Міністерство освіти і науки України

Українська академія друкарства

На правах рукопису

Огірко

Ольга Ігорівна

УДК 004.03:624.15

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ ВИЗНАЧЕННЯ ЕНЕРГЕТИЧНИХ ХАРАКТЕРИСТИК ПОВЕРХНЕВИХ ШАРІВ МЕТАЛІВ

Спеціальність 05.13.06 – інформаційні технології

Дисертація
на здобуття наукового ступеня
кандидата технічних наук

Науковий керівник
Юзевич Володимир Миколайович,
доктор фізико-математичних наук, професор

Львів– 2013

ЗМІСТ

Перелік умовних позначень, , символів, скорочень і термінів …… 4

ВСТУП 10

Розділ І. Огляд і аналіз основ інформаційних технологій для визначення енергетичних характеристик поверхневих шарів металів з використанням алгебри алгоритмів …… 22

1.1. Співвідношення термодинамічної моделі поверхневого шару матеріалу як основа ІТ …….. 22

1.2. Методи подання алгоритмів..... 27

1.3. Подання алгоритмів засобами теорії секвенційних алгоритмів…... 35

1.4. Методи верифікації програм 38

1.5. Висновки до першого розділу 45

Розділ ІІ. ІТ СИНТЕЗУ І МІНІМІЗАЦІЇ СИСТЕМИ ГЕНЕРАЦІЇ ПРОГРАМ З ФОРМУЛ ТЕОРІЇ СЕКВЕНЦІЙНИХ АЛГОРИТМІВ 46

2.1. Принцип побудови системи генерації програм з операцій теорії секвенційних алгоритмів …… 47

2.2. Синтез і мінімізація загальної структури системи …… 51

2.3. Структура генерування програм з операції елімінування …… 57

2.4. Структура синтаксису алгоритму 59

2.5. Висновки до другого розділу..... 61

Розділ ІІІ. МОДЕЛЮВАННЯ СИСТЕМИ ГЕНЕРАЦІЇ ПРОГРАМ З ФОРМУЛ ТЕОРІЇ СЕКВЕНЦІЙНИХ АЛГОРИТМІВ …… 62

3.1. Побудова і дослідження моделі генерації програм з операцій секвентування 65

3.2. Блок-схема системи генерації програм з формул теорії секвенційних алгоритмів …… 72

3.3. Методика роботи з системою 76

3.4. Висновки до третього розділу 79

Розділ IV РЕАЛІЗАЦІЯ ІТ ВИЗНАЧЕННЯ ЕНЕРГЕТИЧНИХ ХАРАКТЕРИСТИК ПОВЕРХНЕВИХ ШАРІВ МЕТАЛІВ ……. 80

4.1. Фізико-математична модель визначення енергетичних характеристик міжфазних шарів металів для аналізу корозійних процесів – приклад реалізації ІТ ……. 80

4.2. Фізико-математична модель активації деформованого металу поблизу вершини корозійної тріщини …… 88

4.3. Застосування алгоритму інформаційної технології для антикорозійного захисту …… 96

4.4. Висновки до четвертого розділу …….. 108

ОСНОВНІ РЕЗУЛЬТАТИ ТА ВИСНОВКИ …….. 110

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ ……... 114

ДОДАТКИ ……. 127

Перелік умовних позначень, символів, скорочень і термінів

АІС – автоматизована інформаційна система;

СУБД – система управління базами даних;

ISO – система міжнародних стандартів

СА – системний аналіз

ЕОМ – електронно-обчислювальна машина

ПЗ – програмне забезпечення

ПН – показники надійності

СКВ – Середнє квадратичне відхилення

ТЗ – Технічний засіб

КС – комп’ютерна система.

АС –автоматизована система

ТЗІ – технічний захист інформації

ЦКС – цифрова керуюча система

СОД – система оброблення даних

ІТ – інформаційна технологія

ІТН – інформаційні технології навчання

ПК – персональний комп’ютер

КС – комп’ютерна система

ІКТ – інформаційно-комунікаційні технології

ОП – оперативна пам’ять

СОО – система оперативної обробки

ЦКС – цифрова керуюча система

САА – система алгоритмічних алгебр

ПЕ – поверхнева енергія

σ_h – поверхневий натяг

γ – поверхнева енергія (ПЕ)

γ_1 – складова ПЕ, яка відповідає вільним (зв’язаним) електричних зарядам (так названа електрична складова)

γ_2 – механічна складова ПЕ

γ_e – електрична складова міжфазної енергії

γ_p – механічна складова міжфазної енергії

w_1, w_2 – густини густини електричної та механічної складових ПЕ

h – ефективна товщина поверхневого шару

$2H$ – ефективна товщина міжфазного шару

$d_\xi \approx 2H$– ефективна товщина подвійного електричного шару

Z_b – ширина (зміщення) подвійного електричного шару на поверхні металу

E_f – енергія Фермі

E_V – робота виходу електрона з поверхні металу

M_{at}, – атомна маса металу

M_{mol} – молярна маса речовини (металу)

pH – водневий показник розчину електроліту

k_f – хвильовий вектор Фермі

σ_{ij} , e_{ij} – компоненти тензорів напружень $\hat{\sigma}$ і деформацій $\hat{e}$ (i, j = 1, 2, 3)

$\sigma_{11} = \sigma_x$; $\sigma_{22} = \sigma_y$ – компоненти тензора механічних напружень

$\sigma_{\theta\theta}$ – компонента тензора механічних напружень у циліндричних координатах

D=2R_t, H_t – діаметр і товщина труби відповідно

d=2a, b – діаметр та глибина заглибини (поверхневої тріщини)

μ_c, c – хімічний потенціал і концентрація домішки

ξ, ξ_m, k, α_t, β, β_c, C_φ, γ_t, η_c, d_c, d_t , z_m – фізичні характеристики матеріалу (металу)

δ_{ij} – символи Кронекера

e – перший інваріант тензора деформацій

ρ – питома густина матеріалу (металу)

ω_v, ω – просторова і масова густини зв’язаних електричних зарядів відповідно

$\varphi = \Phi - \Phi_0$ – відхилення модифікованого потенціалу Φ зв’язаних електричних зарядів від його рівноважного значення Φ_0 в об’ємі тіла далеко від поверхні

Ψ – скалярний потенціал напруженості електричного поля

$\psi = \Psi - \Psi_0$ – відхилення потенціалу Ψ від його рівноважного значення Ψ_0

$\vec{E}_{el} = \nabla\Psi = grad\Psi$ – напруженість електричного поля

ε_0 – електрична постійна (ε_0 = 8,85×10^{-12} Ф/м)

$\Delta T = T - T_0$ – зміна температури

K, G – модулі всестороннього стиску і зсуву відповідно

$$K = \frac{E}{3(1-2\nu)}, \quad G = \frac{E}{2(1+\nu)}$$

E – модуль Юнга

ν – коефіцієнт Пуассона

ξ, k, α_t, β, β_c, C_φ, γ_t, η_c, d_c, d_t – фізичні характеристики матеріалу

σ_m – міжфазний натяг

γ_m – міжфазна енергія

ρ_m – усереднена густина матеріалу (металу) в області подвійного електричного шару

A_{ad} – робота адгезії

γ_{ad} – енергія адгезійних зв'язків

I_{kg} – швидкість корозії металу (густина електричного струму)

I_k – густина корозійного струму

$\overline{E}_{cor}$ – корозійний потенціал

$\overline{E}_p$ – поляризаційний потенціал

N_+, N_- – число частинок (атомів, молекул) в одиниці об'єму речовини

b_{at} – тафелевський нахил поляризаційної кривої

$\Delta\Psi_m$ – стрибок електричного потенціалу в подвійному електричному шарі (електричний бар'єр на границі «метал – розчин електроліту»)

E_k , E_a – електродні потенціали відповідно катодної й анодної ділянок

R_k, R_a – поляризаційні опори катодної і анодної ділянок

R_C – електричний опір середовища (зовнішній)

R_M – електричний опір металу (внутрішній опір корозійної пари)

C_m – ємність міжфазного електричного конденсатора

$C_{\varphi m}$ – питома електроємність подвійного електричного шару (в околі межі «метал–покриття»)

E_p^{MeO} – рівноважний потенціал утворення окислу на поверхні металу

P_{PL} – робота пластичної деформації при утворенні нової поверхні поблизу вершини тріщини

W_{zp} – зміна початкового запасу енергії в околі вершини тріщини

W_{ec} – виділення енергії в процесі електрохімічної реакції в околі вершини тріщини

K_1 — коефіцієнт інтенсивності напружень

K_{1SCC} – мінімальне значення K_1, що відповідає початку поширення корозійної тріщини

z_{si} – формальний заряд сольватованих іонів

F = 96500 C/gr-mol – постійна Фарадея

δ – висота фронту тріщини, що насувається

M – молекулярна маса металу

ζ – перенапруження анодної реакції

J_d – густина потоку дислокацій (дислокаційний струм)

I_C – густина електричного струму при корозії в околі вершини тріщини

A – узагальнена сила (хімічна спорідненість процесів створення і руху дислокацій)

L_A – феноменологічний коефіцієнт, що характеризує провідність електроду

L_d – феноменологічний коефіцієнт, що характеризує поляризованість електроду

σ – перший інваріант тензора механічних напружень

$P_j(\sigma)$ – множина параметрів, що характеризують напружено-деформований стан матеріалу і є функціями прикладених до тіла зовнішніх зусиль, яким відповідають напруження σ

$L_n(\tau)$ – множина параметрів, які визначають у часі τ фізико-хімічні процеси, що проходять між деформованим металом і корозійним середовищем

$B_m(s)$ – множина параметрів, що характеризують стан поверхонь матеріалу «s», які є до руйнування і утворюються під час руйнування металу в процесі корозії

a_t – характерний лінійний розмір тріщини в напрямку її поширення

C_i – множина констант (фізичних характеристик), які відповідають системі «матеріал (метал) – корозійне середовище»

σ_* , σ_{**} – критичні механічні напруження

σ_T – границя текучості матеріалу (металу)

d_* і d_{**} – характерні розміри пластичних зон для плоскої деформації і плоского напруженого стану відповідно в околі вершини тріщини

σ_i – інтенсивність механічних напружень

$\sigma_\tau = \tau_{\max}$ – максимальне дотичне механічн напруження

k – постійна Больцмана

U_0 характеризує енергію активації розриву міжатомних зв'язків і практично співпадає з енергією дисоціації речовини (металу)

$\gamma_d = q \cdot V_a$, де V_a – активаційний об'єм у елементарному акті дисоціації

q – коефіцієнт (безрозмірний), характеризує локальні механічні перенапруження

W_{subl} – енергія сублімації металу

k_1 , k_2 – константи швидкостей анодної та катодної реакцій

C – концентрація (активність) іонів металу (окислювача) в приелектродному шарі електроліту

w_a , w_k – енергії активації анодної та катодної реакцій

R = 8,31 J/(g-mol·K) – універсальна газова стала

α , $(1-\alpha)$ – коефіцієнти переносу частинок через подвійний електричний шар

z – валентність іона

i_p – густина катодного поляризуючого струму

g – коефіцієнта гальмування корозії

P – ступінь захисту металу від корозії

Pz – набір параметрів, які характеризують міжфазні (поверхневі) шари

Nz – набір параметрів, які характеризують фізико-хімічні процеси в дефектах (тріщинах, пітингах)

T – температура

grad$T$ – градієнт температури

$A(L,G,S)$ – корозійні втрати (L, G, S – символічне позначення рідини (liquid L), газу (gas G), твердого тіла (solid S))

Rz – набір параметрів, які характеризують корозійні струми

Ks – символічне позначення корозійної стійкості

γ_{mo} – міжфазна енергія ненавантаженого зразка

W – коефіцієнт конкордації

D_f – фактична дисперсія (середньоквадратичне відхилення) підсумкових (впорядкованих, проранжованих) оцінок, які надані експертами

D_{max} – дисперсія підсумкових (впорядкованих) оцінок за умови, що думки експертів повністю збігаються

a_i – сумарна експертна оцінка, одержана для i-го об'єкту

m – кількість коефіцієнтів вагомості

n – кількість досліджуваних об'єктів (експертів досліджуваних об'єктів

χ^2 – вираз для критерію Пірсона (хі-квадрат)

W_K – коефіцієнт конкордації Кендела

m_k – число аналізованих змінних (коефіцієнтів вагомості)

S – сума квадратів рангів для обчислення коефіцієнта конкордації Кендела

В С Т У П

Актуальність теми. Важливим науковим напрямком інформаційних технологій є розвиток на засадах теорії секвенційних алгоритмів алгоритмічних підходів у різних сферах науки і техніки. Зокрема, такі підходи можуть бути корисними для вивчення особливостей міжфазних ефектів. Відповідні результати доцільно впроваджувати в методиках визначення енергетичних характеристик поверхневих шарів з метою створення інформаційного продукту (алгоритмів) нової якості. З допомогою такого типу алгоритмів на основі залучення масивів параметрів та фізико-хімічних характеристик можна вивчати приховані закономірності корозійних процесів і приймати оптимальні практичні рішення, використовуючи сучасні програмні засоби.

Головною передумовою для реалізації відповідних задач є модифікація математичних моделей систем електрохімзахисту з використанням фундаментальних підходів до розрахунку електрохімічних полів значної складності, розробка науково обґрунтованих методик дослідження поточного стану систем протикорозійного захисту (ізоляційних покриттів в елементах поліграфічних машин), а також використання сучасних методів ідентифікації параметрів математичних моделей.

Дослідження, що проведені в даній праці, спрямовані на автоматизацію задач обробки даних корозійних обстежень та діагностування елементів машин на основі розробки математичних моделей і методів протикорозійного захисту, а також створення інформаційно-аналітичної моделі технологічного комплексу.

Недостатньо розроблені інформаційні технології для електрохімічного захисту від корозії поверхонь конструктивних елементів деталей, в тому числі це стосується неізольованих частин поліграфічних устаткувань, що експлуатуються в корозійно-агресивних електролітах. В рамках господарської проблеми відомо, що відсутня кореляція між корозійною тривкістю металу в ненапруженому стані та границею корозійної втоми в різних середовищах. Фізичними і електрохімічними методами коректно усунути такі

невідповідності проблематично і складно, бо система (метал – електроліт) є в умовах невизначеності та неповноти даних про стан системи. Система змінюється в часі і потоки інформації про стан системи (метал – електроліт) змінюються також. Такими потоками інформації про стан системи потрібно управляти, щоб захистити елементи конструкцій від корозії. Традиційні математичні моделі недостатньо адаптовані до такого управління, бо не враховують стан параметрів на границі розділу (метал – електроліт). Шляхи (алгоритми) – як впорядкувати потоки інформації про стан міжфазних шарів в елементах конструкцій є невідомими, оскільки стан системи «метал – електроліт» постійно змінюється, і на основі вивчення взаємозв’язку потоків інформації потрібно з'ясувати і дати рекомендації про протикорозійний захист.

Використовуючи термодинамічні моделі фізики поверхні, зменшимо ступінь невизначеності інформаційних потоків, які характеризують стан міжфазних шарів металів в агресивних середовищах і фізико-хімічні процеси, що в них відбуваються. Застосовуючи алгоритми теорії секвенційних алгоритмів, можна впорядкувати і фільтрувати потоки інформації. З урахуванням цього актуальною є розробка інформаційної технології визначення параметрів поверхневих шарів конструкційних матеріалів, які знаходяться в агресивних середовищах, з урахуванням моделювання корозійних процесів на основі підходів нерівноважної термодинаміки, фізики поверхневих явищ, теорії планування експерименту, а також застосування апарату підсистеми генерації програм з операцій теорії секвенційних алгоритмів.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконувалась в рамках держбюджетної теми "Модель і програма комп'ютерної системи генерування програм з формул алгоритмів" за номером державної реєстрації 0109Ш01214 на кафедрі АКТ Української академії друкарства (м. Львів) з січня 2009 року до грудня 2011 року. Робота виконана згідно плану науково-технічних робіт, де здобувач працювала як виконавець і розробила функціональну структуру інформаційної технології,

математичну модель і програму системи генерації програм з операцій теорії секвенційних алгоритмів.

Мета роботи – розробка інформаційної технології впорядкування потоків інформації про стан міжфазних шарів в елементах металевих конструкцій поліграфічних машин для забезпечення ефективності протикорозійного захисту.

Для досягнення поставленої мети в дисертаційній роботі вирішуються такі **задачі**:

1. Проаналізувати наявні інформаційні технології і математичні моделі корозіометрії і сконцентрувати увагу на методах оцінювання енергетичних характеристик поверхневих шарів металів. Інформаційне забезпечення такого типу моделей недостатньо розроблено, бо не для всіх характеристик досліджуваних матеріалів розроблено алгоритми і методики розрахунку, а також методи відбору даних.

2. Розробити функціональну структуру інформаційноі технології і інформаційно-аналітичну модель технологічного комплексу протикорозійного захисту.

3. Розробити та дослідити математичну модель системи генерації програм з операцій теорії секвенційних алгоритмів, утворену такими моделями системи генерації програм з операцією секвентування як системи генерації програм з операцією елімінування, системи генерації програм з операцією циклу.

4. Розробити інформаційну технологію визначення параметрів поверхневих шарів конструкційних матеріалів, які знаходяться в агресивних середовищах, з урахуванням моделювання корозійних процесів на основі підходів нерівноважної термодинаміки, фізики поверхневих явищ, теорії планування експерименту, а також застосування апарату підсистеми генерації програм з операцій теорії секвенційних алгоритмів.

5. Використати методику визначення параметрів поверхневих шарів конструкційних матеріалів для удосконалення інформаційної технології протикорозійного захисту елементів конструкцій на підставі використання

відомостей про енергетичні характеристики міжфазних шарів на границі метал – корозійне середовище.

Об'єкт дослідження – процеси інформаційного забезпечення протикорозійного захисту елементів металевих конструкцій.

Предмет дослідження – методи, фізико-математичні моделі та інформаційна технологія для протикорозійного захисту.

Методи дослідження – системний аналіз, синтез, формалізації, моделювання, аналогії, вимірювання, логічний метод, що базуються на принципах інформаційної технології розрахунку та ідентифікації механоелектротермодифузійних параметрів стану елементів конструкцій з дефектами в агресивних середовищах, та алгебрі алгоритмів.

Наукова новизна одержаних результатів.

1. Створено математичну модель, яка акумулює великий обсяг інформації, і запропоновано відповідну концептуальну схему розробки інформаційної технології для аналізу інформаційних потоків в сфері електрохімії та термодинаміки нерівноважних процесів.

2. Вперше для інформаційної технології діагностування корозії поєднано математичні моделі для аналізу параметрів стану та процесів, що характеризують умови розчинення поверхонь зразків сталей різних марок, а також умови адгезії покрить елементів конструкцій.

3. Вперше в інформаційній технології розроблено і доведено до результативного варіанту модель відбору, опрацювання і використання інформації про фізико-хімічні параметри контактуючих середовищ в задачі прийняття рішень щодо оптимального варіанту протикорозійного захисту металів. Відповідні результати дослідження і програму системи генерації програм з операцій теорії секвенційних алгоритмів впроваджено у виробництво і вони використовуються в інформаційній технології, що дозволяє на основі аналізу енергетичних характеристик поверхневих та міжфазних шарів металів приймати рішення щодо покращення умов протикорозійного захисту металів в агресивному електролітичному середовищі.

4. Розроблено концептуальну модель комп’ютерної інформаційної технології, яка передбачає використання фізико-математичних моделей для обчислення та аналізу енергетичних характеристик поверхневих шарів металів в агресивному середовищі. Варіанти моделі орієнтовані на опрацювання великого обсягу вхідних даних та прийняття рішень щодо антикорозійного захисту елементів конструкцій. Модель забезпечує зниження витрат на проведення протикорозійного захисту й ефективніша за наявні.

5. Вперше в запропонованій інформаційній технології поєднано математичну модель визначення енергетичних характеристик поверхневих шарів металів в корозійних середовищах з методами теорії сенквенційних алгоритмів, які забезпечують формалізований опис і тотожні перетворення алгоритмів з метою мінімізації виконання операцій. Таке поєднання забезпечує основу створення програмних комплексів нового типу та більш ефективне використання результатів інформаційної технології. Синтезовано та мінімізовано за кількістю термів структуру системи генерації програм з операцій теорії секвенційних алгоритмів, яка включає операції секвентування, елімінування, циклів і це дає можливість зменшити похибки, точніше проводити операції обчислень енергетичних характеристик поверхневих шарів металів, які знаходяться в агресивному середовищі.

6. Вперше розроблено концепцію і спосіб побудови автоматизованого програмного комплексу, в якому на відміну від відомих поєднано математичні засоби опису фізико-механічних полів, еквівалентних перетворень інформаційних потоків та дослідження математичних моделей корозіометрії з відповідними алгоритмами.

7. Удосконалено функціональну структуру інформаційних потоків ІТ для протикорозійного захисту з використанням даних про енергетичні характеристики поверхневих і міжфазних шарів та активаційних процесів, які характеризують метал, середовище, динаміку корозійних процесів поблизу вершини тріщини в корозійному середовищі, дають можливість керувати

масивами інформації і на основі алгебри алгоритмів приймати оптимальні рішення щодо проведення експериментальних досліджень.

8. Розроблена інформаційна технологія для протикорозійного захисту реалізована у вигляді модуля програмного комплексу та пакету процедур. Програмний комплекс, в основі якого метод розкладу параметрів стану за малим параметром, дає можливість знизити витрати на проведення важливих корозійних обстежень за рахунок адекватної математичної моделі та алгоритмів обробки даних, що дозволяє зменшити обсяги експериментальних вимірювань.

9. Запропоновано математичну модель для діагностування параметрів приграничних шарів елементів конструкцій, поверхня яких покрита захисними плівками і в цих плівках можуть бути дефекти типу тріщин. Оскільки числові розрахунки в цьому випадку складні й громіздкі, то для їх вдосконалення вперше використано методи і засоби автоматизації математичних обчислень на основі алгебри алгоритмів. Модель з комплексом алгоритмів за рахунок використання розширеного набору параметрів стану забезпечує зниження витрат на проведення протикорозійного захисту.

