

## ТЕМПОРАЛЬНІ ПРОЦЕДУРИ ТА АЛГОРИТМИ

В. В.Зубенко

Київський національний університет ім.Тараса Шевченка  
01033, м.Київ, вул.Володимирська, 64  
т/факс: (044) 259 0519, e-mail: vvz@unicyb.kiev.ua

Уводиться поняття темпоральної процедури та алгоритму як засобів подання функціональних співвідношень в абстрактних динамічних системах. Розглядається проблема синтезу та аналізу таких процедур та алгоритмів.

The notions of temporal procedure and algorithm are proposed and studied. The periodical temporal algorithms design and analyze are considered.

## Темпоральні процедури

Темпоральні процедури та алгоритми, введені в [1], призначені для обчислення часових багатозначних функцій в абстрактних динамічних системах [2,3]. Класична теорія алгоритмів, теоретичне програмування з метою більшої прозорості та спрощення моделей, як правило, залишають поза увагою темпоральні елементи обчислень. Деяким виключенням є хіба що питання складності алгоритмів та програм. Але практика програмування, задачі інформаційного моделювання вимагають уточнення та вивчення саме не “оголених”, а реальних темпоральних моделей обчислень та обчислювальності. Особливістю підходу є те, що обчислення розглядаються не в просто часі, а в “подвійному” часовому просторі з урахуванням, так званих, локального і глобального часу. І цей простір не є “декоративним”, як наприклад в автоматних системах [3], а він може активно “впливати” на хід обчислень. Окрім цього обчислення локально можуть без обмежень повертатись “назад” або й просто “застигати” у просторі. Іншою особливістю є те, що темпоральні алгоритми вводяться не як традиційно первинне “наївне” поняття з певними “джентльменським” набором характеристичних ознак і властивостей, а як похідне від таких математичних понять, як множина, темпоральна процедура, конструктивний об’єкт.

Наведемо основні поняття, пов’язані з темпоральними процедурами. Тлумачення деяких з них дещо відрізняється у порівнянні з [1]. Нехай  $S$  – довільна множина станів,  $\langle \Gamma, \mathcal{D} \rangle$  – лінійно впорядкована сукупність, яку будемо трактувати як глобальний часовий простір, а  $T$  – локальний часовий простір. Він теж може бути лінійно впорядкованим, але це не є обов’язковою умовою. Позначимо  $t++$  та  $-t$  безпосередньо наступний та попередній моменти часу. Вважається, що в часових просторах немає найменшого та найбільшого елементів. На локальному часі  $T$  можуть бути визначені й інші структури. Наприклад, структура адитивної абелевої групи або лінійного простору і т.п. Це дозволяє маніпулювати з часом і відстежувати локальну поведінку систем в тих чи інших часових розрізах. Як типові приклади локальних часових перетворень можна навести: облік сумарного часу виконання всіх чи окремої групи операцій обчислення, часові зсуви “вперед” та “назад”  $t \pm t_0$  (переходи до іншого часового поясу), лінійні «стиснення» та «розширення» часу  $k\tau$ , обернення часу  $\tau^{-1}$ , переходи взагалі до іншого часового простору і т.д. Трійка  $\Pi = \langle \Gamma, T, S \rangle$  називається *обчислювальним простором*, елементи декартових добутків  $T \times S$  та  $\Gamma \times (T \times S)$  – *конфігураціями* простору. Конфігурації  $(t, s) \in T \times S$  та  $(\tau, (t, s)) \in \Gamma \times (T \times S)$  будемо позначати відповідно  $s|_t$  та  $s|_t^\tau$ . Нехай  $\tau, \tau_1, \tau_2 \in \Gamma$ . Позначимо  $\Gamma_\tau = \{\tau' \in \Gamma \mid \tau \leq \tau'\}$ ,  $\Gamma_\tau^+ = \Gamma_\tau \setminus \{\tau\}$ ,  $\Gamma_\tau^- = \Gamma \setminus \Gamma_\tau$ ,  $[\tau_1, \tau_2] = \{\tau : \tau_1 \leq \tau \leq \tau_2\}$ . Аналогічні позначення будемо використовувати і для локального часового простору  $T$ . *Процесами* з початком в момент часу  $\tau$  називаються відображення вигляду  $a : \Gamma_\tau \rightarrow T \times S$ . *Процесами* на інтервалі  $[\tau_1, \tau_2]$  називаються відображення вигляду  $a : [\tau_1, \tau_2] \rightarrow T \times S$ . Багатозначні часткові функції надалі будемо називати співвідношеннями. Співвідношення вигляду  $\mathcal{D} : \Gamma \times T \times S \mapsto T \times S$  називаються *функціями переходу*, а пари вигляду  $P = \langle \Pi, \mathcal{D} \rangle$  – *темпоральними процедурами* над простором  $\Pi$ . У разі, коли обчислювальний простір  $\Pi$  зафіксовано, темпоральні процедури будемо ототожнювати з їх функціями переходу. Кожна процедура породжує певну сукупність процесів у глобальному часовому просторі, які називаються обчисленнями. А саме нехай  $\tau \in \Gamma$ ,  $t \in T$  та  $s \in S$  – довільні моменти локального часу та стан. *Обчислення*  $a$  за процедурою  $P$  з початком в момент  $\tau$  та початковою

