і переоцінка методичної основи дадуть потрібний ефект.

Але перспективність цього напрямку дає підставу твердити, що дальший його розвиток має привести до кардинальної ломки усталених понять і вироблення принципово нових теоретичних засад археологічного дослідження.

Розглянемо коротко принципи використання логічних можливостей ЕОМ для потреб нашої науки. За допомогою елементарних прикладів можна наочно продемонструвати метод роботи автомата і відповідно до цього — принципи програмування.

Уявімо собі, що йдеться про логічний (типологічний, структурний і т. д.) аналіз певних археологічних комплексів, скажімо поховань. Кож-

ний конкретний комплекс характеризується певною сумою ознак — спосіб поховання (тілоспалення, тілопокладення), наявність чи відсутність курганного насипу, глибина залягання залишків від сучасної поверхні, наявність чи відсутність інвентаря (його склад — окрема проблема), слідів тризни тощо.

Деякі ознаки є специфічними для того чи іншого типу поховань. Наприклад, кремація може бути на місці або на стороні, в урні, ямці, на вогнищі і т. д. Для тілопокладення актуальна орієнтація, характерна поза небіжчика (випростана, скорчена, із зігнутими ногами), стан кістяка і т. д. Різна градація існує і в оцінці глибини (як правило, тілоспалення розміщуються значно ближче до поверхні, ніж тілопокладення). В разі наявності курганного поховання важливо зазначити рівень залягання — в насипу, на рівні денної поверхні або в ямі тощо.

Подібних ознак може бути багато. Неальтернативність більшості з них робить доцільною формалізацію матеріалу у вигляді позиційного коду. В цьому випадку дуже зручно підігнати розмір коду таким чином, щоб кожна комірочка машинної пам'яті містила інформацію про одне конкретне поховання (якщо однієї комірочки мало, треба брати дві, але бажано завантажити їх повністю).

Скажімо, 18-розрядна комірочка машини «Урал» дає можливість оперувати характеристиками, що складаються з 17 ознак кожна (один розряд, як уже відомо, є знаковим; його теж можна використати, але при цьому треба мати на увазі, що з мінусом машина матиме подекуди відмінний режим роботи).

Наявність таких ознак, актуальних лише для певної частини комплексів, робить доцільною варіацію коду для різних категорій пам'яток; в цьому випадку один і той самий розряд для тілоспалень буде означати одне, а для тілопокладень — інше. Ось який вигляд матиме позиційний код для тілоспалень:

1 розряд — тілоспалення,	10 розряд — спалення на місці,
2 » — тілопокладення,	11 » — спалення на стороні,
3 » — кенотафа,	12 » — урна,
4 » — курган,	13 » — ямка,
5 » — безкурганний,	14 » — вогнище,
6 » — глибина 0,0—0,1,	15 » — сліди тризни,
7 » — глибина 0,1—0,2,	16 » — речовий інвентар,
8 » — глибина 0,2—0,3,	17 » — безінвентарне,
9 » — глибина понад 0,3*,	18 » — знаковий.

Кожний розряд, таким чином, присвячується одній з ознак. Наявність її означає 1, відсутність — 0. Треба зразу ж підкреслити, що альтернативність 1—0 аж ніяк не може свідчити про альтернативність самих ознак навіть там, де вона має дилемний характер. Неприпустимо, зокрема, дилему «тілоспалення» ↔ «тілопокладення» (в розумінні «неспалення») або «курганне» ↔ «безкурганне» вміщувати в один розряд (1 означає «є насип», а 0 — «немає насипу»). В нашому випадку (як і в кожному позиційному коді) 0 вказує на непевність, тобто відсутність *a*, але не наявність не-*a*. Якщо в конкретному розряді, присвяченому, наприклад, курганному насипу, стоїть 0, то це означає: «немає інформації про наявність курганного насипу». В одному випадку це може бути «плоске поховання», в іншому — брак будь-яких надійних відомостей про наявність чи відсутність насипу.