10. Використовуючи можливості алгебри алгоритмів та єдиного підходу інформаційної технології щодо написання алгоритмів стосовно конкретних задачах корозіометрії, проведено моделювання та дослідження вірогідності отриманих результатів. З допомогою засобів програмування розроблено комп’ютерну програму, що дає можливість ефективніше реалізовувати математичні обчислення, які забезпечують достовірні результати аналізу інформаційних потоків для корозіометрії. Програмною реалізацією і апробацією підтверджено достовірність моделі системи генерації програм з операціями теорії секвенційних алгоритмів та коректне функціонування відповідної програми.

11. Набула подальшого розвитку математична модель системи генерації програм з операцій теорії секвенційних алгоритмів, утворена моделями системи генерації програм з операціями секвентування, елімінування та циклу.

Система генерації програм з операцій теорії секвенційних алгоритмів вперше застосована для визначення енергетичних характеристик поверхневих шарів металів.

Практичне значення отриманих результатів. Запропоновані методи, моделі та інформаційна технологія є теоретичною базою практичних аспектів аналізу корозійних процесів для оцінювання станів конструкційних матеріалів, які знаходяться в агресивному середовищі. Розроблена інформаційна технологія для протикорозійного захисту дає можливість знизити витрати на моніторинг корозійного стану за рахунок раціонального планування експерименту. Результати дослідження і програму системи генерацію програм з операцій теорії секвенційних алгоритмів впровадженно в інформаційній технології для корозіометрії, що дозволяє на основі аналізу енергетичних характеристик поверхневих та міжфазних шарів металів приймати рішення щодо покращення умов протикорозійного захисту металів в агресивному електролітичному середовищі. Акт впровадження у ТзОВ НВП «ГАРАС і Л» від 11.02.2013 р. Результати досліджень використовуються і в навчальному процесі підготовки студентів ВНЗ (акти впровадження в дисертації додаються).

Особистий внесок здобувача. Основні результати теоретичних досліджень, які містяться в дисертації, отримані автором особисто. Всі результати дисертації отримані автором самостійно. У роботах [1,65] дисертантові належать інформаційна технологія оцінювання станів об’єктів, реалізація процедури автоматизації математичних обчислень для оцінювання параметрів поверхневих шарів, [14,58,69,70] – використання прогресивних інформаційних технологій. У працях [102,110] автором розроблені моделі, інформаційна технологія, програмний комплекс для збору та обробки даних і доведено до програмної реалізації об'єктно-орієнтоване ядро бази даних автоматизованої системи.

Апробація результатів дисертації. Основні результати дисертації доповідались та обговорювались на: 7 Всеукраїнській науковій конференції “Сучасні проблеми прикладної математики та інформатики” – Львів: ЛНУ ім.

Ів. Франка, 2000 р.; міжнародній конференції «Автоматика-2000», Державний науково-дослідний інститут інформаційної інфраструктури (Львів, 2000); міжнародних конференціях «Комп'ютерні технології друкарства» (Львів, 2000, 2004 рр.); міжнародній науково-практичній конференції "Кваліологія книги" (Львів, 2003); III Всеукраїнській науково-практичній конференції "Сучасні інформаційні технології в економіці, менеджменті та освіті", (Львів, 2012 р.); науково-практичній конференції «Мультимедійні технології в освіті та інших сферах діяльності», (Київ, НАУ, 2012); науково-технічних конференціях, засіданнях, семінарах професорсько-викладацького складу, наукових працівників і аспірантів Української академії друкарства (Львів, 2003, 2006, 2013 рр.).

Публікації. За результатами наукових досліджень по темі дисертаційної роботи опубліковано 16 наукових праць. Із них 12 у фахових виданнях.

Структура та обсяг роботи. Дисертаційна робота складається із вступу, чотирьох розділів і висновків, викладених на 163 сторінках з урахуванням 12 рисунків, 7 таблиць, списку використаної літератури із 124 найменувань і 2 додатків.

Структура та обсяг роботи. Дисертаційна робота складається із вступу, чотирьох розділів і висновків. Повний обсяг дисертації становить 163 сторінки. Робота містить 12 рисунків, 7 таблиць, списку використаної літератури із 124 найменувань і 2 додатки.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У вступі обґрунтовано актуальність роботи, сформульовано мету, визначено основні задачі та методи дослідження, показано наукову новизну і практичну цінність наукових результатів. Висвітлено питання реалізації, апробації та впровадження наукових результатів. Стисло подано предмет і об'єкт досліджень, обґрунтовано актуальність дисертаційної роботи, відзначена наукова новизна і практична цінність отриманих результатів, приведені дані про їхнє використання. Наведено дані про апробацію одержаних результатів, публікації, структуру та обсяг дисертації.

У ***першому*** розділі здійснено огляд основних інформаційно-технологічних етапів. Вагомий внесок щодо розвитку підходів, методів та моделей та інформаційних технологій для поліграфії зробили такі вчені як Афанасьєва О. Ю., Гавенко С. Ф., Дурняк Б. В., Луцків М. М., Овсяк В. К., Сеньківський В. М., Сікора Л.С., Тимченко О. В., Чехман Я. І., Юзевич В. М., Назаркевич М. А. ,Шовгенюк М. В.та інші. Великий внесок у розробку теоретичних і прикладних питань моделювання процесів протикорозійного захисту зробили Панасюк В. В., Похмурський В. І., Джала Р. М., Поляков С. Г., Сопрунюк П.М., Остапенко В.Н., Лукович В. В., Скляров С. О. та інші. Зроблені наступні висновки:
1) системи протикорозійного захисту є досить складними технологічними об'єктами, експлуатація яких пов'язана з великою відповідальністю і потребує від обслуговуючого персоналу глибоких знань і досвіду роботи; 2) для підвищення ефективності роботи служби захисту необхідно створення нової інформаційної технології, за допомогою якої можливо приймати ефективні рішення; 3) для створення такої інформаційної технології необхідно використати адекватний математичний опис об'єкту управління та запропонувати ефективні алгоритми опрацювання даних; 4) існуючі математичні моделі об'єктів протикорозійного захисту не повною мірою адекватні і недостатньо пристосовані до реальних умов експлуатації.

Приведена характеристика технологічного комплексу захисту від корозії. Відзначена інформація про основні чинники, що впливають на інтенсивність та характер корозійних процесів. Проаналізовано існуючі методи захисту від корозії, а також наведені критерії оцінки захищеності металевих споруджень. Проведено аналіз існуючих підходів до моделювання систем протикорозійного захисту і зроблені наступні висновки: 1) інформаційні системи протикорозійного захисту є досить складними об'єктами; 2) для підвищення ефективності роботи служби захисту необхідно створення нової інформаційної технології, за допомогою якої можливо приймати ефективні рішення; 3) для створення такої інформаційної технології необхідно використати адекватний математичний опис об'єкту управління та запропонувати ефективні алгоритми

опрацювання даних; 4) існуючі математичні моделі об'єктів протикорозійного захисту не повною мірою задовольняють вимогам адекватності і недостатньо пристосовані до реальних умов експлуатації. На основі проведених досліджень сформульовані зазначені вище мета та задачі дисертаційної роботи.

В основу інформаційної технології закладено фізико-математичну модель поверхневих явищ (Юзевич В. М., 1999 р.), що описує електричні, теплові, дифузійні і механічні процеси в поверхневих шарах твердих тіл, враховує при цьому особливості електричних взаємодій, поверхневі енергію γ та натяг σ_h, а також модель аналізу корозійних процесів (Джала Р. М., Юзевич В. М., 2011 р.), яка характеризує активаційні процеси деформованого металу поблизу вершини корозійної тріщини.

Для реалізації моделі поверхневих явищ розглянуто багатокомпонентне тверде тіло, розміри якого значно більші за ефективний розмір поверхневої області, в якій зосереджено поверхневий та міжфазний натяг.

Другий розділ. Описано розроблений принцип побудови інформаційної технології для системи генерації програм з операцій теорії секвенційних алгоритмів, що передбачає задання змінних і секвенційних областей їхніх значень, формули секвенційного алгоритму, який забезпечує розпізнавання типу операції та автоматичну генерацію програм з операцій теорії секвенційних алгоритмв на основі алгебри алгоритмів (Овсяк В. К., 1999).

Синтез загальної структури системи генерації програм з операцій теорії секвенційних алгоритмів, якій відповідає форма, утворенна ранговими змінними термами, що описують ідентифікацію та генерацію операцій теорії секвенційних алгоритмів, виконанно секвенційним методом, оскільки відповідні формули є значно компактнішими за таблицю, і ними суттєво легше оперувати.

У третьому розділі описана модель інформаційної технології структури системи генерації програм з операцій теорії секвенційних алгоритмів, якій відповідає формула, отримана заміною в структурі системи генерації програм

рангових змінних термів предметними. Для змінних стану задані секвентні області значень.

Розроблена та досліджена математична модель системи генерації програм з операцій теорії секвенційних алгоритмів, утворена такими моделями системи генерації програм з операцією секвентування як системи генерації програм з операцією елімінування і системи генерації програм з операцією циклу.

У даному розділі виконані дослідження та аналіз існуючих методів корозійних обстежень з метою оцінювання виникаючих похибок та їх характеристик. Результати цих досліджень були використані для урахування характеристик похибок при розробці алгоритмів ідентифікації параметрів. Відповідні методи та алгоритми входять до складу інформаційної технології протикорозійного захисту.

У четвертому розділі описана функціональна структура ІТ системи протикорозійного захисту. До складу системи входять інформаційні функції: які характеризують відбір, збереження і відображення інформації, що отримується під час проведення комплексних корозійних обстежень та діагностики корозійного стану елементів конструкцій поліграфічного матеріалознавства в агресивному середовищі.

Для реалізації вищеназваних функцій та задач було розроблено інформаційну модель бази даних, в основу якої покладено модель системи.

На основі запропонованих і розроблених в дисертації моделей та методів створена інформаційно-комп’ютерна технологія, яка призначена для автоматизації інформаційних функцій і складається з основних програмних модулів: 1) параметризація об'єктів; 2) збирання, накопичення та відображення даних вимірів; 3) діагностика стану контактуючих середовищ; 4) управління даними, розподілення і обмін інформацією між організаційними рівнями.

Аналогічно до розрахунку впливу кількісного фактора на деформаційну характеристику поліуретанового зразка поліграфічної машини (Юзевич В. М., Чехман Я. І., 1987, 1988) проведено обчислення енергетичних характеристик поверхневих шарів поліуретанового зразка. Оцінено з використанням алгебри

алгоритмів роль масштабного фактора при випробуванні поліуретанового зразка різної твердості.

Запропоновано низку фізико-хімічних параметрів і співвідношень, які характеризують метал і середовище поблизу вершини корозійної тріщини. З їх допомогою проведено оцінювання енергетичних характеристик поверхневих шарів для формних циліндрів поліграфічних машин, які функціонують в агресивних середовищах, зокрема, фізико-хімічних характеристик матеріалів в системі «метал – електроліт». Основна увага зосереджена на корозійних струмах, від яких залежить зміна форми тріщини та швидкість її просування в умовах пластичного деформування.

На основі співвідношень фізико-математичної моделі поверхневих шарів металів, які використовуються в корозіометрії, розроблено інформаційну технологію для відбору та опрацювання даних щодо оцінювання енергетичних характеристик міжфазних шарів та активаційних процесів, які характеризують метал, середовище і динаміку корозійних процесів поблизу вершини тріщини в корозійному середовищі.

У перспективі з використанням накопичених на основі експериментальних досліджень даних можна прогнозування корозійну поведінку деталей машин в корозійних середовищах, підданих дії механічного навантаження.

Вірогідність наукових результатів забезпечена строгістю моделей, коректністю використання методів теорії секвенційних алгоритмів та інформаційної технології, практичною реалізацією і апробацією побудови системи генерації програм.

Реалізація результатів роботи. В дисертації на засадах протикорозійного захисту розроблено інформаційну технологію, зокрема, метод, методику та систему алгоритмічного забезпечення відбору інформації про стан енергетичних характеристик поверхневих шарів металів в корозійному середовищі (розчині електроліту). Система генерації відповідних програм з операцій теорії секвенційних алгоритмів програмно реалізована і використана для автоматичної генерації комп’ютерних програм.

РОЗДІЛ І
ОГЛЯД І АНАЛІЗ ОСНОВ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ВИЗНАЧЕННЯ ЕНЕРГЕТИЧНИХ ХАРАКТЕРИСТИК ПОВЕРХНЕВИХ ШАРІВ МЕТАЛІВ З ВИКОРИСТАННЯМ АЛГЕБРИ АЛГОРИТМІВ

У ***першому*** розділі здійснено огляд основних інформаційно-технологічних етапів. В основу інформаційної технології закладено фізико-математичну модель поверхневих явищ (Юзевич В. М., 1999 р.), що описує електричні, теплові, дифузійні і механічні процеси в поверхневих шарах твердих тіл; враховує при цьому особливості електричних взаємодій, поверхневі енергію γ та натяг σ_h, а також модель аналізу корозійних процесів (Джала Р. М., Юзевич В. М., 2011 р.), яка характеризує активаційні процеси деформованого металу поблизу вершини корозійної тріщини.

Також у першому розділі розглянуто різні методи подання алгоритмів. Встановлено, що засобами, які забезпечують формалізований опис та виконання синтезу, мінімізації за вибраними критеріями дослідження математичних моделей алгоритмів комп’ютерних систем корозіометрії є засоби теорії секвенційних алгоритмів.

1.1. Співвідношення термодинамічної моделі поверхневого шару матеріалу як основа ІТ

Для реалізації моделі поверхневих явищ розглянуто багатокомпонентне тверде тіло, розміри якого значно більші за ефективний розмір поверхневої області, в якій зосереджено поверхневий та міжфазний натяг (рис. 1.1).

Співвідношення термодинамічної моделі поверхневого шару матеріалу (металу) подамо у вигляді [22,33, 87,100,105] :

$$Div\hat{\sigma} + \rho \cdot \omega \cdot \vec{E} = 0, \qquad (1)$$

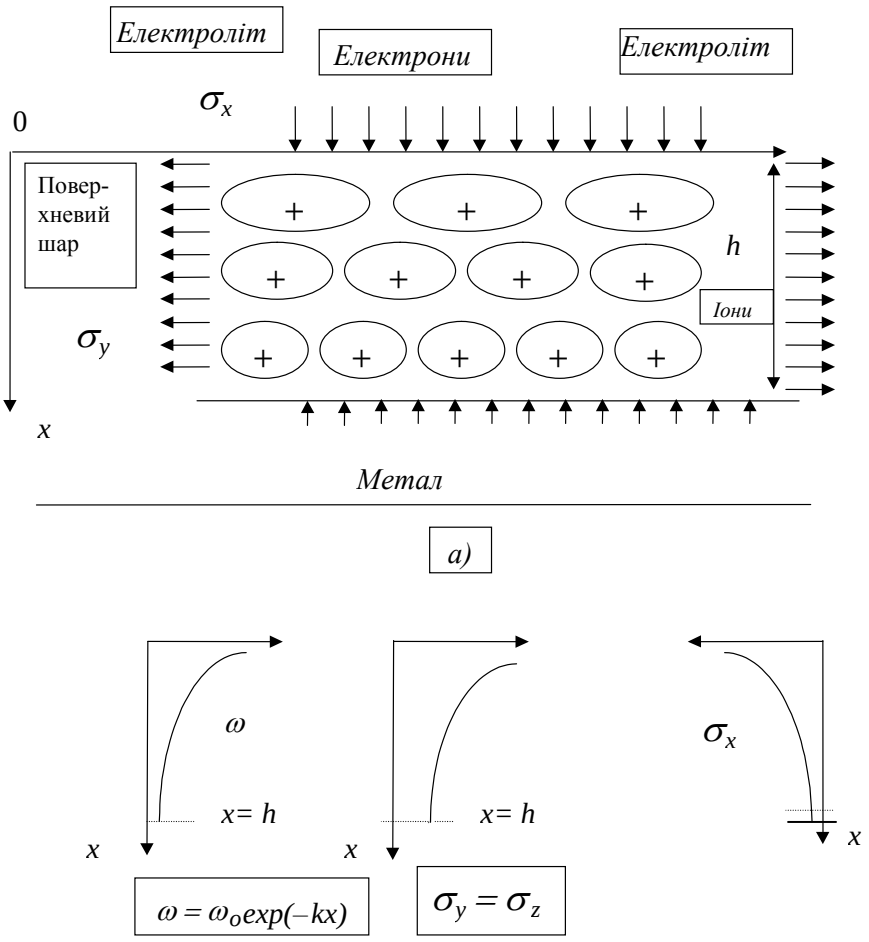


Рис. 1.1. Поверхневий шар металу.

$$\Delta\varphi = \rho \cdot C_\varphi \cdot \varphi / \varepsilon_0, \tag{2}$$

$$\sigma_h = \int_0^h \sigma_y dx, \quad \sigma_y = \sigma_z, \tag{3}$$

$\sigma_y + p = 0$ (для $x = h$) (p = 100 кПа – атмосферний тиск), (4)

$$\gamma = \gamma_1 + \xi\gamma_2, \tag{5}$$

$$\frac{\partial\gamma}{\partial k} = \frac{\partial(\gamma_1 + \xi\gamma_2)}{\partial k} = 0, \quad k = \sqrt{\frac{\rho C_\varphi}{\varepsilon_0}} \tag{6}$$

$$\sigma_{ij} = \left(\left(K - \frac{2}{3} G \right) e - \alpha_t K \cdot \Delta T - K(\beta\varphi + \beta_c c) \right) \delta_{ij} + 2Ge_{ij} , \tag{7}$$

$$\omega_V = \rho\omega = \rho C_\varphi (\varphi - \gamma_t \cdot \Delta T) + \beta K e - \rho\eta_c c . \tag{8}$$

$$\mu_c = d_c c + d_t \cdot \Delta T + \beta_c K \frac{e}{\rho} - \eta_c \varphi, \quad \Delta T = T - T_0 , \tag{9}$$

$$\varphi = -\Phi_0 ; \;\; \varphi + \psi = const ; \quad \sigma_x = -\frac{\varepsilon_0}{2} \cdot \left(\frac{\partial \Psi}{\partial x} \right)^2 \text{ при } x = 0. \tag{10}$$

Тут σ_h – поверхневий натяг; γ – поверхнева енергія (ПЕ); $\gamma_1 = \int_0^h w_1 dx$ – складова ПЕ, яка відповідає вільним (зв’язаним) електричних зарядам (так названа електрична складова); $\gamma_2 = \int_0^h w_2 dx$ – механічна складова ПЕ; $w_1 = \frac{\varepsilon_0}{2} \left(\frac{\partial \Psi}{\partial x} \right)^2$; $w_2 = \frac{\sigma_x (\sigma_x - 4\nu\sigma_y)}{2E} + \frac{(1-\nu)\sigma_y^2}{E}$; w_1, w_2 – густини електричної та механічної складових ПЕ; h – ефективна товщина поверхневого шару; σ_{ij} , e_{ij} – компоненти тензорів напружень $\hat{\sigma}$ і деформацій $\hat{e}$ (i, j = 1, 2, 3); $\sigma_{11} = \sigma_x$; $\sigma_{22} = \sigma_y$; μ_c, c – хімічний потенціал і концентрація домішки; ξ, k, α_t, β, β_c, C_φ, γ_t, η_c, d_c, d_t – фізичні характеристики матеріалу; δ_{ij} – символи Кронекера; e – перший інваріант тензора деформацій; ρ – питома густина матеріалу; ω_v, ω – просторова і масова густини зв’язаних електричних зарядів відповідно; $\varphi = \Phi - \Phi_0$ – відхилення модифікованого потенціалу Φ зв’язаних електричних зарядів від його рівноважного значення Φ_0 в об’ємі тіла далеко від поверхні; Ψ – скалярний потенціал напруженості електричного поля; $\psi = \Psi - \Psi_0$ – відхилення потенціалу Ψ від його рівноважного значення Ψ_0; $\vec{E}_{el} = \nabla\Psi = grad\Psi$ – напруженість електричного поля; ε_0 – електрична постійна; $\Delta T = T - T_0$ – зміна

температури; $K = \frac{E}{3(1-2\nu)}$, $G = \frac{E}{2(1+\nu)}$; K, G – модулі всестороннього стиску і зсуву відповідно; E – модуль Юнга; ν – коефіцієнт Пуассона.

Співвідношення (1)–(6) складають систему рівнянь для визначення фізичних ξ, k, α_t, β, β_c, C_φ, γ_t, η_c, d_c, d_t і геометричної h характеристик поверхневого шару.

Використовуючи рівняння рівноваги (1), співвідношення, яке випливає з рівнянь Максвелла (2), рівняння стану (7)–(9) і граничні умови (10), напруження (і компоненти тензора поверхневого натягу) в поверхневому шарі знаходимо, розкладаючи їх і деформації в ряди за малим параметром $b_м = \beta\Phi_0$.

Вирази (3), (5) подають означення поверхневих натягу σ_h та енергії γ, (6) — умову квазірівноваги поверхневого шару (стосовно перерозподілів механічних напружень і електричних зарядів).

Розроблена інформаційна технологія [1,54,65,66,102] спрямована на відбір даних, щоб забезпечити метод і відповідні методики розрахунку міжфазного натягу і роботи адгезії на границі між кородуючим електропровідним тілом і агресивним середовищем, від яких залежить інтенсивність корозійних процесів.

Сформульовано наукову задачу інформаційної технології корозіометрії, яка полягає в синтезі та дослідженні математичної моделі інформаційної технології системи генерації програм з операцій теорії сенквенційних алгоритмів та обґрунтовано вибір аналітичних засобів для фізико-математичного моделювання енергетичних характеристик поверхневих шарів елементів металевих конструкцій. Відповідна концептуальна модель подана на рис. 1. 2.

У поліграфії [84,105-107], як і в іншому виробництві, де задіюються складні і дорогі верстати, найчастіше, виникають поломки, викликані не лише зносом деталей і вузлів машини, або так званої втоми матеріалу (металу), але і через недбале поводження обслуговуючого персоналу, порушення елементарних правил експлуатації обладнання. Можлива корозія на офсетному

циліндрі друкарської машини при попаданні зволожуючого розчину під гумотканинне полотно (декель), піддаються корозійному впливу деталі зволожувальних апаратів. В агресивному середовищі перебувають вузли машини для електрохімічного нарощування і травлення металів у процесі виготовлення форм глибокого друку.

