УДК 004.4

СТАБІЛЬНІСТЬ ТА МОНОТОННІСТЬ ПРОГРАМ ЩОДО СТРУКТУРНИХ ТРАНСФОРМАЦІЙ ДАНИХ

М.С. Нікітченко, Є.В. Іванов

Київський національний університет імені Тараса Шевченка,
01033, Київ, вул. Володимирська, 64,
e-mail: nikitchenko@unicyb.kiev.ua, ivanov.eugen@gmail.com

Розглядається проблема поведінки програм щодо структурних трансформацій вхідних даних. Проведено уточнення проблеми та її формалізацію в рамках композиційно-номінативного підходу. Досліджено введені поняття та побудовано мову програм, які мають стабільну та монотонну поведінку при зміні структури вхідних даних.

In the article the problem of program behavior with respect to structure transformations of input data is considered. An explication and formalization of the problem in the framework of composition-nominative approach is given. Defined notions are studied and a language of programs having stable and monotone behavior under structure transformations is constructed.

Вступ

Багато мов програмування мають вбудовані операції для роботи з ієрархічними структурами даних. Такі дані зазвичай можуть розглядатися як певні види дерев з вершинами та / або ребрами розміченими значеннями. Часто виконання операцій суттєво залежить від характеру ієрархії (плоска, розгалужена і т. п.) даного, до якого вони застосовані. Ця обставина відбивається на програмуванні: наприклад, програма, яка розрахована на роботу з даними з плоскою ієрархією, може бути не придатною для роботи із даними з розгалуженою ієрархією, і навпаки. Тому програміст має враховувати різні можливі варіанти ієрархії даних і не покладатися на те, що обробка даних з різною ієрархією буде відбуватися потрібним для нього чином.

У деяких мовах програмування присутні засоби, що дозволяють працювати із певними класами даних, не піклуючись про характер ієрархії в них. Для таких мов будемо казати, що програми цих мов мають стабільну поведінку при зміні ієрархії (структури) даних. Наприклад, у мові програмування Pascal багатовимірні масиви можуть бути описані різними способами, і кожному способу опису відповідає своя ієрархія. Так, двовимірний масив можна описати або як матрицю (A: **array** [1..n, 1..m] **of** real), або як масив масивів (A: **array** [1..n] **of array** [1..m] **of** real). За першим способом опису можна вважати, що масив є ієрархією з двома рівнями (зверху вниз): “пара індексів”–“значення”. За другим – можна вважати, що масив є ієрархією з трьома рівнями: “перший індекс”–“другий індекс”–“значення”. Для того, щоб звернутися до елемента масиву із заданими індексами, необхідно пройти вниз за відповідною ієрархією. При цьому, незалежно від способу опису двовимірного масиву, у мові Pascal дозволяється звертатися до його елементів як за нотацією A[i,j], що відповідає плоскій ієрархії, так і за нотацією A[i][j], що відповідає розгалуженій ієрархії. Аналогічна ситуація має місце для довільних багатовимірних масивів. Тобто програміст має можливість писати програми обробки масивів, не піклуючись про спосіб, у який вони описані. При цьому масиви як матриці і масиви масивів можуть подаватися в оперативній пам’яті різними структурами (це залежить від компілятора). Зауважимо, що в мовах програмування, таких як C, C++, C# немає можливостей, аналогічних вищеописаним для мови Pascal.

Незважаючи на те, що деякі практичні мови програмування забезпечують властивість стабільності (або, можливо, обмеженої стабільності) при структурних трансформаціях даних, теорія таких мов програмування не є достатньо розвиненою. Дана робота має на меті зробити певний крок у розбудові зазначеної теорії. Для цього необхідно надати формальні визначення понять ієрархічних даних, програм над ієрархічними даними, стабільності програм при зміні ієрархії даних, побудувати і дослідити мови програм, що забезпечують їх стабільність. Такі дослідження будемо здійснювати в рамках композиційно-номінативного підходу до програмування [1], який надає достатньо розвинені можливості для формалізації необхідних понять. У попередній роботі авторів над цією тематикою [2] було побудовано функціонально повну і структурно адекватну формальну мову програм, що забезпечує їх стабільність при зміні ієрархій даних – так звану мову $SICON_a$ (просту композиційно-номінативну мову програм з асоціативним розіменуванням). У даній роботі досліджуються мови програм, що забезпечують більш сильну властивість, ніж стабільність, а саме – монотонність. Змістовно, монотонність програми означає, що вона стабільна, і при розширенні вхідного даного вихідне дане не зменшується.

Надалі будемо використовувати такі позначення:

- V^+ – множина не порожніх слів в алфавіті V;
- $pref(u) = \{v \in V^+ \mid \exists w \in V^* \; u = vw\}$ – множина не порожніх префіксів слова $u \in V^+$;
- $u \le v$ ($u < v$) – слово u є префіксом (власним префіксом) слова v;
- $f(x)\downarrow$ – функція f визначена на аргументі x;

- $f(x)\uparrow$ – функція f не визначена на аргументі x;
- $f(x) \cong g(y)$ – сильна рівність ($f(x)\downarrow$ тоді і тільки тоді, коли $g(y)\downarrow$, і тоді $f(x) = g(y)$);
- $A \xrightarrow{n} B$ – клас (часткових) функцій з A в B, що мають скінченний графік;
- $\arg(f)$ – область визначеності функції або бінарного відношення;
- $f|_A$ – звуження області визначеності функції або бінарного відношення на множину *A*.

1. Номінативні та складноіменні дані

Для того, щоб формально визначати мови програм, будемо використовувати композиційно-номінативний підхід [1] до визначення понять програмування. Згідно цього підходу, дані уточнюються як певні різновиди номінативних даних (від лат. *nomen* – ім'я), які, неформально, є ієрархічно побудованими об'єктами на основі відношення *ім'я* $\mapsto$ *значення*, а програми уточнюються як функції над номінативними даними, що можуть бути отримані з базових функцій застосуванням композицій (тобто операцій над функціями). Таким чином, розглядаючи множину функцій як носій, а композиції – як операції, отримуємо алгебру функцій (програм). Мова програм визначається класом номінативних даних, над якими діють програми, і складом базових функцій і композицій.

Як вищезазначено, номінативні дані є ієрархічними об'єктами. Їх можна подавати графічно у вигляді орієнтованих дерев із дугами, розміченими іменами, і листками, розміченими значеннями, або за допомогою текстового запису. Наприклад, запис $[a \mapsto 1, b \mapsto [c \mapsto 2]]$ означає, що дане має дві компоненти з іменами a і b, значення першої з яких атомарне, а значення другої саме є номінативним даним. Зауважимо, що у вигляді номінативних даних може бути подано багато видів структур даних, у тому числі записи, файли, масиви, реляції, бази даних тощо.

Формально, номінативні дані визначаються індуктивно на основі множини імен *V* та класу базових значень (атомів) *W* таким чином:

- $ND_0(V,W) = W \cup \{\varnothing\}$ – клас номінативних даних рангу нуль,
- $ND_{k+1}(V,W) = W \cup \left(V \xrightarrow{n} ND_k(V,W)\right)$ для $k \geq 0$ – клас номінативних даних рангу не більше $k+1$.

Тоді $ND(V,W) = \bigcup_{k\geq 0} ND_k(V,W)$ – клас усіх номінативних даних. Класи номінативних даних різного рангу утворюють ланцюг: $ND_0(V,W) \subseteq ND_1(V,W) \subseteq ND_2(V,W) \subseteq \ldots$

Згідно визначення, номінативні дані, що не є атомами (неатомарні дані), є частковими функціями на *V*. При необхідності ми будемо також їх трактувати як бінарні відношення і використовувати теоретико-множинні операції над ними.

Основними операціями над номінативними даними є унарні параметричні операції *іменування* і *розіменування*, і бінарна операція *накладання*. Позначимо літерою d номінативне дане.

Іменування $\Rightarrow v$ з параметром $v \in V^+$ визначається рівністю $\Rightarrow v(d) = [v \mapsto d]$. Тобто ця операція створює дане, у якому v (параметр операції) іменує значення d (аргумент операції).

Розіменування $v \Rightarrow$ з параметром $v \in V^+$ визначається рівністю $v \Rightarrow (d) \cong d(v)$. Тобто ця операція знаходить у даному d компонент з ім'ям v, що є параметром операції.

Накладання ∇ визначене тільки на неатомарних даних і задається рівністю $d_1 \nabla d_2 = d_2 \cup d_1|_{\arg(d_1)\setminus\arg(d_2)}$. Тобто ця операція створює дане, в якому присутні всі компоненти другого аргументу й ті компоненти першого, які не були перевизначені в другому.

