Додаток 1

ПОРІВНЯЛЬНА ТАБЛИЦЯ

із наведенням фрагментів дисертації Марченка В. В.
та відповідних фрагментів опублікованих текстів інших авторів
без зазначення авторства

Збіги текстів виділені жовтим кольором, перефразування та синоніми – бірюзовим, перестановки слів місцями – зеленим; твердження особи про те, що це нібито вона щось пропонує, оцінює, розробила чи робить висновки – фіолетовим.

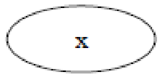
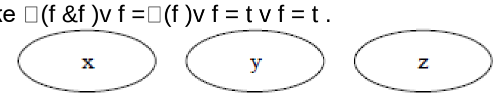
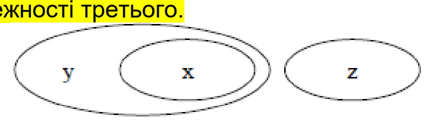
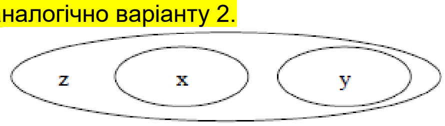
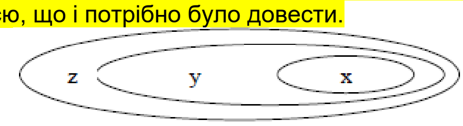
Червоним шрифтом поданий коментар щодо фрагментів дисертації Марченка В. В.

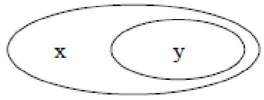
Через великий обсяг кольором виділено не всі збіги.

№	Фрагменти тексту дисертації, у якій виявлено факти порушення академічної доброчесності	Фрагменти опублікованих текстів інших авторів (без зазначення в дисертації Марченка В. В. посилань на джерело)
	Вид виявленого порушення: **плагіат**	
1	**Марченко В. В.** **Метод виявлення шкідливих процесів в інформаційній системі підприємства на основі ідентифікації та діагностування станів логічних об'єктів.** – Дис. … доктора філософії. – Київ, 2023. (https://dut.edu.ua/uploads/p_2625_24461150.pdf)	**Слюняєв А. С.** **Методика побудови інтелектуальної інформаційно-керуючої системи аеропорту на основі мультиагентного підходу.** – Дис. … кандидата технічних наук. – Київ, 2010. (https://nrat.ukrintei.ua/searchdoc/0411U002143/) **Нумерація сторінок – як надруковано у файлі pdf на сторінках справа вгорі.**
	С. 66.	**С. 56.**
	Формалізація компонентів моделі логічного об'єкта. Формалізація компонентів моделі ідентифікації та діагностики логічного об'єкта (ЛО), побудовано в рамках деякої формальної інформаційної системи. Як математичний апарат такої теорії (числення об'єктів) доцільно використати логіку першого порядку. Нехай α – множина функціональних, предикативних і константних символів. Кожному, функціональному символу f Є α можна поставити у відповідність ціле позитивне число F(f) таке, що якщо n = F(f), то f називатимемо n-арним функціональним символом. Кожен предикативний символ R Є α можна пов'язати з позитивним цілим числом F(R); якщо n = F(R), то R називатимемо n-арним предикативним символом. Предикативні символи за своєю суттю є символами відносин. Відповідно до [18,51], під алгебраїчною системою Φ розумітимемо непорожню сукупність M об'єктів, яка є областю дії кванторів, разом з інтерпретацією основних предикативних, функціональних і константних символів з α. **Покликання [18] – це:** Bokr, J., & Jáneš, V. (2006). State of a logical object. Acta Polytechnica, 46(1). https://doi.org/10.14311/808. **Покликання [51] – це:** Lakhno, V., Kazmirchuk, S., Kovalenko, Y., Myrutenko, L., & Zhmurko, T. (2016). Design of adaptive system of detection of cyber-attacks, based on the model of logical procedures and the coverage matrices of features. Eastern-European Journal of Enterprise Technologies, 3(9(81), 30.	**Побудова формальної об'єктної системи.** Необхідним базисом для формалізації інтелектуальних компонентів ІКСА є модель інформаційного об'єкта (ІО), яка повинна будуватися в рамках деякої формальної системи (теорії). Як математичний апарат такої теорії (числення об'єктів) доцільно використати логіку першого порядку. Нехай α – множина функціональних, предикативних і константних символів. Кожному, функціональному символу f Є α можна поставити у відповідність ціле позитивне число #(f) таке, що якщо n = #(f), то f називатимемо n-арним функціональним символом. Кожен предикативний символ R Є α можна пов'язати з позитивним цілим числом #(R); якщо n = #(R), то R називатимемо n-арним предикативним символом. Предикативні символи за своєю суттю є символами відносин. Відповідно до [105], під алгебраїчною системою Φ розумітимемо непорожню сукупність M об'єктів, яка є областю дії кванторів, разом з інтерпретацією основних предикативних, функціональних і константних символів з α. **Покликання [105] – це:** Справочная книга по математической логике: Справочник в 4 ч.: Ч. 1. Теория моделей: пер. с англ. / Под ред. Дж. Барвайса. – М.: Наука, 1982. – 392 с.

	https://doi.org/10.15587/1729-4061.2016.71769. **Ці джерела фальшиві: в них немає слів «algebraic» та «predicative».** **Плагіат.**	
	С. 66–67.	**С. 57.**
	Алгебраїчна система для мови *α* є пара Ф = <M,F>, де M непорожня множина, F – відображення з областю визначення *α* таке, що: 1) якщо RЄα – n-арний предикативний символ, то F(R) □ M^n; 2) якщо f Єα – n-арний функціональний символ, то F(f): Mn —> M ; 3) якщо cЄα – константний символ, то F(c) ЄM. Основними синтаксичними поняттями логіки 1-го порядку є: логічні зв'язки &, v, □, →,=; квантори спільності і існування □ і □ , символи логічних змінних x, y, z... Всяку скінчену послідовність, елементами якої є основні символи або елементи, будемо назвати виразом.	Алгебраїчна система для мови α є пара Ф = <M,F>, де M непорожня множина, F – відображення з областю визначення α таке, що: 1) якщо RЄ α – n-арний предикативний символ, то F(R) □M^n; 2) якщо f Є α – n-арний функціональний символ, то F(f): M^n —> M ; 3) якщо cЄ α – константний символ, то F(c) ЄM. Основними синтаксичними поняттями логіки 1-го порядку є: логічні зв'язки &,v,□,→,=; квантори спільності і існування □ і □, символи логічних змінних x, y, z... Всяку скінчену послідовність, елементами якої є основні символи або елементи, будемо назвати виразом.
	С. 67.	**С. 57.**
	Терми мови *α* утворюють найменшу множину виразів, що містять x, y, z, всі константні символи *α* (якщо такі є) і замкнуту відносно правила утворення: якщо t1,…,tn – терми *α* і якщо f Єα – n-місний функціональний символ, то вираз f(t1,…,tn) є термом мови *α*. Терм [14,22,27], що не містить змінних, називатимемо замкнутим. Атомарна формула мови *α* – це вираз одного з наступних видів: (t1= t2) , R(t 1 , . . . , t n) , де t1, t2 – терми мови, а R Є α — довільний n-місний предикативний символ.	Терми мови α утворюють найменшу множину виразів, що містять x, y, z, всі константні символи α (якщо такі є) і замкнуту відносно правила утворення: якщо t1,…,tn – терми α і якщо f Є α – n-місний функціональний символ, то вираз f(t1,…,tn) є термом мови α . Терм, що не містить змінних, називатимемо замкнутим. Атомарна формула мови α – це вираз одного з наступних видів: (t1= t2) , R(t 1 , . . . , t n) , де t1, t2 – терми мови, а R Є α — довільний n-місний предикативний символ.
	С. 67.	**С. 57.**
	Формули першого порядку мови *α* утворюють найменшу множину виразів, що містить атомарні формули і замкнуту щодо наступного правила утворення: 1) якщо φ і ψ – формули, то вирази –φ , (φ& ψ), (φ v ψ), (φ → ψ) також є формулами. 2) якщо φ – формула і v – змінна, то (□vφ) і (Vvφ) також є формулами. Множина Fv(φ) вільних змінних формули φ визначається таким чином:	Формули першого порядку мови α утворюють найменшу множину виразів, що містить атомарні формули і замкнуту щодо наступного правила утворення: 1) якщо φ і ψ – формули, то вирази –φ, (φ& ψ), (φ v ψ), (φ→ ψ) також є формулами. 2) якщо φ – формула і v – змінна, то (□vφ) і (Vvφ) також є формулами. Множина Fv(φ) вільних змінних формули φ визначається таким чином:
	С. 67.	**С. 58.**
	1) якщо φ – атомарна формула, то Fv(φ) в точності множина змінних, що зустрічаються в φ; 2) Fv(–φ)= Fv(φ); 3) Fv(φ& ψ) = Fv(φv)= Fv(φ)v Fv(ψ); 4) Fv(□vφ) = Fv(Vvφ) = Fv(φ) – {v}. Реченням (першого порядку) мови α називатимемо формулу, що не містить вільних змінних.	1) якщо φ – атомарна формула, то Fv(φ) в точності множина змінних, що зустрічаються в φ; 2) Fv(–φ)= Fv(φ); 3) Fv(φ&ψ)= Fv(φv)= Fv(φ)v Fv(ψ); 4) Fv(□vφ)= Fv(Vvφ)= Fv(φ)–{v}. Реченням (першого порядку) мови α називатимемо формулу, що не містить вільних змінних.
	С. 68.	**С. 58.**
	Будуємо систему алгебри для мови логіки 1-го порядку α над множиною M-логічних об'єктів. Змінними позначаються ЛО [18, 51], далі вводяться предикативні і функціональні символи над множиною M. Логічний об'єкт пропонується визначити наступним чином: O: = <NO, {A}, {O}, {F}> де NO – ім'я об'єкта; {A} – множина атрибутів об'єкта (A0,…,An), де Ai – i-й атрибут ЛО; {O} – множина об'єктів, які структурно входять до даного об'єкта, (O_{NO1},O_{NO2},…,O_{NOm}), де O_{NOi} i-й підпорядко-	Будуємо систему алгебри для мови логіки 1-го порядку α над множиною M-інформаційних об'єктів IKCA. Змінними позначаються IO, далі вводяться предикативні і функціональні символи над множиною M. Інформаційний об'єкт пропонується визначити наступним чином: O: = <NO, {A}, {O}, {F}> де NO – ім'я об'єкта; {A} – множина атрибутів об'єкта (A0,…,An), де Ai – i-й атрибут IO; {O} – множина об'єктів, які структурно входять до даного об'єкта, (O_{NO1},O_{NO2},O_{NOm}), де O_{NOi} i-й підпорядкова-

	ваний об'єкт, об'єкту з ім'ям NO; {F} – множина функцій, які виконує даний ЛО. **До чужого переписаного тексту Марченко додав два покликання.** **Покликання [18] – це:** Bokr, J., & Jáneš, V. (2006). State of a logical object. Acta Polytechnica, 46(1). https://doi.org/10.14311/808. **Покликання [51] – це:** Lakhno, V., Kazmirchuk, S., Kovalenko, Y., Myrutenko, L., & Zhmurko, T. (2016). Design of adaptive system of detection of cyber-attacks, based on the model of logical procedures and the coverage matrices of features. Eastern-European Journal of Enterprise Technologies, 3(9(81), 30. https://doi.org/10.15587/1729-4061.2016.71769. **Ці джерела фальшиві: в них немає слова «predicative» та наведених формул.** **Марченко у переписаному чужому тексті замінив слова «**М-інформаційних об'єктів**» на «**М-логічних об'єктів**», «**Інформаційний об'єкт**» на «**Логічний об'єкт**» та «**ІО**» на «**ЛО**».** **Нахабний плагіат.**	ний об'єкт, об'єкту з ім'ям NO; {F} – множина функцій, які виконує даний ІО.
	С. 68.	**С. 58–59.**
	Атрибут ЛО визначимо як: A = <NA, SA, VA> де NA – ім'я атрибута, SA – множина, на якій визначається значення атрибута, VA – значення атрибута, тобто α Є SA в даний момент часу t. Примітивним ЛО будемо називати такий ЛО, що O = <NO, {A}, ø , {F}>, тобто що має порожню множину вкладених ЛО. Можна ввести структурний ЛО (або пасивний) [14, 22, 27], який визначається як O = <NO, {A}, ø, ø>, тобто який має порожню множину вкладених ЛО і порожню множину функцій, а також ввести однопараметричний ЛО: O = <NO, A, ø, ø> для скорочення запису що позначається як <NO, A>, що має тільки один атрибут. Таке визначення ЛО дозволяє розглядати множину M алгебраїчної системи як актуальну і залишатися в рамках логіки 1-го порядку при розгляді відносин між ЛО. **До чужого переписаного тексту Марченко додав три покликання. Ці покликання фальшиві: 2018-го, 2014-го та 2016-го років.** **Справжнє джерело – це дисертація Слюняєва, яка була захищена 2010 року.** **Плагіат.**	Атрибут ІО визначимо як: A = <NA, SA, VA> де NA – ім'я атрибута, SA – множина, на якій визначається значення атрибута, VA – значення атрибута, тобто αЄSA в даний момент часу t. Примітивним ІО будемо називати такий ІО, що O = <NO, {A}, ø, {F}>, тобто що має порожню множину вкладених ІО. Можна ввести структурний ІО (або пасивний), який визначається як O = <NO, {A}, ø, ø >, тобто який має порожню множину вкладених ІО і порожню множину функцій, а також ввести однопараметричний ІО: O = <NO, A, ø, ø > для скорочення запису що позначається як <NO, A>, що має тільки один атрибут. Таке визначення ІО дозволяє розглядати множину M алгебраїчної системи як актуальну і залишатися в рамках логіки 1-го порядку при розгляді відносин між ІО.
	С. 68–69.	**С. 59.**
	Почнемо введення відносин в дану модель Φ з відношення належності об'єктів (інакше кажучи, відношення структурної підпорядкованості) Rs. Аксіома R.1: Відношення Rs – антирефлексивне: □x(□ Rs (x,x)). Це витікає із змістовного поняття структурної вкладеності (сам предмет не містить самого себе у якості структурного елемента). Аксіома R.2: Відношення Rs – транзитивне: □x□y□z(Rs(x, y)&Rs (y,z)→Rs (x,z)). Аксіома R.3: Відношення Rs – несиметричне: □x□y(Rs(x, y)→□Rs(y,x)). Розглядаючи ці властивості як аксіоми в численні предикатів, покажемо, що вони є загальнозначущими формулами в мові α (або тавтологіями). Для аксіоми R.1: □x(□Rs(x,x)). Скористаємося геометричною інтерпретацією.	Почнемо введення відносин в дану модель Φ з відношення належності об'єктів (інакше кажучи, відношення структурної підпорядкованості) Rs. Аксіома R.1: Відношення Rs – антирефлексивне: □x(□Rs(x, x)). Це витікає із змістовного поняття структурної вкладеності (сам предмет не містить самого себе у якості структурного елемента). Аксіома R.2: Відношення Rs – транзитивне: □x□y□z(Rs(x, y)&Rs(y, z)→Rs(x, z)). Аксіома R.3: Відношення Rs – несиметричне: □x□y(Rs(x, y)→□Rs(y, x)) . Розглядаючи ці властивості як аксіоми в численні предикатів, покажемо, що вони є загальнозначущими формулами в мові α (або тавтологіями). Для аксіоми R.1: □x(□Rs(x, x)). Скористаємося геометричною інтерпретацією:

	Марченко переписав чужий текст зі словами «Скористаємося геометричною інтерпретацією», але сам рисунок не перемалював. Недолугий плагіат.	
	С. 69.	**С. 59–60.**
	Для самого себе об’єкт x не забезпечує істинності Rs. Тому для всіх x аксіома Rs(1) є тавтологією. Для аксіоми R.2 розглянемо наступні варіанти структурних відносин: Варіант 1. Об’єкти x , y і z незалежні один від одного. Формулу (Rs(x, y)&Rs(y,z)→Rs(x,z)) розкриваємо як диз’юнкцію відповідно до правил числення предикатів, тоді отримуємо: (□(Rs(x, y)&Rs(y,z))v Rs(x,z)). При даних відносинах об’єктів маємо істинносні оцінки Rs(x, y)=f , Rs (y,z) = f , Rs(x,z)=f отже □(f &f)vf =□(f)vf = t vf = t . **Марченко переписав чужий текст, але рисунок не перемалював. Недолугий плагіат.**	Для самого себе об’єкт x не забезпечує істинності Rs. Тому для всіх x аксіома Rs(1) є тавтологією. Для аксіоми R.2 розглянемо наступні варіанти структурних відносин: Варіант 1. Об’єкти x, y і z незалежні один від одного. Формулу (Rs(x, y)&Rs(y, z)→Rs(x, z)) розкриваємо як диз’юнкцію відповідно до правил числення предикатів, тоді отримуємо: (□(Rs(x, y)&Rs(y, z))v Rs(x, z)). При даних відносинах об’єктів маємо істинносні оцінки Rs(x, y) = f , Rs(y, z) = f , Rs(x, z)= f отже □(f &f)v f =□(f)v f = t v f = t . 
	С. 69.	**С. 60.**
	Варіант 2. Об’єкт y включає x, а z – незалежний. Тоді Rs(x, y) = t , Rs(y, z) = f , Rs(x,z) = f , і отже □(t&f)v f =□(f)vf = t v f = t . З точністю до перейменування змінних, аналогічні результати виходять для випадків включення одного об’єкта в іншій і незалежності третього. **Марченко переписав чужий текст, але рисунок не перемалював. Недолугий плагіат.**	Варіант 2. Об’єкт y включає x, а z – незалежний. Тоді Rs(x, y) = t , Rs(y, z) = f ,Rs(x, z)= f , і отже □(t&f)v f =□(f)v f = t v f = t . З точністю до перейменування змінних, аналогічні результати виходять для випадків включення одного об’єкта в іншій і незалежності третього. 
	С. 70.	**С. 60.**
	Варіант 3. Два об’єкти x і y входять в об’єкт z. В цьому випадку Rs(x, y) = f Rs(y,z)=t Rs(x,z)=t □(f &t)v t = □(f)v t = tv t = t . З точністю до перейменування змінних, отримаємо докази істинності формул, аналогічно варіанту 2. **Марченко переписав чужий текст, але рисунок не перемалював. Недолугий плагіат.**	Варіант 3. Два об’єкти x і y входять в об’єкт z. В цьому випадку Rs(x, y) = f Rs(y, z) = t Rs(x, z) = t –(f &t)v t =–(f)v t = t v t = t . З точністю до перейменування змінних, отримаємо докази істинності формул, аналогічно варіанту 2. 
	С. 70.	**С. 60.**
	Варіант 4. Об’єкт x входить в об’єкт y , а y входить в z. Маємо Rs(x, y) = t ,Rs(y,z) = t ,Rs(x,z) = t , тоді □(t&t)v t =□(t)v t = f v t = t . Таким чином, при всіх інтерпретаціях x , y , z дана пропозиція є тавтологією, що і потрібно було довести. **Марченко переписав чужий текст, але рисунок не перемалював.**	Варіант 4. Об’єкт x входить в об’єкт y, а y входить в z. Маємо Rs(x, y) = t , Rs(y, z) = t ,Rs(x, z) = t , тоді □(t&t)v t =□ (t)v t = f v t = t . Таким чином, при всіх інтерпретаціях x, y, z дана пропозиція є тавтологією, що і потрібно було довести. 

	Недолугий плагіат.	
	С. 70.	**С. 61.**
	Для аксіоми R.3: □x□y(Rs(x, y))→□Rs(x, y) . Розглядаємо варіанти структурних відносин двох об'єктів x і y . Варіант 1. Об'єкти x і y структурно незалежні. Тоді Rs(x, y) = f , Rs(y,x)= f , і □Rs(x, y)v Rs(y,x) = □f v□f= t v t = t **Марченко переписав чужий текст, але рисунок не перемалював.** **Недолугий плагіат.**	Для аксіоми R.3: □x□y(Rs(x, y))→□Rs(x, y) . Розглядаємо варіанти структурних відносин двох об'єктів x і y. Варіант 1. Об'єкти x і y структурно незалежні. Тоді Rs(x, y) = f , Rs(y, x)= f , і □Rs(x, y)v Rs(y,x) = □f v□f = t v t = t 
	С. 70.	**С. 61.**
	Варіант 2. Один об'єкт включає інший, наприклад x включає y, тоді Rs(x, y) = f , Rs(y,x) = t і □Rs(x, y)v Rs(y,x) = □f v□t = tv f = t . **Марченко переписав чужий текст, але рисунок не перемалював.** **Недолугий плагіат.**	Варіант 2. Один об'єкт включає інший, наприклад x включає y, тоді Rs(x, y)= f , Rs(y, x) = t і □Rs(x, y)v Rs(y, x) = □f v □t = t v f = t . 
	С. 70.	**С. 61.**
	Варіант, коли y включає x, розглядається аналогічно, шляхом перейменування об'єктів. В результаті доведено значимість аксіом R.1 – R.3 в мові α. Як засіб отримання кількісної оцінки структурних відносин об'єктів можна ввести функцію структурної потужності об'єкта, позначивши її функціональним символом Pw. Для цієї функції пропонується наступна система аксіом: Аксіома Pw.1: Pw(x) = 1, якщо x – примітивний ЛО; Аксіома Pw.2: Pw(x) = Pw(x1)+ Pw(x2), якщо x = <NO,{A},(x1, x2),{F}> або, в загальному випадку, x □{<NO> {A}, (x1, x2), {F}>}, де {A} і {F} можуть бути і порожніми множинами. Знак + розуміється тут як символ додавання натуральних чисел. На підставі аксіом Pw.1 І Pw.2 покажемо, що функція Pw є адитивною, тобто справедлива теорема 2.1:	Варіант, коли y включає x, розглядається аналогічно, шляхом перейменування об'єктів. В результаті доведено значимість аксіом R.1 – R.3 в мові α. Як засіб отримання кількісної оцінки структурних відносин об'єктів можна ввести функцію структурної потужності об'єкта, позначивши її функціональним символом Pw. Для цієї функції пропонується наступна система аксіом: Аксіома Pw.1: Pw(x) = 1, якщо x – примітивний ІО; Аксіома Pw.2: Pw(x) = Pw(x1)+ Pw(x2), якщо x = <NO,{A},(x1, x2),{F}> або, в загальному випадку, x □{<NO> {A}, (x1, x2), {F}>}, де {A} і {F} можуть бути і порожніми множинами. Знак + розуміється тут як символ додавання натуральних чисел. На підставі аксіом Pw.1 І Pw.2 покажемо, що функція Pw є адитивною, тобто справедлива теорема 2.1:
	С. 71.	**С. 62.**
	$Pw(x)=\sum_{i=1}^{n} Pw(x_i)$ де x Є{<NO>, {A}, (x1, x2,..., xn), {F}>}. Для доказу скористаємося методом індукції. Для i = 1 маємо: Pw(x) = Pw(x1), у випадку, якщо x1 примітивний ЛО, то Pw(x1) = 1, і отже, Pw(x) = 1. Якщо ж x1 не примітивний ЛО, то повертаємося до початкової посилки. Для i = 2 теорема 2.1 справедлива внаслідок аксіоми Pw(2). Покажемо, що якщо теорема вірна при i = n, то вона вірна і при i = n + 1. В цьому випадку Pw(x) = Pw(<NO>, {A}, (x1, x2, ..., xn, xn+1), {F}>).	$Pw(x)=\sum_{i=1}^{n} Pw(x_i)$ де x Є{<N_O>, {A}, (x1, x2,..., xn), {F}>}. Для доказу скористаємося методом індукції. Для i = 1 маємо: Pw(x) = Pw(x1), у випадку, якщо x1 примітивний ІО, то Pw(x1) = 1, і отже, Pw(x) = 1. Якщо ж x1 не примітивний ІО, то повертаємося до початкової посилки. Для i = 2 теорема 2.1 справедлива внаслідок аксіоми Pw(2). Покажемо, що якщо теорема вірна при i = n, то вона вірна і при i = n + 1. В цьому випадку Pw(x) = Pw(<NO>, {A}, (x1, x2, ..., xn, xn+1), {F}>).
	С. 71.	**С. 62.**
	Оскільки дія функції Pw не зачіпає такі складові об'єкта, як NO, { A } , {F}, то можемо ввести новий	Оскільки дія функції Pw не зачіпає такі складові об'єкта, як NO, {A}, {F}, то можемо ввести новий

	об’єкт x*n, що включає x n і x n + 1 : $x^*_n = \langle No, \{A\}, (x_n, x_{n+1}), \{F\}\rangle$. Тоді $Pw(x) = Pw(\langle No, \{A\}, (x_1, x_2, \ldots, x^*_n), \{F\}\rangle) =$ $= \sum_{i=1}^{n} Pw(x_i) = Pw(x_1) + Pw(x_2) + \ldots + Pw(x^*_n) =$ $= Pw(x_1) + Pw(x_2) + \ldots + Pw(\langle No, \{A\}, (x_n, x_{n+1}), \{F\}\rangle) =$ $= Pw(x_1) + Pw(x_2) + \ldots + Pw(x_n) + Pw(x_{n+1})$ по аксіомі Pw.2. Отже, теорема 2.1 вірна і для довільного натурального числа n. Далі покажемо, що справедлива теорема 2.2: Функція Pw(x) повертає кількість примітивних об’єктів, що містяться в логічному об’єкті X.	об’єкт x*n, що включає x n і x n +1 : $x^*_n = \langle No, \{A\}, (x_n, x_{n+1}), \{F\}\rangle$. Тоді $Pw(x) = Pw(\langle No, \{A\}, (x_1, x_2, \ldots, x^*_n), \{F\}\rangle) =$ $= \sum_{i=1}^{n} Pw(x_i) = Pw(x_1) + Pw(x_2) + \ldots + Pw(x^*_n) =$ $= Pw(x_1) + Pw(x_2) + \ldots + Pw(\langle No, \{A\}, (x_n, x_{n+1}), \{F\}\rangle) =$ $= Pw(x_1) + Pw(x_2) + \ldots + Pw(x_n) + Pw(x_{n+1})$ по аксіомі Pw.2. Отже, теорема 2.1 вірна і для довільного натурального числа n. Далі покажемо, що справедлива теорема 2.2: Функція Pw(x) повертає кількість примітивних об’єктів, що містяться в інформаційному об’єкті X.
	С. 71–72.	**С. 63.**
	Скористаємося індукцією по глибині індекса вкладення об’єктів l. Нехай $l = 1$, тоді $\sum_{i=1}^{n} Pw(x_i) = n$, якщо всі x_i що входять в X є примітивними ЛО. Покажемо, що якщо теорема справедлива при глибині індекса вкладення k, то вона справедлива і при k+1. При глибині вкладення, яка дорівнює k маємо: $\sum_{i=1}^{n} Pw_k(x) = \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} Pw(xl_1l_2 \ldots l_k) = \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} l_k$, коли всі $xl_1l_2 \ldots l_k$ є примітивними ЛО і $l_k = Pw(xl_1l_2 \ldots l_k)$. При глибині вкладення k+1 маємо: $Pw_{k+1}(x) = \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} \sum_{l_{k+1}=1}^{nl_{k+1}} Pw(xl_1l_2 \ldots l_kl_{k+1})$	Скористаємося індукцією по глибині індекса вкладення об’єктів l. Нехай $l = 1$, тоді $\sum_{i=1}^{n} Pw(x_i) = n$ якщо всі x_i що входять в X є примітивними ІО. Покажемо, що якщо теорема справедлива при глибині індекса вкладення k, то вона справедлива і при k+1. При глибині вкладення, яка дорівнює k маємо: $\sum_{i=1}^{n} Pw_k(x) = \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} Pw(xl_1l_2 \ldots l_k) = \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} l_k$, коли всі $xl_1l_2 \ldots l_k$ є примітивними ІО і $l_k = Pw(xl_1l_2 \ldots l_k)$. При глибині вкладення k+1 маємо: $Pw_{k+1}(x) = \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} \sum_{l_{k+1}=1}^{nl_{k+1}} Pw(xl_1l_2 \ldots l_kl_{k+1})$.
	С. 72.	**С. 63.**
	Відповідно до теореми 2.1 можемо цю суму розкрити як $Pw_{k+1}(x) = \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} \left(Pw(xl_1l_2 \ldots l_{k-1}l_{k1}) + Pw(xl_1l_2 \ldots l_{k-1}l_{k2}) + \ldots + Pw(xl_1l_2 \ldots l_{k-1}l_{kl_{k+1}}) \right)$	Відповідно до теореми 2.1 можемо цю суму розкрити як $Pw_{k+1}(x) = \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} \left(Pw(xl_1l_2 \ldots l_{k-1}l_{k1}) + Pw(xl_1l_2 \ldots l_{k-1}l_{k2}) + \ldots + Pw(xl_1l_2 \ldots l_{k-1}l_{kl_{k+1}}) \right)$
	С. 72.	**С. 63.**
	Розкриваючи дужки, отримаємо: $Pw_{k+1}(x) = \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} Pw(xl_1l_2 \ldots l_{k-1}l_{k1}) +$ $+ \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} Pw(xl_1l_2 \ldots l_{k-1}l_{k2}) + \ldots +$ $+ \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} Pw(xl_1l_2 \ldots l_{k-1}l_kl_{k+1})$	Розкриваючи дужки, отримаємо: $Pw_{k+1}(x) = \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} Pw(xl_1l_2 \ldots l_{k-1}l_{k1}) +$ $+ \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} Pw(xl_1l_2 \ldots l_{k-1}l_{k2}) + \ldots +$ $+ \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} Pw(xl_1l_2 \ldots l_{k-1}l_kl_{k+1})$
	С. 72–73.	**С. 64.**
	Для кожного з цих доданків, виходячи із заданої передумови теореми, отримуємо: $Pw_{k+1}(x) = \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} l_{k1} + \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} l_{k2} + \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} l_{kl_{k+1}} =$ $\sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} \sum_{l_{k+1}=1}^{nl_{k+1}} l_{k+1}$ що і потрібно було довести.	Для кожного з цих доданків, виходячи із заданої передумови теореми, отримуємо: $Pw_{k+1}(x) = \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} l_{k1} + \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} l_{k2} + \sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} l_{kl_{k+1}} =$ $\sum_{l_1=1}^{nl_1} \sum_{l_2=1}^{nl_2} \ldots \sum_{l_k=1}^{nl_k} \sum_{l_{k+1}=1}^{nl_{k+1}} l_{k+1}$ що і потрібно було довести.
	С. 73.	**С. 64.**
	Для забезпечення замкнутості числення об’єктів на множині М необхідно поставити умову, щоб ре-	Для забезпечення замкнутості числення об’єктів на множині М необхідно поставити умову, щоб ре-