Для оцінки швидкості корозії використовуються як якісні, так і кількісні характеристики [5,22,25,33,41], однією з яких є корозійна стійкість.

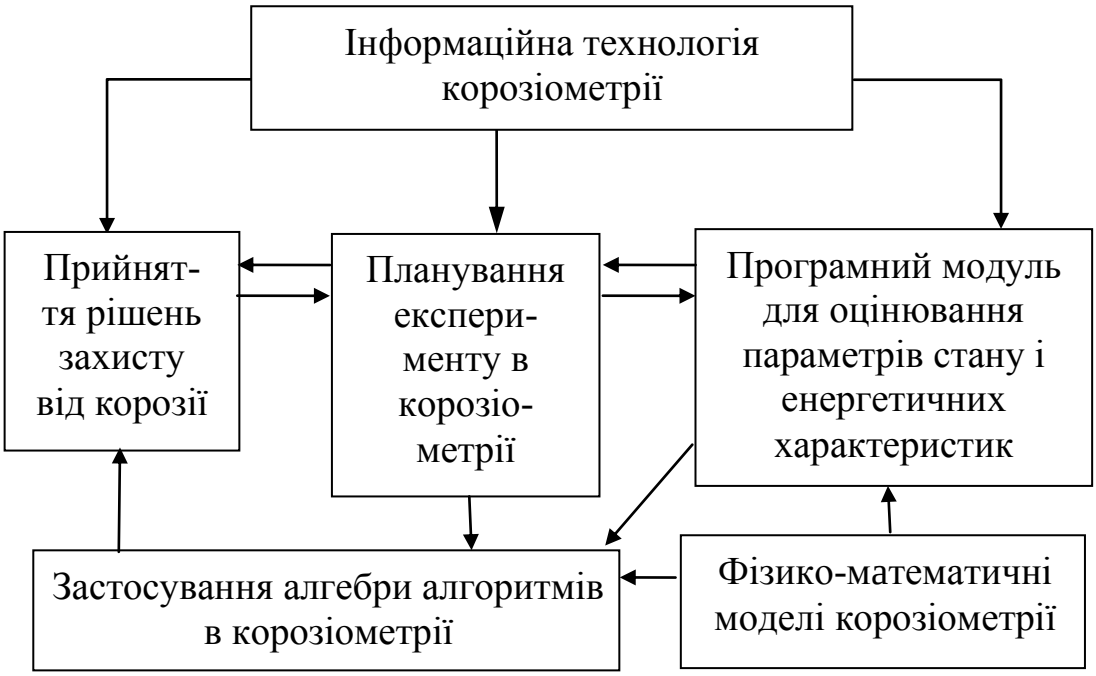


Рис. 1. 2. Структурна схема – концептуальна модель інформаційної технології корозіометрії визначення енергетичних характеристик поверхневих шарів металів з використанням алгебри алгоритмів

Корозійна стійкість – здатність матеріалу (металу) чинити опір корозійній дії середовища, що визначається швидкістю корозії в даних умовах.

Корозія металів – процес хімічного руйнування металів і сплавів при їх взаємодії з зовнішнім середовищем: повітрям, водою, розчинами тощо.

Корозіометрія – це наука на стику теорії корозії та метрології. Для кількісної оцінки характеристик корозіометрії можна використовувати: час, що минув до появи першого осередку корозії; число осередків корозії, що

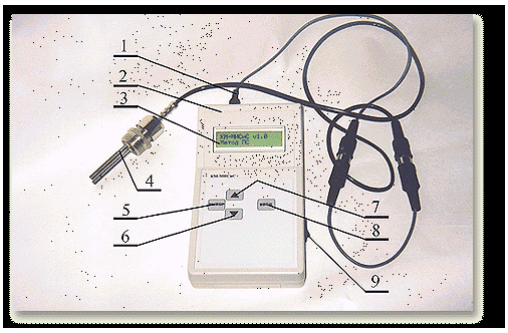
утворилися за певний проміжок часу; зменшення товщини матеріалу протягом заданої одиниці часу; зміна маси металу, віднесена до одиниці поверхні за одиницю часу; об’єм газу, що виділився (або поглинувся) в ході корозії, віднесений до одиниці поверхні за одиницю часу; густина електричного струму, що відповідає інтенсивності даного корозійного процесу; зміна деякої фізичної властивості за певний час корозії. До заходів щодо підвищенню корозійної стійкості відносять: антикорозійні металеві покриття, використання протекторних (анодного і катодного) методів захисту, покриття фарбами, лаками, виготовлення деталей із корозійностійких матеріалів (нержавіючі сталі, пластмаси тощо).

Інформаційні технології доповнюють теоретичні і методологічні основи та інструментальні засоби розроблення і використання критеріїв оцінювання і методів забезпечення якості, надійності, а також принципів оптимізації та моделей і методів прийняття рішень при створенні автоматизованих систем різноманітного призначення [82-86]. Результати дослідження закономірностей побудови інформаційних комунікацій та розроблення засад побудови і впровадження інтелектуальних інформаційних технологій застосовуються для створення систем накопичування, збереження інформації і можуть бути використанні, зокрема, і для аналізу корозійного стану циліндрів друкарських машин. Зображення корозії на офсетному циліндрі представлена на рис. 1.3.

Корозіометрії відповідає комплекс приладів (корозіометрів та адгезиметрів). Деякі з них зображені на рис. 1.4 – 1.6. Корозіометри допомагають вимірювати густину корозійного струму, адгезиметри – міцність покриття, яке захищає метал від негативного впливу агресивного середовища.

1.2. Методи подання алгоритмів

Поняття “алгоритм” в його загальному вигляді [9,13,38,44,47-53,80] належить до основних початкових понять математики і займає одне з центральних місць в сучасній математиці, перш за все в обчислювальній.

Рис. 1.3. Корозія на офсетному циліндрі	 Рис. 1.4. Корозіометр «МК-МИСиС» для корозійного моніторингу

Корозіометри і адгезиметри, ПК є технічними засобами інформаційної технології визначення енергетичних характеристик поверхневих шарів металів з використанням алгебри алгоритмів.

Рис. 1.5. Корозіометр «Експерт-004» в комплекті з сенсорами	Рис. 1.6. Автоматичний адгезиметр PosiTest® AT-A

Це найбільш точне визначення алгоритму, так як при уточненні цього поняття алгоритму мають важливе значення і властивості алгоритму, і параметри алгоритму, що і є перевагою цього визначення.

Алгоритм – точний опис, який задає обчислювальний процес, що починається з довільного початкового даного і направлений на одержання результату, що повністю повністю визначається цими початковими даними.

Алгоритм має такі характеристики (властивості): визначеність, скінченність, результативність, правильність, формальність, масовність [9,38,44].

Визначеність алгоритму. Алгоритм визначений**,** якщо він складається з допустимих команд, які можна виконати для заданих вхідних даних.

Невизначеність виникає, якщо деяка команда буде записана неправильно, бо така команда належить до набору допустимих команд виконавця.

Алгоритм результативний, якщо він дає результати (які можуть виявитися і неправильними).

Правильність алгоритму. Алгоритм правильний, якщо його виконання забезпечує досягнення мети.

Формальність алгоритму. Алгоритм формальний, якщо його можуть виконати не один, а декілька виконавців з однаковими результатами. Цю властивість ще називають однозначністю алгоритму. Вона означає, що коли алгоритм застосовують до двох наборів вхідних даних то й результати мають бути однакові.

Масовість алгоритму. Алгоритм масовий, якщо він придатний для розв’язавання не одної задачі, а низки подібних задач (тобто задач деякого класу).

Масовими є алгоритми розв’язування математичних задач, описаних у загальному вигляді за допомогою формул. Їх можна виконати для різних вхідних даних. Ще донедавна розглядали лише послідовні алгоритми, для яких характерна властивість дискретності: кожна команда починає виконуватись після закінчення попередньої. Сьогодні, крім послідовних, є паралельні й конвеєрні алгоритми, де наступні команди починають виконуватися, до закінчення попередніх.

Алгоритми [95,97,118] є основою інформаційно - технологічних систем. Вони можуть подаватися в різних формах, частково таких як:

1) блок–схеми;
2) граф–схеми;
3) програм, написаних на алгоритмічних мовах;
4) словесний опис;
5) в аналітичному вигляді.

Під час конструювання алгоритмів програмісти керуються в основному інтуїтивним підходом, не замислюючись над тим, чи утворюють певну систему виконувані ними операції, проте така система є, вона формально описана і називається алгоритмічною алгеброю. Наразі розроблено кілька алгоритмічних алгебр [13,95,97], найвідомішими з яких є алгебра Дейкстри, алгебра схем Янова та алгоритміка програм, досліджена у працях В. М. Глушкова і Г. О. Цейтліна.

Дослідження і побудова алгебри алгоритмів або алгоритмічної алгебри почалося з проектування логічних структур ЕОМ під керівництвом академіка В. М.Глушкова. Як результат була створена теорія систем алгоритмічних алгебр (САА), що потім Г. О. Цейтліним була покладена в основу узагальненої теорії структурованих схем алгоритмів і програм, названою ним алгоритмікою [97].

Алгоритміка програм призначена для побудови послідовних і паралельних програм зі структурних схем з використанням апарату формальних алгебраїчних перетворень і канонічних форм опису логічних і операторних виразів. Її основу становлять САА, розширені формалізмами для зображення логічних умов виконання паралельних програм, а також методами символьної мультиобробки [38, 44].

Основними поняттями алгебри алгоритмів [9,13,97] є:

– операції над множинами, булеві операції, предикати, функції й оператори;

– бінарні і n-арні відношення, еквівалентність, частково і цілком упорядковані множини;

– графи-схеми й операції над графовими структурами;

– операції сигнатури САА, аксіоми і правила визначення властивостей програм на основі стратегії згортання, розгортання і їх комбінацій;

– методи синтаксичного аналізу структурних програм і символьна обробка.

Об'єкти алгоритміки – моделі алгоритмів і програм, що подаються у вигляді схем. Алгоритміка базується на комп'ютерній алгебрі, логіці і використовується для формального опису алгоритмів в МП.

У рамках алгоритміки розроблено спеціальні інструментальні засоби реалізації алгоритмів програм, що використовують сучасні об'єктно-орієнтовані засоби і метод моделювання . Цим забезпечується практичне застосування розробленої теорії алгоритміки у реалізації прикладних задач, починаючи з їхньої постановки, формування вимог і розробки алгоритмів і закінчуючи одержанням програм, що розв’язують ці задачі.Алгебра алгоритмів АА = {A, W}, як і будь-яка алгебра, — це основа A і сигнатура W операцій з елементами цієї основи. За допомогою операції сигнатури можна додати довільний елемент q Î AA. Це називається системою утворювальної алгебри.Якщо з цієї системи не може бути виключений жоден елемент без порушення властивостей, то така система утворювальних алгебр називається базисом алгебри. Операції алгебри задовольняють наступні аксіоматичні закони: асоціативності, комутативності, ідемпотентності, закони виключення третього і суперечності. Алгебру, яку задовольняють перераховані операції, називають булевою.В алгебрі алгоритмів використовується алгебра множин, тобто елементи множини й операції над ними (об'єднання, перетину, доповнення і ін.). Вводиться також поняття універсуму – множини операцій постановки і розв’язання деякої задачі [97].Основні об'єкти алгебри алгоритмів – схеми алгоритмів і їх суперпозиції, тобто підстановки одних

схем в інші. З підстановкою зв'язані розгорнення, що відповідає спадному процесу проектування алгоритмів, і згортання, тобто перехід до більш високого рівня специфікації алгоритму [13,95,97]. Схеми алгоритмів відповідають конструкціям структурного програмування.

Послідовне виконання операторів А і В записується у вигляді композиції А*У; альтернативне виконання операторів А и В (u (А, У)) означає, якщо u – істинне, то виконується А, інакше В; цикл (u (А, У)) виконується, поки не стане істинною умова u (u – логічна змінна).За допомогою цих елементарних конструкцій будується більш складна схема Р алгоритму:

P ::= { [u1] A1},

A1 ::= { [u2] A2 * D},

A2 ::= A3 * C,

A3 ::= { [u] A, B},

u ::= u2 Ù u1.

Здійснивши суперпозицію шляхом згортання даної схеми алгоритму Р, одержуємо таку схему:

P ::= {[u1]{ [u2] ([u2 Ù u1] A, У) * З }* D}.

Важливим показником можливостей алгоритміки є проведений порівняльний аналіз алгебри алгоритміки з відомими алгебрами.

Алгебра Дейкстри АД = {АСС, L(2), СИГН} – двохосновна алгебра, елементами якої є множина САА операторів, зображених структурними блок-схемами, множина L(2) булевих функцій у сигнатурі СИГН, в яку входять операції диз’юнкції, кон’юнкції і суперечності, що набувають значення з L(2). За допомогою спеціально розроблених механізмів перетворення АД на алгебру алгоритміки встановлено зв'язок між альтернативою і циклом за формулою {[u] A} = ([u] E, A * {[u] A}), що входить у СИГН. В АД використовуються похідні операції, які можуть бути

отримані в наслідок суперпозиції основних операцій і констант.Операція фільтрації Ф (u) = {[u] E, N} у АД зображена суперпозицією тотожного E для невизначених N операторів і альтернативи в алгебрі алгоритміки, де N – фільтр дозволу виконання операцій обчислень.Оператор циклу while do також представлений суперпозицією операцій композиції і циклу в алгебрі алгоритміки.Алгебра схем Янова АЯ = <{АНС, L(2)}; СИГН>, де АНС – сукупність неструктурних схем, L(2) – сукупність різних булевих функцій, СИГН – сигнатура з композиції A*B і операція неструктурного переходу P (u, F), а також операції диз'юнкції, кон’юнкції і суперечності. АЯ містить у собі операції побудови неструктурних логічних схем програм.Схема Янова складається з предикатних символів множини P(p1, p2,…,), операторних символів множини A{a1, a2,…} і графа переходів. Оператор у даній алгебрі – це пари A{p}, що складається із символів множини A и предикатних символів p. Граф переходу – це орієнтований граф, у вершинах якого розміщуються перетворювачі, розпізнавачі й один оператор зупинки. Дуги графа задаються стрілками і позначаються знаками + і –. Дуги, що виходять з розпізнавача, розрізняються і називаються відповідно плюс-стрілка і мінус–стрілка. Перетворювач має одного нащадка, розпізнавач – двох. Одна вершина в графі переходу зветься вхідною і помічається вхідною стрілкою. Також є вершина зупинення, яка не має нащадків. Кожен розпізнавач містить у собі умови виконання схеми. Перетворювач обробляє оператори, що вміщають логічні змінні, що належать множині (p1, p2,…) [97].Кожна схема АЯ відзначається великою складністю, вимагає серйозного перетворення при переході до задання програми послідовністю дій, умов переходу і безумовного переходу. Для зображення схеми Янова апаратом алгебри алгоритміки сигнатура операцій АЯ вводяться композиції A* B і операція умовного переходу, що залежно від умови u виконує перехід до наступних операторів або до оператора, позначеного міткою (типу goto). Умовний перехід трактується як бінарна операція P (u, F), що залежить від умови u і розміток схеми F. Крім того, виконується заміна альтернативи і циклу типу

while do. Внаслідок виконання бінарних операцій отримується нова схема F’, у якій встановлені бінарні операції P(u) замість мітки і булевих операцій кон’юнкції і суперечності. Еквівалентність операцій перетворення доводить правильність переходу до неструктурного подання програм.

Система алгебр Глушкова АГ = {ОП, УМ, СИГН}, де ОП – множина операторів, що входять у сигнатуру СИГН, і УМ – множина логічних умов, визначених на інформаційній множині ІМ.Сигнатура СИГН = {СИГНад È Прогн.}, де СИГНад – сигнатура операцій Дейкстри, Прогн. – операція прогнозування. Сигнатура САА містить у собі операції алгебри АД, узагальнену тризначну булеву операцію і операцію прогнозування (ліве множення умови на оператор u = (A* u’) з породженням предиката u = УС такого, що u(m) = u’(м'), м' = A(m), A Î ОП). ИМ – множина оброблюваних даних за операціями з множин ОП і УС. Сутність операції прогнозування полягає в перевірці умови u у стані m оператора A і визначення умови u’, обчисленої в стані м' після виконання оператора A. Дана алгебра [13,95,97] орієнтована на аналітичну форму подання алгоритмів і оптимізацію алгоритмів за вибраними критеріями.

Алгебру алгоритміки розширено Г. О. Цейтліним дворівневою алгебраїчною системою і механізмами абстрактного опису даних (класами алгоритмів). Під багатоосновною алгоритмічною системою (БАС) розуміють систему S ={{Di÷ iÎ I }; СИГН0 , СИГНn }, де Di – основи або сорти, СИГН0 , СИГНn – сукупності операцій і предикатів, визначених на Di . Якщо вони порожні, то визначаються багатоосновні моделі – алгебри. Якщо сорти інтерпретуються як множина оброблюваних даних, то БАС є концепцію АТД у вигляді підалгебри, що широко використовується в об'єктному програмуванні. Тим самим установлено зв'язок із сучасними тенденціями розвитку програмування. Практичним результатом досліджень алгебри алгоритміки є побудова оригінальних інструментальних систем проектування програм на основі сучасних засобів підтримки ООП (Rational Rose). Аналіз методів систематичного і теоретичного програмування

свідчить про їх розвиток, удосконалювання, поповнення новими можливостям для застосування. Класичним прикладом є UML, завдяки якому ООП набув нових візуальних можливостей для проектування ПС. Теоретичні методи увібрали в себе не тільки можливості логічного, функціонального і процедурного програмування, а й алгебро-алгоритмічного апарата для абстрактного і формального представлення програм. Визначено сучасні тенденції розвитку мов опису DSL, інженерії ПС і предметних областей. Об'єднуючим ланцюгом у програмуванні є генерувальне програмування, що забезпечує генерацію членів сімейства предметної області з компонентів і ресурсів багаторазового використання, розроблених у різних середовищах за різними методами програмування (об'єктним, аспектним, компонентним і ін.). Дано загальну характеристику [13,95,97] різних формальних теорій програмування, орієнтованих на абстрактне подання програм.

1.3. Подання алгоритмів засобами теорії секвенційних алгоритмів

Аналітичний опис алгоритмів здійснюється засобами теорії секвенційних алгоритмів [47-60]. Дана теорія побудована за аксіоматичним методом. Операції мають властивості, які подаються аксіомами.

Мову (синтаксис і семантика) теорії секвенційних алгоритмів утворюють:

– такі знаки операцій: $\frown$ - секвентування; $\vdash\!\dashv$ елімінування;

$\overline{}$ - інвертування; $/\subset$ - циклічного секвентування ; $/\supset$ - циклічного елімінування; $\overline{||}$ -паралелення; $\not\cup$ - циклічного паралелення;

- незв'язані операціями циклу терми-змінні, коефіцієнти, константи, які позначаються з будь-якими індексами або без них малими літерами із початку латинського алфавіту:

$a, b, c_0, c_1, \ldots, c_j, c^0, c^1, \ldots, c^I, c^i_j, \ldots;$

- зв'язані операціями циклу терми-змінні, які позначаються з будь-якими індексами або без них малими літерами із кінця латинського алфавіту:

$x, y, z_0, z_1, \ldots, z_j, z^0, z^1, \ldots, z^I, z^i_j, \ldots;$

- від однієї чи декількох змінних терм над якими виконуються операції які позначаються з будь-якими індексами або без них великими літерами латинського алфавіту:

$P(a), P(a, b), \ldots;$

-терми умов, які приймають два значення (0 або 1) і позначаються з будь-якими індексами або без них малою літерою латинського алфавіту:

$u, u_0, u_1, \ldots, u_j, u_j^i;$

-*- порожній терм;

-порядку ознаки (секвенти) α і β та змінні секвенти δ, γ, η;

відкрита і закрита круглі дужки, відкрита ($\{$) і закрита ($\}$) фігурні дужки, дві крапки(:), крапка з комою (;) та кома (,), знаки логічних операцій (&, V,-), / - слеш, c_u - ознака кінця циклу за u;

-φ - та ψ - паралени, змінні парален, які позначаються малими літерами із кінця грецького алфавіту ω, σ, ...

Якщо S та R – терми, то $\overset{\frown}{S : R}$ та $\vdash\!\overline{S : R}\!\dashv$ – алгоритми.

Якщо A та B – терми або алгоритми, а u – умова, то

$\vdash\!\overline{A\ ,\ B,\ u\text{-}?}\!\dashv$ та $\vdash\!\overline{B\ ,\ A\ ,\ u\text{-}?}\!\dashv$ – алгоритми.

Якщо A та B – алгоритми, то $\overline{A}, \overline{B}, \overset{\frown}{A : B}, \vdash\!\overline{A : B}\!\dashv$ – алгоритми.

Якщо A_x та A_u – терми або алгоритми, а x – зв'язна змінна та u - зв'язна умова, то $\not\subset$, $\subset xA_x$, $\supset xA_x$, $\not\subset$, U xA_x, $\subset xA_u$, $\supset xA_u$, $\not\subset$, U xA_u – алгоритми.

Теорія секвенційних алгоритмів – наука про властивості секвентування, елімінування, паралелення, інвертування і циклічних операцій над термами та алгоритмами.

Наприклад. Операції теорії секвенціїних алгоритмів є специфічними графічними знаками, яких немає серед стандартних умовних позначень. Тому з метою зменшення трудомісткості набору формул теорії секвенційних

алгоритмів був розроблений редактор формул алгоритмів МОДАЛ [63,64] . Він здійснює автоматичний набір, редагування формул теорії секвенційних алгоритмів.

У розробленому редакторі формул співвідношення розмірів знаків операцій описуються такими виразами

- секвентування $b=2(l+2)$;
- елімінування $b=2(l\backslash 2+2)$;
- паралелення $b=l\backslash 2+2$, де l - довжина, b – ширина.