Модифікація коду для тілопокладень така:

1 розряд — тілоспалення,	4 розряд — курган,
2 » — тілопокладення,	5 » — безкурганний,
3 » — кенотафа,	6 » — глибина 0,0—0,5,

* Цифри взято умовні, в кожному конкретному випадку градація може бути інша.

7 розряд	—глибина 0,5—1,0,	13 розряд	—орієнтація <i>O</i> ,
8 »	—глибина 1,0—1,5,	14 »	—порушений кістяк,
9 »	—глибина понад 1,5,	15 »	—сліди тризни,
10 »	—орієнтація <i>N</i> ,	16 »	—речовий інвентар,
11 »	—орієнтація <i>S</i> ,	17 »	—безінвентарне,
12 »	—орієнтація <i>W</i> ,	18 »	—знаковий.

Звичайно, подібних варіацій може бути багато, і зараз ми аж ніяк не прагнемо їх вичерпати. Побудова коду значною мірою залежить від характеру завдання, від мети, поставленої дослідником. Жодний шаблон тут виключається.

У деяких випадках використовується і знаковий розряд. Можна, наприклад, домовитися, що плюс означатиме «тілопокладення», а мінус — «тілоспалення»; суворо дилемний характер ознаки іноді робить це доцільним. Але при такому варіанті побудова програми матиме свої особливості.

Тепер уявімо собі, що в процесі дослідження нас цікавлять лише поховання з північною орієнтацією*. Отже, з усього масиву інформації (припустимо, з кількох сотень комплексів) треба вибрати ті, в яких є ця ознака. Завдання полягає в тому, щоб відібрати всі коди, де в розряді № 10 є одиниці. Але ж йдеться тільки про тілопокладення, бо спалення, природно, орієнтації не мають; отож, в кодах, присвячених похованням з кремацією, в 10-му розряді значиться зовсім інше (спалення на місці).

Тому перш ніж зробити остаточний висновок, автомат повинен переконатися, що має справу дійсно з інгумацією.

Реалізація цього завдання здійснюється за допомогою перебору. Машина по черзі бере коди, які підлягають перевірці, і випробовує їх на потрібну ознаку; якщо вона відсутня — відкидає, а коли наявна — бере відповідний код до уваги.

Проілюструємо цей процес на прикладі машини «Урал» — однією з найпростіших великих ЕОМ.

Перш за все автомат посилає черговий код у суматор. Для цього потрібна команда, код якої 02, а адреса — номер тієї комірки, де зберігається потрібний код комплексу (скажімо, 3001). Отже, команда матиме вигляд: 02 3001.

Решта кодів, що підлягають перевірці, записані в комірках 3002, 3003, 3004 і т. д. — до 3777; разом їх буде, таким чином, 512.

Далі автомат повинен встановити, що означає код, взятий до суматора, — тілопокладення чи щось інше. Для цього необхідно виділити розряд, присвячений цій ознаці (тобто 2), за допомогою константи 000 000 000 000 000 010 (в двоїчному численні), яка зберігається в спеціально призначеній комірці, наприклад, 0011**. Для цього служить команда «виділення частини числа», яка має код 12, адресу — 0011. В результаті одержуємо в суматорі число, де всі розряди містять нулі, крім потрібного нам 2-го розряду, — в ньому буде записано те, що міститься в первинному коді. Коли це 1 — все лишається, як є, а коли 0, тоді автомат виробить сигнал ω , який впливає на дальшу роботу програми.

* Орієнтація може бути не суворо дотриманою за сторонами світу; в такому випадку одиниці записуються в обидві комірки, що обмежують чверть компаса (наприклад, північно-східна орієнтація передбачає одиниці в комірках 10 і 13). Для спрощення в цьому прикладі «північ» розуміємо широко (включаючи сюди NO та NW), інакше довелось б робити ще додаткову перевірку коду на комірки 12 і 13 (орієнтації NS, зрозуміло, не буває).

** Подібні задачі вимагають повного набору констант, загальна кількість яких дорівнюватиме кількості розрядів стандартної комірки.

Оскільки від змісту розряду 2 залежить, що робити далі, після команди «виділення частини» ставиться команда умовного переходу. Наслідок її дії, як ми вже знаємо, зумовлений наявністю чи відсутністю сигналу ω . У першому випадку управління одержить команда, вказана в адресі умовного переходу, у другому — автомат перейде до виконання чергової команди.