конфігурацією  $s|_t$  визначається рекурентно:  $a(\tau) = s|_t$  і  $a(\zeta) \in \delta(\zeta, a(-\zeta))$  для всіх  $\zeta \in \Gamma_\tau^+$ . Будемо говорити, що обчислення  $a$  в момент часу  $\zeta$  переходить від конфігурації  $a(-\zeta)$  до конфігурації  $a(\zeta)$  і позначати цей факт  $a(-\zeta) \rightarrow_p a(\zeta)$ . Таких переходів в момент часу  $\zeta$  може бути кілька, оскільки функція переходу, взагалі кажучи, багатозначна. Обчислення  $a$  за процедурою  $P$  на інтервалі  $[\tau_1, \tau_2]$  з початковою конфігурацією  $s|_t$  визначається як обмеження відповідного обчислення  $a$  за процедурою  $P$  з початком в момент  $\tau_1$  на інтервал  $[\tau_1, \tau_2]$ . Будемо говорити, що обчислення  $a$  в цьому разі розпочинається конфігурацією  $a(\tau_1)$  і закінчується конфігурацією  $a(\tau_2)$  і позначати цей факт  $a(\tau_1) \Rightarrow_p a(\tau_2)$ . Конфігурація  $a(\tau_2)$  називається заключною, а її стан – результатом обчислення  $a$  на інтервалі  $[\tau_1, \tau_2]$ . Якщо звернутись до загальних динамічних систем, то процедурні системи є різновидом 2-компонентних неавтоматних динамічних систем [3], першій компоненті яких відведено спеціальну роль – локального часового простору.

Нас цікавлять не стільки самі по собі процедури та обчислення, як засоби їх опису та конструювання, а також способи реалізації обчислень. У зв'язку з цим зауважимо, що з функцією переходу пов'язані дві сукупності перетворень – локального часового простору та множини станів. А саме, якщо зафіксувати певний стан  $s_0 \in S$  та моменти часу  $\tau_0 \in \Gamma$ ,  $t_0 \in T$ , то їм відповідають співвідношення  $p_{\tau_0, s_0} : T \mapsto T$  та  $f_{\tau_0, s_0} : S \mapsto S$  такі що, для довільних  $t, h \in T$ ,  $s, u \in S$  :  $p_{\tau_0, s_0}(t) = h$ ,  $f_{\tau_0, s_0}(s) = u$ , якщо існують такі  $h_1 \in T$ ,  $u_1 \in S$ , що  $u_1|_h \in \delta(\tau_0, t, s_0)$  та  $u|_{h_1} \in \delta(\tau_0, t_0, s)$ . Вірно і обернене. Якщо кожній трійці  $(\tau, t, s)$  поставити у відповідність одну або кілька пар відображень  $(p_{\tau, s} : T \rightarrow T, f_{\tau, t} : S \rightarrow S)$ , то цим самим буде визначена певна функція переходу, а саме така, що  $(p_{\tau, s}(t), f_{\tau, t}(s)) \in \delta(\tau, t, s)$ . Щоб спростити опис функції переходу та наблизитись до засобів реалізації реальних обчислювальних процесів, здійснимо певну інтенціоналізацію підходу. А саме розділимо процес переходу від однієї конфігурації обчислення до наступної на дві окремі фази: з'ясування, яка саме дія має відбутися над конфігурацією, та саму дію.

Для цього зафіксуємо  $\Theta = \{p_\alpha : T \rightarrow T \mid \alpha \in A\}$  та  $\Delta = \{f_\beta : S \rightarrow S \mid \beta \in B\}$  - певні сукупності стандартних (елементарних) перетворень моментів часу та станів. Будемо їх вважати однозначними і долучимо до обчислювального простору  $\Pi = \langle \Gamma, T, S, \Theta, \Delta \rangle$ . Нову функцію переходу над простором  $\Pi$  визначимо у