Номінативні дані класу $ND(V,W)$ засновані на неструктурованих (простих) іменах. Проте для цілей даної роботи необхідний клас номінативних даних, в якому можливі еквівалентні структурні перетворення, тобто такі перетворення даного, при яких його інтуїтивний зміст залишається, а характер ієрархії імен може змінюватися. Щоб забезпечити таку можливість, ведемо спеціальний клас номінативних даних зі складними іменами, які задаються як конкатенації простих імен. А саме, *класом даних зі складними іменами* називається клас $NDV(V,W) = ND(V^+,W)$.

Для деяких даних цього класу не виконується *принцип однозначності структурного іменування* (OCI). Наприклад, у даному $[xy \mapsto 1, x \mapsto [y \mapsto 2]]$ доцільно вважати, що складному імені xy відповідає два значення: 1 і 2. Ми будемо розглядати підклас $NDV(V,W)$, для даних якого виконується принцип OCI. Цей підклас назвемо *класом складноіменних даних* та позначимо його як $NDVC(V,W)$. Перед тим, як надати формальне визначення $NDVC(V,W)$, введемо низку допоміжних визначень та позначень, в яких $d \in NDV(V,W)$:

- *іменний шлях* в d – це послідовність імен $(v_1, v_2, .., v_k)$ з V^+ ($k \geq 1$), така, що визначено *значення* $d(v_1, v_2, .., v_k)$ *шляху* в *d*, яке задається рівністю $d(v_1, v_2, .., v_k) \cong (...((d(v_1))(v_2))...(v_k))$; шлях називається *термінальним* в *d*, якщо його значенням в d є дане рангу 0 (в дереві, що відповідає даному *d*, термінальний шлях веде від кореня до листа дерева);

- $rn(d) = \{u \mid d(u)\downarrow\}$ – множина *кореневих імен d*;
- $ra(d) = \{u \mid d(u)\downarrow \in W\}$ – множина *атомних кореневих імен d*;
- $rn_0(d) = \{u \in rn(d) \mid d(u) \notin W\}$ – множина *неатомних кореневих імен d*;
- $rp_0(d) = \{u \in V^+ \mid \exists v \in V^+ (uv \in rn(d))\}$ – слова, які є власними префіксами кореневих імен *d*;
- $rs(d) = rp_0(d) \cup rn(d)V^*$ – слова, які порівнюються з деяким кореневим іменем *d*;
- операція *ділення даного d на ім'я* $u \in V^+$ задається формулою

$$d/u = [v_1 \mapsto d(v) \mid \exists v_1 \in V^+ (v \in rn(d) \wedge v = uv_1)].$$

Формальне визначення класу складноіменних даних подається таким чином: $NDVC(V,W) = W \cup G(V,W)$, де $G(V,W)$ – дані $d \in NDV(V,W) \setminus W$ такі, що має місце властивість: якщо для двох деяких шляхів $(v_1, v_2, .., v_n)$ та $(v_1', v_2', .., v_m')$ в *d* виконується співвідношення $v_1 v_2 ... v_n \in pref(v_1' v_2' ... v_m')$, то $n \le m$ і $v_i = v_i'$ для всіх $i \in \{1, 2, .., n\}$. Позначимо $NDVC_k(V,W) = NDVC(V,W) \cap ND_k(V^+,W)$ – дані класу $NDVC(V,W)$ рангу не більше k.

Змістовно, до класу $NDVC(V,W)$ належать дані зі складними іменами, для яких в дереві, яке відповідає даному, немає двох шляхів від кореня (один з яких не є початком іншого), таких, що конкатенація імен вздовж одного шляху є префіксом конкатенації імен вздовж іншого шляху. Для цих даних виконується принцип OCI.

Домовимося, що надалі літерою d (можливо з індексами) будуть позначатися дані класу $NDVC(V,W)$, якщо явно не вказано інше.

Основними операціями над складноіменними даними є операція іменування, вищевизначена для довільних номінативних даних, унарна параметрична операція *асоціативного розіменування* і бінарна операція *структурного накладання*. Зауважимо, що іменування, застосоване до складноіменного даного, дає складноіменне дане.

Операція *асоціативного розіменування* $v \Rightarrow_a$ з параметром $v \in V^+$ аналогічна звичайному розіменуванню, вищевведеному, але враховує складеність імен. Формально вона визначається індуктивно за довжиною слова-параметра таким чином:

база індукції. Якщо $|v| = 1$, то

- $v \Rightarrow_a (d) = d(v)$, якщо $v \in rn(d)$;
- $v \Rightarrow_a (d) = d/v$, якщо $v \notin rn(d)$ і $d/v \neq \varnothing$;
- $v \Rightarrow_a (d)\uparrow$, якщо $v \notin rn(d)$ і $d/v = \varnothing$.

Індукційний перехід. Якщо $|v| = n \ge 2$, то $v \Rightarrow_a (d) \cong v_1 \Rightarrow_a (x \Rightarrow_a (d))$ для кожного $d \in NDVC(V,W)$, де $x \in V$ і $v_1 \in V^{n-1}$ – такі слова, що $v = xv_1$.

Наведемо приклад дії операції асоціативного розіменування: $(xy) \Rightarrow_a ([x \mapsto [y \mapsto 1], y \mapsto 2]) = 1$ (тут вважається, що $\{1,2\} \subseteq W$ і $\{x, y\} \subseteq V$).

Назва операції $v \Rightarrow_a$ походить від наступної її властивості: $v \Rightarrow_a (d) \cong v_1 \Rightarrow_a (v_2 \Rightarrow_a (..(v_n \Rightarrow_a (d)..))$ для довільного даного $d \in NDVC(V,W)$ і довільного розкладу $v = v_n v_{n-1} .. v_2 v_1$ складного імені $v \in V^+$ на імена $v_1, v_2, .., v_n \in V^+$.

Операція *структурного накладання* також аналогічна накладанню, вищевведеному, але враховує складеність імен. На прикладах вона діє таким чином:

- $[x \mapsto d_1] \nabla_s [y \mapsto d_2] = [x \mapsto d_1, y \mapsto d_2]$, $x, y \in V$, $x \neq y$;
- $[xy \mapsto d_1] \nabla_s [x \mapsto d_2] = [x \mapsto d_2]$, тобто значення під іменем x перевизначає значення під іменами, що є продовженнями x;
- $[x \mapsto d_1] \nabla_s [xy \mapsto d_2] = [x \mapsto (d_1 \nabla_s [y \mapsto d_2])]$ (при $d_1 \notin W$), тобто значення під іменем xy модифікує значення під іменем x.

Формальне визначення структурного накладання подамо індуктивно за рангом першого аргументу. Якщо $d_1 \in NDVC_0(V,W)$, то

- $d_1 \nabla_s d_2 = d_2$, якщо $d_1 = \varnothing$ і $d_2 \in NDVC(V,W) \setminus W$;
- $d_1 \nabla_s d_2 \uparrow$, якщо $d_1 \in W$ або $d_2 \in W$.

Індукційний перехід. Припустимо, що значення $d_1 \nabla_s d_2$ вже визначено для всіх d_1, d_2, таких, що $d_1 \in NDVC_k(V,W)$. Нехай $d_1 \in NDVC_{k+1}(V,W) \setminus NDVC_k(V,W)$.

Тоді покладемо $d_1 \nabla_s d_2 = d$, де дане d визначено таким чином:

1) $d(u)\downarrow = d_2(u)$, якщо u є кореневим іменем в d_2 і не має власного префікса, який би був кореневим іменем в d_1, тобто $u \in rn(d_2) \setminus rn(d_1)V^+$;

2) $d(u)\downarrow = d_1(u)\nabla_s(d_2/u)$, якщо u є неатомним кореневим іменем в d_1 і власним префіксом кореневого імені в d_2, тобто $u \in rn_0(d_1) \cap rp_0(d_2)$;

3) $d(u)\downarrow = d_2/u$, якщо u є атомним кореневим іменем в d_1 і власним префіксом кореневого імені в d_2, тобто $u \in ra(d_1) \cap rp_0(d_2)$;

4) $d(u)\downarrow = d_1(u)$, якщо u є кореневим іменем в d_1 і не порівнюється з жодним кореневим іменем з d_2, тобто $u \in rn(d_1) \setminus rs(d_2)$;

5) $d(u)\uparrow$, в інших випадках, тобто якщо $u \notin rn(d_1) \cup rn(d_2)$ або $u \in rn(d_1) \cap rn(d_2)V^+$ або $u \in rn(d_2) \cap rn(d_1)V^+$.

В подальшому буде потрібна наступна лема, яку наводимо без доведення.