Отже, за умовним переходом йде перша команда наступної операції (перевірка на ознаку «північна орієнтація»), а адресою безумовного переходу є перша команда всього циклу. Іншими словами, при відсутності сигналу ω (тобто, коли йдеться про тілопокладення) автомат продовжить роботу над цим кодом, а при наявності сигналу (якщо мається на увазі тілоспалення або кенотафа) діятиме спочатку: береться в суматор наступний код і робиться перевірка його на «тілопокладення».

Перевірка на «північну орієнтацію» (ознака ця міститься в 10-му розряді) здійснюється за тією самою схемою, тільки команда «виділення частини» містить в своїй адресі іншу комірку — ту, де зберігається потрібна константа 000 000 001 000 000 000 (нехай це буде комірка 0023). Якщо ознака виявиться відсутньою (тобто поховання матиме іншу або невідому орієнтацію), автомат знову-таки перерве роботу і звернеться до наступного коду, а коли в 10-му розряді виявиться 1, він видасть потрібний досліднику код, або його номер, або й те, й друге на друк; чи, може, запише його в потрібну комірку або просто полічить і т. ін. — залежно від мети дослідження. Дальшим етапом буде робота над іншим, наступним комплексом.

Це класичний зразок циклічної операції, що повторюється багаторазово — стільки, скільки є кодів, підлеглих перевірці (у нашому випадку — 512 разів). Тому немає ні потреби, ні можливості включати до програми кожний цикл окремо (для чого потрібно було б кілька тисяч команд). Слід скласти цей блок програми за допомогою команд циклічної дії так, щоб він працював, скільки треба.

При роботі на машині «Урал» з цією метою просто використовується індексний реєстр. До нього записується число, яке показує, скільки разів має повторюватися цикл; відповідна команда має код 25, а її адреса — число n , що на одиницю менше від кількості потрібних операцій (у нашому випадку: $777-1 = 776$)*; отже, в наведеному прикладі ця команда матиме вигляд: 25 0776.

Тепер замість звичайного умовного переходу ставимо умовний перехід за індексом (код цієї команди — 24). Якщо вміст індексного реєстра більший від нуля, ця команда передає управління тій, яка вказана в адресі, але остання виконується так, що її адреса зменшена на число, записане в індексному реєстрі. Після виконання кожної такої команди зміст реєстра автоматично зменшується на одиницю. Коли в ньому залишиться нуль, автомат перейде до реалізації чергової програми.

Отже, у нашому випадку перебір усіх 512 кодів буде здійснюватися таким чином. Уявімо, що перша команда програмного блоку (02 3001) записана в комірку 1001. Тоді команда умовного переходу за індексом має бути 24 1001. Команда, записана в комірку 1001, виконуватиметься так, ніби її адреса зменшена на число, зафіксоване в індексному реєстрі (тобто, в перший раз — 3001—776, другий — 3001—775, третій — 3001—774 і т. д., в останній раз — 3001—0). Тому першу команду слід змінити. Тепер її адресою буде не перша комірка потрібного масиву інформації, а остання, тобто 02 3777. Для забезпечення модифікації вона повинна мати знак мінус (такою є технічна умова: модифікуються тільки негативні команди).

* Восьмеричних.

В цілому весь блок програми матиме такий вигляд:

3000	25 0776	(посилка числа до індексного реєстра),
→3001	—02 3777	(посилка коду в суматор),
3002	12 0011	(виділення 2-го розряду),
3003	21 3011	(умовний перехід),
3004	—02 3777	(посилка того самого коду в суматор),
3005	12 0023	(виділення 10-го розряду),
3006	21 3011	(умовний перехід),
3007	—02 3777	(посилка того самого коду в суматор),
3010	32 0000	(видача коду на друк),
←3011	24 3001	(умовний перехід за індексом).