вигляді  $\delta : \Gamma \times T \times S \rightarrow 2^{\Theta \times \Delta}$ . Її дія тепер обмежується тільки першою фазою переходів конфігурацій. Друга ж фаза “відірвана” від першої і може подаватись та реалізовуватись незалежно від неї, у тому числі операційно, наприклад, у вигляді викликів відповідних алгоритмів, підпрограм і т.п. Обчислення  $a$  за процедурою  $P$  з початком в момент часу  $\tau$  та початковою конфігурацією  $(s_0, t_0)$  визначається рекурентно:  $a(\tau) = (s_0, t_0)$  і для всіх  $\zeta > \tau$   $a(\zeta) = (p_\zeta(t), f_\zeta(s))$ , де  $(p_\zeta, f_\zeta) \in \delta(\zeta, a(-\zeta))$ , а  $t$  та  $s$  – компоненти конфігурації на попередньому кроці, тобто  $a(-\zeta) = s|_t$ . Обчислення  $a$  за процедурою  $P$  на інтервалі  $[\tau_1, \tau_2]$  визначається як обмеження відповідного обчислення  $a$  на інтервал  $[\tau_1, \tau_2]$ . Зазначимо, варіантів пар  $(p_\zeta, f_\zeta)$  на даному кроці обчислення може бути: а) кілька (у випадку недетермінованих процедур), б) рівно одна (у випадку детермінованих процедур) і в) жодної (обчислення зупиняється). Як правило, нас будуть цікавити не всі можливі обчислення в процедурі, а насамперед ті, що розпочинаються з певних виділених вхідних локальних конфігурацій і закінчуються певними виділеними вихідними локальними конфігураціями. Тому введемо поняття ініціальної процедури  $P(C_0, C_{fin})$  над простором  $\Pi$  як трійки  $\langle P, C_0, C_{fin} \rangle$ , де  $P$  – певна процедура над простором  $\Pi$ ,  $C_0 \subseteq T \times S$  та  $C_{fin} \subseteq T \times S$  – виділені підмножини відповідно вхідних та вихідних конфігурацій або як четвірки  $P = \langle \Pi, C_0, C_{fin}, \delta \rangle$ . Обчислення, що розпочинається з певної початкової конфігурації

$a(\tau_0) = s|_t \in C_0$  і закінчується певною вихідною конфігурацією  $a(\tau) = u|_r \in C_{fin}$  і при цьому всі проміжні конфігурації якого не є вихідними, назвемо результативним. Конфігурація  $a(\tau)$  називається заключною, а її стан  $u$  – результатом обчислення на  $a(\tau_0)$ . Заключну конфігурацію позначатимемо  $P^*(a(\tau_0))$ , а результат обчислення –  $P(a(\tau_0))$ . Про обчислення  $a(\tau_0) \Rightarrow_p a(\tau)$  таке, що  $a(\tau_0) \in C_0$ ,  $a(\tau) \in C_{fin}$ , будемо говорити як про гіперрезультативне. В такому обчисленні вихідні конфігурації можуть зустрічатися і серед проміжних. Заключна його конфігурація позначається  $P^{**}(a(\tau_0))$ . Нескінченні обчислення називаються, як правило, безрезультатними. Безрезультатними називатимемо і ті скінченні обчислення, які не мають жодного результативного продовження. В деяких випадках доречно вважати результативними і ті скінченні обчислення, на заключних

конфігураціях яких не визначена функція керування. Нехай  $S_0$  та  $S_{fin}$  – сукупності всіх станів, що належать відповідно вхідним та заключним конфігураціям процедури. Кожна ініціальна процедура  $P(C_0, C_{fin})$  визначає певні багатозначні часові оператори на вхідних конфігураціях  $P^* : \Gamma \times C_0 \mapsto C_{fin}$ ,  $P^{**} : \Gamma \times C_0 \mapsto C_{fin}$ , а саме такі, що їх значення в момент  $\tau_0 \in \Gamma$  на аргументі  $s|_t \in C_0$  складають заключні конфігурації всіх результативних та гіперрезультативних обчислень з початком в момент  $\tau_0$  на  $s|_t \in C_0$ . Позначимо  $P_\tau^* : C_0 \rightarrow C_{fin}$  та  $P_\tau^{**} : C_0 \rightarrow C_{fin}$  результат підстановки в оператори  $P^*$  та  $P^{**}$  часової константи  $\tau \in \Gamma$  замість першої компоненти і  $P_{\tau,t} : S_0 \rightarrow S_{fin}$  – проекцію по другій компоненті результату підстановки в оператор  $P_\tau^*$  константи  $t \in T$  замість першої компоненти. Значеннями оператора  $P_{\tau,t}$  на початковому стані  $s \in S$  стани заключних конфігурацій відповідних результативних обчислень за процедурою  $P$ , що розпочинаються в фіксований момент глобального ( $\tau$ ) та локального ( $t$ ) часу. Про оператори  $P^*$ ,  $P^{**}$ ,  $P_\tau^*$ ,  $P_\tau^{**}$  та  $P_{\tau,t}$  будемо говорити як про такі, що їх обчислює процедура  $P$ . Перші чотири з них є темпоральними, а останні – перетвореннями множини станів. Окремо виділимо ініціальні процедури, що закінчують обчислення виключно за часовими (глобальними чи локальними) критеріями. У першому випадку можуть навіть не фіксуватись заключні конфігурації процедури, а замість них може вводиться, наприклад, певна топологія на сукупності всіх локальних конфігурацій і поняття границі процесу як результату обчислення. У другому випадку сукупність заключних конфігурацій має вигляд  $C_{fin} = T' \times S$  для деякого  $T' \subseteq T$ . Наприклад, процедури що закінчують роботу в певний момент часу  $h$  ( $C_{fin} = \{h\} \times S$ ) або після деякого моменту часу  $h$  ( $C_{fin} = T_h^+ \times S$ ) і т.д. Процедури з однозначними функціями керування будемо називати *детермінованими*. Для таких процедур оператор  $P^*$  є також однозначним. Процедури  $P$  та  $Q$  називаються еквівалентними ( $P \equiv Q$ ), якщо оператори, які вони обчислюють, співпадають як функціональні відношення. Процедури  $P$  та  $Q$  називаються  $\tau$ -еквівалентними ( $P \cong_\tau Q$ ), якщо  $P_\tau^* = Q_\tau^*$ . Процедури  $P$  та  $Q$  називаються  $(\tau, t)$ -еквівалентними ( $P \cong_{\tau,t} Q$ ), якщо  $P_{\tau,t} = Q_{\tau,t}$ . Процедури  $P$  та  $Q$  називаються 2-еквівалентними відносно моменту часу  $\tau$  ( $P \cong_2 Q$ ), якщо  $P_{\tau,t} = Q_{\tau,t}$  для всіх  $t \in T$ . Стан ініціальної процедури  $P(C_0, C_{fin})$  назовемо допустимим, якщо він входить до конфігурації деякого результативного її обчислення. Позначимо  $S_{acc} \subseteq S$  підмножину всіх допустимих станів процедури  $P(C_0, C_{fin})$ . Очевидно, з точки зору операторів, що обчислює ініціальна процедура, стани за межами підмножини  $S_{acc}$  є “зайвими”. Нехай  $P'(C'_0, C'_{fin})$  – ініціальна процедура над простором  $\Pi'$  з множиною станів  $S_{acc}$  та з обмеженими на нього: елементарними операторами, функцією переходу та вхідними і вихідними конфігураціями процедури  $P$ . Очевидно, процедури  $P$  та  $P'$  еквівалентні. Процедура, всі стани якої допустимі, називається *приведеною*. Процедуру назовемо *монотонно зростаючою* (спадуючою), якщо всі стандартні часові перетворення є монотонно зростаючими (спадуючими) і *монотонно не спадуючою* (не зростаючою), якщо всі стандартні часові перетворення є монотонно не спадуючими (не зростаючими). Монотонно зростаючі (спадуючі) процедури будемо називати просто *монотонними*, а монотонно не спадуючі (не зростаючі) – *слабко монотонними*.