	зультат дії функціонального символу Pw також був об’єктом. Тому замість безпосереднього використання натурального числа вважатимемо, що Pw(x) = <NO, A> –однопараметричний об’єкт, де A = < NA, {N}, a>, де {N} – множина натуральних чисел; a Є {N}. У свою чергу, оскільки мова α допускає побудову рекурсивних термів, формально припустимим є терм Pw(Pw(x)). Тому Pw(Pw(x))= Pw(< NO, A >) покладемо рівним x0, де x0 = <ø , ø, ø, ø >, тобто порожній об’єкт [14, 22, 27]. Порожній об’єкт x0 необхідно замкнути сам на себе і вважати, що Pw(x0) = x0, тоді рекурсія виду Pw(Pw... Pw(x)...) завжди дасть x0. З міркувань суворості доведеться доповнити систему аксіом для відношення Rs: Аксіома R.1.1: □x(□Rs(x0,x)) □x(□Rs(x0,x)) ; Аксіома R.1.2: □x(□Rs(x,x0)) □x(□Rs(x,x0)) . Обґрунтування того, що аксіоми R.1.1 і R.1.2 є тавтологіями, абсолютно аналогічно обґрунтуванню аксіоми R.1. **До чужого переписаного тексту Марченко додав три покликання. Ці покликання фальшиві: 2018-го, 2014-го та 2016-го років.** **Справжнє джерело – це дисертація Слюняєва, яка була захищена 2010 року.** **Плагіат.**	зультат дії функціонального символу Pw також був об’єктом. Тому замість безпосереднього використання натурального числа вважатимемо, що Pw(x) = <NO, A> – однопараметричний об’єкт, де A = < NA, {N}, a>, де {N} – множина натуральних чисел; a Є {N}. У свою чергу, оскільки мова α допускає побудову рекурсивних термів, формально припустимим є терм Pw(Pw(x)). Тому Pw(Pw(x))= Pw(< NO, A >) покладемо рівним x0, де x0 = <ø, ø, ø, ø >, тобто порожній об’єкт. Порожній об’єкт x0 необхідно замкнути сам на себе і вважати, що Pw(x0) = x0, тоді рекурсія виду Pw(Pw... Pw(x)...) завжди дасть x0. З міркувань суворості доведеться доповнити систему аксіом для відношення Rs: Аксіома R.1.1: □x(□Rs(x0 , x)) □x(□Rs(x0, x)); Аксіома R.1.2: □x(□Rs(x, x0)) □x (□Rs x, x0)). Обгрунтування того, що аксіоми R.1.1 і R.1.2 є тавтологіями, абсолютно аналогічно обгрунтуванню аксіоми R.1.
	С. 73–74.	**С. 65.**
	Використання числення предикатів з рівністю примушує нас ввести поняття рівності об’єктів. Насправді, трактувати його необхідно як еквівалентність об’єктів. Вважатимемо, що два об’єкти рівні (еквівалентні один одному), якщо $x = <NA_x, \{A\}_x, \{O\}_x, \{F\}_x>$; $y = <NA_y, \{A\}_y, \{O\}_y, \{F\}_y>$; $\{A\}_x = \{A\}_y, \{O\}_x = \{O\}_y, \{F\}_x = \{F\}_y$. Імена об’єктів при цьому можуть бути різними.	Використання числення предикатів з рівністю примушує нас ввести поняття рівності об’єктів. Насправді, трактувати його необхідно як еквівалентність об’єктів. Вважатимемо, що два об’єкти рівні (еквівалентні один одному), якщо $x = <NA_x, \{A\}_x, \{O\}_x, \{F\}_x>$; $y = <NA_y, \{A\}_y, \{O\}_y, \{F\}_y>$; $\{A\}_x = \{A\}_y, \{O\}_x = \{O\}_y, \{F\}_x = \{F\}_y$. Імена об’єктів при цьому можуть бути різними.
	С. 74.	**С. 65.**
	Очевидно, що далі потрібно визначити відношення рівності для множин атрибутів, вкладених об’єктів і моделей функціонування. Дві множини атрибутів {A}x і {A}y вважатимемо рівними (еквівалентними), якщо для всіх AigЄg{A}x і AjgЄg{A}y спостерігається попарна рівність при однаковому значенні індекса i, тобто Aix = Aiy, де i = 1, 2 ... n, при цьому n – потужність множин {A}x і {A}y. Два атрибути Aφ і Aψ назвемо рівними (еквівалентними), якщо NAφ = NAψ, SAφ = SAψ, VAφ = VAψ. Рівність списків вкладених об’єктів {O}x і {O}y визначається аналогічно рівності множин атрибутів, тобто {O}x = {O}y, якщо потужності множин {O}x і {O}y рівні і кожен Oi x = Oiy. Очевидно, що це визначення рекурсивне, оскільки об’єкти Oi x і Oiy можуть мати свої вкладені об’єкти.	Очевидно, що далі потрібно визначити відношення рівності для множин атрибутів, вкладених об’єктів і моделей функціонування. Дві множини атрибутів {A}x і {A}y вважатимемо рівними (еквівалентними), якщо для всіх Ai Є {A}x і Aj Є {A}y спостерігається попарна рівність при однаковому значенні індекса i, тобто Aix = Ai y, де i = 1, 2 ... n, при цьому n – потужність множин {A}x і {A}y. Два атрибути Aφ і Aψ назвемо рівними (еквівалентними), якщо NAφ = NAψ, SAφ = SAψ, VAφ = VAψ. Рівність списків вкладених об’єктів {O}x і {O}y визначається аналогічно рівності множин атрибутів, тобто {O}x = {O}y, якщо потужності множин {O}x і {O}y рівні і кожен Oi x = Oiy. Очевидно, що це визначення рекурсивне, оскільки об’єкти Oi x і Oiy можуть мати свої вкладені об’єкти.
	С. 74.	**С. 65–66.**
	Необхідно також ввести поняття потужності об’єкта Pp, тобто функції, що показує кількість вкладених об’єктів. Введемо наступні аксіоми: Аксіома Pp.1: Pp(x)= ø , якщо x – примітивний ЛО; Аксіома Pp.2: Pp(x)= k, якщо x = <NO, {A}, (x1, x2,..., xk), {F}>, де {A} і {F} можуть бути і порожніми	Необхідно також ввести поняття потужності об’єкта Pp, тобто функції, що показує кількість вкладених об’єктів. Введемо наступні аксіоми: Аксіома Pp.1: Pp(x)= ø , якщо x – примітивний IO; Аксіома Pp.2: Pp(x)= k, якщо x = <NO, {A}, (x1, x2,..., xk), {F}>, де {A} і {F} можуть бути і порожніми

<table>
<tr><td></td><td>множинами.
Можна ввести Pp(x) і конструктивно, тобто використовувати продукційні правила для обчислення k, оскільки це число в явному вигляді може бути і невідоме. За аналогією з функцією Pw вважатимемо, що Pp(x) = < NO, A > – однопараметричний об’єкт, де A – < NA, {N}, a >, де a Є {N}. Pp(x0) покладемо рівним x0.</td><td>множинами.
Можна ввести Pp(x) і конструктивно, тобто використовувати продукційні правила для обчислення k, оскільки це число в явному вигляді може бути і невідоме. За аналогією з функцією Pw вважатимемо, що Pp(x) = < NO, A > – однопараметричний об’єкт, де A – < NA, {N}, a >, де a Є {N}. Pp(x0) покладемо рівним x0.</td></tr>
<tr><td></td><td>С. 75.</td><td>С. 66.</td></tr>
<tr><td></td><td>Розширення алгебраїчної системи Ф можна здійснити шляхом введення нових відношень над ЛО (відповідно нових предикативних символів в мову α), що дозволяє заздалегідь не обмежувати спільність системи Ф.</td><td>Розширення алгебраїчної системи Ф можна здійснити шляхом введення нових відношень над ІО (відповідно нових предикативних символів в мову α), що дозволяє заздалегідь не обмежувати спільність системи Ф. При цьому пропонується наступна схема введення нових відношень:<table><tr><th>Бінарне відношення</th><th>Символ відношення</th><th>Аксіоми відношення</th></tr><tr><td>< ім’я відношення ></td><td>< знак, який позначає відношення у формальній системі ></td><td>< список аксіом ></td></tr></table></td></tr>
<tr><td></td><td>С. 75.</td><td>С. 66.</td></tr>
<tr><td></td><td>Оскільки найбільш поширені відношення мають типові властивості, то можна запропонувати сукупність аксіом, визначених апріорно і відомих формальній системі, щоб згодом задавати їх просто посиланнями. Запишемо їх як
Аксіома А.1: □x□y(xRy→yRx);
Аксіома А.2: □x□y(xRy→□(yRx));
Аксіома А.3: □x(xRx);
Аксіома А.4: □x□(xRx);
Аксіома А.5: □x□y□z((xRy)&(yRz)→xRz).
Тут аксіома А.1 виражає властивість симетричності, аксіома А.2 – асиметричності, аксіома А.3 – рефлексивності, аксіома А.4 – антирефлексивності і аксіома А.5 – транзитивності.
Задаючи таким чином нові відношення (предикативні символи), ми можемо будувати різні, довільні до деякої міри аксіоматизовані системи Ф в мові числення предикатів.</td><td>Оскільки найбільш поширені відношення мають типові властивості, то можна запропонувати сукупність аксіом, визначених апріорно і відомих формальній системі, щоб згодом задавати їх просто посиланнями. Запишемо їх як
Аксіома А.1: □x□y(xRy →yRx);
Аксіома А.2: □x□y(xRy →□(yRx));
Аксіома А.3: □x(xRx);
Аксіома А.4: □x□(xRx);
Аксіома А.5: □x□y□z((xRy)&(yRz)→xRz).
Тут аксіома А.1 виражає властивість симетричності, аксіома А.2 – асиметричності, аксіома А.3 – рефлексивності, аксіома А.4 – антирефлексивності і аксіома А.5 – транзитивності.
Задаючи таким чином нові відношення (предикативні символи), ми можемо будувати різні, довільні до деякої міри аксіоматизовані системи Ф в мові числення предикатів.</td></tr>
<tr><td></td><td>С. 75–76.</td><td>С. 67.</td></tr>
<tr><td></td><td>Відношення рівності (еквівалентності) об’єктів, очевидно, вимагає детальнішого розгляду внаслідок складної структури ЛО. Введемо наступні визначення. Два логічні об’єкти x і y назвемо одновидовими, якщо NOx ≠ NOy, { A } x = { A } y , {O}x = {O}y, {F}x = {F}y, тобто вони еквівалентні з точністю до власних імен об’єктів. Відношення одновидової еквівалентності є симетричним, антирефлексивним і транзитивним (задовольняє аксіомам А.1, А.4, А.5).
Введемо еквівалентність з точністю до імен атрибутів.
Два ЛО x і y називаються об’єктами одного класу, якщо NOx ≠ NOy, {O}x = {O}y, {F}x = {F}y і для всіх і NAxi = NAyi, SAxi = SAyi, де i=1,2...,k, де k – число атрибутів в множині {A}x і {A}y (потужність {A}x = {A}y. Відношення класової еквівалентності також є симетричним, антирефлексивним і транзитивним.</td><td>Відношення рівності (еквівалентності) об’єктів, очевидно, вимагає детальнішого розгляду внаслідок складної структури ІО. Введемо наступні визначення. Два інформаційні об’єкти x і y назвемо одновидовими, якщо $N_{Ox} \neq N_{Oy}$, {A}x = {A}y , {O} x = {O}y, {F}x = {F}y, тобто вони еквівалентні з точністю до власних імен об’єктів. Відношення одновидової еквівалентності є симетричним, антирефлексивним і транзитивним (задовольняє аксіомам А.1, А.4, А.5).
Введемо еквівалентність з точністю до імен атрибутів.
Два ІО x і y називаються об’єктами одного класу, якщо NOx ≠ NOy, {O} x = {O}y, {F}x = {F}y і для всіх і NAxi = NAyi, SAxi = SAyi, де i=1,2...,k, де k – число атрибутів в множині {A}x і {A}y (потужність {A}x = {A}y. Відношення класової еквівалентності також є симетричним, антирефлексивним і транзитивним.</td></tr>
<tr><td></td><td>С. 76.</td><td>С. 67.</td></tr>
<tr><td></td><td>Якщо два об’єкти знаходяться у відношенні одновидової еквівалентності, то вони обов’язково знаходяться і у відношенні класової еквівалентності, тобто справедлива наступна теорема.
Теорема 2.3. $\forall x \forall y \forall z\left(\left(xR_0y\right)\to\left(xR_ky\right)\right)$.
Дана теорема безпосередньо виходить з визна-</td><td>Якщо два об’єкти знаходяться у відношенні одновидової еквівалентності, то вони обов’язково знаходяться і у відношенні класової еквівалентності, тобто справедлива наступна теорема.
Теорема 2.3. $\forall x \forall y \forall z((xR_0y)\to(xR_ky))$
Дана теорема безпосередньо виходить з визна-</td></tr>
</table>

	чень одновидової та класової еквівалентності, оскільки одновидова еквівалентність є окремим випадком класової еквівалентності при VAx і = VAyi. Зворотнє твердження, природньо, є невірним з тих же причин. Тут через R0 позначаємо одновидову еквівалентність, а через Rk – класову еквівалентність.	чень одновидової та класової еквівалентності, оскільки одновидова еквівалентність є окремим випадком класової еквівалентності при VAx і = VAyi. Зворотнє твердження, природньо, є невірним з тих же причин. Тут через R0 позначаємо одновидову еквівалентність, а через Rk – класову еквівалентність.
	С. 76.	**С. 67–68.**
	Два ЛО x і y називаються структурно еквівалентними, якщо NOx ≠ NOy, {O} x = {O}y, {F}x = {F}y, { A } x □ { A } y , де знак □ позначає структурну еквівалентність. СЕ для {A} означає, що потужності множин {A}x і {A}y однакові і SAxi = SAyi при i = 1,2,...,k, де k – потужність {A}x і {A}y відповідно. Таким чином, співпадають тільки визначення атрибутів, а імена і значення їх можуть бути різні. Для множин {O}x і {O}y структурна еквівалентність означає, що первинні потужності Pp(x) і Pp(y) рівні і Oxi = Oyi, де i = 1,2...,m, де m = Pp(x) = Pp(y). Відношення структурної еквівалентності є симетричним, антирефлексивним і транзитивним.	Два ІО x і y називаються структурно еквівалентними, якщо NOx ≠ NOy, {O} x = {O}y, {F}x = {F}y, { A } x □ { A } y , де знак □ позначає структурну еквівалентність. СЕ для {A} означає, що потужності множин {A}x і {A}y однакові і SAxi = SAyi при i = 1,2,...,k, де k – потужність {A}x і {A}y відповідно. Таким чином, співпадають тільки визначення атрибутів, а імена і значення їх можуть бути різні. Для множин {O}x і {O}y структурна еквівалентність означає, що первинні потужності Pp(x) і Pp(y) рівні і Oxi = Oyi, де i = 1,2...,m, де m = Pp(x) = Pp(y). Відношення структурної еквівалентності є симетричним, антирефлексивним і транзитивним.
	С. 76–77.	**С. 68.**
	Функція визначення атрибута. Оскільки при аналізі процесів функціонування буде потрібно порівняння різних характеристик ЛО, введемо функцію визначення атрибута I(p, q), де p – ім’я ЛО, q – ім’я атрибута. $I(p,q)=\begin{cases} x_0, \text{ якщо } \forall N_A(N_A \in p)\wedge(N_A \neq q) \\ O=<p\_N_A, Aq> - \text{у протилежному випадку} \end{cases}$	Функція визначення атрибута. Оскільки при аналізі процесів функціонування буде потрібно порівняння різних характеристик ІО, введемо функцію визначення атрибута I(p, q), де p – ім’я ІО, q – ім’я атрибута. $I(p,q)=\begin{cases} x_0, \text{ якщо } \forall N_A(N_A \in p)\wedge(N_A \neq q) \\ O=<p\_N_A, Aq> - \text{у протилежному випадку} \end{cases}$
	С. 77.	**С. 68.**
	Це означає, що у разі відсутності в множині {A} атрибута з ім’ям q, функція приймає значення порожнього логічного об’єкта x0. Виходить, що функція I(p, q) приймає значення на множині x0 ∪ {Oop}, де {Oop}* – множина усіх можливих однопараметричних ЛО [1-14, 18-22, 51]. Необхідно зазначити апріорно, що взяття функції I від однопараметричного об’єкта дає той же самий однопараметричний об’єкт: I(p,q)=p, якщо p – однопараметричний ЛО. Це дозволяє надалі ввести необхідні математичні операції над однопараметричними ЛО (ОПЛО). Почнемо цей розгляд з операції арифметичного підсумовування. Нехай Σ(X,Y) = Z, де X = <p, A>, Y= <q, B> є ОПЛО з іменами p і q, а також атрибутами A і B, відповідно, при цьому A = <NA, SA, VA>, B = <NB, SB, VB>. Введемо наступні аксіоми: Аксіома В.1: $\forall x((x=x_0)\rightarrow(\Sigma(x,y)=x_0))$; Аксіома В.2: $\forall x((x=x_0)\rightarrow(\Sigma(y,x)=x_0))$. **Марченко у переписаному чужому тексті замінив слова «інформаційного об’єкта» на «логічного об’єкта», «ІО» на «ЛО» та «ОПІО» на «ОПЛО».** **До чужого переписаного тексту Марченко додав 19 фальшивих покликань.** **Справжнє джерело – це дисертація Слюняєва.** **Нахабний плагіат.**	Це означає, що у разі відсутності в множині {A} атрибута з ім’ям q, функція приймає значення порожнього інформаційного об’єкта x0. Виходить, що функція I(p, q) приймає значення на множині x0 ∪ {Oop}, де {Oop}* – множина усіх можливих однопараметричних ІО. Необхідно зазначити апріорно, що взяття функції I від однопараметричного об’єкта дає той же самий однопараметричний об’єкт: I(p,q)=p, якщо p – однопараметричний ІО. Це дозволяє надалі ввести необхідні математичні операції над однопараметричними ІО (ОПІО). Почнемо цей розгляд з операції арифметичного підсумовування. Нехай Σ(X,Y) = Z, де X = <p, A>, Y= <q, B> є ОПІО з іменами p і q, а також атрибутами A і B, відповідно, при цьому A = <NA, SA, VA>, B = <NB, SB, VB>. Введемо наступні аксіоми: Аксіома В.1: $\forall x((x=x_0)\rightarrow(\sum(x,y)=x_0))$; Аксіома В.2: $\forall x((x=x_0)\rightarrow(\sum(y,x)=x_0))$.
	С. 77.	**С. 69.**
	В результаті логічного виводу отримаємо, що Σ(x0, x0) = x0. Об’єкт x0 служить ознакою виник-	В результаті логічного виводу отримаємо, що Σ(x0, x0) = x0. Об’єкт x0 служить ознакою виник-