Лема 1 (властивості структурного накладання). Нехай $d_1 \nabla_s d_2 \downarrow$ і $u \in V^+$. Тоді:

1) якщо $u \Rightarrow_a (d_2)\downarrow \in W \cup \{\varnothing\}$, то $u \Rightarrow_a (d_1 \nabla_s d_2)\downarrow = u \Rightarrow_a (d_2)$, тобто структурне накладання зберігає атомарні значення імен з другого аргументу;

2) якщо $u \Rightarrow_a (d_1 \nabla_s d_2)\downarrow \in W \cup \{\varnothing\}$ і $u \Rightarrow_a (d_2)\uparrow$, то $u \Rightarrow_a (d_1)\downarrow = u \Rightarrow_a (d_1 \nabla_s d_2)$, тобто структурне накладання не створює зайвих (не пов'язаних з жодним з аргументів) атомарних значень;

3) якщо $x \Rightarrow_a (d_2)\uparrow$, де x – перша літера u, то $u \Rightarrow_a (d_1 \nabla_s d_2) \cong u \Rightarrow_a (d_1)$, тобто структурне накладання зберігає значення імен першого аргументу, що не мають спільного префікса з іменами другого.

2. Властивості складноіменних даних

Як зазначалося раніше, до даних класу $NDVC(V,W)$ можуть бути застосовані еквівалентні перетворення, що зберігають їх інтуїтивний зміст, але змінюють характер ієрархії. Щоб формально визначити цю властивість, введемо відповідне відношення еквівалентності на $NDVC(V,W)$ – відношення *номінативної еквівалентності*. Для формального визначення номінативної еквівалентності, попередньо введемо на $NDVC(V,W)$ бінарне відношення *номінативного включення* $\prec$ таким чином:

- якщо $d_1 \in W$ або $d_2 \in W$, то $d_1 \prec d_2$ тоді і тільки тоді, коли $d_1 = d_2$;
- якщо $d_1 \notin W$ і $d_2 \notin W$, то $d_1 \prec d_2$ тоді і тільки тоді, коли для кожного термінального шляху $(v_1, v_2, .., v_k)$ в d_1 існує термінальний шлях $(v_1', v_2', .., v_l')$ в d_2, такий, що $v_1 v_2 .. v_k = v_1' v_2' .. v_l'$ і $d_1(v_1, v_2, .., v_k) \cong d_2(v_1', v_2', .., v_l')$.

З цього визначення випливає, що для відношення номінативного включення множина конкатенацій імен на всіх термінальних шляхах першого даного має входити у таку ж множину другого даного, крім того і відповідні значення мають співпадати. Легко переконатися, що відношення $\prec$ є передпорядком.

Номінативну еквівалентність складноіменних даних $\approx$ визначимо як бінарне відношення на $NDVC(V,W)$, таке, що $d_1 \approx d_2$ тоді і тільки тоді, коли $d_1 \prec d_2$ і $d_2 \prec d_1$. Відношення $\approx$ є еквівалентністю.

На основі номінативної еквівалентності можна визначити поняття стабільності програми при зміні ієрархії даних таким чином: програма (як функція над складноіменними даними) є (номінативно) стабільною, якщо вона на номінативно еквівалентних вхідних даних дає номінативно еквівалентні результати (причому на кожній парі еквівалентних даних вона визначена або не визначена одночасно). В частковому випадку, коли результатами програми є атомарні дані, на номінативно еквівалентних даних номінативно стабільна програма дає однакові результати.

У роботі [2] було побудовано формальну мову програм, які є номінативно стабільними. Далі побудуємо мову програм, які задовольняють більш сильній умові – монотонності. Для цього необхідно визначити відношення на даних, в сенсі якого програми мають бути монотонними. Таким відношенням оберемо відношення *слабкого номінативного включення*, що формально визначено далі.

Введемо на $NDVC(V,W)$ допоміжне бінарне відношення $\prec_A$, вважаючи, що

- якщо $d_1 \in W$ або $d_2 \in W$, то $d_1 \prec_A d_2$ тоді і тільки тоді, коли $d_1 = d_2$;
- якщо $d_1 \notin W$ і $d_2 \notin W$, то $d_1 \prec_A d_2$ тоді і тільки тоді, коли для кожного термінального шляху $(v_1, v_2, .., v_k)$ в d_1, *що закінчується атомом*, існує термінальний шлях $(v_1', v_2', .., v_l')$ в d_2, такий, що $v_1 v_2 .. v_k = v_1' v_2' .. v_l'$ і $d_1(v_1, v_2, .., v_k) \cong d_2(v_1', v_2', .., v_l')$.

Введемо на $NDVC(V,W)$ допоміжне бінарне відношення $\prec_N$, вважаючи, що

- якщо $d_1 \in W$ або $d_2 \in W$, то $d_1 \prec_N d_2$ тоді і тільки тоді, коли $d_1 = d_2$;

- якщо $d_1 \notin W$ і $d_2 \notin W$, то $d_1 \prec_N d_2$ тоді і тільки тоді, коли для всіх $u \in V^+$, таких, що $u \Rightarrow_a (d_1)\downarrow \notin W$, виконується $u \Rightarrow_a (d_2)\downarrow \notin W$.

Тоді введемо на $NDVC(V,W)$ бінарне відношення *слабкого номінативного включення* таким чином: $\prec_w = \prec_A \cap \prec_N$. Назва відношення $\prec_w$ пов'язана з тим, що відношення $\prec$ включається у $\prec_w$. Відмінність $\prec$ від $\prec_w$ проявляється в тих випадках, коли значенням термінального шляху в d_1 є порожнє номінативне дане, бо тоді для відношення $\prec$ цей шлях в d_2 також має давати порожнє дане, а для $\prec_w$ такий шлях в d_2 може мати своїм значенням довільне неатомарне дане. Тобто $\prec_w$ дозволяє "розширювати" порожні дані іменованими компонентами. Наприклад, $[x \mapsto \varnothing] \prec_w [xy \mapsto 1, xz \mapsto 2]$ і $[x \mapsto \varnothing] \prec_w [x \mapsto [y \mapsto 1, z \mapsto 2]]$, але $[x \mapsto \varnothing] \not\prec [xy \mapsto 1, xz \mapsto 2]$ і $[x \mapsto \varnothing] \not\prec [x \mapsto [y \mapsto 1, z \mapsto 2]]$.

Лема 2. Відношення $\prec_w$ є передпорядком і включає передпорядок $\prec$.

Доведення цієї леми легко отримується з означення відношень, тому тут не наводиться.

3. Монотонність операцій над складноіменними даними

Встановимо монотонність операцій над складноіменними даними за відношенням передпорядку $\prec_w$. Функції, монотонні за відношенням $\prec_w$, будемо називати *номінативно монотонними*.

Наведемо дві допоміжні леми без доведень.

Лема 3 (критерій відношення $\prec$). Нехай $d_1, d_2 \in NDVC(V,W) \setminus W$. Тоді $d_1 \prec d_2$ тоді і тільки тоді, коли для всіх $u \in V^+$, таких, що $u \Rightarrow_a (d_1)\downarrow \in W \cup \{\varnothing\}$, виконується $u \Rightarrow_a (d_2)\downarrow = u \Rightarrow_a (d_1)$.

Лема 4 (критерій відношення $\prec_A$). Нехай $d_1, d_2 \in NDVC(V,W) \setminus W$. Тоді $d_1 \prec_A d_2$ тоді і тільки тоді, коли для всіх $u \in V^+$, таких, що $u \Rightarrow_a (d_1)\downarrow \in W$, виконується $u \Rightarrow_a (d_2)\downarrow = u \Rightarrow_a (d_1)$.

Тепер сформулюємо властивості операцій.

Лема 5 (монотонність іменування, слабка монотонність асоціативного розіменування і ділення відносно $\prec_A$). Нехай $v \in V^+$. Тоді

1) якщо $d_1 \prec_A d_2$, то $\Rightarrow v(d_1) \prec_A \Rightarrow v(d_2)$;

2) якщо $d_1 \prec_A d_2$, $v \Rightarrow_a (d_1)\downarrow$ і $v \Rightarrow_a (d_2)\downarrow$, то $v \Rightarrow_a (d_1) \prec_A v \Rightarrow_a (d_2)$;

3) якщо $d_1 \prec_A d_2$ і $d_2 / v \neq \varnothing$, то $d_1 / v \prec_A d_2 / v$.

Доведення.