## Темпоральні алгоритми

Як уже зазначалось, процедури цікавлять нас не самі по собі, а насамперед в контексті їх побудови (програмування) з метою реалізації або моделювання. Коли говориться про реалізацію процедур чи окремої процедури, то мають на увазі існування певного Виконавця обчислень, реального фізичного чи гіпотетичного, здатного фіксувати окремі конфігурації обчислення, знаходити значення функції переходу на кожній з конфігурацій та здійснювати відповідні елементарні перетворення компонентів. Вважається, що Виконавець володіє певною системою для подання (побудови) конфігурацій та правил для проведення елементарних перетворень і обчислення значень функції переходу. Існування та використання подібних правил є принциповим моментом, оскільки елементарні перетворення та функція переходу в більшості своїй є нескінченними, а от правила їх обчислення можуть бути подані в підходящій системі фінітно у вигляді спеціальних об'єктів, наприклад алгоритмів чи програм і т.д. Дана обставина є принциповою з точки зору фізичної реалізації (моделювання) обчислень. Якщо всі об'єкти системи подання є скінченними, то про таку систему будемо говорити як про *конструктивну*, а її об'єкти називати *конструктивними*. Про конструктивність об'єктів (функцій, конфігурацій) можна говори-

ти, наприклад, у випадку алгебричних об'єктів. Для функцій це означає, що вони побудовані з певних базових функцій за допомогою конструкторів-композицій і відомі правила обчислень для найпростіших складових функцій та результатів композицій. Наприклад, як це має місце у випадку арифметичних виразів. Алгебричні об'єкти мають і іншу важливу властивість – їх можна будувати (конструювати, програмувати) зокрема у вигляді слів-термів з базових сигнатурних елементів. Зазначимо, що наше тлумачення системи подання та конструктивності до певної міри інтуїтивне і ми свідомо на даному рівні уникаємо розгляду інших, не алгебричних (в тому числі і генетично обумовлених [4]) структур, пов'язаних з конструктивністю. Говорять, що ініціальна процедура  $A$  є *конструктивною* щодо певної системи подання, якщо її конфігурації та функція керування є конструктивними в цій системі. Конструктивні процедури будемо називати *темпоральними алгоритмами* ( $T$ -алгоритмами). Оператори  $A^*$ ,  $A^{**}$ ,  $A_\tau^*$ ,  $A_\tau^{**}$ ,  $A_{\tau,t}$ ,  $\tau \in \Gamma$ ,  $t \in T$ , що обчислюються темпоральним алгоритмом  $A$ , називаються *алгоритмічно обчислювальними*. Як і процедури, темпоральні алгоритми можуть бути детермінованими і недетермінованими. Наведемо основні властивості темпоральних алгоритмів, що безпосередньо впливають з їх означення.

*Масовість.*  $T$ -алгоритм може бути застосовано до будь-якої вхідної конфігурації з певної їх сукупності.

*Темпоральність.* Обчислення  $T$ -алгоритму відбуваються у глобальному та локальному часових просторах.

*Елементарність.* На кожному кроці обчислень виконуються певні елементарні оператори з фіксованих сукупностей таких операторів ( $\Theta$  та  $\Delta$ ).