Система автоматизованого набору редагування і моделювання формул теорії секвенційних алгоритмів [54-57,60,63,64] має меню, функціями якого є робота з даними системи, редагування знаків операцій: копіювання, вирізання, вставлення, знищення, задання параметрів знаків операцій і елементів термів, встановлення параметрів робочого вікна системи і надання допомоги користувачеві.

Редактор формул алгоритмів реалізовано мовою програмування Delphi. Даних функцій і можливостей є достатньо для програмної реалізації редактора формул теорії секвенційних алгоритмів.

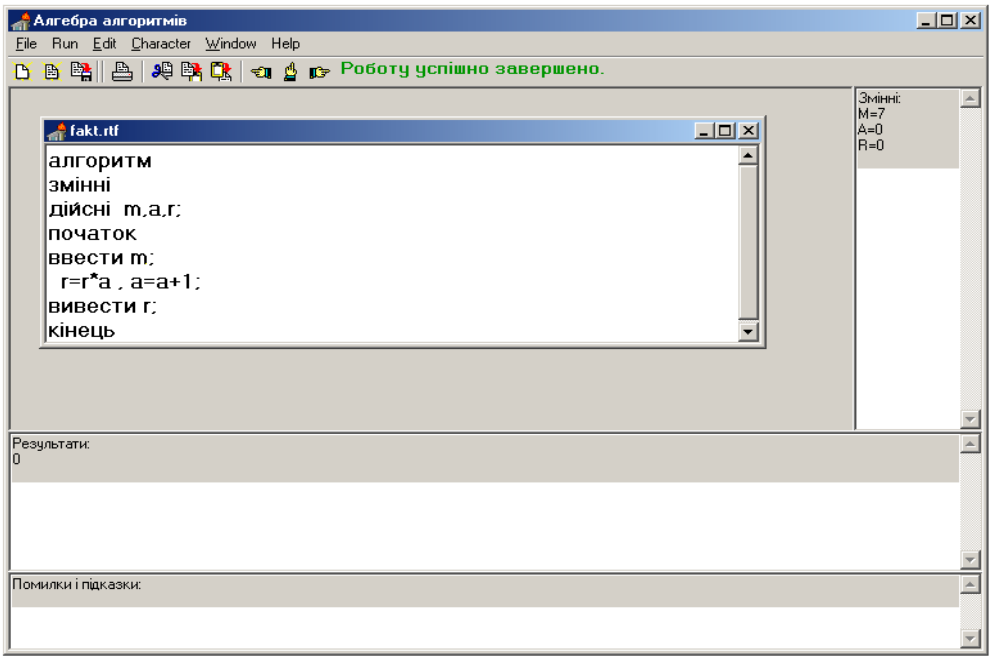


Рис. 1.7. Інтерфейс користувача редактора формул алгоритмів

Аналізом методів подання алгоритмів встановлено, що засоби теорії сенквенційних алгоритмів забезпечують формалізований опис і тотожні перетворення алгоритмів з метою мінімізації.

Системи генерації програм

Сучасна комп'ютерна математика має низку інтегрованих програмних систем для автоматизації математичних обчислень[28-32,64,69,70]: Eureka , Asystant , Derive , MathCad , Mathematica , Maple V .Хоча дані системи розв'язують схожі задачі – розрахунки по формулах, розв'язання систем лінійних і нелінійних рівнянь, мінімізація функції, всі ці системи основані на принципіально різних підходах і базових алгоритмах.Наприклад, пакет **Derive** орієнтований на символьні операції і використовує мову програмування задачі штучного інтелекту MuLISP, Maple V базується на могучому ядрі символьних операцій, система Mathcad містить унікальний формульний інтерпретатор.Основна задача системи **MATLAB** є представлена користувачеві мову програмування орієнтовану на математичні розрахунки.**Пакет Maple V** зручний у роботі, має великий набір функцій та обчислювальних засобів. Пакет дає змогу розв'язати чисельно й аналітично велику кількість математичних задач будь-якого рівня складності. **Mathcad** математично орієнтована універсальна система. З їх допомогою можна підготувати текст статтей, книг, вони крім того полегшують набір складних математичних формул і дають можливість предсталення результатів у графічному вигляді.

Отже, провівши аналіз сучасних програмних системах встановлння, що не має функцій, які б забезпечували генерацію програм з операцій теорії секвенційних алгоритмів.

1.4. Методи верифікації програм

Під верифікацією програми [47-53] розуміється реалізація та апробація математичної моделі системи генерації програм. Сюди відноситься: логічна специфікація програм, аналіз коректності послідовних програм, аксіоматична семантика послідовних програм, автоматизація верифікації програм, доведення

коректності програми в проблемній області. Верифікація розглядається також як технологія програмування.

Проблематику верифікації на основі автоматизованого підходу та формальні постановки задач верифікації (аналіза коректності) відомі автори досліджували в роботах [37,38,47-50]. Враховувався зв'язок коректності та надійності програм.

Одним з методів верифікації є метод індуктивних тверджень, що базується на використанні стратегії оберненого прослідковування шляхів обчислення. Цей метод близький до модифікованого автоматичного підходу [47-52]. Відомі також методи представлення знань про семантику [48-53] типових фрагментів програм основані на теоремі про лінійний перегляд.

Потрібно відмітити, що метод індуктивних тверджень має універсальний підхід. Недолік його в необхідності кожний раз повторяти повністю всю процедуру аналізу, тобто відсутність накопичення знань про виконані раніше доведення.

В роботі [48] для верифікації рекомендується виділяти два види узагальнених знань про властивості коректності.

а) декоративні знання про перетворення програм.;

б) процедурні знання про перетворення програм.

Модулі знань двох видів (фрейми властивостей коректності) повинні зображатися як деякі формальні твердження (теореми) про відповідні властивості, що допускають конкретизацію на програмних об'єктах (змінні, функції, оператори).

Традиційний спосіб забезпечення надійності моделі та генератора програм шляхом тестування не може бути повністю задовільнити зростаючі вимоги практики. Забезпечити вимоги надійності та якості можна шляхом об'єднання методів тестування і верифікації програм.

В загальному випадку верифікація підтверджує коректність моделі та програми та розглядається в роботі як дальший розвиток тестування в новій версії. Коли звичайне тестування обмежено дослідженням окремих виконань

програми, тоді верифікація розглядається як аналіз властивостей всіх допустимих виконань програми з допомогою формальних доведень наявності потрібних властивостей. Верифікація полягає в тому, щоб формально довести відповідність між текстом програми та специфікацією моделі на нові специфікації. По суті модель та програма описують одну і ту ж задачу, але на різних мовах. Мова специфікації, моделі орієнтована на спеціаліста, а мова програмування в більшій степені орієнтована на комп'ютер. З позицій технології програмування верифікація розглядається в роботі як доповнення тестування з новою стратегією відладки програм.

В залежності від етапів [53,60,61,64,67] моделювання та конструювання програми передбачаються декілька підходів.

Аналітичний підхід передбачає верифікацію готової програми та її фрагментів. Правильно здійснене програмування забезпечує доведення властивостей програми.

Згідно синтетичного підходу програма конструюється покроково сумісно з доведенням її коректності.

Верифікація динамічних властивостей обчислень передбачає пошук та знаходження аномалій, наприклад: ділення на нуль, некоректність операцій, незавершення циклу та інші.

Для успішної верифікації в роботі притримувались таких вимог:

- модульний принцип організації програм;
- зручна мова специфікації властивостей програми;
- формальна специфікація проблемної галузі;
- обмеження на керуючі та інформаційні структури мови програмування.

Для верифікації програм [53-55,60,64] використовується проблемно-орієнтований аксіоматичний підхід.

Верифікація звичайно використовує математичну логіку предметом вивчення якої є судження (висловлення, твердження) та способи доведень, що дозволяють виводити одні судження з інших. Використовуються закони

логічного мислення: закон тотожності, закон суперечності, закон виключення третього, закон достатньої підстави, закон комутативності диз'юнкції, закон комутативної кон'юнкції, закони асоціативності, закони поглинання та закони склеювання.

Для формалізації суджень, для умовиводів згідно математичної логіки використовується спеціальна формальна мова. Згідно математичної логіки вивчаються властивості формул моделі.

Методи верифікації використовують алгебру висловлень, бульові операції (логічне заперечення, логічне множення, імплікацію, нерівнозначність, еквівалентність), таблицю істинності та закони алгебри логіки.

Для запису тверджень про властивості програми вигідно користуватися формальною логікою мовою специфікацій. З практичної сторони використання це мова повинна мати такі властивості:

- відсутність, можливість включення в мову нових понять;
- модульність, незалежність частин;
- ієрархічність структури складних понять, що допускають вкладення одних понять в інші;
- зручність, наближення мови до зручного виду.

У зв'язку з цим будемо вважати, що логічна мова специфікацій, математична логіка з її операціями є доповненням до операцій теорії сенквенційних алгоритмів. Таке представлення відкриває широкі можливості для формальних перетворень програм і формальних доведень властивостей програм, що є основою для верифікації та оптимізації програм. Вдала система понять визначає успіх як програми так і її верифікації.

Аналіз програми згідно [53,74,75,79,80] передбачає:

- неформальне та формальне визначення програм;
- логічні формули, що описують виконання програм;;
- аналіз програм з допомогою резолюції;
- завершення роботи та реалізація програм;
- стратегію підтримки;

- правильність та еквівалентність програм;
- спеціалізацію програм.

В роботі [53] описані принципи синтезу програм, застосування аксіом індукції до синтезу програм, а також програма для ЕОМ, що використовує одиничну бінарну резолюцію.

Пошук висновку та автоматизація доведень програм передбачає:

- теорію пошуку висновку та обернелий метод;
- ідею метазмінних в локальних методах пошуку висновку.

В роботах [48-53] використовуються теорія графів, традиційні блок-схеми та записи програм подібні до теорії сенквенційних алгоритмів.

Генерація здійснюється програмою на основі алгоритму. Побудова і дослідження алгоритму є важливими науково-технічними задачами, розв’язання яких потребує відповідних методів. Для здійснення синтезу і переробки алгоритмів потрібні належні математичні засоби. Відомі такі підходи формалізації алгоритмів[47,52], як машина Поста , машина Тюрінга , алгоритми Маркова, машина Калмогорова, рекурсивні функції, теорія секвенційних алгоритмів [53].

Підходи формалізації алгоритмів, які викладені у роботах [7,9,10,53-55], забезпечують формалізованого опису алгоритмів, тотожних перетворень з метою мінімізації за вибраними критеріями, дослідження з метою виявлення і вилучення помилок.

В теорії секвенційних алгоритмів [53] введенні операції впроваджених термінів, які забезпечують формалізований опис і тотожні перетворення алгоритмів. Розроблені також методи синтезу, мінімізації і дослідження алгоритмів. Відомі методики застосування теорії секвенцій них алгоритмів до формалізації системи моделювання електромеханічних схем, формалізації алгоритмів функціонування мікропроцесорних схем, побудови і дослідження математичної моделі редактора формул сквенційних алгоритмів [9,10]. Теорія секвенційних алгоритмів вивчає загальні властивості алгоритмів, виникла в 80-х роках 20 століття. Вперше поняття алгоритму з’явилося в працях Е. Бореля

(1912) та Г. Вейля (1921).Першими формальними моделями алгоритмічно обчислюваних функцій були λ-означувані функції (Алонзо Черч, 1932) та загальнорекурсивні функції (Курт Гедель, 1934). В 1936 році Стівен Коул Кліні поширив поняття загальнорекурсивної функції на випадок часткових функцій, ввівши поняття частково рекурсивної функції, та описав клас таких функцій в чисто функціональних термінах. В 1943 році Еміль Пост запропонував модель обчислюваних функцій на основі введеного ним числення спеціального вигляду (канонічних систем). Теорія секвенційних алгоритмів виникла як розділ математичної логіки, поняття алгоритму тісно пов'язане з поняттям числення. Теорія секвенційних алгоритмів є теоретичним фундаментом програмування, вона має застосування всюди, де зустрічаються алгоритмічні проблеми , теорія інформації, теорія керування, конструктивний аналіз, обчислювальна математика, та ін. Основними поняттями алгебри секвенційних алгоритмів є:– операції над множинами, булеві операції, предикати, функції й оператори;– бінарні і n-арні відношення, частково і цілком упорядковані множини;– операції сигнатури САА, аксіоми і правила визначення властивостей програм на основі стратегії згортання, розгортання і їх комбінацій;– методи синтаксичного аналізу структурних програм і символьна обробка.Теорія секвенційних алгоритмів і проектування комп’ютерних систем.Засоби еквівалентних перетворень алгоритмів. Методи підвищення ефективності математичного моделювання алгоритмів інформаційно-технологічних систем.

Проектування комп'ютерних систем-створенню мереж на основі канального і мережного рівня моделі OSI. Одна з перших класифікацій комп'ютерних систем була запропонована Д. Шорам на початку 70-х років. Вона цікава тим, що є спробою виділення типових способів компонування комп'ютерних систем на основі фіксованого числа базових блоків: пристрою керування, арифметико-логічного пристрою, пам'яті команд і пам'яті даних. Вивчається механізм роботи системи DNS, питання побудови і настроювання основного, допоміжного серверу і серверу кешування DNS, функції керованих

комутаторів, зокрема віртуальні мережі, керування трафиком, фільтрація трафика, приорітизація трафика, агрегування каналу. Розглядаються принципи маршрутизації, протоколи маршрутизації, реалізація міжмережної взаємодії засобами TCP/IP, дистанційно-векторний протокол RIP і протокол стану зв'язків OSPF динамічної маршрутизації. Вивчаються функції маршрутизатора на рівнях інтерфейсів, мережного протоколу, протоколів маршрутизації,засоби керування і моніторингу комп'ютерними мережами на базі протоколу SNMP. Практично за допомогою пакета MRTG і протоколу SNMP графічно досліджується завантаження інтерфейсів комунікаційних пристроїв. Вивчаються питання установки в мережі і настроювання проксі сервера SQUID для забезпечення виходу мережі організації в мережу Internet, механізм роботи системи DNS, питання побудови і настроювання основного, допоміжного серверу і серверу кешування DNS.

Суттєвий внесок зробив В. К. Овсяк: інтенсифікація технологічних процесів в поліграфії, моделювання та оптимізація процесів, теорія секвенційних алгоритмів і проектування комп'ютерних систем, принципи побудови комп'ютерної бібліотеки абстрактних алгоритмів [47-53].

Як бачимо єдиними засобами, які забезпечують виконання синтезу, мінімізації за вибраними критеріями і дослідження математичних моделей алгоритмів комп’ютерних систем є засоби теорії секвенційних алгоритмів. Тому для синтезу, мінімізації за кількістю термінів і дослідження математичної моделі системи генерації програм з операції теорії секвенцій них алгоритмів використовуватимемо теорію секвенцій них алгоритмів.

1.5. Висновки до першого розділу

1) Для забезпечення функціонування ІТ впорядковано набір параметрів стану, фізичних констант і енергетичних характеристик поверхневих та міжфазних шарів, які формують оптимальний набір фізичних величин для системи вихідних даних.

2) Для забезпечення ефективності ІТ прийняті за основу математичні моделі об'єктів протикорозійного захисту і встановлено, що вони не повною мірою задовольняють вимогам адекватності, громіздкі і недостатньо пристосовані до аналізу реальних умов експлуатації устаткування, тому потребують удосконалення та інформаційного наповнення.

3) На основі проведених досліджень уточнено мету та задачі дисертаційної роботи, що стосуються інформаційного забезпечення фізико-математичної моделі контактуючих середовищ (моделі поверхневих явищ).

4) Для опису процесів ІТ на основі моделі поверхневих явищ розглянуто багатокомпонентне тверде тіло (метал), розміри якого значно більші за ефективний розмір поверхневої області h, в якій зосереджено поверхневий σ_h та міжфазний σ_m натяги, поверхнева γ та міжфазна γ_m енергії.

5) Проаналізовано методи подання алгоритмів. Встановлено, що засоби теорії сенквенційних алгоритмів забезпечують формалізований опис і тотожні перетворення алгоритмів з метою мінімізації. На основі аналізу відомих комп’ютерних систем встановлено, що ними не забезпечується генерація програми з операцій теорії сенквенційних алгоритмів.

РОЗДІЛ II
ІТ СИНТЕЗУ І МІНІМІЗАЦІЇ СИСТЕМИ ГЕНЕРАЦІЇ ПРОГРАМ З ФОРМУЛ ТЕОРІЇ СЕКВЕНЦІЙНИХ АЛГОРИТМІВ

У другому розділі описано принцип побудови системи генерації програм з операцій теорії секвенційних алгоритмів, синтезовано і мінімізовано за кількістю термів структури системи генерації.

Для синтезу формул теорії секвенційних алгоритмів [55,57,60,63,64] розробленно два методи, це секвенційний та табличний

Використання табличного методу синтезу передбачає побудову таблиці істиності, та побудову формул.

Таблиця істиності будується так:

1. Записується одна за другою змінні умови.
2. Для змінних умов утворюються всі можливі комбінації значення їх істиності.
3. За змінними умовами записуються змінні терми.
4. Для кожної із комбінацій значачень істиності задаються виконуванні терми.
5. Із умов і термів будується таблиця.

Побудова формули:

1. На кожній із комбінацій значень істинності умов віднайти терми, значення істиності яких збігаються.
2. Утворити секвенції з термів із збіжними значеннями істиності.
3. Віднайти умову, за якою виключаються (при різних значеннях істинності умови секвенції мають значення істиності 1) дві утворені секвенції.
4. Виключені секвенції елімінувати.
5. Віднайти умову, за якою виключаються дві утворенні секвенції, або секвенція та побудоване елімінування, чи елімінування з елімінуванням. Проелімінувати.

6. Пункт 4 та 5 виконувати до тих пір, поки не буде побудована формула, яка має всі значення істиності 1 термів таблиці.

Синтез секвенціїним методом супроводжується утворенням, на перщому етапі, формул-секвенцій, якими описуються послідовності термів і, на другому етапі, елімінуванням утворених секвенцій за умовами.

Синтез структури системи генерації програм з операцій теорії секвенційних алгоритмів виконаний як секвенційним так і табличним методом.

В розділі використовуються деякі позначення і поняття теорії секвенційних алгоритмів : секвентування; елімінування; інвертування; $\subset$ - циклічного секвентування ; $\supset$ - циклічного елімінування; | | - паралелення.

Терми – вирази мови. Структура алгоритму – формула утворена, ранговими змінними термами. Секвенти (та) – ознаки порядку. Ранг – встановленна операціами секвентування послідовність і секвент.

Ранговий терм – терм, для якого встановлений його ранг. Константи – терми, значення яких не змінюються. Секвентна область значень – константи з рангами, встановленими операціями секвентування.

Змінні терми, значення яких змінюються. Предметні терми – терми з фіксованими властивостями. Змінні терми – терми, властивості яких не фіксовані.

Мінімізацією формул секвенційних алгоритмів системи генерації програм виконано за кількістю термів за властивостями комутативності, асоціативності, поглинання термів і дистрибутивності операцій теорії секвенційних алгоритмів.

2.1. Принцип побудови системи генерації програм з операцій теорії секвенційних алгоритмів

Система генерації передбачає автоматичну генерацію програм [55,57,64] з формул теорії секвенційних алгоритмів.

Відомо, що актуальною є проблема організації інтерфейсу і підтримки раціонального обміну даними та інформацією. Тому інтерфейс системи генерації праграм з формул теорії секвенційних алгоритмів також повинен забезпечувати обмін інформацією між користувачам і програмою. Для цього інтерфейс системи генерації програм з формул теорії секвенційних алгоритмів повинен містити:

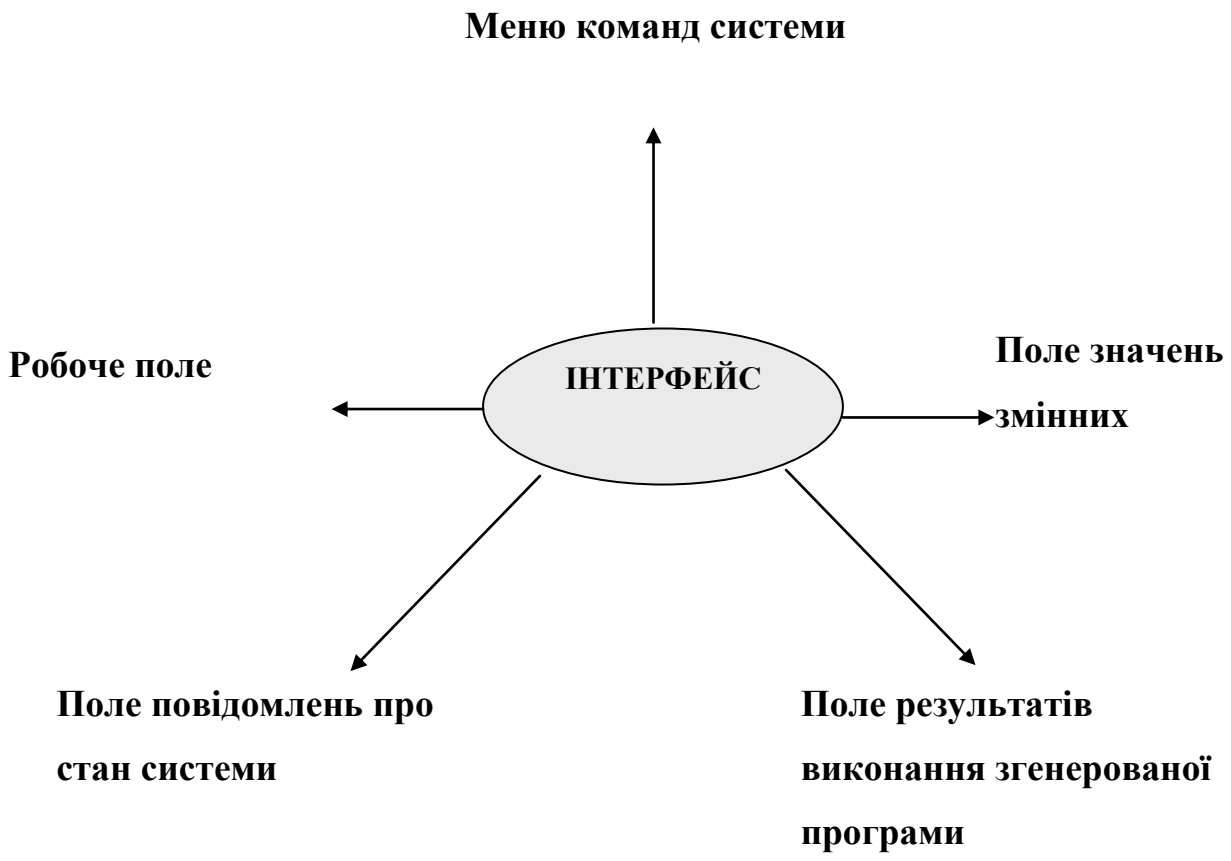


Рис. 2.1. Інтерфейс системи генерації програм з формул теорії секвнеційних алгоритмів

Рис. 2.2. Підсистема перевірки наявності всіх розділів алгоритму

Здійснення генерації передбачає запис алгоритму у вигляді формул теорії секвенційних алгоритмів.