Таким чином, весь цикл складається з 10 команд. Але це не значить, що при перегляді кожного коду потрібно 10 тактів машинної праці; навпаки, такий варіант виключений. Перша команда (засилання числа індексу) виконується лише один раз на весь перебір. Далі довжина циклу визначається змістом коду. Тілоспалення, наприклад, потребуватиме чотири такти команди (3001, 3002, 3003 та 3011), а тілопокладення з не-північною орієнтацією — семи тактів. І лише поховання з кістяком, орієнтованим на північ (тобто те, що потрібно досліднику), примусить машину виконати дев'ять команд.

«Урал-1» — найповільніша ЕОМ — виконує 100 команд за 1 сек; переважна більшість автоматів, в тому числі й наступні моделі «Уралу», працюють набагато швидше. Отже, за найгірших умов, витрата часу на цикл не досягатиме 0,1 сек. Це значить, що весь масив інформації з 512 кодів «Урал-1» переробить за 30 сек, включаючи і видачу на друк (якщо б усі комплекси виявилися з північною орієнтацією, обробка продовжувалася 45 сек). Машина з середньою швидкістю дії (10 000 команд на 1 сек) виконає подібне завдання менше ніж за 1 сек, але тут реальним обмеженням буде друкування наслідків.

У наведеному прикладі ми свідомо обрали не найкоротший шлях, маючи на меті розкриття послідовності логічної думки. На практиці досвідчений програміст об'єднає обидві ознаки в одну константу. Теоретичною основою цього є можливість подати умову задачі у вигляді кон'юнкції ознак: $X = a \wedge b$, де X — комплекс, що нас цікавить, a — тілопокладення, b — північна орієнтація.

Суть цього варіанта полягає у тому, що спочатку машина побудує код — константу кон'юнкції, а далі перебір масиву інформації здійснюватиме за допомогою отриманого результату.

Як зазначалось, у розв'язанні задачі беруть участь дві константи — 000 000 000 000 000 010 та 000 000 001 000 000 000.

Іх можна з'єднати методом простого додавання:

$$\begin{array}{r} 000\ 000\ 000\ 000\ 000\ 010 \\ +\ 000\ 000\ 001\ 000\ 000\ 000 \\ \hline 000\ 000\ 001\ 000\ 000\ 010 \end{array}$$

Отже, в скороченому вигляді програма буде така:

3000	25 0776	(посилка числа в індексний реєстр),
3001	02 0011	(посилка в суматор константи),
3002	01 0023	(додавання другої константи),
3003	16 0007	(посилка нової константи в пам'ять),
→3004	—02 3777	(посилка коду в суматор),
3005	12 0007	(виділення частини—обох розрядів),
3006	14 0007	(порівняння з константою),
3007	21 3012	(умовний перехід),
3010	—02 3777	(посилка того ж коду в суматор),
3011	32 0000	(видача на друк),
←3012	24 3004	(умовний перехід за індексом).

Правда, тепер у загальному вигляді блок складається з 11 команд (замість 10), але чотири з них виконуються лише один раз до початку

перебору. Потрібний код вимагає семи тактів (замість дев'яти), а непотрібний — п'ять (замість семи або чотирьох). В підсумку економія часу буде досить помітна.

Розглянемо ще один варіант програмування тієї самої задачі — у випадку, коли є можливість використати знаковий розгляд для кодування інформації. Таким шляхом можна фіксувати ще одну ознаку — момент, дуже важливий з огляду на обмеженість оперативної пам'яті ЕОМ.

Для запису у знаковий розряд придатні лише суворо дилемні ознаки. Так, якщо могильник добре досліджений і в ньому представлені лише тілопокладення і тілоспалення, причому наявна інформація має вичерпний характер (отже, не буде непевності в жодному випадку), ми можемо прийняти такий варіант коду, в якому мінус означатиме тілоспалення, а плюс — тілопокладення.

При засиланні до суматора конкретного коду, присвяченого похованню, машина буде виробляти (або не виробляти) сигнал ω , бо при команді 02 ситуація (S) ≤ 0 є умовою $\omega = 1$. Завдяки цьому відпадає потреба спеціального випробовування на ознаку «тілопокладення» і зразу ж після посліжки коду в суматор можна ставити команду умовного переходу. Програма спрощується.