*Визначеність.* Порядок застосування елементарних операторів в обчисленні не є довільним, а визначається функцією переходу за поточними моментом часу та конфігурацією обчислення.

*Результативність.* Є механізм завершення обчислень.

*Фінітність.* Скінченність подання конфігурацій та функції переходу як конструктивних об'єктів.

*Релятивність.* Відносність поняття  $T$ -алгоритму. Конструктивність процедури тісно пов'язана з системою подання, і процедура, будучи  $T$ -алгоритмом відносно даної системи подання, не обов'язково буде  $T$ -алгоритмом відносно якоїсь іншої системи. Іншим аспектом релятивності алгоритмів є взаємовідносини елементарних операторів та системи подання. В деяких випадках елементарні оператори (всі чи їх частина) можуть бути неконструктивними відносно системи подання. Про такі оператори говорять як про *оракули*, що знаходяться за межами системи подання, а про самі алгоритми – як про алгоритми з оракулами.

Як бачимо, основні властивості класичних числових та словарних алгоритмів притаманні й темпоральним алгоритмам. Але в останніх з'явилися і нові специфічні риси – темпоральність та релятивність. Перша узагальнює дискретність класичних алгоритмів та “прив'язує” їх обчислення до локального часового простору, друга акцентує увагу на відносному характері поняття алгоритму, на його до певної міри «несамостійності» й залежності від тієї чи іншої системи подання. Оскільки темпоральні алгоритми обчислюють і позачасові оператори, то вони можуть слугувати платформою для уточнення і загального поняття обчислювального оператора в довільній області. Важливо, що цей шлях є прямим і не спирається на ті чи інші універсальні числові чи словарні моделі таких алгоритмів. Фіксуючи конкретні обчислювальні простори та вибираючи конкретні системи подання, можна отримати весь спектр класичних алгоритмічних систем (машини Тьюрінга, алгоритми Маркова, алгоритми Колгомова-Успенського, дискретні перетворювачі, різні класи схем програм, формальні граматики та числення і інші). Так, у випадку машин Тьюрінга станами їх процедурних еквівалентів виступають стани робочої стрічки машини (=слова в робочому алфавіті з відміченою позицією робочої комірки), локальний часовий простір  $T$  – скінчений і його складають внутрішні стани машини,  $\Gamma = N$ , значення функції переходу на конфігурації визначається локальним часом та станом поточної комірки на стрічці. При цьому це значення ніяким чином не залежить від номера кроку (глобальної часової компоненти!) обчислення. Загальний варіант темпоральної машини Тьюрінга можна отримати на основі 2-стрічкової її моделі, якщо одну зі стрічок використовувати як звичайну робочу, а іншу – як лічильник натуральних чисел у якості локальної часової компоненти. Команди такої темпоральної квазімашини Тьюрінга враховують (та змінюють) стани поточної комірки, робочої голівки та лічильника. З відомих алгоритмічних систем найбільш природну процедурну інтерпретацію мають дискретні перетворювачі В.М.Глушкова. Як відомо, дискретний перетворювач є системою двох з'єднаних між собою автоматів: керуючого  $X - Y$ -автомата Мілі та операційного  $Y - X$ -автомата Мура. Операційний автомат здійснює перетворення своїх поточних станів (станів інформаційного середовища перетворювача) та посиляє вхідні сигнали керуючому автомату про стан інформаційного середовища. Останній, виходячи з цієї інформації та зі свого поточного стану, вибирає наступне елементарне перетворення для операційного автомата. Процедурний еквівалент дискретного перетворювача вхідні сигнали керуючого автомата інтерпретує як локальні часові константи (тобто  $X = T$ ), глобальний часовий простір  $\Gamma$  є таким, як і у випадку машин Тьюрінга, –  $N$ . Стани процедури – 2-компонентні і складаються зі станів керуючого та операційного автоматів. Функція переходу реалізує функцію керуючого автомата, за виключенням безпосереднього перетворення його внутрішнього стану. Вона тільки передає інформацію про таке перетворення та про відповідні перетворення операційного середовища та локального часу. Зазначимо, що функції переходу машини Тьюрінга чи дискретного перетворювача є автоматними, тобто наступні їх дії не залежать від моменту часу (глобального) та попередньої історії обчислення. Якщо звернутись до таких систем як, наприклад, ігрові числення, то в їх правилах та стратегіях

ях часові фактори відіграють дуже важливу роль. Це означає, що вони як динамічні системи є темпоральними. В багатьох іграх важливим локальним фактором є черговість ходу. Але є й інші принципові обставини, пов'язані з темпоральністю. Наприклад, в шахах рокіровка короля залежить від того, чи рухався він до цього. На стратегію вибору чергового ходу може впливати обмеженість часу, відведеного на шахову партію (гра в цейтноті), і т.д.