Алгоритм містить ключові слова і саму формулу записану операціями теорії секвенційних алгоритмів [55,57,60,64]. Вхідними даними для процесу генерації є ім”я алгоритму, змінні, секвенційна формула алгоритму, які ідентифікуються такими словами:

алгоритм

змінні

дійсні a,b,c,d;

початок

кінець

Структура системи генерації програм з формул теорії секвенційних алгоритмів містить такі підсистеми:

- підсистему перевіркинаявності всіх розділів алгоритму;
- підсистему ідентифікації та генерування, яка отримує як параметр кожен рядок алгоритму і в залежності від типу рядка виконбує певні дії.

Дана підсистема включає опрацювання рядка з опиисом змінних, опрацювання операції введеня, операції елімінування, секвентування, циклу, операції виведення. Це показано на рис. 2.3.

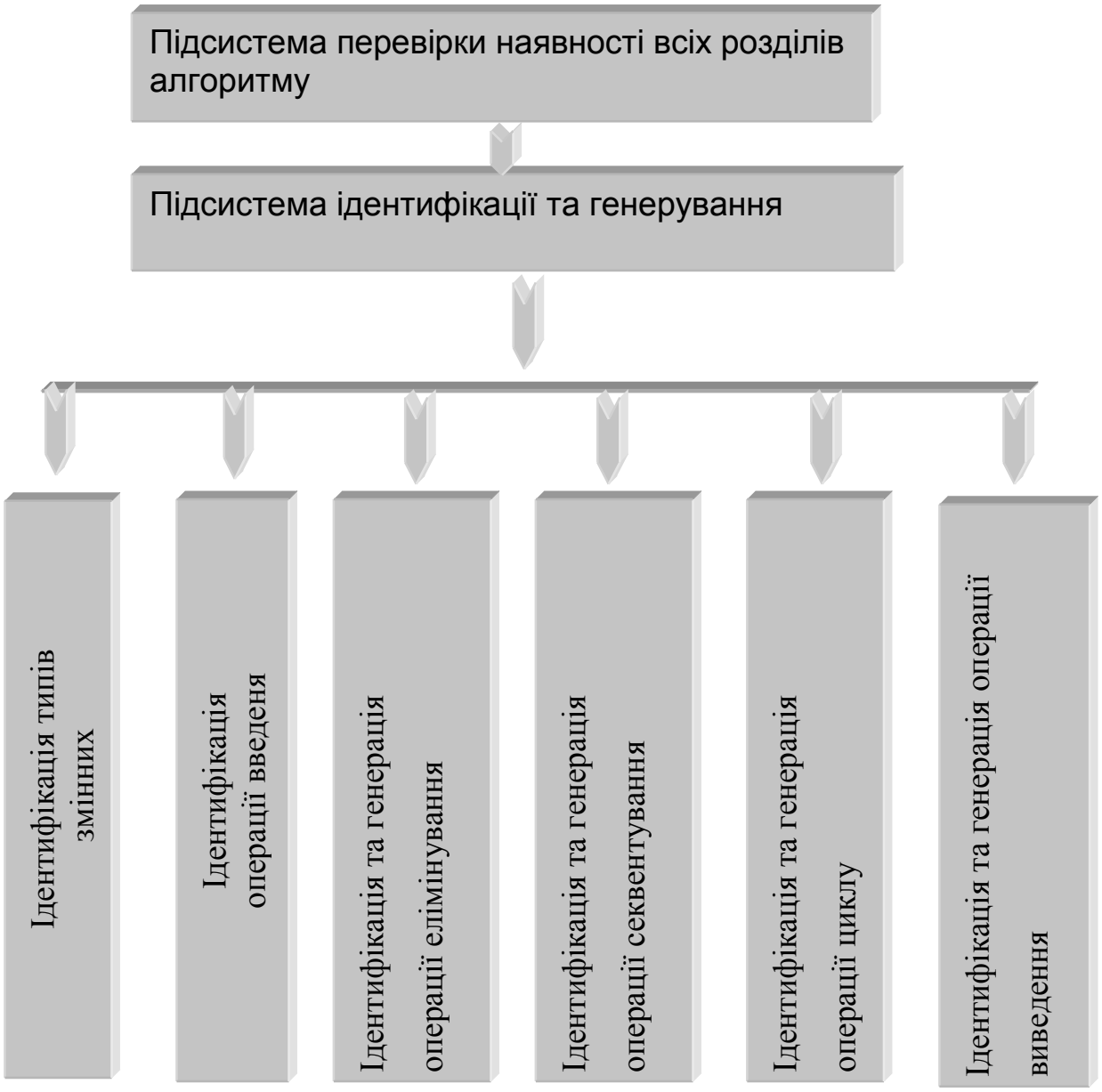


Рис. 2.3. Блок-схема системи генерації програм з формул теорії секвнеційних алгоритмів

2.2. Синтез і оптимізація загальної структури системи

Отже, принцип побудови системи генерації програм з операцій теорії секвенційних алгоритмів передбачає задання типу секвенційних алгоритмів, змінних, секвенційних областей значень змінниих, формули секвенційних алгоритмів; розпізнавання типу операції і генерацію послідовності термів [61,62].

Синтез загальної структури системи генерації програм з операцій теорії секвенційних алгоритмів будемо виконувати секвенційним методом, оскільки формули є значно компактнішими за таблицю, і ними суттєво легше оперувати.

Синтез секвенцій

Насамперед описується ідентифікація і діагностування ключового слова “змінні” та опису самих змінних (S_1). Після чого ідентифікуються операції (терми) введення (S_2), умови (S_3), циклу (S_4), виведення (S_5), присвоєння та виконання обчислень (S_6). У вигляді секвенкій це опишеться так:

$$F_1 = S_1 \; ; S_2 \; ; S_3 \; ; \; S_4 \; ; S_5 ; S_6$$

Якщо в системі відсутня операція виконання обчислень, то опустивши в секвенції F_1 терм S_6 одержимо секвенці F_2.

$$F_2 = S_1 \; ; \; S_2 \; ; \; S_3 \; ; \; S_4 \; ; \; S_5$$

Отже якщо якась із данних операцій відсутня, то опустивши її одержимо секвенції, які запишуться аналогічно до секвенції F_2.

Всього секвенцій буде 64 і це випливає з означення комбінації:

$$C_0^6 + C_1^6 + C_2^6 + C_3^6 + C_4^6 + C_5^6 + C_6^6 = 64.$$

Синтез елімінування

Дві перші секвенції F_1і F_2 елімінуємо за умовою u_6 - чи присутня операція виконання обчислень і присвоєння,матимемо таку формулу:

$$N_1 = \vdash F_1 ,\ F_2 ,\ u_6\text{-}? \dashv .$$

За цією умовою елімінуємо ще такі секвенції.

$$N_n = \vdash F_{2n-1} ,\ F_{2n} ,\ u_6\text{-}? \dashv . \qquad \text{,де } n=1,\ 2,\ ,\ \ldots,\ 32;$$

Елімінуванням формул N_1 та N_2 за умовою u_5- перевірка присутності операції виведення синтезуємо такий вираз:

$$M_1 = \vdash N_1 ,\ N_2 ,\ u_5\text{-}? \dashv .$$

Елімінування N_{2s-1} та N_{2s} , де $s=1,\ 2,\ 3,\ \ldots,\ 16$ елімінуємо за тією самою умовою і одержимо такі елімінування:

$$M_s = \vdash N_{2s-1} ,\ N_{2s},\ u_5\text{-}? \dashv . \qquad \text{,де } s=1,\ 2,\ 3,\ \ldots,\ 16;$$

За умовою u_4-наявність операції циклу елімінуємо такі елімінування M_{2z-1} та M_{2z} ,де $z=1,\ 2,\ 3,\ \ldots,\ 8$. Це описано такою формулою:

$$V_z = \vdash M_{2z-1} ,\ M_{2z} ,\ u_4\text{-}? \dashv . \qquad \text{, де } z=1,\ 2,\ 3,\ \ldots,\ 8;$$

Проелімінувавши за умовою u_3-перевірка присутності операції умови утворенні елімінування одержимо такі формули:

$$P_1 = \vdash V_1 ,\ V_2 ,\ u_3\text{-}? \dashv . \qquad P_3 = \vdash V_5 ,\ V_6 ,\ u_3 \dashv .$$

$P_2 = \vdash V_3 , V_4 , u_3\text{-}? \dashv .$ $\qquad$ $P_4 = \vdash V_7 , V_8 , u_3\text{-}? \dashv .$

За умовою u_2 (перевірка наявності операції введення) елімінуємо P_1 , P_2 та P_3 , P_4, що опишеться такими формулами:

$$R_1 = \vdash P_1 , P_2 , u_2\text{-}? \dashv ,$$

$$R_2 = \vdash P_3 , P_4 , u_2\text{-}? \dashv .$$

Елімінування R_1 і R_2 елімінуємо за умовою u_1-наявність опису змінних, одержимо.

$$D_1 = \vdash R_1 , R_2 , u_6\text{-}? \dashv .$$

На цьому синтез загальної структури системи генерації програм з операцій теорії секвенційних алгоритмів завершено.

Мінімізація елімінування.

Розглянемо мінімізацію елімінування

$$N_1 = F_1 , F_2 , u_1\text{-}?$$

На основі підстановок замість F_1 і F_2 у відповідні секвенції отримаємо таке співвідношення (формулу):

$$\left(\left(\left(S_1, ; S_2\right) ; \left(S_3 ; S_4\right) ; \left(S_5 ; S_6\right)\right)\right) \left(\left(S_1, ; S_2\right) ; \left(S_3 ; S_4 ; S_5\right)\right) \quad u_1\text{-?}$$

Застосувавши до останнього виразу п’ять разів властивість винесення переднього символу за знак елімінування, отримаємо таку формулу:

$$\left(\left(S_1 ; S_2\right) ; \left(\left(S_3 ; S_4 ; S_5 ; S_6\right), *, u_1\text{-?}\right)\right)$$

На цьому мінімізація даного елімінування завершена.

Аналогічно проводим мінімізацію синтезованого виразу, в результаті чого отримаємо таку формулу:

$$\left(\left(\left(\vdash S_1\,,\,*\,,\,u_1\text{-}?\,\right.;\left.\vdash S_2\,,\,*\,,\,u_2\text{-}?\right);\left(\vdash S_3\,,\,*\,,\,u_3\text{-}?\,\right.;\left.\vdash S_4\,,\,*\,,\,u_4\text{-}?\right)\right);\left(\vdash S_5\,,\,*\,,\,u_5\text{-}?\,\right.;\left.\vdash S_6\,,\,*\,,\,u_6\text{-}?\right)\right)$$

Мінімізацією зменшено кількість термів у загальній структурі системи генерації програм з формул теорії секвнеційних алгоритмів в 5,7 рази.

Синтез підсистеми опису змінних

Синтез підсистеми опису змінних [55,60,64] проводимо за допомогою табличного методу.

Для цього складаємо таблицю істиності. В ній будемо мати дві змінні умови u_{11}, u_{12}. Умови виконуються або ні. Тому їх значення істиності є $2^2=4$.

Задамо такі змінні терми $P(b,\ c_1)$, $P(i,\ c_1)$, $P_0(p,\ c_1)$, c_v, C_iF_i і порожній терм. Для кожної із комбінацій значень істиності умов задаємо 1 або 0 для символів-операторів.

Утворюємо такі секвенції

$$B_3 = \overbrace{P(b, c_1)\ ,\ c_v}$$

$$B_4 = \overbrace{\overbrace{P(b, c_1)\ ,\ P(i, c_1)}\ ,\ C_iF_{i.}P(b, c_1)}$$

При невиконанні умови u_{11}, отримаємо порожній символ тому будемо мати $B_1 = B_2 = *$.

Секвенції B_3 *і* B_4 виключаються за умвою u_{12}. За цією умовою виключаються і секвенції. B_1 , B_2.

Синтезовані елімінування виключаються за умовою u_{11}. Проелімінувавши формули за даними умовою будемо мати такий синтезований алгоритм:

$$\vdash \overbrace{B_4\ ,\ B_3.,\ u_{12}\text{-}?}\ ;\ \overbrace{B_1\ ,\ B_2.,\ u_{12}\text{-}?}\ ;\ u_{11}\text{-}? \dashv .$$

Підставивши замість B_1, B_2, B_3, B_4 відповідні секвенції одержимо.

$$\vdash \overbrace{\left(\begin{array}{l} P(b, c_1)\ , \\ , \\ P(i, c_1) \end{array}\right.\ \left(\begin{array}{l} P_0(p, s) \\ \\ c_v \end{array}\right.\ u_{12}\text{-}?}\ ,\ \overbrace{*\ ,\ *\ ,\ u_{12}\text{-}?}\ ,\ u_{11}\text{-}? \dashv .$$

Провівши мінімізацію одержимо.

$$\vdash \overbrace{\left(\left(\begin{array}{l} P(b, c_1)\ , \\ , \\ P(i, c_1) \end{array}\right.\ \left(\begin{array}{l} P_0(ps, s) \\ , \\ c_v \end{array}\right.\ u_{12}\text{-}?\right.}\ ,\ *\ ,\ u_{11}\text{-}? \dashv .$$

$$\left.\begin{array}{l} , \\ C_iF_{i.} \end{array}\right.$$

Синтезуємо такі секвенції, які проелімінуємо за відповідними умовами, отримаємо таку формулу:

$$M_8, M_7, u_{15}\text{-}?\ ,\ M_6, M_5, u_{15}\text{-}?, u_{14}\text{-}?\ ,\ M_4, M_3, u_{15}\text{-}?\ ,\ M_2, M_1, u_{15}\text{-}?, u_{14}\text{-}?, u_{14}\text{-}?\ \text{де}$$

$$M_8 = M_7 = P_1(p)\ ,\ P(b_{11}, c_0)\ ,\ P(b, c_0)$$

$$M_6 = M_5 = c_i$$

$$M_4 = M_2 = D(n, j)\ ,\ P(V, p)\ ,\ P_0(p, s)\ ,\ c_z$$

$$M_3 = M_1 = P_0(p, s)\ ,\ c_z$$

Провівши мінімізацію одержаної формули одержимо:

$$M_8\ ,\ M_6\ ,\ u_{14}\text{-}?\ ,\ M_4\ ,\ M_3\ ,\ u_{15}\text{-}?\ ,\ u_{13}\text{-}?$$

Мінімізацією зменшено кількість термів у підсистемі опису змінних системи генерації програм з формул теорії секвнеційних алгоритмів в 1,7 раза.

2.3. Структура генерування програм з операції елімінування

Синтез струкрури генератора з операції еламінування проводимо табличним методом .

Записуємо умовну змінну і змінні терми [55,57,63].

Для кожної із значень істиності умови задамо 1 або 0 для символів-операторів.

В результаті матимемо таку таблицю (2.1).

Таблиця. 2.1.

Таблиця істиності

u_{31}	S_{31}	$D_1(p_1)$	$D_1(p_2)$
0	1		1
1	1	1	

В ній збігаються значення істиності таких термів S_{31}, $D_1(p_1)$ *і* S_{31}, $D_1(p_2)$

Утворюємо з термів таблиці секвенції:

$$\overbrace{S_{31}, D_1(p_1)} \qquad \overbrace{S_{31}, D_1(p_2)}$$

Проелімінувавши їх за умовою отримаємо формулу

$$\vdash \left(\begin{matrix} S_{31} \\ , \\ D_1(op1) \end{matrix} \right. \left(\begin{matrix} S_{31}, \\ , \\ D_1(op2) \end{matrix} \right. u_{31}\text{-?} \vdash .$$

Мінімізуємо даний алгоритм.

За властивістю операцій теорії секвенційних алгоритмів здійснюються такі перетворення.

$$\left(\begin{matrix} S_{31} \\ , \\ \vdash D_1(op1) \quad D_1(op2) \quad u_{31}\text{-?} \vdash \end{matrix} \right. .$$

Зауважимо що S_{31} –відповідна секвенція.

2.4. Структура синтаксису алгоритму

Синтез загальної структури системи перевірки наявності всіх розділів алгоритму. Даний синтез описується такими секвенціями:

Синтез секвенцій

$$A = P(g_1,c_0)\ ,\ P(g_2,c_0)\ ,\ P(g_3,c_0)\ ,\ P(g_4,c_0)$$

Секвенція перевірки першого рядка запишеться так:

$$S_1 = A;\ B\ ;\ P_1(s_1)$$

Аналогічно запишуться і наступні секвенції

$$S_2 = A;B;\ P_1(s_2)\ ; \qquad S_3 = A;\ B\ ;\ P_1(s_3)\ ,$$

$$S_4 = A;\ B\ ;\ P_1(s_4); \qquad S_5 = A;\ B\ ;\ P_1(s_5)$$

Ще одна секвенція запишеться так

$$S_6 = A\ ;\ B\ ,\ \text{де}\ B = Ci\ P(g_i,c_1).$$

Тепер проведемо синтез елімінування.

Секвенції S_4 *і* S_5 елімінуємо за умови u_{41}.

Проелімінувавши, за умовою u_{31} формулу і секвенцію S_3 , отримаємо вирази :

$$\left(\left(A\ ;\ B\right);\ P_1(s_4)\right),\ \left(\left(A\ ;\ B\right);\ P_1(s_5)\right),\ u_{41}\text{-?}$$

$$A\ ;\ B\ ;\ P_1(s_3),\ A\ ;\ B\ ;\ P_1(s_4),\ A\ ;\ B\ ;\ P_1(s_5),\ u_{41}\text{-?},\ u_{31}\text{-?}$$

Перевіряючи наступні умови u_{21} та u_{11} , одержано елімінування, описане громісткою формулою, тому наступний крок – оптимізація.

Розглянемо мінімізацію останньої формули згідно з інформацією, викладеною у працях [55,60,64].

Застосувавши до даної формули властивість винесення переднього символу за знак елімінування, отримаємо вираз.

$$\left(\left(A\ ;\ B\right);\ P_1(s_4),\ P_1(s_5),\ u_{41}\text{-?}\right)$$

На цьому оптимізація даного елімінування завершена.

Аналогічно проводиться оптимізація інших формул. Отже, оптимізований алгоритм буде мати вигляд:

A;

B ;

$P_1(s_1), P_1(s_2), P_1(s_3), P_1(s_4), P_1(s_5), u_{41}$-? , u_{31}-?, u_{21}-?, u_{11}-?

Мінімізацією зменшено кількість термів у системі синтаксису системи генерації програм з формул теорії секвнеційних алгоритмів у 2,3 рази.

2.6. Висновки до другого розділу

1. Розроблено принцип побудови системи генерації програм з формул теорії секвенційних алгоритмів, який передбачає задання змінних і секвенційних областей їхніх значень, а також формул для секвенційного алгоритму, що забезпечують розпізнавання типу операції, генерацію послідовності операторів програми.

2. Синтизовано та мінімізовано за кількістю термів структуру системи генерації програм з операцій теорії секвенційних алгоритмів, яка включає операцію секвентування, операцію елімінування та операцію циклічного елімінування.

Розділ III

МОДЕЛЮВАННЯ СИСТЕМИ ГЕНЕРАЦІЇ ПРОГРАМ З ФОРМУЛ ТЕОРІЇ СЕКВЕНЦІЙНИХ АЛГОРИТМІВ

Згідно інформації, викладеній у працях [39,52,63,69,70], математичні моделі корозіометрії охоплюють клас абстрактних, символьних математичних об’єктів, таких як числа або вектори, і відношення між цими об’єктами. Математичні відношення – це гіпотетичні правила, що зв’язують два або більше символьних об’єктів. Багато відношень можуть бути описані з допомогою математичних операцій, що зв’язують один або декілька об’єктів з іншим об’єктом або множиною об’єктів.

Під математичним моделюванням розуміється процес встановлення відповідності певної системи чи процесу [52,63,70] деякому математичному об’єкту, що називається математичною моделлю, і на основы выдповыдних математичних спыввыдношень проводимо дослідження цієї моделі.

Моделлю деякої системи або об’єкта [39,69] називається штучна система, що може замінити систему (оригінал чи об’єкт). Алгоритм, тобто певна послідовність дій, також є моделлю, яка відтворює життэвий цикл певної діяльності. Алгоритми складних систем будуються в два етапи. На першому етапі створюється спрощений алгоритм, закладається головна ідея, будується його узагальнена структура, яка описується формулою теорії секвенційних алгоритмів, утвореною змінними термами.

На основі впорядкування операцій теорії секвенційних алгоритмів змінних термів утворюється структура алгоритму [47-53].

Структура алгоритму, всі змінні терми якої замінені предметними термами так, що будь який змінний терм від n змінних замінюється предметним термом з кількістю змінних m, є такою, що для неї $m>n$, а для кожної змінної та предметного терму задані секвентні області значень. Отриманий результат відповідає моделі структури алгоритму.

Модель структури алгоритму, всі рангові терми якої приймають повний ансамбль можливих значень тільки із заданих рангових секвентних областей значень, називається вірогідною.

Математична модель структури [39,63,70] будується з використанням операції заміни змінних термів структури предметними і заданням секвентних областей значень як незв'язаних операцій циклів так і зв'язаних, а також предметних термів.