	нення помилки. Аксіома B.3: $\forall x\forall y(\neg(S_A = S_B)\rightarrow(\Sigma(X,Y)=x_0))$; Аксіома B.4: $\forall x\forall y\forall z((S_A = S_B)\&(V_C \in S_A)\rightarrow(\Sigma(X,Y)=Z))$; де $Z = \langle p\text{-}q,\ \Sigma A, B\rangle$, де $\Sigma A,\ B{=}C{=}\langle N_C, S_C, V_C\rangle$, $N_C = N_A - N_B$, $S_C = S_A = S_B$, $V_C = \Sigma V_A, V_B$; Аксіома B.5: $\forall x\forall y\forall z((S_A = S_B)\&\neg(V_C \in S_A)\rightarrow(\Sigma(X,Y)=x_0))$. Це випадок, коли VC лежить за межами множини SA	нення помилки. Аксіома B.3: $\forall x\forall y(\neg(S_A = S_B)\rightarrow(\Sigma(X,Y)=x_0))$; Аксіома B.4: $\forall x\forall y\forall z((S_A = S_B)\&(V_C \in S_A)\rightarrow(\Sigma(X,Y)=Z))$; де $Z = \langle p\text{-}q,\ \Sigma A, B\rangle$, де $\Sigma A,\ B{=}C{=}\langle N_C, S_C, V_C\rangle$, $N_C = N_A - N_B$, $S_C = S_A = S_B$, $V_C = \Sigma V_A, V_B$; Аксіома B.5: $\forall x\forall y\forall z((S_A = S_B)\&\neg(V_C \in S_A)\rightarrow(\Sigma(X,Y)=x_0))$. Це випадок, коли VC лежить за межами множини SA.
	С. 78.	**С. 69.**
	VC = ΣVA, VB розуміється як арифметична сума, яка може бути побудована конструктивно для відповідних видів чисел. Операцію можна зробити і n-арною, використовуючи рекурсивне визначення: $\sum_{i=1}^{n} x_i = \Sigma\left(\sum_{i=1}^{n-1} x_i\right), x_n$ Легко відмітити, що для введеного таким чином додавання ОПІО не виконується комутативний закон: Σ(X,Y) ≠ Σ(Y,X) , оскільки результатом лівої суми є Z = <p –q,ΣA,B>, а результатом правої суми Z′ = <q – p , ΣB,A>. Та зате для додавання ОПІО виконується асоціативний закон $\Sigma(X,Y,Z)=\Sigma(X,\Sigma(Y,Z))=\Sigma(\Sigma(X,Y),Z)$. З рекурсивного визначення суми отримуємо: $\Sigma(X,Y,Z)= \Sigma(X,\Sigma(Y,Z))$. Покажемо, що це в точності дорівнює $\Sigma(\Sigma(X,Y),Z)$ Нехай $X=\langle p, A\rangle$, $Y=\langle q, B\rangle$, $Z=\langle 1, C\rangle$, тоді $\Sigma(X, \Sigma(Y, Z))$ набуває вигляду $\Sigma(X, \Sigma(Y, Z)) = \Sigma(X, \langle q-1, \Sigma B, C\rangle) = \Sigma(X, \langle q-1, \langle N_B - N_C, S_B, \Sigma V_B, V_C\rangle\rangle)$. У свою чергу $\Sigma(\Sigma(X,Y),Z) = \Sigma\langle p-q, \Sigma A, B\rangle,\ Z=\Sigma\langle p\text{-}q, N_A-N_B, S_A, \Sigma V_A, V_B\rangle\rangle$, $Z = \langle p\text{-}q\text{-}1, N_A-N_B-N_C, S_A, \Sigma(\Sigma V_A, V_B), V_C\rangle$	VC = ΣVA, VB розуміється як арифметична сума, яка може бути побудована конструктивно для відповідних видів чисел. Операцію можна зробити і n-арною, використовуючи рекурсивне визначення: $\sum_{i=1}^{n} x_i = \Sigma\left(\sum_{i=1}^{n-1} x_i\right), x_n$. Легко відмітити, що для введеного таким чином додавання ОПІО не виконується комутативний закон: Σ(X,Y) ≠ Σ(Y,X) , оскільки результатом лівої суми є Z =<p –q,ΣA,B>, а результатом правої суми Z′ =<q – p , ΣB,A>. Та зате для додавання ОПІО виконується асоціативний закон $\Sigma(X,Y,Z)=\Sigma(X,\Sigma(Y,Z))=\Sigma(\Sigma(X,Y),Z)$. З рекурсивного визначення суми отримуємо: $\Sigma(X,Y,Z)= \Sigma(X,\Sigma(Y,Z))$. Покажемо, що це в точності дорівнює $\Sigma(\Sigma(X,Y),Z)$ Нехай $X=\langle p, A\rangle$, $Y=\langle q, B\rangle$, $Z=\langle 1, C\rangle$, тоді $\Sigma(X, \Sigma(Y, Z))$ набуває вигляду $\Sigma(X, \Sigma(Y, Z)) = \Sigma(X, \langle q-1, \Sigma B, C\rangle) = \Sigma(X, \langle q-1, \langle N_B - N_C, S_B, \Sigma V_B, V_C\rangle\rangle)$. У свою чергу $\Sigma(\Sigma(X,Y),Z) = \Sigma\langle p-q, \Sigma A, B\rangle,\ Z=\Sigma\langle p\text{-}q, N_A-N_B, S_A, \Sigma V_A, V_B\rangle\rangle$, $Z = \langle p\text{-}q\text{-}1, N_A-N_B-N_C, S_A, \Sigma(\Sigma V_A, V_B), V_C\rangle$
	С. 78.	**С. 70.**
	Оскільки VA, VB, VC – це числа, то для них асоціативний закон виконується і ΣVA,ΣVB,VC = Σ(ΣVA,VB), VC. Таким чином, доведена справедливість асоціативного закону для ОПІО. Покажемо справедливість закону нульової множини для складання ОПІО. Теорема 2.4. Для всіх ОПІО виконується сувора рівність (еквівалентність): $\Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_0\right) = x_0$ **Марченко пише «Таким чином, доведена», нібито це він робить якийсь висновок і щось там доводить, а насправді переписує чужу дисертацію.** **Плагіат.**	Оскільки VA, VB, VC – це числа, то для них асоціативний закон виконується і ΣVA,ΣVB,VC = Σ(ΣVA,VB), VC. Таким чином, доведена справедливість асоціативного закону для ОПІО. Покажемо справедливість закону нульової множини для складання ОПІО. Теорема 2.4. Для всіх ОПІО виконується сувора рівність (еквівалентність): $\Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_0\right) = x_0$
	С. 79.	**С. 70.**
	Для доказу скористаємося індукцією по довжині суми. При n = 2 маємо Σ(x),x0 = x0 згідно Аксіоми B.2. Припустимо, що наше твердження справедливе при довільному n, і покажемо, що з цього виходить його істинність при n + 1: $\Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_0\right) = \Sigma\left(\Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_{n+1}\right), x_0\right)$ Розглянемо два можливі випадки:	Для доказу скористаємося індукцією по довжині суми. При n = 2 маємо Σ(x),x0 = x0 згідно Аксіоми B.2. Припустимо, що наше твердження справедливе при довільному n, і покажемо, що з цього виходить його істинність при n + 1: $\Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_0\right) = \Sigma\left(\Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_{n+1}\right), x_0\right)$ Розглянемо два можливі випадки:

	а) сума $\sum_{i=1}^{n-1} x_i$ дає деякий ОПІО, який позначимо як M. Тоді отримуємо: $\Sigma(\Sigma(M, x_{n+1}), x_0) = \Sigma(M, \Sigma(x_{n+1}, x_0))$, $\Sigma\left(\Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_{n+1}\right), x_0\right) = \Sigma(\Sigma(M, x_{n+1}), x_0)$ згідно з асоціативним законом. За аксіомою В.2 отримуємо $\Sigma(x_{n+1}, x_0)$ і тоді $\Sigma(M, x_0) = \Sigma\left(\sum_{i=1}^{n} x_i\right), x_0 = x_0$ за посилкою індукції; б) сума $\sum_{i=1}^{n-1} x_i$ дає в результаті x_0. Тоді $\Sigma\left(\left(\sum_{i=1}^{n+1} x_i\right), x_0\right) = \Sigma\left(\Sigma\left(\left(\sum_{i=1}^{n} x_i\right), x_{n+1}\right), x_0\right) = \Sigma(\Sigma(x_0, x_{n+1}), x_0)$ За аксіомою В.1 отримуємо: $\Sigma(\Sigma(x_0, x_{n+1})), x_0 = \Sigma(x_0, x_0) = x_0$	а) сума $\sum_{i=1}^{n-1} x_i$ дає деякий ОПІО, який позначимо як M. Тоді отримуємо: $\sum(\sum(M, x_{n+1}), x_0) = \sum(M, \sum(x_{n+1}, x_0))$, $\sum\left(\sum\left(\left(\sum_{i=1}^{n-1} x_i\right), x_{n+1}\right), x_0\right) = \sum(\sum(M, x_{n+1}), x_0)$ згідно з асоціативним законом. За аксіомою В.2 отримуємо $\Sigma(x_{n+1}, x_0)$ і тоді $\sum(M, x_0) = \sum\left(\sum_{i=1}^{n} x_i\right), x_0 = x_0$ за посилкою індукції; б) сума $\sum_{i=1}^{n-1} x_i$ дає в результаті x_0. Тоді $\sum\left(\left(\sum_{i=1}^{n+1} x_i\right), x_0\right) = \sum\left(\sum\left(\left(\sum_{i=1}^{n} x_i\right), x_{n+1}\right), x_0\right) = \sum(\sum(x_0, x_{n+1}), x_0)$ За аксіомою В.1 отримуємо: $\sum(\sum(x_0, x_{n+1})), x_0 = \sum(x_0, x_0) = x_0$
	С. 79–80.	**С. 71.**
	Теорема 2.5. Для всіх ОПІО виконується сувора рівність (еквівалентність): ∑x1, …,xk, x0,x_{k+2}, …,xn) = x0 при довільних k і n, що є натуральними числами. Для доказу цього твердження перетворюватимемо n-арну суму відповідно до рекурсивного визначення: $\Sigma(x_1,\dots,x_k,x_0,x_{k+2},\dots,x_n) = \Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_n\right) = \Sigma\left(\Sigma\left(\sum_{i=1}^{n-2} x_i\right), x_{n-1}\right) =$ $= \Sigma\left(\Sigma\left(\dots\left(\sum_{i=1}^{k}(x_k, x_0), x_{k+2}\right)\dots\right), x_n\right)$	Теорема 2.5. Для всіх ОПІО виконується сувора рівність (еквівалентність): ∑(x1, …,xk ,x0,x_{k+2}, …,xn) = x0 при довільних k і n, що є натуральними числами. Для доказу цього твердження перетворюватимемо n-арну суму відповідно до рекурсивного визначення: $\sum(x_1,\dots,x_k,x_0,x_{k+2},\dots,x_n) = \sum\left(\left(\sum_{i=1}^{n-1} x_i\right), x_n\right) = \sum\left(\sum\left(\sum_{i=1}^{n-2} x_i\right), x_{n-1}\right) =$ $= \sum\left(\sum\left(\dots\left(\sum_{i=1}^{k}(x_k, x_0), x_{k+2}\right)\dots\right), x_n\right)$
	С. 80.	**С. 71.**
	Оскільки тут змінюються два індекси k і n, то необхідно використовувати подвійну індукцію. Оскільки k зобов'язаний знаходитися в межах 0 < k < n, то скористаємося зворотною індукцією по k: а) k = 1, n = 3: $\Sigma x_0, x_2, x_3 = \Sigma(\Sigma(x_0, x_2), x_3) = \Sigma x_0, x_3 = x_0$ Це витікає з асоціативного закону і Аксіоми В.1. Слід довести, що теорема справедлива при k = 1 і довільному n. Тому робимо наступний крок; б) k = 1, n = m, вважаючи, що твердження істинне при n = m, доведемо, що воно істинне і при n = m + 1: $\Sigma x_0, x_2, x_3, \dots, x_{m+1} = \Sigma(\Sigma(x_0, \dots, x_m), x_{m+1}) = \Sigma x_0, x_{m+1} = x_0$ Далі переходимо до зворотної індукції по k. в) перевіряємо справедливість теореми при k = n – 1: $\Sigma(x_1, \dots, x_0, x_n) = \Sigma(\Sigma(x_1, \dots, x_0), x_n) = \Sigma x_0, x_n = x_0$ відповідно до закону нульової множини.	Оскільки тут змінюються два індекси k і n, то необхідно використовувати подвійну індукцію. Оскільки k зобов'язаний знаходитися в межах 0 < k < n, то скористаємося зворотною індукцією по k: а) k = 1, n = 3: $\sum x_0, x_2, x_3 = \sum(\sum(x_0, x_2), x_3) = \sum x_0, x_3 = x_0$ Це витікає з асоціативного закону і Аксіоми В.1. Слід довести, що теорема справедлива при k = 1 і довільному n. Тому робимо наступний крок; б) k = 1, n = m, вважаючи, що твердження істинне при n = m, доведемо, що воно істинне і при n = m + 1: $\sum x_0, x_2, x_3, \dots, x_{m+1} = \sum(\sum(x_0, \dots, x_m), x_{m+1}) = \sum x_0, x_{m+1} = x_0$. Далі переходимо до зворотної індукції по k. в) перевіряємо справедливість теореми при k = n – 1: $\sum(x_1, \dots, x_0, x_n) = \sum(\sum(x_1, \dots, x_0), x_n) = \sum x_0, x_n = x_0$ відповідно до закону нульової множини.
	С. 80.	**С. 72.**
	г) покажемо, що якщо теорема вірна для довільного k, 1 < k < n, то вона вірна і для k – 1: $\Sigma(x_1, \dots, x_k, x_0, x_{k+2}, x_n) = \underbrace{\Sigma(\Sigma(\dots\Sigma}_{n-k\text{ раз}}\left(\sum_{i=1}^{k-1} x_i, x_k\right), x_0), x_{k+2}), \dots), x_n) = x_0 \qquad (2.1)$ $\Sigma(x_1, \dots, x_{k-1}, x_0, x_{k+1}, \dots, x_n) = \underbrace{\Sigma(\Sigma(\dots\Sigma}_{n-k\text{ раз}}\left(\sum_{i=1}^{k-1} x_i, x_0\right), x_{k+1}), x_{k+2}), \dots), x_n) =$ $\underbrace{\Sigma(\Sigma(\dots\Sigma}_{n-k\text{ раз}}(x_0, x_{k+1}), x_{k+2}), \dots), x_n) = \underbrace{\Sigma(\Sigma(\dots\Sigma}_{n-k-1\text{ раз}}(x_0, x_{k+2}), x_{k+3}), \dots), x_n) \qquad (2.2)$	г) покажемо, що якщо теорема вірна для довільного k, 1 < k < n, то вона вірна і для k – 1: $\sum(x_1, \dots, x_k, x_0, x_{k+2}, x_n) = \underbrace{\sum(\sum(\dots\sum}_{n-k\text{ раз}}\left(\sum_{i=1}^{k-1} x_i, x_k\right), x_0), x_{k+2}), \dots), x_n) = x_0 \qquad (2.1)$ $\sum(x_1, \dots, x_{k-1}, x_0, x_{k+1}, \dots, x_n) = \underbrace{\sum(\sum(\dots\sum}_{n-k\text{ раз}}\left(\sum_{i=1}^{k-1} x_i, x_0\right), x_{k+1}), x_{k+2}), \dots), x_n) =$ $\underbrace{\sum(\sum(\dots\sum}_{n-k\text{ раз}}(x_0, x_{k+1}), x_{k+2}), \dots), x_n) = \underbrace{\sum(\sum(\dots\sum}_{n-k-1\text{ раз}}(x_0, x_{k+2}), x_{k+3}), \dots), x_n) \qquad (2.2)$
	С. 81.	**С. 72.**
	Суму (2.1) приведемо до вигляду (2.2), використовуючи закон нульової множини:	Суму (2.1) приведемо до вигляду (2.2), використовуючи закон нульової множини:

	$$\underbrace{\Sigma(\Sigma(\ldots\Sigma}_{n-k\text{ раз}}\left(\sum_{i=1}^{k-1} x_i, x_k\right), x_0), x_{k+2}), \ldots), x_n) =$$ $$= \underbrace{\Sigma(\Sigma(\ldots\Sigma}_{n-k\text{ раз}}\left(\sum_{i=1}^{k-1} x_i, x_0\right), x_0), x_{k+2}), \ldots), x_n) =$$ $$= \underbrace{\Sigma(\Sigma(\ldots\Sigma}_{n-k-1\text{ раз}}\left(x_0, x_{k+2}\right), x_{k+3}), \ldots), x_n)$$	$$\underbrace{\Sigma(\Sigma(\ldots\Sigma}_{n-k\text{ раз}}\left(\sum_{i=1}^{k-1} x_i, x_k\right), x_0), x_{k+2}), \ldots), x_n) =$$ $$= \underbrace{\Sigma(\Sigma(\ldots\Sigma}_{n-k\text{ раз}}\left(\sum_{i=1}^{k-1} x_i, x_0\right), x_0), x_{k+2}), \ldots), x_n) =$$ $$= \underbrace{\Sigma(\Sigma(\ldots\Sigma}_{n-k-1\text{ раз}}\left(x_0, x_{k+2}\right), x_{k+3}), \ldots), x_n)$$
	С. 81.	**С. 72.**
	Видно, що суми (2.1) і (2.2) набувають ідентичного вигляду, отже, твердження z справедливе, і, отже, істинна теорема в цілому. Слід зазначити, що комутативний закон для додавання ОПЛО виконується для відношення структурної еквівалентності: $(\Sigma(X,Y)R_{ST}(\Sigma(Y,X))$. **Марченко переписав чужий текст разом із висловом «твердження z», у той час як цього «твердження z» у тексті ніде немає, а є «твердження г», яке було в дисертації Швецова (2004), звідки з помилкою – «твердження z» – переписав текст Слюняєв, а вже від Слюняєва ця помилка переїхала до Марченка.** **Недолугий смішний плагіат.**	Видно, що суми (2.1) і (2.2) набувають ідентичного вигляду, отже, твердження z справедливе, і, отже, істинна теорема в цілому. Слід зазначити, що комутативний закон для додавання ОПІО виконується для відношення структурної еквівалентності: $(\Sigma(X,Y)R_{ST}(\Sigma(Y,X))$.
	С. 81.	**С. 72–73.**
	Опис функціонування логічного об'єкта інформаційної системи. Множина функцій логічного об'єкта. Введення поняття "множина функцій" дозволяє природним чином розділити ЛО на два класи: активні ({F} ≠ ø) і пасивні ({F} = ø). Активними ЛО (АЛО) називатимемо такі ЛО, у яких множина функцій не порожня, тобто вони володіють власною поведінкою і можуть виконувати деякі активні дії. Пасивними ЛО (ПЛО) назвемо ЛО, що не володіють власною поведінкою, пасивно беруть участь в реалізації деяких дій. **Марченко у переписаному чужому тексті замінив слова «інформаційного об'єкта» на «логічного об'єкта», «ІО» на «ЛО», «АІО» на «АЛО» та «ПІО» на «ПЛО».** **Нахабний плагіат.**	**Множина функцій інформаційного об'єкта.** Введення поняття "множина функцій" дозволяє природним чином розділити ІО на два класи: активні ({F} ≠ ø) і пасивні ({F} = ø). Активними ІО (АІО) називатимемо такі ІО, у яких множина функцій не порожня, тобто вони володіють власною поведінкою і можуть виконувати деякі активні дії. Пасивними ІО (ПІО) назвемо ІО, що не володіють власною поведінкою, пасивно беруть участь в реалізації деяких дій.
	С. 81–82.	**С. 73.**
	Припустимо, що взаємодія АЛО здійснюється через прийом і передачу ПЛО, при цьому в множині функцій АЛО можуть породжуватися необхідні ПЛО і передаватися іншому ЛО. Цей інформаційний обмін можна трактувати як передачу повідомлень, обмін сигналами, зміну вхідних сигналів і тому подібне, що дозволяє розглядати систему "джерело – приймач" в більш широкому сенсі ніж парадигма "клієнт – сервер". Традиційним засобом опису функціонування об'єктів в об'єктно-орієнтованих методах і системах є модель скінченого автомата. Як більш універсальний засіб визначення функціонування ЛО пропонується використовувати апарат канонічних числень Е. Поста [27]. Канонічним численням (КЧ) називатимемо четвірку вигляду	Припустимо, що взаємодія АІО здійснюється через прийом і передачу ПІО, при цьому в множині функцій АІО можуть породжуватися необхідні ПІО і передаватися іншому ІО. Цей інформаційний обмін можна трактувати як передачу повідомлень, обмін сигналами, зміну вхідних сигналів і тому подібне, що дозволяє розглядати систему "джерело – приймач" в більш широкому сенсі ніж парадигма "клієнт – сервер". Традиційним засобом опису функціонування об'єктів в об'єктно-орієнтованих методах і системах є модель скінченого автомата. Як більш універсальний засіб визначення функціонування ІО пропонується використовувати апарат канонічних числень Е. Поста [106]. Канонічним численням (КЧ) називатимемо четвірку вигляду

	(A,a,P,G), (2.3) де A – алфавіт числення, a – список аксіом КЧ, P – алфавіт змінних, G – список правил виводу, кожне з яких має вигляд.2 $$\frac{\begin{array}{l}G_{11}p_{11},G_{12}p_{12},...,G_{1n_1}p_{1n_1}G_{1n_{1+1}};\\ G_{21}p_{21},G_{22}p_{22},...,G_{2n_2}p_{2n_2}G_{2n_{2+1}};\\ G_{m1}p_{m1},G_{m2}p_{m2},...,G_{mn_m}p_{mn_m}\end{array}}{G_1p_1,G_2p_2,...,G_np_nG_{n+1}}$$	(A,a,P,G), (2.3) де A – алфавіт числення, a – список аксіом КЧ, P – алфавіт змінних, G – список правил виводу, кожне з яких має вигляд $$\frac{\begin{array}{l}G_{11}p_{11},G_{12}p_{12},...,G_{1n_1}p_{1n_1}G_{1n_{1+1}};\\ G_{21}p_{21},G_{22}p_{22},...,G_{2n_2}p_{2n_2}G_{2n_{2+1}};\\ G_{m1}p_{m1},G_{m2}p_{m2},...,G_{mn_m}p_{mn_m}\end{array}}{G_1p_1,G_2p_2,...,G_np_nG_{n+1}}$$
	С. 82–83.	**С. 74–75.**
	де Gij(i =1,...,m, j =1,...,n) і G(k =1,...,n+1) – деякі конкретні слова в алфавіті A, у тому числі і порожнє слово. Далі скористаємося визначеннями, введеними в [14-22]. Число т називається індексом схеми (2.4). Реалізуючим набором схеми (2.4) називатимемо вираз вигляду $$\begin{array}{c}p^{(1)},p^{(2)},...,p^{(f)};\\ P_1,P_2,...,P_f\end{array}$$ де p(1),p(2),...,p(f) – список (без повторень) всіх змінних, що входять в (2.4), а Pi – слово в алфавіті A, яке називається значенням змінної в даному реалізуючому наборі (i=1,2...,f). Якщо замість кожного входження кожної змінної в схему (2.4) підставити значення цієї змінної в R (R – це конкретний реалізуючий набір схеми (2.4)), то всі рядки схеми перетворяться на слова в алфавіті A. Так виходить реалізація продукційної схеми (2.4) реалізуючим набором R. Конструктивний об’єкт називатимемо реалізацією схеми (2.4), якщо він є реалізацією схеми яким-небудь реалізуючим набором. Слово Q називатимемо словом, що безпосередньо виводиться по схемі (2.4) із слів Q1,Q2,...,Qm, якщо вираз $$Q_1,Q_2,...,Q_m \Rightarrow Q$$ є реалізацією схеми (2.4). Список слів називається виводом в КЧ, якщо кожне слово списку є аксіомою даного КЧ або безпосередньо виводиться по якій-небудь схемі із попередніх слів в даному списку. Довжиною виводу називатимемо число слів у виводі. Слово P називатимемо таким, що виводиться в КЧ, якщо можна побудувати вивід в КЧ, останнім словом якого є P.	де Gij (i =1,...,m, j =1,...,n) і G(k =1,...,n +1) – деякі конкретні слова в алфавіті A, у тому числі і порожнє слово. Далі скористаємося визначеннями, введеними в [107–108]. Число т називається індексом схеми (2.4). Реалізуючим набором схеми (2.4) називатимемо вираз вигляду $$\begin{array}{c}p^{(1)},p^{(2)},...,p^{(f)};\\ P_1,P_2,...,P_f\end{array}$$ де p(1) , p(2) ,..., p(f) – список (без повторень) всіх змінних, що входять в (2.4), а Pi – слово в алфавіті A, яке називається значенням змінної в даному реалізуючому наборі (i=1,2...,f). Якщо замість кожного входження кожної змінної в схему (2.4) підставити значення цієї змінної в R (R – це конкретний реалізуючий набір схеми (2.4)), то всі рядки схеми перетворяться на слова в алфавіті A. Так виходить реалізація продукційної схеми (2.4) реалізуючим набором R. Конструктивний об’єкт називатимемо реалізацією схеми (2.4), якщо він є реалізацією схеми яким-небудь реалізуючим набором. Слово Q називатимемо словом, що безпосередньо виводиться по схемі (2.4) із слів Q1,Q2 ,...,Qm , якщо вираз $$Q_1,Q_2,...,Q_m \Rightarrow Q$$ є реалізацією схеми (2.4). Список слів називається виводом в КЧ, якщо кожне слово списку є аксіомою даного КЧ або безпосередньо виводиться по якій-небудь схемі із попередніх слів в даному списку. Довжиною виводу називатимемо число слів у виводі. Слово P називатимемо таким, що виводиться в КЧ, якщо можна побудувати вивід в КЧ, останнім словом якого є P.
	С. 83.	**С. 75.**
	Поняття виводу і виводимості в КЧ дозволяють визначити еквівалентність числень і їх відношення з поняттям множини (2.4). Якщо алфавіт містить A, то воно називається численням над A. Два числення над A еквівалентні відносно A, якщо будь-яке слово в A виводиться в першому численні тоді і тільки тоді, коли воно виводиться в другому. Пара (A,T), (2.5) де T – канонічне числення над A, є поданням множини що виводяться в T слів алфавіту A. При цьому A називатимемо основним алфавітом подання, а доповнення до повного алфавіту числення T називатимемо допоміжним алфавітом, при цьому говорять, що T строго представляє множину слів, що виводяться в ньому. Множина M слів в A злічима, якщо для неї існує подання (2.5).	Поняття виводу і виводимості в КЧ дозволяють визначити еквівалентність числень і їх відношення з поняттям множини (2.4). Якщо алфавіт містить A, то воно називається численням над A. Два числення над A еквівалентні відносно A, якщо будь-яке слово в A виводиться в першому численні тоді і тільки тоді, коли воно виводиться в другому. Пара (A,T), (2.5) де T – канонічне числення над A, є поданням множини що виводяться в T слів алфавіту A. При цьому A називатимемо основним алфавітом подання, а доповнення до повного алфавіту числення T називатимемо допоміжним алфавітом, при цьому говорять, що T строго представляє множину слів, що виводяться в ньому. Множина M слів в A злічима, якщо для неї існує подання (2.5).
	С. 84.	**С. 75–76.**
	Для формалізації функціональної моделі ЛО	Для формалізації функціональної моделі ІО