## Періодичні алгоритми

Одним із шляхів до спеціалізації та вивчення тих чи інших підкласів темпоральних процедур та алгоритмів є факторизація областей визначення функцій переходу. Розглянемо як приклад, так звані, періодичні алгоритми. Нехай далі локальний часовий простір є ізоморфним адитивній групі  $Z$  цілих чисел і нехай  $P$  – довільна процедура над простором  $\Pi = \langle \Gamma, Z, S, \Delta, \Theta \rangle$ ,  $\pi$  – певне відношення еквівалентності на станах з  $S$ . Факторизуємо функцію переходу за часовими аргументами та станами. Зафіксуємо деяку локальну часову константу  $k$ . Будемо говорити, що функція переходу  $\delta$  є періодичною по модулю  $\pi$  з періодом  $k$  (або просто  $k$ -періодичною по модулю  $\pi$ ), якщо для будь-яких еквівалентних станів  $p$  та  $s$  та довільних моментів часу  $\tau, \tau' \in \Gamma$ ,  $t \in Z$  виконується  $\delta(\tau, t, p) = \delta(\tau', t \pm k, s)$ . З періодичності функції  $\delta$  випливає, що вона фактично не залежить від глобального часу, тобто в будь-який момент часу вибір чергових перетворень визначається виключно поточною конфігурацією. З 1-періодичності  $\delta$  випливає незалежність такого вибору і від локального часу. Темпоральні процедури та алгоритми з  $k$ -періодичними по модулю  $\pi$  функціями переходу називаються *періодичними*. Еквівалентність  $\pi$  назвемо розв'язною, якщо кожен з її класів еквівалентності є розв'язним. Індексом багатозначної функції назвемо максимальну потужність сукупностей значень функції на окремих аргументах. У разі, коли функція переходу періодичного алгоритму та еквівалентність  $\pi$  мають скінчений індекс і еквівалентність є розв'язною, алгоритм називається *табличним*. Табличні алгоритми задаються скінченими таблицями, стовпчики яких пронумеровані числами від 0 до  $k-1$ , рядки відповідають класам еквівалентності  $\pi$ , а в клітинах таблиці розміщуються значення функції переходу. Традиційні алгоритмічні системи, як правило, є табличним як темпоральні алгоритми. У випадку машин Тюрінга дві конфігурації є еквівалентними, якщо стани їх робочих головок та комірок співпадають. Ці два елементи повністю визначають чергове перетворення конфігурації машини.

Нехай  $\Pi = \langle \Gamma, T, S, \Theta, \Delta \rangle$  та  $\Pi' = \langle \Gamma, T', S', \Theta', \Delta' \rangle$  – обчислювальні простори, такі, що  $T \subseteq T', S \subseteq S'$ ,  $A(C_0, C_{fin})$  та  $B(C_0, C_{fin})$  – алгоритми відповідно над просторами  $\Pi$  та  $\Pi'$  і нехай ці алгоритми є еквівалентними. Тоді алгоритм  $B(C_0, C_{fin})$  називається *розширенням* алгоритму  $A(C_0, C_{fin})$ .

Нехай  $M(T, A)$  сукупність всіх часткових багатозначних темпоральних операторів ( $T$  – операторів) вигляду  $f : T \times A \rightarrow T \times A$ . Для того чи іншого підкласу операторів  $M' \subseteq M(T, A)$  проблема його *процедуризації* (*синтезу*) полягає в побудові того чи іншого класу ініціальних темпоральних процедур, що обчислюють (моделюють) всі його функції. В [1] показано, якщо дозволити процедурам переходити в процесі обчислень в додаткові ізоморфні простори, то сукупність операторів, які обчислюються такими розширеними  $T$  – процедурами, замкнена відносно композицій-зсувів (часових), дзеркального відображення, множення, прямого множення, об'єднання, прямої суми, розгалуження, вибору,  $\alpha$  – ітерації, булевих композицій. Аналогічний результат має місце і для періодичних та табличних алгоритмів. Але при умові, якщо у базовому обчислювальному просторі є тотожні елементарні оператори  $f_0 : S \rightarrow S$ ,  $p_0 : Z \rightarrow Z : f_0(s) = s$ ,  $p_0(t) = t$  для будь-яких  $s \in S$ ,  $t \in Z$ . Такі обчислювальні простори назвемо невивродженими. Покладемо  $\Xi_0$  – сукупність наведених вище композицій. Позначимо  $\tilde{M}(Z, A)$  сукупність темпоральних операторів над  $Z \times A$  та їх прямих добутків. Функціональну алгебру  $R = \langle \tilde{M}(Z, A), \Xi_0 \rangle$  будемо називати *регулярною алгеброю  $T$  – операторів*. Елементи підалгебри  $R(F)$  алгебри  $R$  з множиною твірних  $F$ , де  $F = \{f_1, \dots, f_n, \dots\}$  довільна сукупність  $T$  – операторів та предикатів над  $Z \times A$ , будемо називати *регулярними  $T$  – операторами над базисом  $F$* .

**Теорема 1.** Підклас темпоральних операторів, що обчислюються розширеними періодичними процедурами над невивродженим простором  $\Pi = \langle \Gamma, Z, S, \Theta, \Delta \rangle$ , утворює підалгебру регулярної алгебри  $R = \langle \tilde{M}(Z, S), \Xi_0 \rangle$ .