	3000	25 0776	(посилка числа в індексний реєстр),
→	3001	-02 3777	(посилка коду в суматор),
	3002	21 3007	(умовний перехід),
	3003	12 0023	(виділення 10-го розряду),
	3004	21 3007	(умовний перехід),
	3005	-02 3777	(посилка того самого коду в суматор),
	3006	32 0000	(видача на друк),
←	3007	24 3001	(умовний перехід за індексом).

Таким чином, програма скорочується на дві команди, крім того, звільнюються розряди 1 та 2, в які можна записати дві зайві ознаки — елементи комплексу.

Цей елементарний приклад демонструє принципи дискретної дії автомата. Сукупність команд, реалізованих машиною, забезпечує можливість здійснення будь-якої операції, моделювання будь-якого розумового процесу. Кожний висновок (який тільки має суворо логічну основу) може бути представлений в цілком формалізованому вигляді і, отже, введений до ЕОМ. Теоретичних обмежень тут не існує; єдиним лімітом є тільки практична сторона.

Наведений приклад є характерним зразком логічного блоку програми. Він покликаний забезпечити виконання автоматом однієї конкретної операції (виборки комплексів з певною ознакою з нерозчленованого масиву інформації). Нормально кожне дослідницьке завдання складається з цілої серії таких блоків. Отже, відповідно і вся програма повинна мати аналогічну блок-схему, яка є формалізованим виявом прийнятого алгоритму. Саме з її побудови слід починати складання програми.

Тепер розглянемо більш складну ситуацію, що передбачає елементи навчання і логічного аналізу. Як приклад візьмемо реалізований нами експеримент по автоматизації бібліографічного пошуку в галузі археології України.

Цей експеримент полягає у тому, щоб довідки, які видаватиме автомат, мали не формальний, а осмислений характер. Пошук провадився по кількох параметрах, у різних сферах знання (тип пам'ятки, її культурна належність, територія, хронологія тощо). Звичайно, завдання легко розв'язується на термінологічному рівні: скажімо, коли потрібна література, присвячена Київській Русі, ЕОМ без будь-яких труднощів відбере коди з терміном «Київська Русь». Але цього мало, бо певна кількість бібліо-

графічних позицій (книг, статей, заміток) може бути закодована на інші терміни, що, однак, матиме пряме відношення до теми запиту. Наприклад, якщо у коді стоїть «давньоруське городище IX—X ст.», то він, безперечно, належить до числа потрібних, хоч термін «доба Київської Русі» у ньому відсутній.

Аналогічно в географічній номенклатурі поняття «Поросся» є структурною частиною більш широкого — «Правобережний Лісостеп», а останнє — ще ширшого, «Правобережжя». Отже, якщо дослідника цікавлять пам'ятки, розташовані на Правобережжі, то позиції, закодовані на «Поросся», мусять бути видані автоматом на друк. Те саме — з типом пам'ятки або її інвентарем: городище є конкретною формою поселення, а, скажімо, наральник — різновидом більш загального поняття «сільськогосподарські інструменти». Тому насамперед потрібно навчити автомат хронології, географії, археології.

Робиться це таким чином. До пам'яті ЕОМ вводиться словник, який встановлює зв'язки між термінами (і відповідними їм поняттями). В нашому випадку був застосований метод формалізації матеріалу через розкриття складу; інформацію зручно подавати у вигляді діз'юнкції типу: $x = a \vee b \vee c \dots \vee n$, де x — загальне поняття, a, b, c і т. д. — його структурні частини (практичне застосування теорії множин).

Для характеристики бібліографічних позицій був обраний анкетний метод кодування. Повна 36-розрядна комірка машини «Урал» (для експерименту використана саме ця модель) була розбита на кілька зон, кожна з яких присвячувалася конкретній ознаці (тип пам'ятки, культура, хронологія і т. п.); іншими словами, відповідна інформація записувалася в точно визначені заздалегідь комірки пам'яті (наприклад, географічні дані — в комірках 19—30).