	введемо: {R} – множина ПЛО, що приймаються даним об'єктом; {T} – множина передаваних даним об'єктом ПЛО. Вважаємо, що функціональна модель оперує з множинами {A}, {R} і {T}. Оскільки для кожного атрибута Ai множина, на якій він визначений, – SAi, може мати різну природу [22, 27], то елементи цієї множини можна інтерпретувати досить широко: як програмні коди, виклики функцій операційної системи, графічні структури та ін. Також виділимо типи станів ЛО у функціональній моделі: а) стани, в яких можливий прийом елементів множини {R}, надалі позначених як Ri; б) стани, в яких неможливий прийом Ri. Оскільки множина функцій повинна враховувати співвідношення атрибутів і зміст Ri, то необхідно ввести предикати, які утворюватимуть множину допустимих предикатів: $\{Pr\} = (Pr_1, Pr_2, \dots, Pr_\varphi)$. **Марченко у переписаному чужому тексті замінив слова «ІО» на «ЛО» та «ПІО» на «ПЛО».** **До чужого переписаного тексту Марченко додав два фальшиві покликання 2014-го та 2016-го років.** **Справжнє джерело – це дисертація Слюняєва, яка була захищена 2010 року.** **Плагіат.**	введемо: {R} – множина ПІО, що приймаються даним об'єктом; {T} – множина передаваних даним об'єктом ПІО. Вважаємо, що функціональна модель оперує з множинами {A}, {R} і {T}. Оскільки для кожного атрибута Ai множина, на якій він визначений, – SAi, може мати різну природу, то елементи цієї множини можна інтерпретувати досить широко: як програмні коди, виклики функцій операційної системи, графічні структури та ін. Також виділимо типи станів ІО у функціональній моделі: а) стани, в яких можливий прийом елементів множини {R}, надалі позначених як Ri; б) стани, в яких неможливий прийом Ri. Оскільки множина функцій повинна враховувати співвідношення атрибутів і зміст Ri, то необхідно ввести предикати, які утворюватимуть множину допустимих предикатів: $\{Pr\} = (Pr_1, Pr_2, \dots, Pr_\varphi)$.
	С. 84–85.	**С. 76.**
	Для аналізу складних умов і співвідношень будуватимемо формули над предикатами в мові числення висловлювань і позначати їх F(Pr) або просто F. Алфавіт числення КФМ визначимо таким чином: А = ({R}, {T}, {A}, {S}, $\{Pr\}, \&, \vee, (,), \neg, \rightarrow, \Im, \nabla, \varnothing)$, де {S} – множина станів ЛО, J – символ порожнього слова. До нього включаємо символи мови ЛО для побудови формул F. Алфавіт змінних включатиме змінні P = (p, q, f, hA), де p – послідовність вхідних ПЛО, q – послідовність вихідних ПЛО, f – послідовність формул з предикатами Pr в ЛО, hA – список атрибутів ЛО, для якого будується функціональна модель. Аксіому числення задамо як $A = (\varnothing \Im S_0 \Im hA(0) \Im \varnothing \Im \varnothing)$, де ø означає порожній стан змінної, а під hA(0) розуміється список вигляду $hA(0) = <<N_{A1}, S_{A1}, V_{A1}(0)>, <N_{A2}, S_{A2}, V_{A2}(0)>, \dots <N_{A\theta}, S_{A\theta}, V_{A\theta}(0)>>$, де VAi(0) позначає значення i-го атрибуту у момент часу t = 0, тобто у момент початку функціонування ЛО.	Для аналізу складних умов і співвідношень будуватимемо формули над предикатами в мові числення висловлювань і позначати їх F(Pr) або просто F. Алфавіт числення КФМ визначимо таким чином: А = ({R}, $\{T\}, \{A\}, \{S\}, \{Pr\}, \&, \vee, (,), \neg, \rightarrow, \xi, \nabla, \varnothing)$, де {S} – множина станів ІО, ξ – символ порожнього слова. До нього включаємо символи мови ІО для побудови формул F. Алфавіт змінних включатиме змінні P = (p, q, f, hA), де p – послідовність вхідних ПІО, q – послідовність вихідних ПІО, f – послідовність формул з предикатами Pr в ІО, hA – список атрибутів ІО, для якого будується функціональна модель. Аксіому числення задамо як $A = (\varnothing \xi S_0 \xi hA(0) \xi \varnothing \xi \varnothing)$, де ø означає порожній стан змінної, а під hA(0) розуміється список вигляду $hA(0) = <<N_{A1}, S_{A1}, V_{A1}(0)>, <N_{A2}, S_{A2}, V_{A2}(0)>, \dots <N_{A\theta}, S_{A\theta}, V_{A\theta}(0)>>$, де VAi(0) позначає значення i-го атрибуту у момент часу t = 0, тобто у момент початку функціонування ІО.
	С. 85.	**С. 77.**
	Правила виводу для числення КФМ будуватимемо як схеми правил, оскільки в конкретній функціональній моделі виходитиме різна кількість правил виводу, які мають вигляд, що задовольняє запропонованим схемам: Схема 1: $R_i p \Im S_0 \Im hA(0) \Im q \Im f \Rightarrow p \Im S_i \Im hA(R_i) \Im q, T_i \Im f, F_i$; (2.6) Схема 2: $R_i p \Im S_0 \Im hA(0) \Im q \Im f \Rightarrow \nabla p \Im S_i \Im hA(R_i) \Im q, T_i \Im f, F_i$. (2.7)	Правила виводу для числення КФМ будуватимемо як схеми правил, оскільки в конкретній функціональній моделі виходитиме різна кількість правил виводу, які мають вигляд, що задовольняє запропонованим схемам: Схема 1: $R_i p \xi S_0 \xi hA(0) \xi q \xi f \Rightarrow p \xi S_i \xi hA(R_i) \xi q, T_i \xi f, F_i$; (2.6) Схема 2: $R_i p \xi S_0 \xi hA(0) \xi q \xi f \Rightarrow \nabla p \xi S_i \xi hA(R_i) \xi q, T_i \xi f, F_i$. (2.7)
	С. 85.	**С. 77.**
	Схеми (2.6) і (2.7) задають правила, що виво-	Схеми (2.6) і (2.7) задають правила, що виво-

	дять зі стану S0 в стани типу а) і б) відповідно. Для позначення неможливості обробки вхідної послідовності Ri використовується службовий символ □. У цих схемах породжується вихідний ПЛО Ti, і формула Fi, при цьому ми допускаємо можливість завдання Ti = ø і Fi = ø , що дозволяє уникнути зайвих схем виводу [18, 51]. Схема 3: $R_1 p\exists S_i \exists hA \exists q \exists f \Rightarrow p\exists S_j \exists hA(R_1) \exists q.T_j \exists f.F_j$; (2.8) Схема 4: $R_1 p\exists S_1 \exists hA \exists q \exists f.F_1 \Rightarrow p\exists S_j \exists hA(R_1) \exists q.T_j \exists f$. (2.9) **Марченко у переписаному чужому тексті замінив слова «ПІО» на «ПЛО».** **До чужого переписаного тексту Марченко додав два фальшивих джерела: [18], у якому немає такої фрази й формул, та [51], написане 2016-го року.** **Справжнє джерело – це дисертація Слюняєва, яка була захищена 2010 року.** **Нахабний плагіат.**	дять зі стану S0 в стани типу а) і б) відповідно. Для позначення неможливості обробки вхідної послідовності Ri використовується службовий символ □. У цих схемах породжується вихідний ПІО Ti, і формула Fi, при цьому ми допускаємо можливість завдання Ti = ø і Fi = ø , що дозволяє уникнути зайвих схем виводу. Схема 3: $R_1 p\exists S_i \exists hA \exists q \exists f \Rightarrow p\exists S_j \exists hA(R_1) \exists q.T_j \exists f.F_j$; (2.8) Схема 4: $R_1 p\exists S_i \exists hA \exists q \exists f.F_i \Rightarrow p\exists S_j \exists hA(R_1) \exists q.T_j \exists f$. (2.9)
	С. 85.	**С. 77.**
	Схеми (2.8) і (2.9) визначають переходи із станів типу а) в стани типу а), з аналізом істинності Fi або без аналізу. Допускаємо також, що в Fi може бути задана формула (□Fi), тобто перевіряється істинність заперечення деякої формули. Таке розширення допустиме, оскільки у численні висловлювань істинність або помилковість будь-якого вислову може бути точно встановлена. Оброблена формула Fi виключається з подальшого процесу виводу. При циклічній поведінці ЛО необхідна формула може знову породжуватися схемами (2.8).	Схеми (2.8) і (2.9) визначають переходи із станів типу а) в стани типу а) , з аналізом істинності Fi або без аналізу. Допускаємо також, що в Fi може бути задана формула (□Fi), тобто перевіряється істинність заперечення деякої формули. Таке розширення допустиме, оскільки у численні висловлювань істинність або помилковість будь-якого вислову може бути точно встановлена. Оброблена формула Fi виключається з подальшого процесу виводу. При циклічній поведінці ІО необхідна формула може знову породжуватися схемами (2.8).
	С. 86.	**С. 78.**
	Схема 5: $R_1 p\exists S_1 \exists hA \exists q \exists f \Rightarrow \nabla p\exists S_j \exists hA(R_1) \exists q.T_j \exists f.F_j$; (2.10) Схема 6: $R_1 p\exists S_1 \exists hA \exists q \exists f.F_1 \Rightarrow \nabla p\exists S_j \exists hA(R_1) \exists q \exists f$. (2.11) Схема (2.10) задає перехід із стану типу а) в стан типу б) без аналізу F, а схема (2.11) – з аналізом F. Схема 7: $\nabla p\exists S_1 \exists hA \exists q \exists f \Rightarrow p\exists S_j \exists hA(R_1) \exists q.T_j \exists f.F_j$; (2.12) Схема 8: $\nabla p\exists S_1 \exists hA \exists q \exists f.F_j \Rightarrow p\exists S_j \exists hA(S_1) \exists q.T_j \exists f$; (2.13) Схема 9: $\nabla p\exists S_1 \exists hA \exists q \exists f \Rightarrow \nabla p\exists S_j \exists hA(S_1) \exists q.T_j \exists f.F_j$; (2.14) Схема 10: $\nabla p\exists S_1 \exists hA \exists q \exists f.F_1 \Rightarrow \nabla p\exists S_j \exists hA(S_1) \exists q.T_j \exists f$. (2.15)	Схема 5: $R_1 p\exists S_1 \exists hA \exists q \exists f \Rightarrow \nabla p\exists S_j \exists hA(R_1) \exists q.T_j \exists f.F_j$; (2.10) Схема 6: $R_1 p\exists S_1 \exists hA \exists q \exists f.F_1 \Rightarrow \nabla p\exists S_j \exists hA(R_1) \exists q \exists f$. (2.11) Схема (2.10) задає перехід із стану типу а) в стан типу б) без аналізу F, а схема (2.11) – з аналізом F. Схема 7: $\nabla p\exists S_1 \exists hA \exists q \exists f \Rightarrow p\exists S_j \exists hA(R_1) \exists q.T_j \exists f.F_j$; (2.12) Схема 8: $\nabla p\exists S_1 \exists hA \exists q \exists f.F_j \Rightarrow p\exists S_j \exists hA(S_1) \exists q.T_j \exists f$; (2.13) Схема 9: $\nabla p\exists S_1 \exists hA \exists q \exists f \Rightarrow \nabla p\exists S_j \exists hA(S_1) \exists q.T_j \exists f.F_j$; (2.14) Схема 10: $\nabla p\exists S_1 \exists hA \exists q \exists f.F_1 \Rightarrow \nabla p\exists S_j \exists hA(S_1) \exists q.T_j \exists f$. (2.15)
	С. 86.	**С. 78.**
	Схеми (2.12), (2.13) визначають переходи із станів типу б) в стани типу а), а схеми (2.14) і (2.15) із станів типу б) в стани типу б). У цих схемах правил закладається можливість повернення в стан S0 і зупинки при переході в такий стан Sj з якого немає можливості подальшого виводу. У схемах правил виводу (2.6) – (2.15) задаються в загальному вигляді функціональні перетворення hA(Si), які можна визначити як перетворення над значеннями атрибутів: VAi := f(k)(VAφ1,...,VAφk) , де k – арність функціонального символу, VAφi – і-й аргумент функції f(k), узятий із списку значень атрибутів.	Схеми (2.12), (2.13) визначають переходи із станів типу б) в стани типу а) , а схеми (2.14) і (2.15) із станів типу б) в стани типу б) . У цих схемах правил закладається можливість повернення в стан S0 і зупинки при переході в такий стан Sj з якого немає можливості подальшого виводу. У схемах правил виводу (2.6) – (2.15) задаються в загальному вигляді функціональні перетворення h A (Si), які можна визначити як перетворення над значеннями атрибутів: Vai := f(k) (VAφ1,...,VAφk), де k – арність функціонального символу, VAφi – і-й аргумент функції f(k), узятий із списку значень атрибутів.
	С. 91.	**С. 86.**
	2.4. Розробка моделі процесу функціонування ідентифікації та діагностування стану логічних об’єктів	**2.3. Розробка моделі процесу пошуку рішення інтелектуальним агентом інформаційно-керуючої системи аеропорту**

	В ході аналізу динамічної моделі функціонування ЛО встановлено той факт, що в загальних моделях ідентифікації та діагностування (рис. 2.5, рис. 2.6) застосовуються різні способи функціонування. Під функціонуванням слід розуміти знаходження шляху досягнення мети або цілей даними. Оскільки різні логічні об’єкти володіють своєю специфікою, то навряд чи є можливим застосування деякого універсального методу функціонування для всіх підсистем.	В ході аналізу відомих моделей ІА встановлено той факт, що в даних моделях застосовуються різні способи пошуку рішення. Під пошуком рішення слід розуміти знаходження шляху досягнення мети або цілей даним ІА в поточному стані МА-оточення. Оскільки різні структурні підрозділи аеропорту володіють своєю специфікою в т.ч. і при прийнятті рішень, то навряд чи є можливим застосування деякого універсального методу пошуку рішення для всіх підсистем ІКСА.
	С. 94–95.	**С. 86–87.**
	Більш доцільним є розробка на базі динамічної моделі функціонування ЛО, що дозволить створювати різні моделі ідентифікації та діагностування (МІД). З погляду реалізації дій моделі слід розбити на три класи: 1) з визначеною кінцевою множиною елементарних дій [1-35]; 2) з множиною планів [51-58]; 3) з довільними повідомленнями і діями в логічній мові [27]. Узагальнюючи ці моделі функціонування, в узагальненій моделі МІД пропонується наступний варіант функціонування. Вважаємо, що МІД має визначену множину статичних цілей GS ={gsi|i =1,...,n}. Апріорі відомі шляхи досягнення цілей, тобто побудовані логічні об’єкти і (IOi|=1,...,n), функціонування яких повинне вести до gsi . Тут кожен ЛО покриває деякий план, слідуючи термінології [51-58]. Усередині ж цього плану, тобто в моделі поведінки ЛО, можуть бути сформовані довільні повідомлення і довільні послідовності дій, як показано у розд. 2.3.	Більш доцільним є розробка на базі узагальненої моделі ІА “бібліотеки” моделей, які довз олять створювати різні конкретні моделі ІА. З погляду реалізації дій відомі моделі слід розбити на три класи: 1) з визначеною кінцевою множиною елементарних дій [112–117]; 2) з множиною планів [118–119]; 3) з довільними повідомленнями і діями в логічній мові [120–121]. Узагальнюючи ці моделі пошуку рішення, в узагальненій моделі ІА пропонується наступний варіант пошуку рішення. Вважаємо, що ІА має визначену множину статичних цілей GS {gs^i|i = 1,...,n}. Апріорі відомі шляхи досягнення цілей, тобто побудовані інформаційні об’єкти (IO^i|i = 1,..., n), функціонування яких повинне вести до gs i. Тут кожен ІО покриває деякий план, слідуючи термінології [122–124]. Усередині ж цього плану, тобто в моделі поведінки ІО, можуть бути сформовані довільні повідомлення і довільні послідовності дій, як показано у розд. 2.1.
	С. 95.	**С. 87.**
	Тоді модель функціонування задається функцією функціонування SR:GS→VO де VO – множина вкладених ЛО i-го МІД. Це відображення однозначне, але не взаємно, оскільки можливо, що декілька цілей досягаються одним і тим же ЛО. Модель активних дій визначається відображенням AD:GA →VO, яке вибирає необхідні для запуску у нинішній момент ЛО. Процес запуску ЛО реалізується на стадії функціонування СБ. Зміна активних цілей (зміна множини GA) приводитиме до зупинки і запуску відповідних ЛО.	Тоді модель пошуку рішення задається функцією пошуку рішення SR :GS→VO де VO – множина вкладених ІО i-го ІА. Це відображення однозначне, але не взаємно, оскільки можливо , що декілька цілей досягаються одним і тим же ІО. Модель активних дій визначається відображенням AD:GA →VO, яке вибирає необхідні для запуску у нинішній момент ІО. Процес запуску ІО реалізується на стадії функціонування ІКСА. Зміна активних цілей (зміна множини GA) приводитиме до зупинки і запуску відповідних ІО.
	С. 95–96.	**С. 87–88.**
	Далі розробляється продукційна формалізація функціонування агента моніторингу стану логічних об’єктів в узагальненій моделі МІД. Введемо наступні позначення: SA(0) – початкова конфігурація структури атрибутів МІД, Lvo – список вкладених ЛО, SMA(0) – початкова конфігурація МІД-оточення, Ls – список стратегій МІД, Lsg – список статичних цілей, Ltg – список цілей, що отримуються, Ldg – список цілей, що передаються, Ldag – список автономних динамічних цілей, Lag – список активних цілей. Аксіома числення KMVB у момент первинного запуску МІД приймає вигляд $AA = \{SA(0); Lvo; SMA(0); Ls(0); Lsg(0); \varnothing; \varnothing; \varnothing; \varnothing\}$ (2.16)	Далі розробляється продукційна формалізація функціонування інтелектуального агента в узагальненій моделі ІА. Введемо наступні позначення: SA(0) – початкова конфігурація структури атрибутів ІА, Lvo – список вкладених ІО, SMA(0) – початкова конфігурація МА-оточення, Ls – список стратегій ІА, Lsg – список статичних цілей, Ltg – список цілей, що отримуються “зверху”, Ldg – список цілей, передаваних “вниз”, Ldag – список автономних динамічних цілей, Lag – список активних цілей. Аксіома числення KMVB у момент первинного запуску ІА приймає вигляд $AA = \{SA(0); Lvo; SMA(0); Ls(0); Lsg(0); \varnothing; \varnothing; \varnothing; \varnothing\}$
	С. 96.	**С. 88.**
	Для спрощення запису інформаційного простору V і його стану SV введемо	Для спрощення запису інформаційного простору V і його стану SV введемо