Дослідженням математичної моделі структури, ще до практичної реалізації та апробації виявляються допущені на етапі синтезу структури і побудови її моделі помилки. Дослідження виконується за принципом математичної індукції. Формулюється він так: твердження $T(x)$, де x пробігає натуральні числа , має місце для будь-яких натуральних чисел, якщо:

а) встановлено, що воно справджується для початкового значення x;

б) із припущення цого справедливості для якого-небудь n встановлено, що воно має місце і для $T(n+1)$.

Якщо $T(x)$ для початкового та $(n+1)$-го значення доводиться аналогічно, то має місце трансфінітна індукція, яка формулюється так: твердження $T(x)$ має місце для будь-якого натурального x , якщо із припущення його істиності для будь-якого натурального $x<n$ витікає, що $T(x)$ істине і при $x=n$.

Бувають випадки, як правило, в моделях складних систем, в яких кількість понять і тверджень, означуваних і доводимих сукупною математичною індукцією, записується тризначним числом. В таких випадках використання принципу математичної індукції називається доведенням складною сукупною індукцією.

Методика побудови математичної моделі алгоритму містить такі кроки [39,53,63]:

1 Описати кожне розгалуження окремим секвентуванням.

2 Із кожного секвентування виділити його початкові дані.

3 Виписати одним (інтегральним) секвентуванням початкові дані всіх простих секвентувань.

4 Замінити початкові дані кожного окремого простого секвентування інтегральними початковими даними.

5 Якщо допустити паралелення, прості впоряднування паралеляться. Якщо ж паралелення недопустиме, то встановити ознаки за якими розрізняються прості секвентування за встановленими ознакоми виключного впорядкування.

Одним із методів виявлення помилок є математичне моделювання. Процес математичного моделювання трактується як вивчення явищ з допомогою математичних моделей [69,70].

Для синтезу формул, алгоритмів інформаційно-технологічних систем розробленні секвенційний і табличний методи синтезу. А дослідження математичних моделей секвенційних алгоритмів виконувалось методом математичної індукції. Результатом такого дослідження є встановлення вірогідності математичної моделі, яка трактується як обмеження значень змінних і термів значеннями їх секвентних областей.

Трансфінітний метод дослідження алгоритмів: Подана формолою теорії секвенційних алгоритмів і залежна від натурального параметра *n* модель алгоритму *A(n)*, якою описується процес або явище *F(n)* досліджується так:

1. Встановити, що для початкового значення *n* модель алгоритму *A(n)* описується процес або явище *F(n).*

2. Виходячи із припущення опису моделлю алгоритму процесу або явища для будь-якого i , такого, що i>m, встановити опис моделлю опису або явищща для *i=m*.

Якщо *A(x,y)* – модель алгоритму, яка залежить від двох змінних і описує процес або явище *F(x,y)*, то вона досліджується так:

1. Встановити, що для початкових значень x та y модель алгоритму *A(x,y)* описується процес або явище *F(x,y).*

2. Для початкових значень *y* та будь-яких значень *x* встановити, що моделлю алгоритму *A(x,y)* описується процес або явище *F(x,y).*

3. Для початкових значень *x* та будь-яких значень *y* встановити, що моделлю алгоритму *A(x,y)* описується процес або явище *F(x,y)*.

4. Встановити, що для яких-небудь значень *x* та *y* моделлю *A(x,y)* описується процес або явище *F(x,y)*.

Подібним способом встановлюють опис моделлю *A(x,y,…,z)* процесу або явища *F(x,y,…,z)*.

3.1. Побудова і дослідження моделі генерації програм з операцій секвентування

У системі генерації програм з формул теорії секвенційних алгоритмів реалізовано підмножину операцій теорії секвенційних алгоритмів [67-70], а також для зручності користувача вводимо деякі допоміжні функції.

Загальна структура системи передбачає розпізнавання типу операцій, і в залежності від типу операції виконуються певні обчислення [63,64].

Дана система містить такі операції [67,68]:

- опису змінних;
- введення данних;
- елімінування;
- циклу;
- секвентування;
- виведення данних;

Секвенційним методом побудовано модель системи генерації програм [63,64,67,68] з формул теорії секвенційних.

Дана модель має такий вигляд:

$$
\left(\left(\left(\left(\left(\vdash\!\!\overline{\quad S_1\ ,\ *\ ,\ u_1\text{-}?\quad}\right|.\ ;\ \vdash\!\!\overline{\quad S_2\ ,\ *\ ,\ u_2\text{-}?\quad}\right|.\right)\ ;\ \vdash\!\!\overline{\quad S_3\ ,\ *\ ,\ u_3\text{-}?\quad}\right|.\right)\ ;\ \vdash\!\!\overline{\quad S_4\ ,\ *\ ,\ u_4\text{-}?\quad}\right|.\right)\ ;\ \vdash\!\!\overline{\quad S_5\ ,\ *\ ,\ u_5\text{-}?\quad}\right|.\ ;\ \vdash\!\!\overline{\quad S_6\ ,\ *\ ,\ u_6\text{-}?\quad}\right|.
$$

Побудову моделі виконуємо заміною змінних термів предметними із заданням змінним та термам секвентних областей їх значень; а також заміною умовних термів, конкретними термальними умовами.

Двомісний терм *P(p, s)* - заміняємо термальним оператором присвоєння, який позначимо :=.Отже змінній p присвоюється значення s.

$P(s_1, s)$ означає– $s_1=s$;

$P(b, c_0)$ – $b=c_0$ c_0 приймає значення константи, $c_0=0$.

Для b задамо таку секвентну область значень $b_i \in Q_1 = \overset{\frown}{0\ ,\ 1}$

$s, p, s_1 \in Q_2$ де

$$
Q_2=\overset{\frown}{\overset{\frown}{\overset{\frown}{e_1\ ,\ \overset{\frown}{e_2 e_3}}\ ,\ \overset{\frown}{e_4 e_5 e_6\ ,\ e_7 e_8 e_9 e_{10}}}\ ,\ e_{11}\text{''},\text{''}e_{12}\ ,\ e_{13}\text{''},\text{''}e_{14} e_{15}}\ ,\ \ldots\ ,e_n e_{n+1} e_{n+2} e_{n+3}
$$

де $e_1, e_2, e_3, \ldots \in Q_3$

$$Q_3 = 0\ ,\ 1\ ,\ 2\ ,\ \dots\ ,\ 9\ \ ;\ \ a\ ,b\ ,\ \dots\ ,\ z;$$

Змінний умовний терм u_1 замінюємо конкретною термальною умовою (p=*”дійсні”*)-?

Аналогічно замінюємо і наступні умовні терми:

u_2-? – (p=*”ввести”*)-?;

u_3-? – (p=*” ____”*) -?;

u_4-? – (p=*”$”*) - ?;

u_5-? – (p= *“вивести”*) - ?;

u_6-? – (b_1=c_0)&(p=0)-?.

S_1 – підсистема опрацювання рядка опису змінних;

S_2 – підсистема опрацювання рядка з оператором введення;

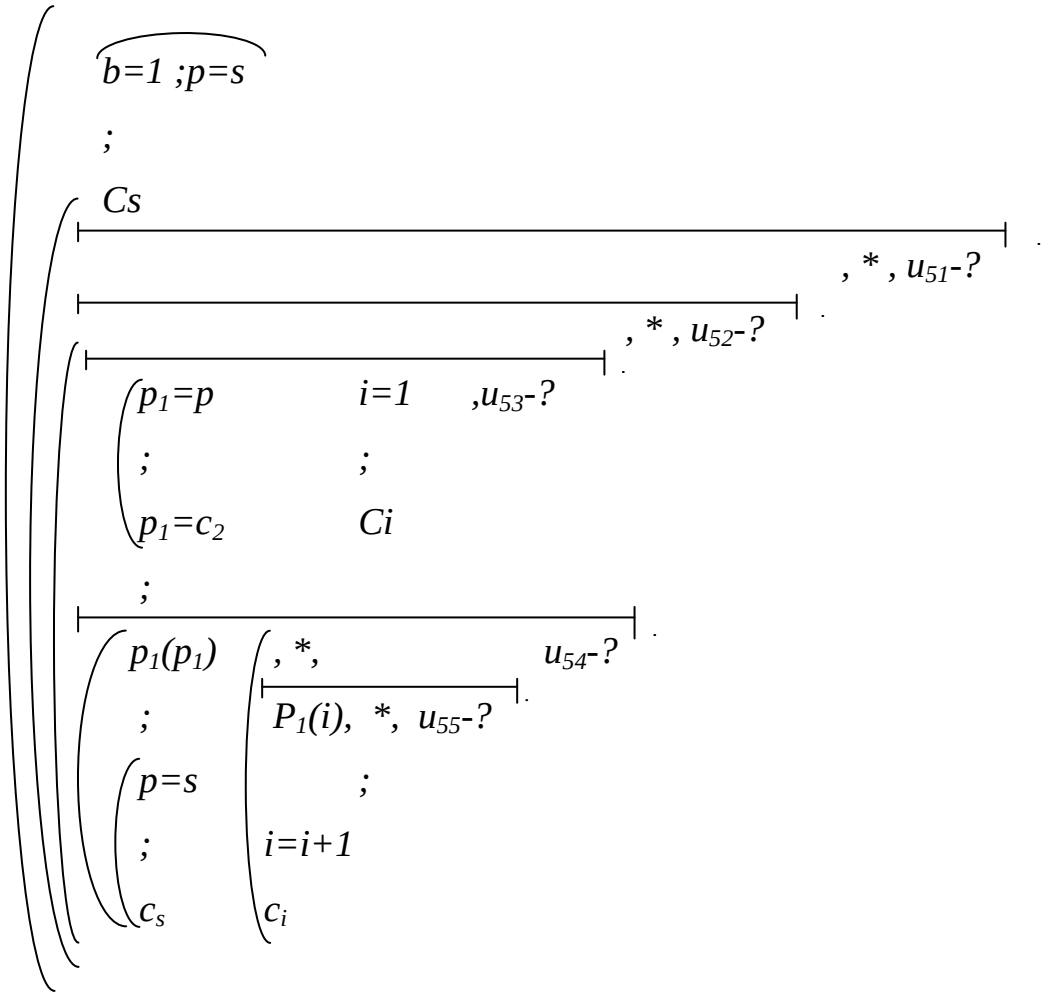
S_3 – підсистема опрацювання рядка з оператором умови;

S_4 – підсистема опрацювання рядка з оператором циклу;

S_5 – підсистема опрацювання рядка з оператором виведення;

S_6 – підсистема опрацювання рядка виклнання обчислень і присвоєння.

Підсистема опрацювання рядка оператора [63,64,67] введення S_2 описується такою формулою:



Одномісний терм $P(p_1)$ замінюємо термом виводу на друк значення змінної p_1.

Відповідно і наступні підсистеми описуються певними формулами.

Із алгоритму підстановкою змінних терм, введених операцій отримаємо математичну модель підсистеми генерації програм.

S_1 , * , (p=”дійсні”)-?

;

S_2 , * , (p=”ввести”)-?

;

S_3 , * , (p=” ____ ”) -?

;

S_4 , * , (p=”$”) - ?

;

S_5 , * , (p= “вивести”) - ?

;

S_6 , * , (b_1=c_0)&(p=0)-?

Твердження: остання формула описує генерацію програм з операцій теорії секвенційних алгоритмів [63,64,67].

Послідовність моделювання структури алгоритму зображенно на рис. 3.1.

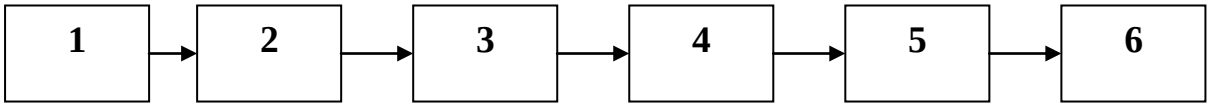


Рис. 3.1. Послідовність етапів моделювання

де 1 – блок моделювання підсистеми генерації програм з операцєю опису змінних;

2 – блок моделювання підсистеми генерації програм з операцєю введення данних;

3 – блок моделювання підсистеми генерації програм з операцєю елімінування;

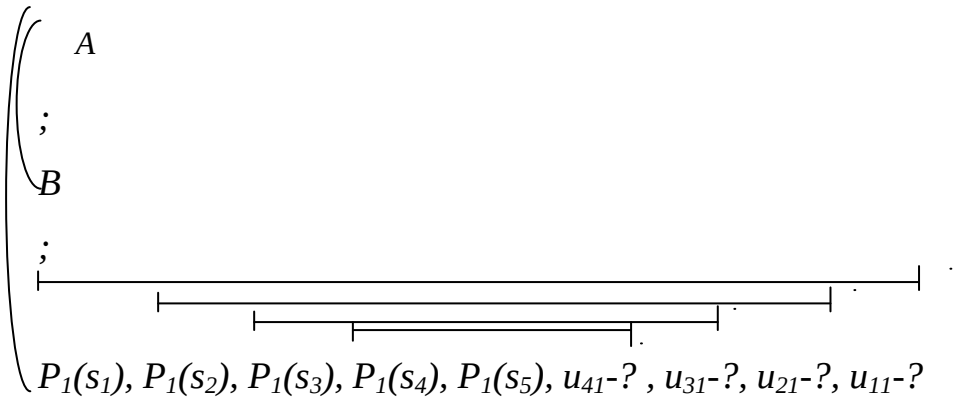
4 – блок моделювання підсистеми генерації програм з операцєю циклу;

5 – блок моделювання підсистеми генерації програм з операцєю секвентування;

6 – блок моделювання підсистеми генерації програм з операцєю виведення данних.

Перевірка наявності всіх розділів алгоритму системи

Алгоритм системи перевірки наявності всіх розділів алгоритму системи має вигляд [63,64,68].



Виконуємо побудову моделі, заміною змінних термів предметними із заданням змінним та термам секвентних областей їх значень; а також заміною змінних умов термальними умовами.

Двомісний терм $P(g_i, c_o)$ - заміняємо термальним оператором присвоєння, який позначимо :=;

Нехай в операторі $g_i := c_o$, значення $c_o \in 0$.

Змінній g_i терму $P(g_i, c_o)$ для задамо таку секвентну область значень

$$g_i \in Q_1 = 0 , 1 \;;\; де \quad i \in Q_2 = 1 ; 2 ;3;4$$

Одномісний терм $P_1(s)$, заміняємо термом виводу на друк значення s.

де s_1 – константа “відсутність заголовку”;

s_2 – константа “відсутність розділу змінних”;

s_3 – константа “відсутність початку алгоритму”;

s_4 – константа “відсутністькінця алгоритму”;

s_5 – константа “всі розділи алгоритму присутні".

Умовний терм u_{41} заміняємо термальноюумовою $(g_4=0)$-?;

Умовний терм u_{31} заміняємо термальною умовою $(g_3=0)$-?;

Умовний терм u_{21} заміняємо термальною умовою $(g_2=0)$-?;

Умовний терм u_{11} заміняємо термальною умовою $(g_1=0)$-?.

Із алгоритму підстановкою змінних терм, введених операцій отримаємо математичну модель підсистеми перевірки всіх розділів алгоритмів, яка наведена формулою

$$
\begin{cases}
\overbrace{\overbrace{g_1:=0,\ g_2:=0}\ ,\ \overbrace{g_3:=0,\ g_4:=0}} \\
; \\
Ci \\
\overline{g_i:=1,\ c_i\ ,\ \ u_i\text{-?}} \\
; \\
P_1(s_1),\ P_1(s_2),\ P_1(s_3),\ P_1(s_4),\ P_1(s_5),\ g_4=0\text{-?}\ ,\ g_3=0\text{-?}\ ,\ g_2=0\text{-?}\ ,\ g_1=0\text{-?}
\end{cases}
$$

Встановимо чи моделю описується перевірка наявності всіх розділів алгоритму.

Дослідження виконуємо методом трансфінітної індукції. Доводимо при початкових значеннях змінних. після виконання чотирьох початкових термів $(g_1=0\ ,\ g_2=0\ ,\ g_3=0\ ,\ g_4=0)$. Ці значення належать секвентній області Q_1. Після цього виконується цикл за змінною i що належить секвентній області Q_2. Всього можливих комбінацій може бути $4!=24$.

Розглянемо випадок коли $g_1=0\ ,\ g_2=0\ ,\ g_3=0\ ,\ g_4=0$.Ці значення належать секвентній області Q_1. Отже в даному випадку моделю описується перевірка наявності всіх розділів алгоритму.

Припустимо, що для інших значень g_i модель описує перевірку наявності всіх розділів алгоритму.

Перевіримо чи моделю буде описуватися перевірка всіх розділів алгоритму коли $g_1=1$, $g_2=1$, $g_3=1$, $g_4=1$. Всі ці значення належать секвентній області Q_1. І на екрані буде виведено повідомлення “*Всі розділи алгоритму присутні*”. Отже дана модель описує перевірку наявності всіх розділів алгоритму.

3.2 Блок-схема системи генерації програм з формул теорії секвенційних алгоритмів

Блок-схема системи генерації програм з формул теорії секвенційних алгоритмів [63,64,67,68] будується на основі загальної структури системи генерації програм з формул теорії секвенційних алгоритмів.

Дана блок-схема системи генерації програм з формул теорії секвенційних алгоритмів складається з шести основних блоків , а саме:

- блоку перевірки службових слів;
- блоку перевірки оператора введення;
- блоку перевірки оператора виведення;
- блоку перевірки операції присвоєння;
- блоку перевірки операції елімінування;
- блоку перевірки операції циклу.

Блок перевірки оператора введення забезпечує діалогове вікно для введеня змінних, зупиняється обробка оператора, при знаходженні “;”.

Блок перевірки оператора виведення забезпечуе виведення конкретного значення (число) або символьного рядка на екран.

Блок перевірки операції присвоєння служить для генерації операції секвентування.

Блок перевірки операції елімінування здійснює генерацію операції елімінування.

Блок перевірки операції циклу служить для генерації операції циклу.

Програмна реалізація математичної моделі системи

Створення програми в Delphi базується на використанні об”єктно-орієнтованої технології. Структурною одиницею візуального програмування є компонент. Компонент [63,64,67] предстевляє різновид об”єкта, який можна переносити в програму з спеціальної палітри компонентів, а також має набар властивостей, які можна визначити, не змінивши вихідний код програми. Компонент містить методи-обробники події, кожний з яких відповідає за реалізацію компонента на деяку подію. Компоненти є візуальні та невізуальні. Візуальні призначені для організації інтерфейсу з користувачем. невізуальні компоненти відповідають за доступ до системних ресурсів: драйверів, протоколів DDE, OLE.

Головні риси об”єктно-орієнтованого програмування [8,18,29] це механізми інкапсуляції, спадковості, динамічного зв”язування функцій.

Проблема організації інтерфейсу для підтримки раціонального обміну даними та інформацією є дуже актуальною. Це пояснюється ускладненням архітектури обчислювальних систем, режимами управління процесами обміну та обробки даних. Є необхідність відповідно до формалізованої мови, що включає поняття; відношення та формулювання основних задач предметної оюласті розробити раціональний інтерфейс.

Для цього пропонується використати експертну оцінку [60,61,65] спеціалістів, що включає:

1. Розробку змісту та структуру анкети для експерта.
2. Залучення необхідної кількості спеціалістів-експертів.
3. Аналіз відповідей в анкетах.
4. На основі отриманих результатів методу експертних оцінок розробка інтерфейсу для алгебри алгоритмів.
5. Розробка алгоритмічного та програмного забезпечення предметної області.
6. Реалізація задач та аналіз результатів.

7. Можливість корекції програмного забезпечення в залежності від результатів проведеного експерименту.

Робота з експертом полягає у зборі та введені їх оцінок, аналізу даних, а також у знаходженні раціонального рішення. Як відомо сучасні інформаційні технології **[43,74,100]** складають бази даних, системи управління ними, експертні системи для зберігання та обробки інформації.

Для реалізації зручного інтерфейсу, надійності роботи, сумісності з іншими програмами було вибрано інструментальні засоби **[20,94]** системи розробки програм Delphi фірми Borland, що працює під операційною системою Microsoft Windows.

Головне вікно системи генерації програм з операцій теорії секвенційних алгоритмів **[63-65]** створюємо на базі екранної форми мови Delphi, яка служить контейнером для компонентів – елементів управління. Екранна форма є найбільш важливим об”єктом Delphi, і в проекті Delphi представляє не тільки зовнішній вигляд, але і є повноцінним компонентом, який володіє власними властивостями і подіями.

Складові системи генерації програм з операцій теорії секвенційних алгоритмів:

Файл MDIFrame. Робота головного вікна програми.

Файл MDIEdit. Робота дочірних вікон і процедури інтерпритатора.

Процедури інтерпритатора.:

Процедура перевірки навності всіх розділів алгоритму.

Процедура, яка отримує як параметр кожен рядок алгоритму і в залежності від типу рядка виконує певні дії. вона включає:

- ✓ опрацювання рядка опису змінних;
- ✓ опрацювання оператора введення;
- ✓ опрацювання рядка з операцією елімінування;
- ✓ опрацювання рядка з операцією циклу;
- ✓ опрацювання рядка з операцією секвентування;
- ✓ опрацювання оператора виведення;

а також:

- ✓ функцію, що розбиває рядок на окремі слова і розділювачі;
- ✓ функцію, що опрацьовує рядок умови.

Програма **[63-65]** включає:

- процедуру, що ініціалізує виконання алгоритму;
- функцію, що виконує присвоєння та обрахунок виразів;
- функцію, яка розпазнає стандартні ідентифікатори;
- процедуру, показує поточні значення змінних у основному вікні;
- процедуру, що запускає на виконання попередню функцію;
- функцію, що перетворює дійсні числа до вигляду, який зрозумілий в середовищі Delphi.