1) нехай $d_1 \prec_A d_2$ і $v \in V^+$. Позначимо $e_i = \Rightarrow v(d_i) = [v \mapsto d_i]$, $i = 1,2$. Якщо хоч одне з d_1, d_2 належить W, то $d_1 = d_2$, тому $\Rightarrow v(d_1) \prec_A \Rightarrow v(d_2)$. Розглянемо випадок, коли $d_1 \notin W$ і $d_2 \notin W$. Нехай $(v_1,..,v_n)$ – термінальний шлях в e_1, що закінчується атомом. Тоді $v_1 = v$. Оскільки $d_1 \notin W$, то $n > 1$. Тоді $(v_2,..,v_n)$ – термінальний шлях в d_1 і оскільки $d_1 \prec_A d_2$, то існує термінальний шлях $(v_2',..,v_m')$ в d_2, такий, що $v_2..v_n = v_2'..v_m'$ і $d_2(v_2',..,v_m') \cong d_1(v_2,..,v_n)$. Тоді $(v, v_2',..,v_m')$ є термінальним шляхом в e_2 і $e_2(v, v_2',..,v_m') \cong e_1(v, v_2,.., v_n)$. Отже $e_1 \prec_A e_2$, тобто $\Rightarrow v(d_1) \prec_A \Rightarrow v(d_2)$;

2) нехай $d_1 \prec_A d_2$, $v \Rightarrow_a (d_1)\downarrow$ і $v \Rightarrow_a (d_2)\downarrow$. Нехай $u \in V^+$ і $u \Rightarrow_a (v \Rightarrow_a (d_1))\downarrow \in W$. Тоді $(vu) \Rightarrow_a (d_1)\downarrow \in W$. Оскільки $d_1 \prec_A d_2$, то $(vu) \Rightarrow_a (d_2)\downarrow = (vu) \Rightarrow_a (d_1)$ за лемою 4. Тоді $u \Rightarrow_a (v \Rightarrow_a (d_2))\downarrow = u \Rightarrow_a (v \Rightarrow_a (d_1))$. Отже $v \Rightarrow_a (d_1) \prec_A v \Rightarrow_a (d_2)$ за лемою 4 в силу довільності u;

3) Нехай $d_1 \prec_A d_2$ і $d_2 / v \neq \varnothing$. Тоді $d_1, d_2 \notin W$. Нехай $(v_1,..,v_n)$ – термінальний шлях в d_1 / v, що закінчується атомом. Тоді $(vv_1,..,v_n)$ – термінальний шлях в d_1, що закінчується атомом і оскільки $d_1 \prec_A d_2$, то існує термінальний шлях $(v_1',..,v_m')$ в d_2, такий, що $vv_1..v_n = v_1'..v_m'$ і $d_1(vv_1,..,v_n) \cong d_2(v_1',..,v_m')$. Припустимо, що $v_1' \leq v$. Оскільки $v_1' \in rn(d_2)$, то в $rn(d_2)$ не існує імені, яке є продовженням v, тому $d_2 / v = \varnothing$, що суперечить припущенню $d_2 / v \neq \varnothing$. Таким чином, відношення $v_1' \leq v$ не виконується, а тому $v < v_1'$. Тоді існує слово v_1'', таке, що $vv_1'' = v_1'$ і $(v_1'', v_2'..,v_m')$ є термінальним шляхом в d_2 / v, таким, що $(d_1 / v)(v_1, v_2,..,v_n) \cong (d_2 / v)(v_1'', v_2',..,v_m')$. Отже $d_1 / v \prec_A d_2 / v$.

Лема доведена.

Лема 6 (монотонність іменування, слабка монотонність асоціативного розіменування і ділення відносно $\prec_N$). Нехай $v \in V^+$. Тоді

1) якщо $d_1 \prec_N d_2$, то $\Rightarrow v(d_1) \prec_N \Rightarrow v(d_2)$;

2) якщо $d_1 \prec_N d_2$, $v \Rightarrow_a (d_1)\downarrow$ і $v \Rightarrow_a (d_2)\downarrow$, то $v \Rightarrow_a (d_1) \prec_N v \Rightarrow_a (d_2)$;

3) якщо $d_1 \prec_N d_2$ і $d_2 / v \neq \varnothing$, то $d_1 / v \prec_N d_2 / v$.

Доведення.

1) нехай $d_1 \prec_N d_2$ і $v \in V^+$. Позначимо $e_i ==\Rightarrow v(d_i) = [v \mapsto d_i]$, $i = 1,2$. Якщо хоч одне з d_1, d_2 належить W, то $d_1 = d_2$, тому $\Rightarrow v(d_1) \prec_N \Rightarrow v(d_2)$. Розглянемо випадок, коли $d_1 \notin W$ і $d_2 \notin W$. Нехай $u \in V^+$ – слово, таке, що $u \Rightarrow_a (e_1)\downarrow \notin W$. Тоді або $u \le v$, або $v \le u$. Якщо $u \le v$, то $u \Rightarrow (e_2)\downarrow \notin W$, бо $d_2 \notin W$. Якщо $v < u$, то існує слово $v_1 \in V^+$, таке, що $vv_1 = u$ і $u \Rightarrow_a (e_1) = v_1 \Rightarrow_a (d_1)\downarrow \notin W$. Оскільки $d_1 \prec_N d_2$, то $v_1 \Rightarrow_a (d_2)\downarrow \notin W$. Тоді $u \Rightarrow_a (e_2) \cong v_1 \Rightarrow_a (d_2)\downarrow \notin W$. Отже $e_1 \prec_N e_2$, тобто $\Rightarrow v(d_1) \prec_N \Rightarrow v(d_2)$;

2) нехай $d_1 \prec_N d_2$, $v \Rightarrow_a (d_1)\downarrow$ і $v \Rightarrow_a (d_2)\downarrow$. Нехай $u \in V^+$ і $u \Rightarrow_a (v \Rightarrow_a (d_1))\downarrow \notin W$. Тоді $(vu) \Rightarrow_a (d_1)\downarrow \notin W$. Оскільки $d_1 \prec_N d_2$, то $(vu) \Rightarrow_a (d_2)\downarrow \notin W$. Тоді $u \Rightarrow_a (v \Rightarrow_a (d_2))\downarrow \notin W$. Отже $v \Rightarrow_a (d_1) \prec_N v \Rightarrow_a (d_2)$ в силу довільності u;

3) нехай $d_1 \prec_N d_2$ і $d_2 / v \ne \varnothing$. Тоді $d_1, d_2 \notin W$. Нехай $u \in V^+$ – слово, таке, що $u \Rightarrow_a (d_1 / v)\downarrow \notin W$. Тоді $(vu) \Rightarrow_a (d_1)\downarrow \notin W$ і оскільки $d_1 \prec_N d_2$, то $(vu) \Rightarrow_a (d_2)\downarrow \notin W$. Тоді $u \Rightarrow_a (v \Rightarrow_a (d_2))\downarrow \notin W$. Оскільки $d_2 / v \ne \varnothing$, то $v \Rightarrow_a (d_2)\downarrow = d_2 / v$, тому $u \Rightarrow_a (d_2 / v)\downarrow \notin W$. Отже $d_1 / v \prec_N d_2 / v$ в силу довільності u.

Лема доведена.

Лема 7 (монотонність іменування, асоціативного розіменування і ділення відносно $\prec_w$). Нехай $v \in V^+$. Тоді

1) якщо $d_1 \prec_w d_2$, то $\Rightarrow v(d_1) \prec_w \Rightarrow v(d_2)$;

2) якщо $d_1 \prec_w d_2$ і $v \Rightarrow_a (d_1)\downarrow$, то $v \Rightarrow_a (d_2)\downarrow$ і $v \Rightarrow_a (d_1) \prec_w v \Rightarrow_a (d_2)\downarrow$;

3) якщо $d_1 \prec_w d_2$ і $d_2 / v \ne \varnothing$, то $d_1 / v \prec_w d_2 / v$.

Доведення випливає безпосередньо з лем 5 і 6.

Лема 8 (монотонність структурного накладання за першим аргументом відносно $\prec_A$). Нехай $d_1 \prec_A d_1'$ і $d_1 \nabla_s d_2 \downarrow$. Тоді $d_1' \nabla_s d_2 \downarrow$ і $d_1 \nabla_s d_2 \prec_A d_1' \nabla_s d_2$.

Доведення. Припустивши, що $d_1' \nabla_s d_2 \uparrow$, отримаємо $d_1' \in W$ або $d_2 \in W$, тому $d_1 \in W$ (в силу того, що $d_1 \prec_A d_1'$) або $d_2 \in W$, а отже $d_1 \nabla_s d_2 \uparrow$, що суперечить умові леми. Тому $d_1' \nabla_s d_2 \downarrow$, і достатньо довести, що $d_1 \nabla_s d_2 \prec_A d_1' \nabla_s d_2$.