	$\mathrm{Lia}=\{\mathrm{NIA}_j,\{A^{\psi}_{IA_j}\,\vert\,\psi=1,\ldots,k\}\,\vert\,j=1,\ldots,l\}$ $\mathrm{Lio}=\{\mathrm{NIO}_j,\{A^{\psi}_{IO_j}\,\vert\,\psi=1,\ldots,k\}\,\vert\,j=1,\ldots,l\}$ V представляємо двома списками – Lia, Lio, а SV – списками атрибутів Laia, Laio. Тоді $(V,SV)\Leftrightarrow \mathrm{Lia},\ \mathrm{Lio},\ \mathrm{Laia},\ \mathrm{Laio}$.	$\mathrm{Lia}=\{\mathrm{NIA}_j,\{A^{\psi}_{IA_j}\,\vert\,\psi=1,\ldots,k\}\,\vert\,j=1,\ldots,l\}$, $\mathrm{Lio}=\{\mathrm{NIO}_j,\{A^{\psi}_{IO_j}\,\vert\,\psi=1,\ldots,k\}\,\vert\,j=1,\ldots,l\}$. V представляємо двома списками – Lia, Lio, а SV – списками атрибутів Laia, Laio. Тоді $(V,SV)\Leftrightarrow \mathrm{Lia},\ \mathrm{Lio},\ \mathrm{Laia},\ \mathrm{Laio}$.
	С. 96.	**С. 88.**
	Формування першої ситуації МІД-оточення описуємо правилом $(1)\ \mathrm{AA}\Rightarrow(2)\mathrm{Lia},\mathrm{Lio},\varnothing,\varnothing;\mathrm{AA}$ Виконання обсервації визначиться правилом (2) $(2)\ \mathrm{Lia},\mathrm{Lio},\varnothing,\varnothing;\mathrm{AA}\Rightarrow(3)\mathrm{FV}(\mathrm{lia},\mathrm{lio},\varnothing,\varnothing);\mathrm{AA}$	Формування першої ситуації МА-оточення описуємо правилом $(1)\ \mathrm{AA}\Rightarrow(2)\mathrm{Lia},\mathrm{Lio},\varnothing,\varnothing;\mathrm{AA}$. Виконання обсервації визначиться правилом (2) $(2)\mathrm{Lia},\mathrm{Lio},\varnothing,\varnothing;\mathrm{AA}\Rightarrow(3)\mathrm{FV}(\mathrm{lia},\mathrm{lio},\varnothing,\varnothing);\mathrm{AA}$.
	С. 96–97.	**С. 88–89.**
	Далі для скорочення запису позначатимемо складові антецедента кутовими дужками з нижнім індексом як номер, і, якщо вміст даного компоненту не змінюється в даному правилі, то просто повторюватимемо Ri . В правилах числення приписуємо на початок антецедента номер правила в круглих дужках (...), а в консеквенті формуватимемо номер наступного правила. Тоді формування початкового стану МІД-оточення опишеться правилом $(3)\ \mathrm{Lia},\mathrm{Lio},\mathrm{Laia},\mathrm{Laio};\mathrm{AA}\Rightarrow(4)\ \Re_1\langle S_A\rangle_2\langle \mathrm{lvo}\rangle_3$ $\langle\varnothing;\mathrm{Lia}':=\mathrm{FP}(\Re_1),\mathrm{Lio}':=\mathrm{FP}(\Re_1),\mathrm{Laia}':=\mathrm{FP}(\Re_1),\mathrm{Laio}':=\mathrm{FP}(\Re_1)\rangle_4$ $\langle \mathrm{ls}\rangle_5\langle \mathrm{lsg}\rangle_6\langle\varnothing\rangle_7\langle\varnothing\rangle_8\langle\varnothing\rangle_9\langle\varnothing\rangle_{10}$	Далі для скорочення запису позначатимемо складові антецедента кутовими дужками з нижнім індексом як номер, і, якщо вміст даного компоненту не змінюється в даному правилі, то просто повторюватимемо Ri . В правилах числення приписуємо на початок антецедента номер правила в круглих дужках (...), а в консеквенті формуватимемо номер наступного правила. Тоді формування початкового стану МА-оточення опишеться правилом $(3)\ \mathrm{Lia},\mathrm{Lio},\mathrm{Laia},\mathrm{Laio};\mathrm{AA}\Rightarrow(4)\ \Re_1\langle S_A\rangle_2\langle \mathrm{lvo}\rangle_3$ $\langle\varnothing;\mathrm{Lia}':=\mathrm{FP}(\Re_1),\mathrm{Lio}':=\mathrm{FP}(\Re_1),\mathrm{Laia}':=\mathrm{FP}(\Re_1),\mathrm{Laio}':=\mathrm{FP}(\Re_1)\rangle_4$ $\langle \mathrm{ls}\rangle_5\langle \mathrm{lsg}\rangle_6\langle\varnothing\rangle_7\langle\varnothing\rangle_8\langle\varnothing\rangle_9\langle\varnothing\rangle_{10}$
	С. 97.	**С. 89.**
	Вибір стратегії задається правилом (4) $(4)\ \Re_1\Re_2\Re_3\Re_4\langle \mathrm{ls}\rangle_5\Re_6\Re_7\Re_8\Re_9\Re_{10}\Rightarrow$ $\Rightarrow(5)\ \Re_1\Re_2\Re_3\Re_4\langle s':=\mathrm{FSS}(\mathrm{ls},\Re_4,\Re_{10});\mathrm{ls}\rangle_5\Re_6\Re_7\Re_8\Re_9\Re_{10}$ Для прийому цілей, що отримуються введемо операцію читання буфера цілей – read(bltg). Відповідне правило приймає вигляд $(5)\ \Re_1\Re_2\Re_3\Re_4\Re_5\Re_6\Re_7\Re_8\Re_9\Re_{10}\Rightarrow$ $\Rightarrow(6)\ \Re_1\Re_2\Re_3\Re_4\Re_5\Re_6\Re_7\langle \mathrm{Ltg}:=\mathrm{read}(\mathrm{bltg})\rangle_8\Re_9\Re_{10}$	Вибір стратегії задається правилом (4) $(4)\ \Re_1\Re_2\Re_3\Re_4\langle \mathrm{ls}\rangle_5\Re_6\Re_7\Re_8\Re_9\Re_{10}\Rightarrow$ $\Rightarrow(5)\ \Re_1\Re_2\Re_3\Re_4\langle s':=\mathrm{FSS}(\mathrm{ls},\Re_4,\Re_{10});\mathrm{ls}\rangle_5\Re_6\Re_7\Re_8\Re_9\Re_{10}$ Для прийому цілей, що отримуються “зверху” введемо операцію читання буфера “верхніх” цілей – read(bltg). Відповідне правило приймає вигляд $(5)\ \Re_1\Re_2\Re_3\Re_4\Re_5\Re_6\Re_7\Re_8\Re_9\Re_{10}\Rightarrow$ $\Rightarrow(6)\ \Re_1\Re_2\Re_3\Re_4\Re_5\Re_6\Re_7\langle \mathrm{Ltg}:=\mathrm{read}(\mathrm{bltg})\rangle_8\Re_9\Re_{10}$
	С. 97.	**С. 89–90.**
	Далі можна сформувати цілі передавані, динамічні цілі з множини $G^{AVT(D)}$ і вибрати статичні цілі $(6)\ \Re_1\Re_2\Re_3\Re_4\Re_5\Re_6\Re_7\Re_8\Re_9\Re_{10}\Rightarrow$ $\Rightarrow(7)\ \Re_1\Re_2\Re_3\Re_4\Re_5\Re_6\Re_7\langle \mathrm{Ldg}:=h^{DOWN}(\Re_4,s',U,\mathrm{Lag})\rangle_8$ $\langle \mathrm{Ldag}:=h^{AVT(D)}(\Re_4,s',U,\mathrm{Lag})\rangle_9\Re_{10}\langle \mathrm{Lavtgs}:=h^{AVT(D)}(\Re_4,s',U,\mathrm{Lag})\rangle_{11}$ Тепер може бути побудована множина активних цілей $(7)\ \Re_1\Re_2\Re_3\Re_4\Re_5\Re_6\Re_7\Re_8\Re_9\Re_{10}\Re_{11}\Rightarrow(8)\ \Re_1\Re_2\Re_3\Re_4\Re_5\Re_6\Re_7\Re_8$ $\Re_9\langle \mathrm{Lag}:=h^{A}(s',\Re_9,\Re_{10},\Re_{11},Z)\rangle_{10}\Re_{11}$	Далі можна сформувати цілі передавані “вниз”, динамічні цілі з множини $G^{AVT(D)}$ і вибрати статичні цілі $(6)\ \Re_1\Re_2\Re_3\Re_4\Re_5\Re_6\Re_7\Re_8\Re_9\Re_{10}\Rightarrow$ $\Rightarrow(7)\ \Re_1\Re_2\Re_3\Re_4\Re_5\Re_6\Re_7\langle \mathrm{Ldg}:=h^{DOWN}(\Re_4,s',U,\mathrm{Lag})\rangle_8$ $\langle \mathrm{Ldag}:=h^{AVT(D)}(\Re_4,s',U,\mathrm{Lag})\rangle_9\Re_{10}\langle \mathrm{Lavtgs}:=h^{AVT(D)}(\Re_4,s',U,\mathrm{Lag})\rangle_{11}$ Тепер може бути побудована множина активних цілей $(7)\ \Re_1\Re_2\Re_3\Re_4\Re_5\Re_6\Re_7\Re_8\Re_9\Re_{10}\Re_{11}\Rightarrow(8)\ \Re_1\Re_2\Re_3\Re_4\Re_5\Re_6\Re_7\Re_8$ $\Re_9\langle \mathrm{Lag}:=h^{A}(s',\Re_9,\Re_{10},\Re_{11},Z)\rangle_{10}\Re_{11}$
	С. 98.	**С. 90.**
	Для кожної активної мети gaj вибираємо підмножину ЛО для яких апріорі припускаємо, що вони можуть вести до даної активної мети. Вважаємо, що цілі, що містяться в Lag не суперечливі, виконання цієї вимоги повинне забезпечити перетворення hA що використовує онтологію активізації цілей Z. Отже, множина VO розбивається на n непересічних підмножин, таких що	Для кожної активної мети gaj вибираємо підмножину ІО для яких апріорі припускаємо, що вони можуть вести до даної активної мети. Вважаємо, що цілі, що містяться в Lag не суперечливі, виконання цієї вимоги повинне забезпечити перетворення hA що використовує онтологію активізації цілей Z. Отже, множина VO розбивається на n непересічних підмножин, таких що

	$VO \subseteq \bigcup_{j=1}^{n} VOga^{j}$	$VO \subseteq \bigcup_{j=1}^{n} VOga^{j}$.
	С. 98.	**С. 90–91.**
	Такі підмножини VOga j називатимемо модельною множиною вкладених ЛО і позначати ModelSet j. ЛО, що входять в j-ту модельну множину позначатимемо IOm j,i де нижній індекс t показує [18, 51], що даний ЛО моделюється в процесі функціонування. Для здійснення такого розбиття потрібна функція FMVO = hVO(Lvo,Lag,X) , де X – онтологія, що описує логіку такого розбиття і необхідна тому, що потенційна множина динамічних цілей може бути нескінченною. Формулюємо правило (8) $\Re_1\Re_2\langle lvo\rangle_3\Re_4\Re_5\Re_6\Re_7\Re_8\Re_9\langle Lag\neq\varnothing\rangle_{10}\Re_{11}\Rightarrow$ $\Rightarrow\Re_1\Re_2\langle h^A(Lvo,Lag,X)\rangle_3\Re_4\Re_5\Re_6\Re_7\Re_8\Re_9\langle Lag\neq\varnothing\rangle_{10}\Re_{11}\Rightarrow$ $\Rightarrow$ (9) $\Re_1\Re_2\langle ModelSet,Lvo\rangle_3\Re_4\Re_5\Re_6\Re_7\Re_8\Re_9\Re_{10}\Re_{11}$ **Марченко у переписаному чужому тексті замінив слова «IO» на «ЛО».** **До чужого переписаного тексту Марченко додав два фальшивих джерела: [18], у якому немає такої фрази, слів «ModelSet», «index» і формул, та [51], написане 2016-го року.** **Справжнє джерело – це дисертація Слюняєва, яка була захищена 2010 року.** **Нахабний плагіат.**	Такі підмножини VOga j називатимемо модельною множиною вкладених IO і позначати ModelSet j. IO, що входять в j-ту модельну множину позначатимемо IOm j,i де нижній індекс т показує, що даний IO моделюється в процесі пошуку рішення. Для здійснення такого розбиття потрібна функція FMVO = h VO (Lvo,Lag,X), де X – онтологія, що описує логіку такого розбиття і необхідна тому, що потенційний а множина динамічних цілей може бути нескінченною. Формулюємо правило (8) $\Re_1\Re_2\langle lvo\rangle_3\Re_4\Re_5\Re_6\Re_7\Re_8\Re_9\langle Lag\neq\varnothing\rangle_{10}\Re_{11}\Rightarrow$ $\Rightarrow\Re_1\Re_2\langle h^A(Lvo,Lag,X)\rangle_3\Re_4\Re_5\Re_6\Re_7\Re_8\Re_9\langle Lag\neq\varnothing\rangle_{10}\Re_{11}\Rightarrow$ $\Rightarrow$ (9) $\Re_1\Re_2\langle ModelSet,Lvo\rangle_3\Re_4\Re_5\Re_6\Re_7\Re_8\Re_9\Re_{10}\Re_{11}$
	С. 98–99.	**С. 91.**
	Логічні об’єкти відповідно до своєї моделі поведінки формують ПЛО (пасивні логічні об’єкти) і передають їх як повідомлення; а також виконують перетворення hA над структурою своїх атрибутів. Тому в процесі функціонування ми моделюємо поведінку всіх ЛО, що належать даній модельній множині для пошуку шляху до мети gaj на глибину в gp кроків. Вважаємо, що параметр gp визначається вибраною стратегією si і вводимо два лічильники: лічильник модельних кроків Cmgp і лічильник актуальних кроків Cagp. Виконуємо модельні кроки для модельних множин на глибину gp. Після кожного кроку обчислюємо передбачувану оцінку стану МА-оточення для всіх модельованих ЛО і робимо наступний модельний крок. Повторюємо модельні кроки до досягнення Cmgp = gp. Далі обчислюємо передбачувану оцінку перетворень цілей для всіх ЛО і відхилення від цілей (міру близькості до мети). Визначаємо найменше відхилення $\Delta_{min}=\min(\Delta(IO_m^{J,1}) \mid IO_m^{J,1})\in ModelSet^J$	Інформаційні об’єкти відповідно до своєї моделі поведінки формують ПІО (пасивні інформаційні об’єкти) і передають їх як повідомлення; а також виконують перетворення hA над структурою своїх атрибутів. Тому в процесі пошуку рішення ми моделюємо поведінку всіх IO, що належать даній модельній множині для пошуку шляху до мети gaj на глибину в gp кроків. Вважаємо, що параметр gp визначається вибраною стратегією si і вводимо два лічильники: лічильник модельних кроків Cmgp і лічильник актуальних кроків Cagp. Виконуємо модельні кроки для модельних множин на глибину gp. Після кожного кроку обчислюємо передбачувану оцінку стану МА-оточення для всіх модельованих IO і робимо наступний модельний крок. Повторюємо модельні кроки до досягнення Cmgp = gp. Далі обчислюємо передбачувану оцінку перетворень цілей для всіх IO і відхилення від цілей (міру близькості до мети). Визначаємо найменше відхилення $\Delta_{min}=\min(\Delta(IO_m^{j,i}) \mid IO_m^{j,i})\in ModelSet^{j}$.
	С. 99–100.	**С. 91–92.**
	Ці дії формалізуємо правилами (9)	Ці дії формалізуємо правилами (9)