**Доведення.** Проводиться аналогічно, як і у випадку загальних темпоральних алгоритмів [1].  $\in$

Відзначимо також композицію збільшення періоду, яка, не змінюючи позачасові оператори обчислювача  $k$  – періодичним алгоритмом  $A = \langle \Pi, C_0, C_{fin}, \delta \rangle$ , збільшує його період на 1. Новий алгоритм позначимо  $A^{+1}$ . Визначимо функцію переходу  $\delta'$  на збільшеному основному періоді  $[0, k] \times S$  наступним чином:  $\delta'(r+1, s) = \{p+1, f\} : (p, f) \in \delta(r, s), 0 \leq r < k, s \in S\}$ ,  $\delta'(0, s) = (p_0, f_0)$  для всіх  $s \in S$ . За межами основного періоду функція переходу  $\delta'$  продовжується як  $(k+1)$  – періодична. Покладемо  $A^{+1} = \langle \Pi, C'_0, C'_{fin}, \delta' \rangle$ , де початкові та заключні конфігурації алгоритму  $A$  зсунуті в часі вперед на 1. За побудою, алгоритми  $A$  і  $A^{+1}$  обчислюють однакові позачасові оператори на станах. Композиція збільшення періоду дозволяє у разі необхідності вирівнювати періоди двох або більше алгоритмів.

Проблема аналізу процедур, що належать певному класу  $K$ , полягає у пошуку тих чи інших алгебричних форм подання операторів, обчислювальних процедурами з  $K$ . При цьому таких форм, що існують для кожної процедури. Дану проблему можна також сформулювати як проблему конструктивізації операторів, обчислюваних процедурами класу  $K$ . Теорема 2 розв'язує проблему аналізу для монотонних табличних алгоритмів. Конфігурації називаються сусідніми, якщо вони належать до одного часового періоду. Це означає, що їх часові компоненти  $t_1$  та  $t_2$  задовольняють умові:  $t_1 \div k = t_2 \div k$ , де  $\div$  – операція цілочислового ділення. Нехай  $(p, f) \in \Theta \times \Delta$  – довільна пара елементарних перетворень. Позначимо  $p \otimes f$  – темпоральний оператор, що здійснює покомпонентне перетворення конфігурацій:  $p \otimes f(t, s) = (p(t), f(s))$ . Будемо називати його також елементарним. Покладемо  $\Delta \otimes \Theta = \{p \otimes f \mid (p, f) \in \Theta \times \Delta\}$ .

**Теорема 2** (про регуляризацію монотонних табличних алгоритмів). Нехай  $A(C_0, C_{fin})$  –  $k$  – періодичний по модулю  $\pi$  монотонний табличний алгоритм над певним обчислювальним простором  $\Pi = \langle \Gamma, Z, S, \Theta, \Delta \rangle$  і  $\tau \in \Gamma$ . Тоді оператор  $A_\tau^*$  є регулярним відносно базису, який складають елементарні оператори  $\Delta \otimes \Theta$ , характеристичні предикати класів розбиття  $\pi$ , сукупностей початкових і заключних конфігурацій, а також предикат сусідства конфігурацій.

**Доведення.** Нехай  $[S]_\pi = \{R_1, R_2, \dots, R_n\}$  – розбиття станів за еквівалентністю  $\pi$ ,  $[Z \times S]_\pi = \{Z \times R_1, \dots, Z \times R_n\}$  – розбиття конфігурацій за еквівалентністю  $\pi$ ,  $P = \{p_1, \dots, p_n, p_0, p_{fin}, \infty\}$  – сукупність характеристичних предикатів класів  $Z \times R_i$ , множин  $C_0$  та  $C_{fin}$  та відношення сусідства. Нехай  $R(Z, \Delta \cup \Theta \cup P)$  – підалгебра регулярною алгебри  $R$  з множиною твірних  $\Delta \otimes \Theta \cup P$ . Враховуючи наявність композиції обходу в алгебрі  $R$ , для доведення теореми достатньо розглянути випадок, коли  $C_0 = T \times S$  і показати, що функція  $A_\tau^*$  належить підалгебрі  $R(Z, \Delta \cup \Theta \cup P)$ . Виберемо  $s_1 \in R_1, \dots, s_n \in R_n$  – певну сукупність канонічних представників класів еквівалентності. Позначимо  $B_i$  – обмеження функції  $A_\tau^*$  на клас еквівалентності  $Z \times R_i$ . Покажемо, що функції  $B_i, i = \overline{1, n}$ , задовольняють певній системі праволінійних рівнянь в алгебрі  $R(Z, \Delta \cup \Theta \cup P)$ . Покладемо  $D_T$  оператор, що обчислює алгоритм  $A$  на кожному з періодів, не попадаючи в вихідний стан, тобто коли заключними оголошено конфігурації множини  $\overline{C} = Z \times S \setminus C_{fin}$ .