Нехай $u \in V^+$ – таке слово, що $u \Rightarrow_a (d_1 \nabla_s d_2)\downarrow \in W$. Можливі 3 випадки:

1) $u \Rightarrow_a (d_2)\downarrow \in W \cup \{\varnothing\}$. За лемою 1. 1), $u \Rightarrow_a (d_1 \nabla_s d_2)\downarrow = u \Rightarrow_a (d_2)$ і $u \Rightarrow_a (d_1' \nabla_s d_2)\downarrow = u \Rightarrow_a (d_2)$, а отже $u \Rightarrow_a (d_1' \nabla_s d_2)\downarrow = u \Rightarrow_a (d_1 \nabla_s d_2)$;

2) $u \Rightarrow_a (d_2)\downarrow \notin W \cup \{\varnothing\}$. У цьому випадку існує слово $u_1 \in uV^+$, таке, що $u_1 \Rightarrow_a (d_2)\downarrow \in W \cup \{\varnothing\}$. Тоді за лемою 1. 1), $u_1 \Rightarrow_a (d_1 \nabla_s d_2)\downarrow$, що суперечить припущенню $u \Rightarrow_a (d_1 \nabla_s d_2) \in W$, оскільки u є власним префіксом u_1. Тобто припущення $u \Rightarrow_a (d_2)\downarrow \notin W \cup \{\varnothing\}$ виконуватися не може;

3) $u \Rightarrow_a (d_2)\uparrow$. За лемою 1. 2), $u \Rightarrow_a (d_1)\downarrow = u \Rightarrow_a (d_1 \nabla_s d_2) \in W$. Оскільки $d_1 \prec_A d_1'$, то $u \Rightarrow_a (d_1')\downarrow = u \Rightarrow_a (d_1) \in W$ за лемою 4.

Можливі такі випадки:

a) $u \Rightarrow_a (d_1' \nabla_s d_2)\downarrow \in W \cup \{\varnothing\}$. У цьому випадку за лемою 1. 2), $u \Rightarrow (d_1' \nabla_s d_2) = u \Rightarrow (d_1') = u \Rightarrow (d_1 \nabla_s d_2)$;

b) $u \Rightarrow_a (d_1' \nabla_s d_2)\downarrow \notin W \cup \{\varnothing\}$. У цьому випадку існує слово $v \in V^+$, таке, що $(uv) \Rightarrow_a (d_1' \nabla_s d_2)\downarrow \in W \cup \{\varnothing\}$. Оскільки $(uv) \Rightarrow_a (d_2)\uparrow$, то за лемою 1. 2) отримуємо $(uv) \Rightarrow_a (d_1')\downarrow \in W \cup \{\varnothing\}$, звідки $u \Rightarrow_a (d_1')\downarrow \notin W$, що суперечить вищевказаній належності $u \Rightarrow_a (d_1')\downarrow \in W$;

c) $u \Rightarrow_a (d_1' \nabla_s d_2)\uparrow$, але існує u_0 – власний префікс u, такий, що $u_0 \Rightarrow_a (d_1' \nabla_s d_2)\downarrow$. У цьому випадку існує слово v_0, таке, що u_0 є префіксом v_0 і $v_0 \Rightarrow_a (d_1' \nabla_s d_2)\downarrow \in W \cup \{\varnothing\}$, причому v_0 є власним префіксом u. Позначимо $v_1 \in V^+$ – слово, таке, що $u = v_0 v_1$. Якщо $v_0 \Rightarrow_a (d_2)\uparrow$, то за лемою 1. 2), $v_0 \Rightarrow_a (d_1')\downarrow = v_0 \Rightarrow (d_1' \nabla_s d_2) \in W \cup \{\varnothing\}$, що суперечить вищевказаній належності $u \Rightarrow_a (d_1') \in W$. Якщо ж $v_0 \Rightarrow_a (d_2)\downarrow$, то існує слово $v_1' \in v_0 V^*$, таке, що $v_1' \Rightarrow_a (d_2)\downarrow \in W \cup \{\varnothing\}$. Тоді за лемою 1. 1), $v_1' \Rightarrow_a (d_1' \nabla_s d_2)\downarrow = v_1' \Rightarrow_a (d_2) \in W \cup \{\varnothing\}$. Оскільки $v_0 \in pref(v_1')$ і $v_0 \Rightarrow_a (d_1' \nabla_s d_2)\downarrow \in W \cup \{\varnothing\}$, то $v_1' = v_0$. Тоді $v_0 \Rightarrow_a (d_2)\downarrow \in W \cup \{\varnothing\}$, звідки за лемою 1. 1), $v_0 \Rightarrow_a (d_1 \nabla_s d_2)\downarrow = v_0 \Rightarrow_a (d_2) \in W \cup \{\varnothing\}$. Але $u \Rightarrow_a (d_1 \nabla_s d_2)\downarrow \in W$ і v_0 – власний префікс u, отже отримали протиріччя. Тобто припущення даного випадку виконуватися не може;

d) $u \Rightarrow_a (d_1' \nabla_s d_2) \uparrow$ для всіх префіксів u, зокрема $x \Rightarrow_a (d_1' \nabla_s d_2) \uparrow$, де x – перша літера u. У цьому випадку $x \Rightarrow_a (d_2) \uparrow$, бо інакше існує слово $v \in xV^*$, таке, що $v \Rightarrow_a (d_2) \downarrow \in W \cup \{\varnothing\}$, і за лемою 1. 1), $v \Rightarrow_a (d_1' \nabla_s d_2) \downarrow$, а отже $x \Rightarrow_a (d_1' \nabla_s d_2) \downarrow$. Тоді за лемою 1. 3), $u \Rightarrow_a (d_1 \nabla_s d_2) \cong u \Rightarrow_a (d_1)$ і $u \Rightarrow_a (d_1' \nabla_s d_2) \cong u \Rightarrow_a (d_1')$. Оскільки $u \Rightarrow_a (d_1 \nabla_s d_2) \downarrow \in W$ і $d_1 \prec_A d_1'$, то $u \Rightarrow_a (d_1' \nabla_s d_2) \downarrow = u \Rightarrow_a (d_1 \nabla_s d_2)$.

Таким чином, у всіх можливих випадках $u \Rightarrow_a (d_1' \nabla_s d_2) \downarrow = u \Rightarrow_a (d_1 \nabla_s d_2)$, а отже $d_1 \nabla_s d_2 \prec_A d_1' \nabla_s d_2$ за лемою 4, в силу довільності слова u. Лема доведена.

Лема 9. Нехай $d_1 \prec_w d_1'$ і $d_1 \nabla_s d_2 \downarrow$. Тоді $d_1' \nabla_s d_2 \downarrow$ і $d_1 \nabla_s d_2 \prec_N d_1' \nabla_s d_2$.

Доведення. Припустивши, що $d_1' \nabla_s d_2 \uparrow$, отримаємо $d_1' \in W$ або $d_2 \in W$, тому $d_1 \in W$ (в силу того, що $d_1 \prec_N d_1'$) або $d_2 \in W$, а отже $d_1 \nabla_s d_2 \uparrow$, що суперечить умові леми. Тому $d_1' \nabla_s d_2 \downarrow$, і достатньо довести, що $d_1 \nabla_s d_2 \prec_N d_1' \nabla_s d_2$.

Нехай $\bar{u} \in V^+$ – таке слово, що $\bar{u} \Rightarrow_a (d_1 \nabla_s d_2) \downarrow \notin W$. Тоді існує слово $u \in \bar{u}V^*$, таке, що $u \Rightarrow_a (d_1 \nabla_s d_2) \downarrow \in W \cup \{\varnothing\}$. Можливі 3 випадки:

1) $u \Rightarrow_a (d_2) \downarrow \in W \cup \{\varnothing\}$. За лемою 1. 1), $u \Rightarrow_a (d_1 \nabla_s d_2) \downarrow = u \Rightarrow_a (d_2)$ і $u \Rightarrow_a (d_1' \nabla_s d_2) \downarrow = u \Rightarrow_a (d_2)$, а отже $u \Rightarrow_a (d_1' \nabla_s d_2) \downarrow = u \Rightarrow_a (d_1 \nabla_s d_2)$. Тому $\bar{u} \Rightarrow_a (d_1' \nabla_s d_2) \downarrow \notin W$;

2) $u \Rightarrow_a (d_2) \downarrow \notin W \cup \{\varnothing\}$. У цьому випадку існує слово $u_1 \in uV^+$, таке, що $u_1 \Rightarrow_a (d_2) \downarrow \in W \cup \{\varnothing\}$. Тоді за лемою 1. 1), $u_1 \Rightarrow_a (d_1 \nabla_s d_2) \downarrow$, що суперечить припущенню $u \Rightarrow_a (d_1 \nabla_s d_2) \in W \cup \{\varnothing\}$, оскільки u є власним префіксом u_1. Тобто припущення $u \Rightarrow_a (d_2) \downarrow \notin W \cup \{\varnothing\}$ виконуватися не може;

3) $u \Rightarrow_a (d_2) \uparrow$. За лемою 1. 2), $u \Rightarrow_a (d_1) \downarrow = u \Rightarrow_a (d_1 \nabla_s d_2) \in W \cup \{\varnothing\}$.