Для ілюстрації роботи автомата візьмемо одну з ознак — хронологічну, скориставшись кодом, опублікованим нами в «Археології» (т. XXI, стор. 41—43). Цей код має чотиріступеневу ієрархію термінів, як, наприклад, кам'яний вік — палеоліт — нижній палеоліт — шель. Тут 81 термін, яким зручно присвоїти порядкові восьмеричні номери від 001 до 121. При цьому нумерація йде від вищого ступеня (кам'яний вік, бронза, залізо) до другого (палеоліт, енеоліт, гальштат тощо), потім третього і т. д.

Застосовуючи згаданий вище метод формалізації через розкриття складу, одержуємо, наприклад: $004 = 014 \vee 015 \vee 016$, де 004 означає палеоліт, 014 — нижній палеоліт, 015 — середній, а 016 — верхній палеоліт. Аналогічним чином: $014 = 037 \vee 040 \vee 041$, де 014 — нижній палеоліт, 037 — дошель, 040 — шель, 041 — ашель.

Такі формули вводяться до пам'яті автомата. Короткі комірки машини «Урал» мають 18 двоїчних розрядів або 6 восьмеричних. Це дає можливість в кожну комірку записати два тризначні восьмеричні числа (при цьому перша цифра першого числа не повинна перевищувати 3, щоб лишити вільним знаковий розряд — саме та умова, якій задовольняє наш код). Кожну з одержаних формул, отже, записуємо в три комірки: 014 037; 014 040; 014 041.

Уявімо тепер, що дослідника цікавить література, присвячена палеоліту. Тоді йому потрібні всі коди, де значиться не лише «палеоліт», а й «верхній палеоліт», «нижній палеоліт», «шель», «мустье», «солютре» і т. п. Отже, діставши такий запит, автомат перед тим, як перейти до перегляду основного масиву інформації, мусить відібрати всі терміни, які підходять до теми запиту, по всіх ознаках.

Коли черга доходить до хронології, автомат спочатку бере в суматор код запиту і виділяє ту його зону, де містяться хронологічні відомості (це можуть бути розряди 10—18); одержаний наслідок записується в спеціально призначену комірку пам'яті. Далі починається аналіз даного терміна, він полягає у відборі всіх інших, які своїм змістом пов'язані з терміном запиту. Машина по черзі бере коди частини словника, при-

свяченої датам, і випробовує їх. Записавши в суматор черговий словниковий код, вона спочатку виділяє першу його половину (розряди 10—18) і порівнює з наявним кодом запиту. Якщо числа збігаються, комп'ютер, взявши другу половину, зсуває її на 9 розрядів вліво (тобто переносить новоодержаний код в розряди 10—18), а наслідок фіксує в одній з призначених для цього комірок (кількість останніх повинна бути достатньою для того, щоб записати максимально потрібне число кодів).

В результаті аналізу терміна «палеоліт» (004) будуть відібрані нижній палеоліт (014), середній палеоліт (015) і верхній палеоліт (016). Далі шляхом такої самої операції щодо кожного з них відбираються потрібні терміни третього ступеня (шель, дошель, ашель, мустье, оріньяк, солютре, мадлен).

Завершивши цей процес, автомат матиме вичерпну інформацію для пошуку — кожна бібліографічна позиція тепер буде за кожною ознакою перевірятися не на один, на всі відібрані коди.

Теоретично автомат здатний виконати будь-яку логічну операцію. Але практично звертатися до послуг ЕОМ доцільно в тих випадках, коли дослідження пов'язане з переробкою (особливо багаторазовою — по різних параметрах) великого масиву монотонної інформації, або тоді, коли складна сукупність ознак може стати причиною логічних помилок дослідника.

М. Ю. БРАЙЧЕВСКИЙ

Археология и кибернетика

II. Применение больших ЭВМ в археологическом исследовании

Резюме

Статья посвящена изложению принципов работы больших электронно-вычислительных машин (ЭВМ) и принципов программирования применительно к археологической проблематике. Автор придерживается точки зрения, согласно которой наиболее перспективным направлением является исследование качественных отношений с использованием логических возможностей автоматов. Реализация этой задачи требует серьезного пересмотра методических основ археологической науки.