Процедура перевірки наявності всіх розділів алгоритму [63,64] подана у вигляді фрагменту програми:

```
procedure TEditForm.Testing1Click(Sender: TObject);
var
i,poch:integer;
g1,g2,g3,g4,gg:boolean;
begin
FrameForm.Memo1.Lines.Clear;
FrameForm.Memo2.Lines.Clear;
FrameForm.Memo3.Lines.Clear;
FrameForm.Memo2.Lines.Add('Результати:');
FrameForm.Memo1.Lines.Add('Помилки і підказки:');
FrameForm.Memo3.Lines.Add('Змінні:');
g1:=false;
g2:=false;
g3:=false;
g4:=false;
for i:=0 to editor.Lines.Count do
begin
if editor.Lines.Strings[i]='алгоритм' then g1:=true;
if editor.Lines.Strings[i]='змінні' then g2:=true;
if editor.Lines.Strings[i]='початок' then
begin
g3:=true;
```

```
poch:=i;
end;
if editor.Lines.Strings[i]='кінець' then g4:=true;
end;

if (g1=false) then FrameForm.Memo1.Lines.Add('Відсутній заголовок');
if (g2=false) then FrameForm.Memo1.Lines.Add('Відсутній розділ змінних');
if (g3=false) then FrameForm.Memo1.Lines.Add('Відсутній початок алгоритму');
if (g4=false) then FrameForm.Memo1.Lines.Add('Відсутній кінець алгоритму');
if ((g1=true)and(g2=true)and(g3=true)and(g4=true)) then
FrameForm.Memo1.Lines.Add('Всі розділи алгоритму присутні');

end;
```

3.3. Методика роботи з системою

Програма системи генерації програм з операцій теорії секвенційних алгоритмів створена для практичного використання на комп'ютері теорії секвенційних алгоритмів. У програмі [60] реалізовано лише підмножину операцій вказаної теорії, а також для зручності користувача введено деякі допоміжні функції.

Нижче приведено опис синтаксису алгоритму для використання у програмі :

<алгоритм>::=алгоритм змінні
<список_змінних>початок<тіло_алгоритму>кінець
<список_змінних>::=дійсні <змінні>
<змінні>::=<змінна>[,<змінна>];
<тіло_алгоритму>::=<оператор>|<оператор>[,<оператор>]
<оператор>::=<оп_вводу>|<оп_виводу>|<оп_присвоєнння>|<оп_циклу>|<оп_умови>
<оп_вводу>::=ввести <змінна>|[,<змінна>];
<оп_виводу>::=вивести <змінна>|[,<змінна>];

<оп_присвоєння>::=<формула>;

<оп_циклу>::=|- <формула>,<формула>,<умова> -|;

<оп_умови>::=$ <змінна>|<ціле> <формула>[,<формула>];

<формула>::= <змінна>=<оп_1>|<змінна>=<оп_1><операція><оп_2>;

<операція>::=+ | - | * | /

<умова>::=<оп_1> <зн_нерівності> <оп_2>

З даного опису випливає що структура алгоритму [63-65], написаного для виконання у програмі, виглядає наступним чином:

алгоритм

змінні

дійсні a,b,c,d;

початок

кінець

Попередній фрагмент є прикладом коректного опису структури алгоритму, але ні до яких дії не призведе. Щоб алгоритм виконував певні дії потрібно дописати рядки з операторами [53,54]. Наприклад:

алгоритм

змінні

дійсні a,b,c,d;

початок

ввести a,b,c,d;

вивести a,b,c,d;

кінець

Далі наведено приклад алгоритму з використанням всіх можливих опереторів:

алгоритм

змінні

дійсні a,b,c,d,r,g,h,t,q,w,a1,b1,b2,b3,b4,b5;

дійсні aaa,bbbb,cccc;

початок

ввести a,b,c,d;

r=1.4;

r=-sin_a+cos_b;

вивести r;

вивести &Перевірка_процедури_DoIt,r;

|- r=a, r=b, a > b -|;

вивести &Перевірка_оператора_|-.....-|,r;

r=1;

*$ d r=r*a , a=a+1;*

вивести &Перевірка_оператора_$_x.....,r;

кінець

Зауваження до даного фрагменту:

1. Рядок з присвоєнням, і взагалі присвоєння які використовуються у інших операторах, не повинен містити пробілів.

2. Рядок умови в операторі перевірки умови пишеться через пробіли, тобто спочатку перше значення потім пробіл, потім знак нерівності , знов пробіл і друге значення.

3. При використанні тригонометричних функцій параметр цих функцій пишеться через знак "_".

4. Дійсні числа пишуться з використвнням крапки а не коми, оскільки кома в програмі використовується як роздільник.

5. При виведені на друк рядка який починається з & виводиться сам рядок а не значення змінної.

3.4. Висновки до третього розділу

1. Для інформаційної технології з метою визначення енергетичних характеристик поверхневих шарів металів з використанням алгебри алгоритмів побудовано підсистеми генерації програм з операції секвентування, підсистеми генерації програм з операції елімінування, підсистеми генерації програм з операцій циклів.

2. Для ІТ здійснено програмну реалізацію і апробацію математичної моделі системи генерації програм з операцій теорії секвенційних алгоритмів [61-65].

3. Показано, шо вона дійсно забезпечує автоматичну генерацію програм з формул теорії секвенційних алгоритмів. Цим підтверджена достовірність математичної моделі системи генерації програм з формул теорії секвенційних алгоритмів і її програмна реалізація.

4. Використовуючи можливості алгебри алгоритмів та єдиного підходу щодо написання алгоритмів в конкретних задачах корозіометрії, на основі бібліотеки програм символьної математики побудовано структури алгоритмів, їх моделей, проведено моделювання та дослідження вірогідності отриманих результатів. З допомогою засобів об’єктно-орієнтованого програмування розроблено допоміжну програму, яка дає можливість ефективніше реалізовувати математичні обчислення. Програмною реалізацією і апробацією математичної моделі системи генерації програм з операцій теорії секвенційних алгоритмів підтвердженно достовірність моделі системи генерації програм з операцій теорії секвенційних алгоритмів та коректне функціонування її програми в конкретних задачах.

РОЗДІЛ IV
РЕАЛІЗАЦІЯ ІТ ВИЗНАЧЕННЯ ЕНЕРГЕТИЧНИХ ХАРАКТЕРИСТИК ПОВЕРХНЕВИХ ШАРІВ МЕТАЛІВ

4.1. Фізико-математична модель визначення енергетичних характеристик міжфазних шарів металів для аналізу корозійних процесів – приклад реалізації ІТ

Енергетичні характеристики міжфазних шарів металів [22,23,25,33,100-103], на поверхні яких знаходиться покриття, або адсорбовані з електроліту частинки (рис. 4.1), для аналізу корозійних процесів означимо з використанням інформації, поданої у підрозділі 1.1.

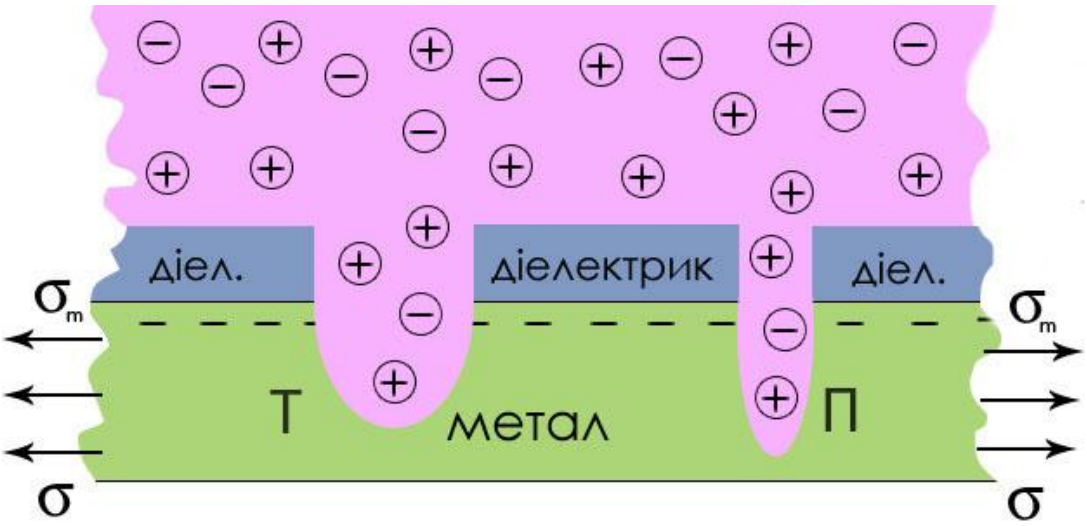


Рис. 4.1. Поверхневий шар металу з діелектричним покриттям і дефектами в електроліті (Т –тріщина, П – пітинг, σ – зовнішнє навантаження)

Пітинги – виразки, порожнини в металі, що розпочинаються з його поверхні. Зовні пітинг проявляється у вигляді появи заглиблень на поверхні сталі. Пітинги виникають в основному в захисному шарі (нанесеному або такому, що утворився природним чином) в місцях різних дефектів (тріщин від внутрішніх напружень, пор, мікровключень, виходу на поверхню металу зерен, дислокацій тощо).

Міжфазні енергію γ_m та натяг σ_m подамо аналогічно як (1.1), (1.3) з допомогою співвідношень [100-103]:

$$\gamma_m = \gamma_e + \xi_m \cdot \gamma_p\,; \;\; \gamma_e = \int_{-H}^{H} w_1 dx; \quad \gamma_p = \int_{-H}^{H} w_2 dx; \quad \sigma_m = \int_{-H}^{H} \sigma_y dx\,. \tag{4.1}$$

Тут ξ_m , $z_m = \gamma_e/\gamma_m$ – фізичні характеристики міжфазного шару; $2H$ – ефективна товщина міжфазного шару; γ_e – електрична складова міжфазної енергії; γ_p – механічна складова міжфазної енергії; σ_y – механічні напруження розтягу, які діють вздовж границі розділу середовищ (в результаті сумування (інтегрування) компоненти σ_y в околі границі розділу середовищ визначають міжфазний натяг σ_m (четверте співвідношення із (4.1))).

Умову рівноваги міжфазного шару для γ_m та наближені умови на границях міжфазного шару (при $x = H$ та $x = -H$) запишемо так [100-103]:

$$\frac{\partial \gamma_m}{\partial x} = \frac{\partial \gamma_e}{\partial x} + \xi_m \cdot \frac{\partial \gamma_p}{\partial x} = 0;$$

$$\sigma_y^+ + p = 0 \quad (x = +H)\,; \qquad \sigma_y^- + p = 0 \quad (x = -H)\,. \tag{4.2}$$

Тут індекс (+) відповідає параметрам металу, а (–) – параметрам покриття (або електроліту – агресивного середовища).

Подамо граничні умови для межі розділу середовищ метал – покриття (електроліт) [100]:

$$\varphi_+ = \varphi_-; \;\; \sigma_{x+} = \sigma_{x-}\,, \; \sigma_{y+} = \sigma_{y-} \text{ при } (x = 0), \tag{4.3}$$

де σ_x – механічні напруження стиску, які діють перпендикулярно до границі розділу середовищ (рис 1.).

Необхідно врахувати зміщення Z_b подвійного електричного шару в глибину металу. Для зміщення Z_b використовуємо відому формулу [100]:

$$Z_b = (3/(4k_f))(\pi/2 + (E_f/E_v - 1)arctg(E_v/E_f)^{1/2} - (E_f/E_v)^{1/2}). \qquad (4.4)$$

Тут Z_b – ширина подвійного електричного шару на поверхні металу (аналітичний вираз для Z_b отримано Sugiyama A.); E_f – енергія Фермі;

E_v – робота виходу електрона з металу; k_f – хвильовий вектор Фермі. Числові значення E_f, , E_v , k_f , Z_b для заліза (Fe), золота (Au), срібла (Ag), алюмінію (Al) приведено в додатках монографії [100].

Співвідношення (1.1)–(1.10), (4.1)–(4.4) створюють основу замкнутої системи рівнянь для визначення фізичних характеристик ξ_+, ξ_-, ξ_m, $b = (b_+, b_-)$, $k = (k_+, k_-)$ і товщин $h = (h_+, h_-)$, $2H$ для поверхневих та міжфазного шарів.

Роботу адгезії A_{ad} і енергію адгезійних зв'язків γ_{ad} системи метал – покриття (електроліт) визначимо на основі відомих співвідношень [100,110]:

$$A_{ad} = \sigma_{h+} + \sigma_{h-} - \sigma_m , \qquad (4.5)$$

$$\gamma_{ad} = \gamma_+ + \gamma_- - \gamma_m . \qquad (4.6)$$

Для запропонованої інформаційної технології розглядаємо три типи контактуючих середовищ: 1) метал (який кородує) – метал (покриття);
2) метал (який кородує) – діелектрик (покриття); 3) метал (який кородує) – розчин електроліту (покриття відсутнє). У першому випадку це може бути, наприклад, сталь – срібне покриття, у третьому – метал (сталь), який контактує безпосередньо з розчином електроліту в тріщині чи пітингу (рис. 1).

Алгоритм розв'язування задачі (4.1)–(4.6).

1) вибір системи контактуючих середовищ, наприклад «залізо (кородує) – метал (Au або Ag або Al – покриття)», «залізо (кородує) – електроліт», …

2) оцінка фізичних характеристик і параметрів (з таблиць фізичних величин, довідників, наукових статтей, монографій, Internet), зокрема згідно [100,110]:

$$E_+ = 177 \text{ GPa}, \quad \nu_+ = 0{,}334, \quad N_+ = 8{,}5\cdot 10^{28} \text{ 1/m}^3,$$

$$\sigma_{h+} = 2{,}5 \text{ N/m}, \quad \gamma_+ = 2{,}5 \text{ J/m}^2, \quad Z_{b+} = 0{,}0236 \text{ nm} \quad \text{(Fe)};$$

$$E_{-} = 80 \text{ GPa}, \quad \nu_{-} = 0{,}42, \quad N_{-} = 8{,}45 \cdot 10^{28} \text{ 1/m}^3,$$

$$\sigma_{h-} = 1{,}89 \text{ N/m}, \quad \gamma_{-} = 2{,}52 \text{ J/m}^2, \quad Z_{b-} = 0{,}034 \text{ nm} \quad (\text{Au});$$

$$E_{-} = 77 \text{ GPa}, \quad \nu_{-} = 0{,}37, \quad N_{-} = 5{,}85 \cdot 10^{28} \text{ 1/m}^3,$$

$$\sigma_{h-} = 1{,}5 \text{ N/m}, \quad \gamma_{-} = 2{,}52 \text{ J/m}^2, \quad Z_{b-} = 0{,}0318 \text{ nm} \quad (\text{Ag});$$

$$E_{-} = 70 \text{ GPa}, \quad \nu_{-} = 0{,}34, \quad N_{-} = 5{,}8 \cdot 10^{28} \text{ 1/m}^3,$$

$$\sigma_{h-} = 1{,}05 \text{ N/m}, \quad \gamma_{-} = 1{,}043 \text{ J/m}^2, \quad Z_{b-} = 0{,}0133 \text{ nm} \quad (\text{Al});$$

$$E_{-} = 7{,}3 \text{ GPa}, \quad \nu_{-} = 0{,}3, \quad N_{-} = 10{,}037 \cdot 10^{28} \text{ 1/m}^3,$$

$$\sigma_{h-} = 0{,}5 \text{ N/m}, \quad \gamma_{-} = 0{,}5 \text{ J/m}^2 \text{ (вода)}, \tag{4.7}$$

де N_{+}, N_{-} – число частинок (атомів) в одиниці об’єму;

3) визначення на основі комп’ютерної програми стрибка потенціалів $\Delta\Psi$ на міжфазних границях в системі «метал – електроліт» («метал – покриття»), а також енергетичних характеристик міжфазних шарів σ_m, γ_m, A_{ad}, γ_{ad} . Комп’ютерна програма розроблена з використанням співвідношень (1.1)–(1.10), (4.1)–(4.6) і її структура подана на рис. 4.2.

На основі $\Delta\Psi$ (на границі метал – розчин електроліту) визначаємо зміни густини I_k корозійного струму [100]:

$$I_k = \frac{E_k - E_a}{R_k + R_a + R_C + R_M}, \tag{4.8}$$

Рис. 4.2. Структура інформаційно-аналітичної моделі відбору, опрацювання і використання інформації про фізико-хімічні параметри контактуючих середовищ в технологічному комплексі протикорозійного захисту (E_V – робота виходу електрона з поверхні металу; M_{at}, – атомна маса металу; M_{mol} – молярна маса речовини; pH – водневий показник розчину електроліту) (Примітка: розроблено автором).

де E_k , E_a – електродні потенціали відповідно катодної й анодної ділянок; R_k, R_a, R_C, R_M – поляризаційні опори катодної і анодної ділянок, опір середовища (зовнішній) та опір металу (внутрішній опір корозійної пари).

Густина I_k корозійного струму є важливим параметром процесу, оскільки його величина є основним критеріальним значенням для визначення швидкості залишкової корозії металу в дефекті ізоляційного покриття [100]:

$$I_k = I_{kg} \cdot 10^{V_{pt}}; \quad V_{pt} = \frac{\overline{E}_{cor} - \overline{E}_p}{b_{at}}. \tag{4.9}$$

Тут I_{kg} – швидкість ґрунтової корозії металу; $\overline{E}_{cor}$ – корозійний потенціал; $\overline{E}_p$ – поляризаційний потенціал; b_{at} – тафелевський нахил анодної поляризаційної кривої.

Для критерію (4.9) на рис. 4.3 зображно алгоритм.

З допомогою енергетичних характеристик міжфазних шарів σ_m, γ_m, A_{ad}, γ_{ad} визначаємо умови адгезії, механічної міцності (і, відповідно, умови руйнування) покрить, які захищають метал від корозійного середовища.

З використанням даних (4.7) і комп'ютерної програми отримано для двох системи отримано: А) «сталь – срібло (покриття)» –

$$\sigma_m = 1{,}5426\, N/m, \quad \gamma_m = 1{,}5905\, J/m^2, \quad A_{ad} = 2{,}457\, N/m, \quad \gamma_{ad} = 3{,}430\, J/m^2,$$

$$z_\gamma = \frac{\gamma_m}{\gamma_{ad}} = 0{,}464; \quad z_A = \frac{\sigma_m}{A_{ad}} = 0{,}628; \quad \Delta\Psi_m = 1{,}659\, V; \quad Q_m = 0{,}232\, \frac{C}{m^2};$$

В) «сталь – алюміній (покриття)» –

$$\sigma_m = 1{,}455\, N/m, \quad \gamma_m = 1{,}0363\, J/m^2, \quad A_{ad} = 2{,}095\, N/m, \quad \gamma_{ad} = 2{,}507\, J/m^2,$$

$$z_\gamma = \frac{\gamma_m}{\gamma_{ad}} = 0{,}413; \quad z_A = \frac{\sigma_m}{A_{ad}} = 0{,}694; \quad \Delta\Psi_m = 2{,}146\, V; \quad Q_m = 0{,}209\, \frac{C}{m^2}. \tag{4.10}$$

алгоритм
змінні
дійсні $\overline{E}_{cor}, \overline{E}_{cor}, b_{at}, I_{kg}, I_k, V_{pt}$ *,a1,b1,b2,b3,b4,b5;*
дійсні aaa,bbbb,cccc;
початок
ввести $\overline{E}_{cor}, \overline{E}_{cor}, b_{at}, I_{kg}$ *;*
$I_k = 1.4$*;*
$V_{pt} = \left(\overline{E}_{cor} - \overline{E}_p\right) / b_{at}; \quad I_k = I_{kg} \cdot 10^{V_{pt}}$;
вивести I_k *;*
вивести &Перевірка_процедури_DoIt, I_k *;*
|- $I_k = \alpha_1$, $I_k = \alpha_2$, $\alpha_1 > \alpha_2$ -|;
вивести &Перевірка_оператора_|-.....-|, I_k *;*
$I_k = 1$*;*
$ d $I_k = I_k$ *a , a=a+1;*
вивести &Перевірка_оператора_$_x....., I_k *;*
кінець

Рис. 4.3. Алгоритм визначення швидкості залишкової корозії металу в дефекті ізоляційного покриття (за струмом корозії I_k)

Тут Q_m – заряд обкладки подвійного електричного шару (в околі межі «метал (залізо) – покриття». Ці значення (4.10) можна використати як тестові приклад для запропонованої інформаційної технології. Для вимірювання адгезії покриття відносно металу, який захищають від корозії [22,23,62], використовують прилади – адгезиметри.

Для прикладу [102] $\Delta\Psi_m$ – електричний бар'єр на границі «метал (сталь 3) – водний розчин слабкого електроліту» (стрибок потенціалу в подвійному електричному шарі) приймає значення:

$$\Delta\Psi_m = 5{,}3 \text{ В}. \tag{4.11}$$

Це значення (4.11) можна розглядати також як тестовий приклад для запропонованої інформаційної технології.

Використовуємо для позначення ємності міжфазного електричного конденсатора позначення C_m , а також формули, аналогічні [100]:

$$\gamma_{m1} = \frac{Q_m^2}{2 \cdot C_m} = \frac{Q_m \cdot \Delta\Psi_m}{2}, \quad C_m = \frac{Q_m}{\Delta\Psi_m} = \frac{1}{2}\varepsilon_0 \cdot k_m, \quad d_\xi = \frac{2}{k_m}, \quad k_m = \sqrt{\frac{\rho_m \cdot C_{\varphi m}}{\varepsilon_0}}, \quad (4.12)$$

де γ_{m1} – електрична складова міжфазної енергії; $d_\xi = 2H$– ефективна товщина подвійного електричного шару; ρ_m – усереднена густина матеріалу в області подвійного електричного шару ($\rho_m \approx 9187$ кг/м^2). Вирази (4.12) випливають із відомих співвідношень електростатики і (1.1)-(1.10).

Підставляючи Q_m і $\Delta\Psi_m$ в (4.12), отримуємо:

А) «сталь – срібло (покриття)» –

$$\gamma_{m1} = 0{,}308\ \text{Дж/м}^2, \quad C_m = 86{,}8\ \text{мФ/м}^2, \quad k_m = 1{,}96 \cdot 10^{10}\ 1/\text{м},$$

$$d_\xi = 0{,}102\ \text{нм}, \quad C_{\varphi m} = 3{,}70 \cdot 10^5\ \text{Ф/кг};$$

В) «сталь – алюміній (покриття)» –

$$\gamma_{m1} = 0{,}224\ \text{Дж/м}^2, \quad C_m = 97{,}28\ \text{мФ/м}^2, \quad k_m = 2{,}20 \cdot 10^{10}\ 1/\text{м},$$

$$d_\xi = 0{,}091\ \text{нм}, \quad C_{\varphi m} = 8{,}09 \cdot 10^5\ \text{Ф/кг}. \quad (4.13)$$

Тут $C_{\varphi m}$ – питома електроємність подвійного електричного шару (в околі межі «метал–покриття»).