Нехай  $Q_i, i = \overline{1, n}$ , – оператор, що обчислює алгоритм  $A$  на кожному з періодів своєї області визначення з початковими конфігураціями, що належать класу розбиття  $Z \times R_i$ . Враховуючи монотонність та “табличність” алгоритму  $A$ , можна перевірити, що всі оператори  $D_T, Q_i, i = \overline{1, n}$ , є певними суперпозиціями обмежень відповідних елементарних операторів на області, що задаються тими чи іншими комбінаціями базових характеристичних предикатів підалгебри  $R(Z, \Delta \cup \Theta \cup P)$ . А це означає, що вони належать підалгебрі  $R(Z, \Delta \cup \Theta \cup P)$ . Покажемо, що функції  $B_i, i = \overline{1, n}$ , задовольняють наступній системі праволінійних рівнянь:

$$(*) \{X_i = \bigcup_{j=0}^{k-1} \bigcup_{(p,f) \in \delta(j, [s_i])} p \otimes f \mid_{\overline{C}} \circ D_T \circ (X_1 \cup X_2 \cup \dots \cup X_n) \cup Q_i, i = \overline{1, n},$$

Нехай  $B_i(t, s) = (t', s')$  для деяких  $(t, s), (t', s') \in Z \times S$ ,  $1 \leq i \leq n$ . Тобто пара  $(s|_t, s'|_{t'})$  належить лівій частині  $i$ -ї рівності (\*) при підстановці функції  $B_i$  замість змінної  $X_i$  в (\*). Це означає, що існує таке обчислення  $s|_t \xRightarrow{p} s'|_{t'}$ , що  $s \in R_i$  і  $s'|_{t'} \in C_{fin}$  для деякого  $\lambda \in \Gamma$ . Покажемо, що дана пара належить і правій частині рівняння. Зазначимо, що  $Q_i \subseteq B_i, i = \overline{1, n}$ . Тому розглянемо тільки випадок, коли  $s|_t \notin Dom(Q_i)$ . Враховуючи монотонність алгоритму  $A$ , це означає, що обчислення  $s|_t \xRightarrow{p} s'|_{t'}$  виходить за межі одного періоду. Розглянемо його префікс  $s|_t \xRightarrow{p} s_l|_{t_l}^{\xi}$ ,  $k \leq t_l$ , що закінчується “стрибком” за межі першого періоду та його хвіст  $s_l|_{t_l}^{\xi} \xRightarrow{p} s'|_{t'}$ . За побудовою,  $s_l|_{t_l} = fD_T(t, s)$ ,  $s'|_{t'} = B_m(t_l, s_l)$  для деякого  $1 \leq m \leq n$ . Значить пара  $(s|_t, s'|_{t'})$  належить правій частині  $i$ -ї рівності (\*). Вірно і обернене твердження, якщо пара  $(s|_t, s'|_{t'})$  належить правій частині  $i$ -ї рівності при  $X_i = B_i, i = \overline{1, n}$ , то вона належить і її лівій частині. Доведення те саме, тільки проводиться у зворотному порядку. Таким чином, система функцій  $B_i, i = \overline{1, n}$ , є розв’язком системи (\*). Більше того, вона є її найменшим розв’язком. Дійсно, розглянемо довільний розв’язок системи (\*)  $E_i, i = \overline{1, n}$ . Ясно,  $Q_i \subseteq E_i, i = \overline{1, n}$ . Визначимо сукупності станів  $Q_i^m, i = \overline{1, n}, m \geq 0$ , поклавши  $Q_i^0 = Q_i, Q_i^{m+1} = \bigcup_{j=0}^{k-1} \bigcup_{(p,f) \in \delta(j, [s_i])} p \otimes f|_{\bar{c}} D_T(Q_i^m \cup \dots \cup Q_n^m)$ . За побудовою,  $Q_i^m \subseteq E_i$ . Тоді  $Q_i^\infty = \bigcup_{m=0}^{\infty} Q_i^m \subseteq E_i, i = \overline{1, n}$ , є розв’язком системи (\*), а значить, і найменшим її розв’язком. Але за побудовою, сукупності  $Q_i^\infty$  співпадають з  $B_i$ . Для закінчення доведення теореми 1 залишилось, зазначити, що: 1) найменший розв’язком праволінійної системи функціональних рівнянь вигляду (\*) є кортеж регулярних функцій над базисом, що складається з її коефіцієнтів та вільних членів, та 2)  $A_\tau^* = \bigcup_{i=1}^n (p_i \rightarrow B_i)$ . Отже, функція  $A_\tau^*$  належить алгебрі  $R(Z, \Delta \cup \Theta \cup P)$ .

1. Зубенко В.В. Про темпоральні процедури та алгоритми // Вісн. КНУ ім. Тараса Шевченка. Серія: Кібернетика. – 2005. – №6. – С. 20-26.
2. Месарович М., Такаха Я. Общая теория систем: математические основы // М.: Мир. –1978. – 308 с.
3. Капитонова Ю.В., Лещевский А.А. Математическая теория проектирования вычислительных систем // М.: Наука. –1988. – 295 с.
4. Редько В.Н. Экспликативное программирование: ретроспективы и перспективы // Тр. 1-й Межд. науч.-практ. конф. по программированию УкрПРОГ’98. – Киев: Ин-т проблем программирования НАНУ, 1998. – С. 3-24.