Якщо $u \Rightarrow_a (d_1 \nabla_s d_2) \in W$, то оскільки $d_1 \prec_A d_1'$, то $u \Rightarrow_a (d_1') \downarrow \in W$ за лемою 4 і крім того, $\bar{u} < u$, бо $\bar{u} \Rightarrow_a (d_1 \nabla_s d_2) \notin W$. Якщо $u \Rightarrow_a (d_1 \nabla_s d_2) = \varnothing$, то $u \Rightarrow_a (d_1') \downarrow \notin W$, бо $d_1 \prec_N d_1'$. В обох випадках $u \Rightarrow_a (d_1') \downarrow$. Можливі такі випадки:

1) $u \Rightarrow_a (d_1' \nabla_s d_2) \downarrow \notin W \cup \{\varnothing\}$. Оскільки $\bar{u} \le u$, то $\bar{u} \Rightarrow_a (d_1' \nabla_s d_2) \downarrow \notin W$;

2) $u \Rightarrow_a (d_1' \nabla_s d_2) \downarrow \in W \cup \{\varnothing\}$. Якщо $u \Rightarrow_a (d_1 \nabla_s d_2) \in W$, то $\bar{u} < u$, тому $\bar{u} \Rightarrow_a (d_1' \nabla_s d_2) \downarrow \notin W$. Якщо $u \Rightarrow_a (d_1 \nabla_s d_2) = u \Rightarrow_a (d_1) = \varnothing$, то за лемою 1. 1), $u \Rightarrow_a (d_1') \downarrow = u \Rightarrow_a (d_1' \nabla_s d_2)$ і оскільки $d_1 \prec_N d_1'$, то $u \Rightarrow_a (d_1') \downarrow \notin W$ і $u \Rightarrow_a (d_1' \nabla_s d_2) \notin W$. Тоді $\bar{u} \Rightarrow_a (d_1' \nabla_s d_2) \downarrow \notin W$;

3) $u \Rightarrow_a (d_1') \in W$ і $u \Rightarrow_a (d_1' \nabla_s d_2) \uparrow$, але існує u_0 – власний префікс u, такий, що $u_0 \Rightarrow_a (d_1' \nabla_s d_2) \downarrow$. Тоді існує слово v_0, таке, що u_0 є префіксом v_0 і $v_0 \Rightarrow_a (d_1' \nabla_s d_2) \downarrow \in W \cup \{\varnothing\}$, причому v_0 є власним префіксом u. Позначимо $v_1 \in V^+$ – слово, таке, що $u = v_0 v_1$. Якщо $v_0 \Rightarrow_a (d_2) \uparrow$, то за лемою 1. 2), $v_0 \Rightarrow_a (d_1') \downarrow = v_0 \Rightarrow (d_1' \nabla_s d_2) \in W \cup \{\varnothing\}$, що суперечить тому, що $u \Rightarrow_a (d_1') \downarrow$. Якщо ж $v_0 \Rightarrow_a (d_2) \downarrow$, то існує слово $v_1' \in v_0 V^*$, таке, що $v_1' \Rightarrow_a (d_2) \downarrow \in W \cup \{\varnothing\}$. Тоді за лемою 1. 1), $v_1' \Rightarrow_a (d_1' \nabla_s d_2) \downarrow = v_1' \Rightarrow_a (d_2) \in W \cup \{\varnothing\}$. Оскільки $v_0 \in pref(v_1')$ і $v_0 \Rightarrow_a (d_1' \nabla_s d_2) \downarrow \in W \cup \{\varnothing\}$, то $v_1' = v_0$. Тоді $v_0 \Rightarrow_a (d_2) \downarrow \in W \cup \{\varnothing\}$, звідки за лемою 1. 1), $v_0 \Rightarrow_a (d_1 \nabla_s d_2) \downarrow = v_0 \Rightarrow_a (d_2) \in W \cup \{\varnothing\}$. Але $u \Rightarrow_a (d_1 \nabla_s d_2) \downarrow \in W \cup \{\varnothing\}$ і v_0 – власний префікс u, отже отримали протиріччя. Тобто припущення даного випадку виконуватися не може;

4) $v \Rightarrow_a (d_1' \nabla_s d_2) \uparrow$ для всіх префіксів $v \in pref(u)$, зокрема, $x \Rightarrow_a (d_1' \nabla_s d_2) \uparrow$, де x – перша літера u. У цьому випадку $x \Rightarrow_a (d_2) \uparrow$, бо інакше існує слово $v \in xV^*$, таке, що $v \Rightarrow_a (d_2) \downarrow \in W \cup \{\varnothing\}$, і за лемою 1. 1), $v \Rightarrow_a (d_1' \nabla_s d_2) \downarrow$, а отже $x \Rightarrow_a (d_1' \nabla_s d_2) \downarrow$. Тоді за лемою 1. 3), $u \Rightarrow_a (d_1 \nabla_s d_2) \cong u \Rightarrow_a (d_1)$ і $u \Rightarrow_a (d_1' \nabla_s d_2) \cong u \Rightarrow_a (d_1')$. Оскільки $u \Rightarrow_a (d_1 \nabla_s d_2) \downarrow$ і $d_1 \prec_w d_1'$, то $u \Rightarrow_a (d_1' \nabla_s d_2) \downarrow$ – отримали протиріччя з припущенням $x \Rightarrow_a (d_1' \nabla_s d_2) \uparrow$.

В усіх можливих випадках $\bar{u} \Rightarrow_a (d_1' \nabla_s d_2) \downarrow \notin W$, а отже $d_1 \nabla_s d_2 \prec_N d_1' \nabla_s d_2$ в силу довільності слова u.

Лема доведена.

Лема 10 (монотонність структурного накладання за першим аргументом відносно $\prec_w$). Нехай $d_1 \prec_w d_1'$ і $d_1 \nabla_s d_2 \downarrow$. Тоді $d_1' \nabla_s d_2 \downarrow$ і $d_1 \nabla_s d_2 \prec_w d_1' \nabla_s d_2$.

Доведення випливає безпосередньо з лем 8 і 9.

Лема 11 (обмежена монотонність структурного накладання за другим аргументом відносно $\prec_A$). Нехай $d_1, d_2, d_2' \in NDVC(V,W) \setminus W$, $d_2 \prec_A d_2'$ і $rn(d_2) = rn(d_2')$. Тоді $d_1 \nabla_s d_2 \prec_A d_1 \nabla_s d_2'$.

Доведемо індукцією за k, що якщо $d_1 \in NDVC_k(V,W) \setminus W$, $d_2, d_2' \in NDVC(V,W) \setminus W$, $d_2 \prec_A d_2'$ і $rn(d_2) = rn(d_2')$, то $d_1 \nabla_s d_2 \prec_A d_1 \nabla_s d_2'$.

База індукції. Якщо $k=0$, то $d_1=\varnothing$ і $d_1\nabla_s d_2=d_2\prec_A d_2'=d_1\nabla_s d_2'$.

Індукційний перехід. Припустимо, що твердження вже доведено для всіх $d_1\in NDVC_k(V,W)\setminus W$. Оберемо $d_1\in NDVC_{k+1}(V,W)\setminus NDVC_k(V,W)$ і $d_2,d_2'\notin W$, такі, що $d_2\prec_A d_2'$, $rn(d_2)=rn(d_2')$ і доведемо, що $d_1\nabla_s d_2\prec_A d_1\nabla_s d_2'$.

Нехай $u\in V^+$ – слово, таке, що $u\Rightarrow_a(d_1\nabla_s d_2)\downarrow\in W$. Тоді існує слово $v\in pref(u)$, що є кореневим ім'ям $d_1\nabla_s d_2$. Можливі такі випадки:

1) слово v не порівнюється з жодним кореневим іменем d_2. В цьому випадку $v\in rn(d_1)$ (бо інакше $(d_1\nabla_s d_2)(v)\uparrow$ за правилом 5 визначення операції ∇_s) і $d_1(v)=(d_1\nabla_s d_2)(v)$ за правилом 4 визначення операції ∇_s. Оскільки $rn(d_2)=rn(d_2')$, то u не порівнюється з жодним кореневим іменем d_2'. Тоді $(d_1\nabla_s d_2')(v)\downarrow=d_1(v)$, звідки $u\Rightarrow_a(d_1\nabla_s d_2')\downarrow=u\Rightarrow_a(d_1\nabla_s d_2)$, бо $v\in pref(u)$;

2) слово v є власним префіксом деякого кореневого імені v_1 даного d_2 і неатомним кореневим ім'ям в d_1. В цьому випадку $(d_1\nabla_s d_2)(v)=d_1(v)\nabla_s(d_2/v)$ за правилом 2 визначення операції ∇_s. Оскільки $rn(d_2)=rn(d_2')$, то $d_2'(v_1)\downarrow$ і $(d_1\nabla_s d_2')(v)=d_1(v)\nabla_s(d_2'/v)$. Оскільки $d_2'/v\neq\varnothing$ і $d_2\prec_A d_2'$, то $d_2/v\prec_A d_2'/v$ за лемою 5. 3). Оскільки $rn(d_2)=rn(d_2')$, то $rn(d_2/v)=rn(d_2'/v)$. Оскільки $d_1(v)\in NDVC_k(V,W)$, то за припущенням індукції, $(d_1\nabla_s d_2)(v)\prec_A(d_1\nabla_s d_2')(v)$. Тоді $u\Rightarrow_a(d_1\nabla_s d_2')\downarrow=u\Rightarrow_a(d_1\nabla_s d_2)$ за лемою 4, оскільки $v\in pref(u)$ і $u\Rightarrow_a(d_1\nabla_s d_2)\in W$;