	(9) $\mathfrak{R}_1\mathfrak{R}_2(\text{LModelSet};\text{Lvo})_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2(\text{LModelSet};\text{Lvo})_3\mathfrak{R}_4\langle s';\text{Ls};\text{Cmgp}:=h^{gp}(s');\text{Cagp}:=h^{gp}(s')\rangle_5$ $\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2\left\langle\text{LModelSet}(K_{\text{МП}}(IO_m^{j,i})\mid IO_m^{j,i}\in\text{ModelSet}^j;j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\langle s';\text{Ls};\text{Cmgp}:=\text{Cmgp}-1;\text{Cagp}\rangle_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2$ $\left\langle\text{LModelSet}(K_{\text{МП}}(IO_m^{j,i}),FP(Rt,K_{\text{МП}}(IO_m^{j,i}))\mid IO_m^{j,i}\in\text{ModelSet}^j;j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2$ $\left\langle\text{LModelSet}(K_{\text{МП}}(K_{\text{МП}}(IO_m^{j,i}),Ri)\mid IO_m^{j,i}\in\text{ModelSet}^j;j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\langle s';\text{Ls};\text{Cmgp}:=\text{Cmgp}-1;\text{Cagp}\rangle_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow\dots\Rightarrow\dots\Rightarrow\dots\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2\left\langle\text{LModelSet}(K_{\text{МП}}(K_{\text{МП}}(IO_m^{j,i}),Ri)\mid IO_m^{j,i}\in\text{ModelSet}^j;j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\langle s';\text{Ls};\text{Cmgp}=0;\text{Cagp}\rangle_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow\mathfrak{R}_1\mathfrak{R}_2$ $\left\langle\text{LModelSet}(h^{ga^j}(K_{\text{МП}}(K_{\text{МП}}(IO_m^{j,i}),Ri))\mid IO_m^{j,i}\in\text{ModelSet}^j;j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2\left\langle\text{LModelSet}(\Delta(ga^j,ga^j)\mid IO_m^{j,i}\in\text{ModelSet}^j;j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow$(10) $\mathfrak{R}_1\mathfrak{R}_2\left\langle\text{LModelSet}(\Delta_{\min}:=\min(\Delta(ga^j,ga^j))\mid j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$	(9) $\mathfrak{R}_1\mathfrak{R}_2(\text{LModelSet};\text{Lvo})_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2(\text{LModelSet};\text{Lvo})_3\mathfrak{R}_4\langle s';\text{Ls};\text{Cmgp}:=h^{gp}(s');\text{Cagp}:=h^{gp}(s')\rangle_5$ $\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2\left\langle\text{LModelSet}(K_{\text{МП}}(IO_m^{j,i})\mid IO_m^{j,i}\in\text{ModelSet}^j;j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\langle s';\text{Ls};\text{Cmgp}:=\text{Cmgp}-1;\text{Cagp}\rangle_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2$ $\left\langle\text{LModelSet}(K_{\text{МП}}(IO_m^{j,i}),FP(Rt,K_{\text{МП}}(IO_m^{j,i}))\mid IO_m^{j,i}\in\text{ModelSet}^j;j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2$ $\left\langle\text{LModelSet}(K_{\text{МП}}(K_{\text{МП}}(IO_m^{j,i}),Ri)\mid IO_m^{j,i}\in\text{ModelSet}^j;j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\langle s';\text{Ls};\text{Cmgp}:=\text{Cmgp}-1;\text{Cagp}\rangle_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow\dots\Rightarrow\dots\Rightarrow\dots\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2\left\langle\text{LModelSet}(K_{\text{МП}}(K_{\text{МП}}(IO_m^{j,i}),Ri)\mid IO_m^{j,i}\in\text{ModelSet}^j;j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\langle s';\text{Ls};\text{Cmgp}=0;\text{Cagp}\rangle_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow\mathfrak{R}_1\mathfrak{R}_2$ $\left\langle\text{LModelSet}(h^{gp^j}(K_{\text{МП}}(K_{\text{МП}}(IO_m^{j,i}),Ri))\mid IO_m^{j,i}\in\text{ModelSet}^j;j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2\left\langle\text{LModelSet}(\Delta(ga^j,ga^j)\mid IO_m^{j,i}\in\text{ModelSet}^j;j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow$(10) $\mathfrak{R}_1\mathfrak{R}_2\left\langle\text{LModelSet}(\Delta_{\min}:=\min(\Delta(ga^1,ga^2))\mid j=1,\dots,n);\text{Lvo}\right\rangle_3$ $\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$
	С. 100.	**С. 92–93.**
	Після цього кроку слід сформувати множину актуальних ЛО, які діятимуть і посилатимуть повідомлення, таким чином, переходячи до фази активних дій і реалізації моделі ідентифікації та діагностування: (10) $\mathfrak{R}_1\mathfrak{R}_2\left\langle(\Delta^j_{\min}\mid j=1,\dots,n),\text{lvo}\right\rangle_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2\left\langle(IO_m^{1,1},\dots,IO_m^{1,\min},\dots,IO_m^{1,k_1})(IO_m^{2,1},\dots,IO_m^{2,\min},\dots,IO_m^{2,k_2})\dots\right.$ $\left.(IO_m^{n,1},\dots,IO_m^{n,\min},\dots,IO_m^{n,k_n});\text{lvo}\right\rangle_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow$(11) $\mathfrak{R}_1\mathfrak{R}_2\left\langle\text{lvo}:=h^R(\text{lvo})\right\rangle_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$	Після цього кроку слід сформувати множину актуальних ІО, які діятимуть і посилатимуть повідомлення, таким чином, переходячи до фази активних дій і реалізації моделі МА: (10) $\mathfrak{R}_1\mathfrak{R}_2\left\langle(\Delta^j_{\min}\mid j=1,\dots,n),\text{lvo}\right\rangle_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2\left\langle(IO_m^{1,1},\dots,IO_m^{1,\min},\dots,IO_m^{1,k_1})(IO_m^{2,1},\dots,IO_m^{2,\min},\dots,IO_m^{2,k_2})\dots\right.$ $\left.(IO_m^{n,1},\dots,IO_m^{n,\min},\dots,IO_m^{n,k_n});\text{lvo}\right\rangle_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow$(11) $\mathfrak{R}_1\mathfrak{R}_2\left\langle\text{lvo}:=h^R(\text{lvo})\right\rangle_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$
	С. 100–101.	**С. 93.**
	Зміст цього правила полягає в тому, що ЛО, відповідні ΔjMin позначаються як актуальні, а решта IOm j,i заміщають своїх батьків в списку Lvo за допомогою перетворення зворотньої підстановки hR. Вміст компонента R3 приймає вигляд IO1,..., IOia,..., IOja,..., IOl, де l – потужність множини вкладених логічних об’єктів {VO}. Далі включаємо в цю множину і виділений логічний об’єкт IO 0, що представляє дії ІА над своїми атрибутами, який активізується за умовчанням. Описуємо фазу активних дій наступним правилом: (11) $\mathfrak{R}_1\mathfrak{R}_2\left\langle IO^a_0,\dots,IO^a_j,\dots\right\rangle_3\mathfrak{R}_4\langle s';\text{Ls};\text{Cagp}\rangle_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2\left\langle K_{\text{МП}}(IO^a_0),\dots,K_{\text{МП}}(IO^a_j),\dots\right\rangle_3\mathfrak{R}_4\langle s';\text{Ls};\text{Cagp}:=\text{Cagp}-1\rangle_5$ $\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow\dots\Rightarrow\dots\Rightarrow\dots\Rightarrow$ $\Rightarrow$(12) $\mathfrak{R}_1\mathfrak{R}_2\left\langle K_{\text{МП}}(K_{\text{МП}}\dots(IO^a_0)),\dots,K_{\text{МП}}(K_{\text{МП}}\dots(IO^a_j)),\dots\right\rangle_3\mathfrak{R}_4$ $\langle s';\text{Ls};\text{Cagp}=0\rangle_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$	Зміст цього правила полягає в тому, що ІО, відповідні Δjmin позначаються як актуальні, а решта IOm j,i заміщають своїх батьків в списку Lvo за допомогою перетворення зворотньої підстановки hR. Вміст компонента R3 приймає вигляд IO1,..., IOia,..., IOja,..., IOl, де l – потужність множини вкладених інформаційних об’єктів {VO}. Далі включаємо в цю множину і виділений інформаційний об’єкт IO0 , що представляє дії ІА над своїми атрибутами, який активізується за умовчанням. Описуємо фазу активних дій наступним правилом: (11) $\mathfrak{R}_1\mathfrak{R}_2\left\langle IO^a_0,\dots,IO^a_j,\dots\right\rangle_3\mathfrak{R}_4\langle s';\text{Ls};\text{Cagp}\rangle_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow$ $\Rightarrow\mathfrak{R}_1\mathfrak{R}_2\left\langle K_{\text{МП}}(IO^a_0),\dots,K_{\text{МП}}(IO^a_j),\dots\right\rangle_3\mathfrak{R}_4\langle s';\text{Ls};\text{Cagp}:=\text{Cagp}-1\rangle_5$ $\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}\Rightarrow\dots\Rightarrow\dots\Rightarrow\dots\Rightarrow$ $\Rightarrow$(12) $\mathfrak{R}_1\mathfrak{R}_2\left\langle K_{\text{МП}}(K_{\text{МП}}\dots(IO^a_0)),\dots,K_{\text{МП}}(K_{\text{МП}}\dots(IO^a_j)),\dots\right\rangle_3\mathfrak{R}_4$ $\langle s';\text{Ls};\text{Cagp}=0\rangle_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$
	С. 101–102.	**С. 93–94.**
	Після того, як активні дії закінчилися, можна виконати перевірку і зняти мітки активності з ЛО:	Після того, як активні дії закінчилися, можна виконати перевірку і зняти мітки активності з ІО:

	$(12)\ \langle Lia,Lio,Laia,Laio\rangle_1 \mathfrak{R}_2 \langle IO_0^a,\ldots,IO_j^a,\ldots\rangle_3 \mathfrak{R}_4 \langle s';Ls;Cagp=0\rangle_5$ $\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow (13)\ [\langle Lia',Lio',Laia',Laio' := FV(Lia,Lio,Laia,Laio)\rangle_1$ $\mathfrak{R}_2 \langle (IO_0,\ldots,IO_j,\ldots\rangle_3 \mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$ Формуємо новий поточний стан МІД-оточення за допомогою правил (13) і (14) $(13)\ \langle Lia',Lio',Laia',Laio'\rangle_1 \mathfrak{R}_2\mathfrak{R}_3 \langle Pa;Rt;Fu\rangle_4 \langle s';Ls\rangle_5$ $\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow (14)\ \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3$ $\langle Rt,Pa;Lia',Lio',Laia',Laio';Fu := FP(Lia',Lio',Laia',Laio')\rangle_4$ $\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$ $(14)\ \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle Pa;Rt;Fu\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \left\langle (g\hat{a}^j := h^{ga_j}(Pa,Rt,Fu) \mid j=1,\ldots,n);Pa;Rt;Fu\right\rangle_4$ $\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow (15)\ \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \left\langle \Delta(ga^j,g\hat{a}^j) \mid j=1,\ldots,n);Pa;Rt;Fu\right\rangle_4$ $\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$	$(12)\ \langle Lia,Lio,Laia,Laio\rangle_1 \mathfrak{R}_2 \langle IO_0^a,\ldots,IO_j^a,\ldots\rangle_3 \mathfrak{R}_4 \langle s';Ls;Cagp=0\rangle_5$ $\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow (13)\ [\langle Lia',Lio',Laia',Laio' := FV(Lia,Lio,Laia,Laio)\rangle_1$ $\mathfrak{R}_2 \langle (IO_0,\ldots,IO_j,\ldots\rangle_3 \mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$ Формуємо новий поточний стан МА-оточення за допомогою правил (13) і (14) $(13)\ \langle Lia',Lio',Laia',Laio'\rangle_1 \mathfrak{R}_2\mathfrak{R}_3 \langle Pa;Rt;Fu\rangle_4 \langle s';Ls\rangle_5$ $\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow (14)\ \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3$ $\langle Rt,Pa;Lia',Lio',Laia',Laio';Fu := FP(Lia',Lio',Laia',Laio')\rangle_4$ $\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$ $(14)\ \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle Pa;Rt;Fu\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \left\langle (g\hat{a}^j := h^{ga_j}(Pa,Rt,Fu) \mid j=1,\ldots,n);Pa;Rt;Fu\right\rangle_4$ $\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow (15)\ \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \left\langle \Delta(ga^j,g\hat{a}^j) \mid j=1,\ldots,n);Pa;Rt;Fu\right\rangle_4$ $\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$
	С. 102.	**С. 94–95.**
	Обчислюємо оцінки стану цілей від досягнутого стану МІД-оточення і визначаємо відхилення від мети [18]. Оскільки цілей може бути декілька (у разі j > 1), необхідно сформувати комплексну оцінку положення МІД в просторі цілей $\Delta_k = \sum_{j=1}^{n} a_i \Delta(ga^j, g\hat{a}^j)$ що враховує значущість цілей, ai – вагові коефіцієнти важливості цілей, n – потужність множини активних цілей. Тому вводимо правило (15) $(15)\ \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \left\langle \Delta(ga^j,g\hat{a}^j) \mid j=1,\ldots,n);Pa;Rt;Fu\right\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \nabla\mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle \Delta k;Pa;Rt;Fu\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$	Обчислюємо оцінки стану цілей від досягнутого стану МА-оточення і визначаємо відхилення від мети. Оскільки цілей може бути декілька (у разі j > 1), необхідно сформувати комплексну оцінку положення ІА в просторі цілей $\Delta_k = \sum_{j=1}^{n} a_i \Delta(ga^j, g\hat{a}^j),$ що враховує значущість цілей, ai – вагові коефіцієнти важливості цілей, n –потужність множини активних цілей. Тому вводимо правило (15) $(15)\ \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \left\langle \Delta(ga^j,g\hat{a}^j) \mid j=1,\ldots,n);Pa;Rt;Fu\right\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \nabla\mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle \Delta k;Pa;Rt;Fu\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$
	С. 102.	**С. 95.**
	Проводимо порівняння з допустимим комплексним відхиленням Δd за допомогою наступних правил: $(16)\ \nabla\mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle \Delta k > \Delta d;Pa;Rt;Fu\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow (4)\ \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle Pa;Rt;Fu\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$ $(17)\ \nabla\mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle \Delta k \le \Delta d;Pa;Rt;Fu\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow (5)\ \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle Pa;Rt;Fu\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$	Проводимо порівняння з допустимим комплексним відхиленням Δd за допомогою наступних правил: $(16)\ \nabla\mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle \Delta k > \Delta d;Pa;Rt;Fu\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow (4)\ \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle Pa;Rt;Fu\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$ $(17)\ \nabla\mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle \Delta k \le \Delta d;Pa;Rt;Fu\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow (5)\ \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle Pa;Rt;Fu\rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$
	С. 103.	**С. 95.**
	При виконанні умови Δkd> Δd переходимо до правила (4), тобто переоцінюємо стратегію формування цілей; якщо ж Δk≤Δd то функціонування МІД продовжується у тій же самій стратегії. Побудована в даному розділі модель функціонування в узагальненій моделі ідентифікації та діагностування (ІД) дозволяє описати такі відомі класи моделей реалізації поведінки як моделі із зумовленою кінцевою множиною елементарних дій; моделі з множиною планів; моделі з довільними повідомленнями і діями. На базі даної моделі можуть створюватися нові моделі реалізації поведінки МІД, що поєднують механізми різних класів.	При виконанні умови Δk > Δd переходимо до правила (4), тобто переоцінюємо стратегію формування цілей; якщо ж Δk ≤ Δd то функціонування ІА продовжується у тій же самій стратегії. Побудована в даному розділі модель пошуку рішення в узагальненій моделі ІА дозволяє описати такі відомі класи моделей реалізації поведінки як моделі із зумовленою кінцевою множиною елементарних дій; моделі з множиною планів; моделі з довільними повідомленнями і діями. На базі даної моделі можуть створюватися нові моделі реалізації поведінки ІА, що поєднують механізми різних класів.