3) слово v є власним префіксом деякого кореневого імені v_1 даного d_2 і атомним кореневим ім'ям в d_1. У цьому випадку $(d_1\nabla_s d_2)(v)=d_2/v$ за правилом 3 визначення операції ∇_s. Оскільки $rn(d_2)=rn(d_2')$, то $d_2'(v_1)\downarrow$ і $(d_1\nabla_s d_2')(v)=d_2'/v$. Оскільки $d_2'/v\neq\varnothing$ і $d_2\prec_A d_2'$, то $d_2/v\prec_A d_2'/v$ за лемою 5. 3). Тоді $u\Rightarrow_a(d_1\nabla_s d_2')\downarrow=u\Rightarrow_a(d_1\nabla_s d_2)$ за лемою 4, оскільки $v\in pref(u)$ і $u\Rightarrow_a(d_1\nabla_s d_2)\in W$;

4) не виконуються умови 1–3. Тоді за правилами 1 та 5 визначення операції ∇_s, $v\in rn(d_2)\setminus rn(d_1)V^+$ і $(d_1\nabla_s d_2)(v)=d_2(v)$. Оскільки $rn(d_2)=rn(d_2')$, то $v\in rn(d_2')\setminus rn(d_1)V^+$. Тоді $(d_1\nabla_s d_2')(v)\downarrow=d_2'(v)$. Оскільки $v\in pref(u)$ і $d_2\prec_A d_2'$, то $u\Rightarrow_a(d_2')\downarrow=u\Rightarrow_a(d_2)\in W$ за лемою 4. Тоді $u\Rightarrow(d_1\nabla_s d_2')\downarrow=u\Rightarrow(d_2')=u\Rightarrow(d_2)=u\Rightarrow(d_1\nabla_s d_2)$.

Отже в усіх випадках $u\Rightarrow_a(d_1\nabla_s d_2)\downarrow=u\Rightarrow_a(d_1\nabla_s d_2')$, звідки $d_1\nabla_s d_2\prec_A d_1\nabla_s d_2'$ за лемою 4 в силу довільності слова u. Індукційний перехід завершено. Лема доведена.

Лема 12. (обмежена монотонність структурного накладання за другим аргументом відносно $\prec_N$). Нехай $d_1,d_2,d_2'\in NDVC(V,W)\setminus W$, $d_2\prec_N d_2'$ і $rn(d_2)=rn(d_2')$. Тоді $d_1\nabla_s d_2\prec_N d_1\nabla_s d_2'$.

Доведемо індукцією за k, що якщо $d_1\in NDVC_k(V,W)\setminus W$, $d_2,d_2'\in NDVC(V,W)\setminus W$, $d_2\prec_N d_2'$ і $rn(d_2)=rn(d_2')$, то $d_1\nabla_s d_2\prec_N d_1\nabla_s d_2'$.

База індукції. Якщо $k=0$, то $d_1=\varnothing$ і $d_1\nabla_s d_2=d_2\prec_N d_2'=d_1\nabla_s d_2'$.

Індукційний перехід. Припустимо, що твердження вже доведено для всіх $d_1\in NDVC_k(V,W)\setminus W$. Оберемо $d_1\in NDVC_{k+1}(V,W)\setminus NDVC_k(V,W)$ і $d_2,d_2'\notin W$, такі, що $d_2\prec_N d_2'$, $rn(d_2)=rn(d_2')$ і доведемо, що $d_1\nabla_s d_2\prec_N d_1\nabla_s d_2'$.

Нехай $u\in V^+$ – слово, таке, що $u\Rightarrow_a(d_1\nabla_s d_2)\downarrow\notin W$. Тоді існує слово $v\in pref(u)$, що є кореневим ім'ям $d_1\nabla_s d_2$. Можливі такі випадки:

1) слово v не порівнюється з жодним кореневим іменем d_2. В цьому випадку $v\in rn(d_1)$ (бо інакше $(d_1\nabla_s d_2)(v)\uparrow$ за правилом 5 визначення операції ∇_s) і $d_1(v)=(d_1\nabla_s d_2)(v)$ за правилом 4 визначення операції ∇_s. Оскільки $rn(d_2)=rn(d_2')$, то u не порівнюється з жодним кореневим іменем d_2'. Тоді $(d_1\nabla_s d_2')(v)\downarrow=d_1(v)$, звідки $u\Rightarrow_a(d_1\nabla_s d_2')\downarrow=u\Rightarrow_a(d_1\nabla_s d_2)$, бо $v\in pref(u)$, тому $u\Rightarrow_a(d_1\nabla_s d_2')\downarrow\notin W$;

2) слово v є власним префіксом деякого кореневого імені v_1 даного d_2 і неатомним кореневим ім'ям в d_1. У цьому випадку $(d_1\nabla_s d_2)(v)=d_1(v)\nabla_s(d_2/v)$ за правилом 2 визначення операції ∇_s. Оскільки $rn(d_2)=rn(d_2')$, то $d_2'(v_1)\downarrow$ і $(d_1\nabla_s d_2')(v)=d_1(v)\nabla_s(d_2'/v)$. Оскільки $d_2'/v\neq\varnothing$ і $d_2\prec_N d_2'$, то $d_2/v\prec_N d_2'/v$ за лемою 5. 3). Оскільки $rn(d_2)=rn(d_2')$, то $rn(d_2/v)=rn(d_2'/v)$. Оскільки $d_1(v)\in NDVC_k(V,W)$, то за припущенням індукції, $(d_1\nabla_s d_2)(v)\prec_N(d_1\nabla_s d_2')(v)$. Тоді $u\Rightarrow_a(d_1\nabla_s d_2')\downarrow\notin W$, оскільки $v\in pref(u)$ і $d_2\prec_N d_2'$;

3) слово v є власним префіксом деякого кореневого імені v_1 даного d_2 і атомним кореневим ім'ям в d_1. В цьому випадку $(d_1\nabla_s d_2)(v) = d_2/v$ за правилом 3 визначення операції ∇_s. Оскільки $rn(d_2) = rn(d_2')$, то $d_2'(v_1)\downarrow$ і $(d_1\nabla_s d_2')(v) = d_2'/v$. Оскільки $d_2'/v \neq \varnothing$ і $d_2 \prec_N d_2'$, то $d_2/v \prec_N d_2'/v$ за лемою 5. 3). Тоді $u \Rightarrow_a (d_1\nabla_s d_2')\downarrow \notin W$, оскільки $v \in pref(u)$ і $u \Rightarrow_a (d_1\nabla_s d_2) \notin W$.

4) не виконуються умови 1–3. Тоді за правилами 1 та 5 визначення операції ∇_s, $v \in rn(d_2) \setminus rn(d_1)V^+$ і $(d_1\nabla_s d_2)(v) = d_2(v)$. Оскільки $rn(d_2) = rn(d_2')$, то $v \in rn(d_2') \setminus rn(d_1)V^+$. Тоді $(d_1\nabla_s d_2')(v)\downarrow = d_2'(v)$. Оскільки $v \in pref(u)$, то $u \Rightarrow_a (d_2)\downarrow \notin W$ і оскільки $d_2 \prec_N d_2'$, то $u \Rightarrow_a (d_2')\downarrow \notin W$. Тоді $u \Rightarrow (d_1\nabla_s d_2')\downarrow = u \Rightarrow (d_2') \notin W$.

Отже в усіх випадках $u \Rightarrow_a (d_1\nabla_s d_2')\downarrow \notin W$, звідки $d_1\nabla_s d_2 \prec_N d_1\nabla_s d_2'$ в силу довільності слова u. Індукційний перехід завершено. Лема доведена.

Лема 13 (обмежена монотонність структурного накладання за другим аргументом відносно $\prec_w$). Нехай $d_1, d_2, d_2' \in NDVC(V,W) \setminus W$, $d_2 \prec_w d_2'$ і $rn(d_2) = rn(d_2')$. Тоді $d_1\nabla_s d_2 \prec_w d_1\nabla_s d_2'$.

Доведення випливає безпосередньо з лем 11 і 12.

Таким чином, з наведених лем випливає, що операції іменування і асоціативного розіменування є монотонними за відношенням $\prec_w$, а операція структурного накладання монотонна за першим аргументом і монотонна в обмеженому сенсі за другим аргументом.

4. Мова $SICON_T$

Визначимо мову програм над складноіменними даними, яку назвемо $SICON_T$, програми якої будуть монотонними за відношенням $\prec_w$. Як зазначалося раніше, для цього необхідно вказати базові функції і композиції. Оберемо деяке фіксоване значення $T \in W$, яке інтерпретується як логічна константа, і позначимо $F = NDVC(V,W) \to NDVC(V,W)$.

До базових функцій мови $SICON_T$ належать:

- сім'я операцій іменування $\Rightarrow v$ з параметром $v \in V^+$;
- сім'я операцій асоціативного розіменування $v \Rightarrow_a$ з параметром $v \in V^+$;
- функція-константа $\overline{T}$, яка на всіх вхідних даних приймає значення T.

Як композицію мови $SICON_T$ оберемо найпростіші композиції структурного програмування.

1. М*ноження* – це бінарна композиція, яка за функціями $f, g \in F$ будує функцію $h \in F$ (позначається $h = f \bullet g$), таку, що $h(d) \cong g(f(d))$.

2. *Цикл* – це бінарна композиція, яка за функціями $p \in F$ (умова) і $g \in F$ (тіло циклу) будує функцію $h \in F$ (позначається $h = p^* g$), таку, що

- $h(d) = d$, якщо $p(d)\downarrow = T$;
- $h(d) = g^{(k)}(d)$, якщо $p(g^{(k)}(d))\downarrow = T$ і $p(g^{(i)}(d))\downarrow \neq T$ для всіх $i \in \{1,2,..,k-1\}$ (тут $g^{(k)}(d) = g(g(..g(d)..))$ – k-кратне застосування функції g до даного d);
- $h(d)$ не визначено, в інших випадках.

3. *Присвоєння* Asg^v, залежне від параметру $v \in V^+$ – унарна композиція на F, що задається формулою $Asg^v(f)(d) \cong d\nabla_s[v \mapsto f(d)]$.

Тоді всі функції (програми), які виражаються в мові $SICON_T$, складають множину $SFA^T(V,W) = \bigcup_{k\geq 0} SFA_k^T(V,W)$, де

- $SFA_0^T(V,W) = \{\overline{T}\} \cup \{\Rightarrow u \mid u \in V^+\} \cup \{u \Rightarrow_a \mid u \in V^+\}$;
- $SFA_{k+1}^T(V,W) = SFA_k^T(V,W) \cup \{f \bullet g \mid f, g \in SFA_k^T(V,W)\} \cup$
$$\cup \{p^* g \mid p, g \in SFA_k^T(V,W)\} \cup \{Asg^v(f) \mid f \in SFA_k^T(V,W) \wedge v \in V^+\}.$$

Тобто $SFA^T(V,W)$ – це функції, що можуть бути отримані з базових застосуванням композицій.

Незважаючи на простоту, мова $SICON_T$ є достатньо потужною. По-перше, як і в мові $SICON_a$ [2], в ній можна промоделювати всі частково рекурсивні функції, тобто ця мова є функціонально повною. По-друге, в мові $SICON_a$ можна промоделювати похідними композиціями багато композицій, що використовуються у програмуванні – різноманітні умовні оператори, цикли тощо. Це дозволяє стверджувати, що мова $SICON_T$ є структурно адекватною. По-третє, вона очевидно, має стислий та формальний опис. Наступна теорема показує, що всі програми цієї мови є монотонними за відношенням $\prec_w$ (і як наслідок, номінативно стабільними).

Теорема 1. Програми мови $SICON_T$ є монотонними за відношенням $\prec_w$, тобто довільна функція $f \in SFA^T(V,W)$ є монотонною в такому сенсі:

якщо $d_1 \prec_w d_2$ і $f(d_1)\downarrow$, то $f(d_2)\downarrow$ і $f(d_1) \prec_w f(d_2)$.

Доведемо індукцією за рангом d_1, що кожна функція $f \in SFA_k^T(V,W)$ монотонна.

База індукції. При $k=0$, $SFA_0^T(V,W) = \{\overline{T}\} \cup \{\Rightarrow u \mid u \in V^+\} \cup \{u \Rightarrow_a \mid u \in V^+\}$, тому монотонність функцій з $SFA_0^T(V,W)$ випливає з леми 5.

Індукційний перехід. Припустимо, що для деякого k твердження вже доведено, і доведемо його для кожної функції з $SFA_{k+1}^T(V,W)$. Нехай $f \in SFA_{k+1}^T(V,W) \setminus SFA_k^T(V,W)$.

Можливі такі випадки:

1) $f = g \bullet h$ для деяких $g,h \in SFA_k^T(V,W)$. Нехай $d_1 \prec_w d_2$ і $f(d_1)\downarrow$. Тоді $g(d_1)\downarrow$ і за індукційним припущенням, $g(d_2)\downarrow$ і $g(d_1) \prec_w g(d_2)$. Звідси $f(d_2) \cong h(g(d_2))\downarrow$ і $h(g(d_1)) = f(d_1) \prec_w f(d_2)$. Отже, функція f монотонна;

2) $f = g * h$ для деяких $g,h \in SFA_k^T(V,W)$. Нехай $d_1 \prec_w d_2$ і $f(d_1)\downarrow$. Тоді для деякого $m \in Nat$ виконується $h^{(m)}(d_1)\downarrow = f(d_1)$, $g(h^{(m)}(d_1))\downarrow = T$ і $g(h^{(i)}(d_1))\downarrow \neq T$ для всіх $i \in \{0,1,..,m-1\}$. Використовуючи m разів монотонність функції h, отримуємо, що $h^{(i)}(d_2)\downarrow$ і $h^{(i)}(d_1)\downarrow \prec_w h^{(i)}(d_2)$ для $i \in \{0,1,..,m\}$. Із монотонності g випливає, що $g(h^{(i)}(d_2))\downarrow \neq T$ для всіх $i \in \{0,1,..,m-1\}$ і $g(h^{(m)}(d_2))\downarrow = T$. Тоді за визначенням циклу, $f(d_2) \cong (g*h)(d_2)\downarrow = h^{(m)}(d_2)$ і $f(d_1) \prec_w f(d_2)$. Отже f монотонна;

3) $f = Asg^u(g)$ для деякого $u \in V^+$ і $g \in SFA_k(V,W)$. Нехай $d_1 \prec_w d_2$ і $f(d_1)\downarrow$. Тоді $g(d_1)\downarrow$ і за монотонністю g, $g(d_2)\downarrow$ і $g(d_1) \prec_w g(d_2)$. Крім того, $f(d_1) = d_1 \nabla_s [u \mapsto g(d_1)]$, $f(d_2) \cong d_2 \nabla_s [u \mapsto g(d_2)]$. Оскільки $d_1 \notin W$, то $d_2 \notin W$ і $f(d_2)\downarrow$. Оскільки $g(d_1) \prec_w g(d_2)$, то за лемою 7, $[u \mapsto g(d_1)] \prec_w [u \mapsto g(d_2)]$. Тоді з лем 10 і 13 випливає, що $f(d_1) = d_1 \nabla_s [u \mapsto g(d_1)] \prec_w d_1 \nabla_s [u \mapsto g(d_2)] \prec_w d_2 \nabla_s [u \mapsto g(d_2)] = f(d_2)$. Отже функція f монотонна. Індукційний перехід завершено. Теорема доведена.

Наслідком отриманого результату є наступне твердження.

Теорема 2. Програми мови $SICON_T$ є номінативно стабільними за відношенням $\approx$, тобто довільна функція $f \in SFA^T(V,W)$ є стабільною в такому сенсі:

якщо $d_1 \approx d_2$ і $f(d_1)\downarrow$ або $f(d_2)\downarrow$, то обидва значення $f(d_1)$ та $f(d_2)$ визначені і $f(d_1) \approx f(d_2)$.

Значення отриманих результатів полягає в тому, що програміст при написанні програм мови $SICON_T$ (або подібних мов) може не займатися "ручним" узгодженням ієрархій даних між різними компонентами програм (програмних систем), бо таке узгодження відбувається автоматично.

Висновки

У роботі побудовано та досліджено достатньо потужну формальну мову $SICON_T$ номінативно стабільних та монотонних програм щодо структурних трансформацій даних. Така мова може стати теоретичною основою для практичних мов програмування, які дозволять спростити процес програмування та зробити його більш ефективним.

1. *Никитченко Н.С.* Композиционно-номинативный подход к уточнению понятия программы // Проблеми програмування.– 1999.– № 1. – С. 16–31.
2. *Нікітченко М.С., Іванов Є.В.* Властивості композиційно-номінативних мов програм з асоціативним розіменуванням // Матеріали XVI Всеукраїнської наук. конф. "Сучасні проблеми прикладної математики та інформатики". 8–9 жовтня 2009. – Львів, 2009. – С. 157–158.