Параметри $\Delta\Psi_m$ і C_m важливі для вивчення механізмів антикорозійного захисту. Срібло можна використовувати як покриття тільки в тому випадку, якщо буде суцільне покриття без дефектів. Вигідніше покривати залізо алюмінієм, оскільки в алюмінію більш від’ємний електродний потенціал [100] і, крім того, алюміній значно дешевший.

Але в практиці можуть бути покриття з домішками [22,23,85], а домішки можуть як збільшувати, так і зменшувати електродний потенціал, і відповідна методика, яка апробована для покриттів з двома протилежними значеннями відносного електродного потенціалу (4.10), (4.13) допоможе на основі результатів стимуляційних програм прийняти правильне рішення, що забезпечить оптимальний варіант антикорозійного захисту.

4.2. Фізико-математична модель активації деформованого металу поблизу вершини корозійної тріщини

Утворення оксиду в околі вершини тріщини можливе внаслідок взаємодії металу (*Me*) з іонами OH^- [100]:

$$n \cdot Me + 2mOH^- \Rightarrow Me_nO_m + m \cdot H_2O + 2m \cdot q_0, \qquad (4.14)$$

де q_0 = 1,6·10^{-19} С – елементарний заряд (заряд електрона); n, m – натуральні числа.

Рівноважний потенціал утворення оксиду E_p^{MeO} на поверхні металу залежить від рівноважного потенціалу (електродного) процесу E_p і кислотності pH зовнішнього корозійного середовища [100]:

$$E_p^{MeO} = E_p + 0{,}826 - 0{,}059 \cdot pH. \qquad (4.15)$$

Тут в (4.15) значення потенціалів виражені у вольтах.

В результаті розв’язання в циліндричних координатах (r, θ, z) задачі розподілу механічних напружень в області тріщини отримано наближене співвідношення для нормованих напружень [100]:

$$\alpha_1 = \max(\overline{\sigma}_1) = \max(\sigma_{\theta\theta}(r, z, \pi/2)/\sigma_1) = 1 + 2 \cdot \overline{b}^{\,\overline{a}/(\overline{a}+0{,}25)}, \qquad (4.16)$$

де $\bar{a} = a / H_t$, $\bar{b} = b / H_t$; θ =π/2 у вершині тріщини (в початку координат (див. рис. 1)); $\sigma_{\theta\theta}$ – компонента механічних напружень у циліндричних координатах (r, θ, z); D=2R_t, H_t – діаметр і товщина труби відповідно; d=2a, b – діаметр та глибина заглибини (поверхневої тріщини).

Енергетичний баланс процесу поширення корозійної тріщини в металі можна представити аналогічно як у ситуації корозійного розтріскування: робота пластичної деформації – P_{PL}, зміна початкового запасу енергії W_{zp}, виділення енергії в процесі електрохімічної реакції W_{ec} [100]:

$$P_{PL} = W_{zp} + W_{ec} = \frac{K_1^2(1-\nu^2)}{E} + \frac{z_{si}F\rho}{\zeta} \cdot \frac{\delta}{M}, \tag{4.17}$$

де K_1 — коефіцієнт інтенсивності напружень; ρ – густина матеріалу; z_{si} – формальний заряд сольватованих іонів; F = 96500 C/gr-mol – постійна Фарадея; δ – висота фронту тріщини, що насувається; M – молекулярна маса металу; ζ – перенапруження анодної реакції. Величиною поверхневої енергії в (4.17) можна нехтувати, оскільки вона мала в порівнянні з роботою пластичної деформації P_{PL} [100].

При пороговому значенні коефіцієнта інтенсивності напружень K_{1SCC}, тобто при мінімальному значенні K_1, що відповідає початку поширенню корозійної тріщини, рівняння (4.17) набуває наступного вигляду [100]:

$$K_{1SCC} = \sqrt{\frac{E}{1-\nu^2} \cdot \left(\frac{P_{PL} - z_{si}F\rho\delta\zeta}{M} \right)} = \sqrt{C_{k1}(P_{PL} - C_{k2} \cdot \zeta)}, \tag{4.18}$$

де $C_{k1} = \dfrac{E}{M(1-\nu^2)}$; $C_{k2} = \dfrac{z_{si}F\rho\delta E}{M(1-\nu^2)}$.

Із виразу (4.18) видно, що P_{PL}, ζ – основні величини, від яких залежить K_{1SCC}. Зменшення значення роботи пластичної деформації P_{PL} відбуватиметься в результаті збільшення або межі текучості, або швидкості механічного

зміцнення у вершині тріщини [100]. В результаті кожен з цих чинників при постійному значенні ζ знижуватиме величину K_{1SCC} і, отже, знижуватиме ступінь опору щодо поширення корозійної тріщини [100]. Збільшення перенапруження анодної реакції ζ (при цьому потенціал металу стає більш електропозитивним) при певному значенні роботи пластичної деформації P_{PL}, згідно співвідношення (4.18), приводитиме до пониження опору поширенню тріщини. Величина анодного перенапруження ζ є функцією електрохімічних умов усередині тріщини і, відповідно, умови поширення корозійної тріщини будуть залежати від pH (водневого показника) розчину електроліту, активності аніонів, електродного потенціалу E_p, хімічного складу металу, поверхневого натягу і поверхневої енергії поблизу вершини тріщини, а також умов формування пасивувального шару [100].

Напруження, що виникають в металі, і швидкість росту корозійного дефекту зв’язують за допомогою співвідношень нерівноважної термодинаміки [100]. Деформація металу ε (ε – перший інваріант тензора деформацій) розглядається на стадії лінійного зміцнення, коли дислокації шикуються і рухаються в системі паралельних площин ковзання за відсутності поперечного ковзання. З математичних міркувань випливає ряд феноменологічних рівнянь, що відображають взаємозв'язок процесів пластичної деформації і стресс-коррозії в металах [100]. Такі рівняння при одночасному процесі деформування і електрохімічної корозії приймають наступний вигляд [22,23,100]:

$$J_d = L_A A + L_B \zeta; \quad I_C = L_C \varepsilon + L_d \zeta; \tag{4.19}$$

де J_d – густина потоку дислокацій (дислокаційний струм); I_C – густина електричного струму при корозії; A – узагальнена сила (хімічна спорідненість процесів створення і руху дислокацій); феноменологічні коефіцієнти: L_A – коефіцієнт, що характеризує провідність; L_d – характеризує поляризованість електроду; L_C = L_B регулюють ступінь взаємного впливу деформації та електрохімічної корозії і виражають кількісно явище механохімічного ефекту.

Фізичний сенс цього коефіцієнта (L_C) полягає в зміні хімічного потенціалу металу в результаті його пластичної деформації, яка пов'язана з ослабленням міжатомних зв'язків в місцях скупчення дислокацій [100].

Використаємо базову концепцію механіки корозійного руйнування конструкційних матеріалів [22,23,100]. Згідно цієї концепції стан зони передруйнування матеріалу в околі вершини тріщини можна описати такими групами параметрів [100]: $P_j(\sigma)$ – параметри, що характеризують напружено-деформований стан матеріалу і є функціями прикладених до тіла зовнішніх зусиль, яким відповідають напруження σ (σ – перший інваріант тензора механічних напружень); $L_n(\tau)$ – параметри, які визначають у часі τ фізико-хімічні процеси, що проходять між деформованим металом і корозійним середовищем; $B_m(s)$ – параметри, що характеризують стан поверхонь матеріалу «s», які є до руйнування і утворюються під час руйнування. Швидкість локального корозійного руйнування матеріалу (тобто швидкість розвитку корозійної тріщини, яка виникне у вершині поверхневого дефекту) подамо аналогічно як для корозійно-втомної тріщини [100]:

$$\frac{da_t}{d\tau} = f(C_i, P_j(\sigma), L_n(\tau), B_m(s)), \tag{4.20}$$

де a_t – характерний лінійний розмір тріщини в напрямку її поширення; C_i – константи, що характеризують систему «матеріал (метал) – корозійне середовище»; i, j, n, m – 1, 2, 3, …. Наближено для зони передруйнування металу (сталі) співвідношення (4.20) запишемо у вигляді:

$$\frac{da}{d\tau} = f(E, \nu, K_1, pH, E_p, \gamma_m, \sigma_m, \gamma_S), \tag{4.21}$$

де E, ν, K_1, pH, E_p, γ_m, σ_m, γ_s – фізичні характеристики матеріалу і параметри стану; γ_m, σ_m – міжфазні енергія та натяг; $\gamma_s = \gamma_s(\gamma_m)$ – питома робота пластичної деформації на одиницю поверхні тріщини при утворенні нової деформаційно

активованої поверхні. Параметр γ_s входить у відому формулу Гріфітса-Орована [100]:

$$\sigma_* = \sqrt{\frac{2E(\gamma_m + \gamma_S)}{\pi \cdot a_t(1-\nu^2)}}, \quad \sigma_{**} = \sqrt{\frac{2E(\gamma_m + \gamma_S)}{\pi \cdot a_t}}, \tag{4.22}$$

Тут перша формула (4.22) записана для плоскої деформації, друга – для плоского напруженого стану; σ_*, σ_{**} – критичні напруження ($\sigma_* = \sigma_T$, σ_T – границя текучості).

Аналогічно як у [100] подамо оцінки довжин пластичних зон в околі вершини тріщини:

$$d_* = 0{,}578\frac{K_1^2}{\pi \cdot \sigma_T^2}, \qquad d_{**} = \frac{\pi \cdot K_1^2}{8 \cdot \sigma_T^2}, \tag{4.23}$$

де d_* і d_{**} – характерні розміри пластичних зон для плоскої деформації і плоского напруженого стану відповідно.

Для оцінювання граничного стану матеріалу (сталі) поблизу вершини тріщини в умовах складного напруженого стану використовуємо критерій Губера-Мізеса (для σ_i – інтенсивності напружень) або Кулона-Треска (для σ_τ – максимального дотичного напруження) [100]:

$$\sigma_i = \frac{\sqrt{2}}{2}\sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_1 - \sigma_3)^2 + (\sigma_2 - \sigma_3)^2} = const,$$

$$\sigma_\tau = \tau_{\max} = \frac{\sigma_1 - \sigma_3}{2} = const, \tag{4.24}$$

в яких σ_1, σ_2, $\sigma_3 = \sigma_{zz}$ – головні напруження.

Для моделювання особливостей пластичного деформування у вершині тріщини використаємо інтегральну характеристику швидкості процесу, що веде до розриву тіла, тобто “довговічність” (τ_d) – час між моментом прикладання

навантаження σ і моментом розриву тіла. Цей час обернено пропорційний середній швидкості процесу руйнування [22,100]:

$$\tau_d(\sigma,T) = \tau_0 \, exp \frac{U_0 - \gamma_d \cdot \sigma}{kT}, \qquad (4.25)$$

де k – постійна Больцмана; $\tau_0 \sim 10^{-13}$ с – час, за порядком величини близький до періоду коливань атомів у твердих тілах (періоду коливань максимальної частоти в дебаєвському спектрі); U_0 характеризує енергію активації розриву міжатомних зв'язків і практично співпадає з енергією дисоціації речовини (металу); $\gamma_d = q \cdot V_a$, де V_a – активаційний об'єм у елементарному акті дисоціації, q коефіцієнт (безрозмірний) локальних механічних перенапружень (досягає за рахунок мікродефектів в реальних тілах значень 10÷100 і вище, що і характеризує відмінність експериментальної величини міцності від теоретичної) [22]. Зокрема, для сталі (заліза) U_0 = 4,4 еВ (1 еВ = $1{,}6 \cdot 10^{-19}$ Дж), $V_a = 1{,}2 \cdot 10^{-2}$ нм$^{-3}$ [2]. Необхідно відзначити, що U_0 близька до енергії сублімації W_{subl} більщості металів, яка для заліза приймає значення $W_{subl}(Fe)$ = 4,2 еВ [22].

Пластичне деформування металу супроводжується послідовними елементарними актами розриву напружених міжатомних зв'язків в результаті флуктуацій теплової енергії атомів. Залежність (4.25) увійшла до сфери наукових досліджень про міцність (руйнування) як формула Журкова [22].

Процесу електрохімічної (корозійної) реакції в системі «метал – електроліт» відповідають густини анодного i_a і катодного i_k струмів, які характеризують швидкості окислення та відновлення [22,23]:

$$i_a = k_1 \exp(-w_a/(RT)), \quad i_k = k_2 C \exp(-w_k/(RT)). \qquad (4.26)$$

Тут k_1 , k_2 – константи швидкостей анодної та катодної реакцій; C – концентрація (активність) іонів металу (окиснювача) в приелектродному шарі електроліту; w_a , w_k – енергії активації анодної та катодної реакцій;

R = 8,31 J/(g-mol·K) – універсальна газова стала; T – абсолютна температура. У вираз для i_a (4.26) у явному вигляді не входить концентрація атомів металу, бо вона під час реакції практично не змінюється і включена в константу k_1.

Енергії активації електрохімічного процесу з урахуванням стрибка потенціалу напруженості електричного поля в подвійному електричному шарі $\Delta\Psi_z$, тобто змін, викликаних в результаті поляризації об'єкту від зовнішнього джерела $\Delta\Psi_e$, кривизни поверхні $\Delta\Psi_r$, зміщення електродного потенціалу $\Delta\Psi_s$ і механічним навантаженням $\Delta\Psi_p$, запишемо у вигляді [22,100]:

$$w_a = w_{0a} - \alpha zF(\Delta\Psi_e + \Delta\Psi_r + \Delta\Psi_s + \Delta\Psi_p),$$

$$w_k = w_{0k} + (1-\alpha)zF(\Delta\Psi_e + \Delta\Psi_r + \Delta\Psi_s), \qquad (4.27)$$

де α , (1–α) – коефіцієнти переносу; z – валентність іона; w_{0a} , w_{0k} – постійні величини для конкретних умов системи «метал – корозійне середовище (електроліт)». У [22,23] наведено приклади методики теоретичного оцінювання $\Delta\Psi_e$ для металів. Зокрема, для сталі (заліза) $\Delta\Psi_e$ = 5,3 В (4.11).

Методика оцінювання зміни електродного потенціалу $\Delta\Psi_r$, викликаного кривизною поверхні, і відповідний конкретний приклад для металевого тіла з порожниною еліптичного перерізу наведено у праці [16]. На основі числових розрахунків встановлено, що перевищення електродного потенціалу у вершині еліпса в порівнянні з коловим отвором сягає більше 5 разів [16].

Як відзначено у монографії [100], поверхня, заповнена пасивною плівкою, має властивості катодної області, а ювенільна поверхня поблизу вершини тріщини є анодною.

При катодній поляризації зменшується густина струму анодного розчинення. При цьому виконується умова $i_a < i_{cor}$, де i_a – густина анодного струму, i_{cor} – струм корозії. Зниження i_a досягають шляхом накладання катодного поляризуючого струму з густиною i_p . Практично важливою є залежність між ними. Ефективність катодного захисту оцінюють на основі

використання коефіцієнта гальмування корозії g, а також ступеня захисту P [100]

$$g = \frac{i_{cor}}{i_a}, \quad P = 1 - \frac{1}{g}. \tag{4.28}$$

На основі співвідношень (4.1)–(4.27) можна буде оцінити енергетичні характеристики міжфазних шарів та активаційних процесів, які характеризують метал, середовище і динаміку корозійних (анодного та катодного) процесів поблизу вершини тріщини в навантаженому металі. Відповідна схема з урахуванням корозійних процесів поблизу вершини тріщини подана на рис. 4.4.

Рис. 4.4. Структурна схема інформаційно-комп'ютерної технології корозіометрії, якщо в покритті є тріщина і метал безпосередньо контактує з розчином електроліту

Основне завдання запропонованої інформаційної технології [1,110]– відбір і використання даних, а також теорії сенквенційних алгоритмів для фізико-механічних моделей, які відображені співвідношеннями (1.1)–(1.10), (4.1)–(4.27).

З допомогою інформаційно-комп'ютерної технології корозіометрії [62,65,102], яка враховує корозійні процеси поблизу вершини тріщини можна оцінювати якість протикорозійного захисту. Сформулюємо відповідні співвідношення концептуальної моделі для оцінювання якості протикорозійного захисту:

$$Ks = Ks(T, gradT, A(L,G,S), \sigma_{ij}),$$

$$Nz = Nz(T, gradT, A(L,G,S), \sigma_{ij}),$$

$$Pz = Pz(Nz, Ks), \qquad Rz = Rz(Nz, Ks), \tag{4.29}$$

де σ_{ij} – компоненти тензора механічних напружень; Pz – набір параметрів, які характеризують міжфазні (поверхневі) шари; Nz – набір параметрів, які характеризують фізико-хімічні процеси в дефектах (тріщинах, пітингах); T – температура; grad$T$ – градієнт температури [23]; $A(L,G,S)$ – корозійні втрати (L,G,S – символічне позначення рідини (liquid), газу (gas), твердого тіла (solid)); Rz – набір параметрів, які характеризують корозійні струми; Ks – символічне позначення корозійної стійкості, для визначення якої можна використовувати критерій (4.9) і параметри g, P (4.28).

4.3. Застосування алгоритму інформаційної технології для антикорозійного захисту

Загальний традиційний підхід до формування виробничої технології захисту металів від корозії включає наступні етапи [62,65,102] (рис. 4.5).

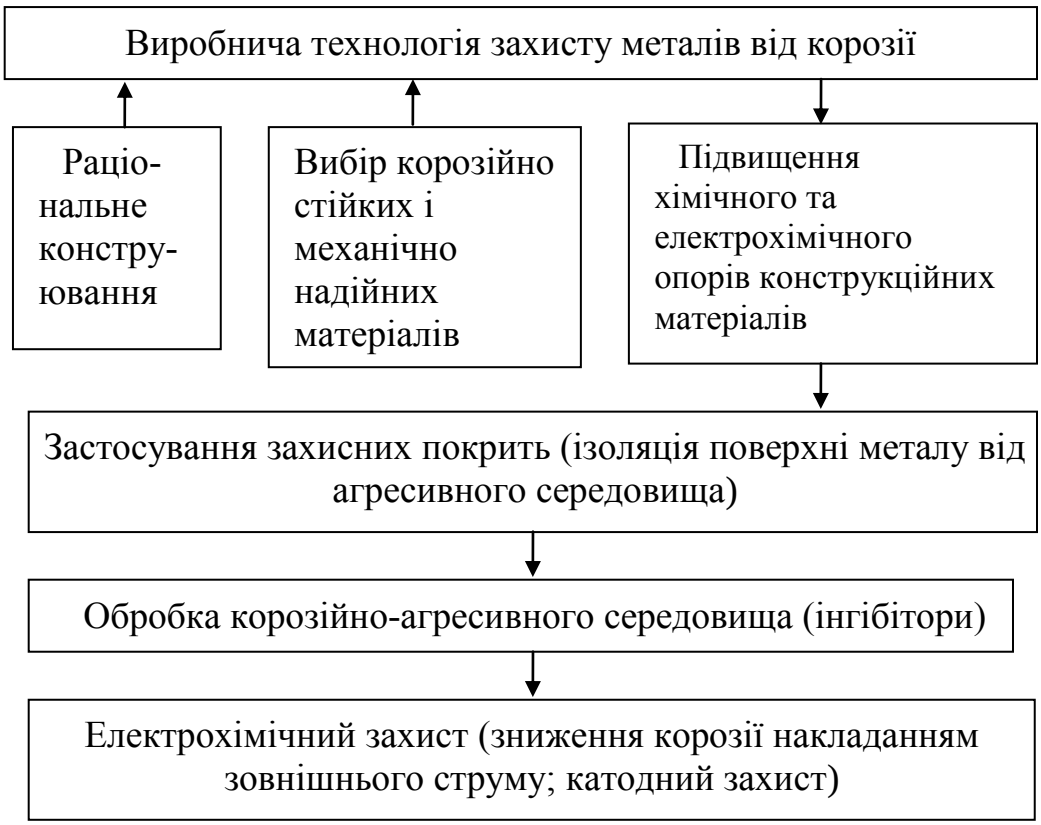


Рис. 4.5. Етапи виробничої інформаційної технології для захисту металів поліграфічного матеріалознавства від корозії

Елементи такого типу інформаційного забезпечення і відповідні методи розв’язання конкретних задач відомі, хоча не всі вони враховують елементи інших етапів [83]. Але така інформаційна технологія буде громіздкою і неефективною. Тому доцільно розробити для кожного етапу свою інформаційну технологію з урахуванням можливості підключення інших етапів, а потім по можливості об’єднати їх в єдине ціле.

Розглянемо задачу розроблення інформаційної технології для етапу «Вибір корозійно стійких і механічно надійних матеріалів», використавши результати праці [105]. При цьому передбачаємо подальший етап «Застосування захисних покрить».

На основі рівнянь (1.1)–(1.10), (4.1)–(4.6) а також числових значень констант (4.7), [105]:

$$E = 200 \text{ ГПа}, \ \nu = 0{,}26, \ Z_b = 0{,}014 \text{ нм}; \ N = 8{,}5\cdot10^{28} \text{ 1/м}^3; \ E_v = 4{,}16 \text{ еВ};$$

$$E_f = 11{,}6 \text{ еВ}; \ k_f = 1{,}714\cdot10^{10} \text{ 1/м}; \ \sigma_h = 4{,}18 \text{ Н/м}; \ \gamma = 2{,}26 \text{ Дж/м}^2, \quad (4.30)$$

з допомогою автоматизованих способів алгебри алгоритмів для температури T = 300 К для сталі 20 встановлено:

$$k_1 = 1{,}33\cdot10^{10} \text{ 1/м}, \quad \xi_1 = 15{,}43, \quad \beta_1 = -0{,}120 \text{ 1/В}; \quad \Delta\Psi_1 = 8{,}69 \text{ В};$$

$$k_2 = 2{,}04\cdot10^{10} \text{ 1/м}, \quad \xi_2 = 34{,}07, \quad \beta_2 = -0{,}116 \text{ 1/В};$$

$$\Delta\Psi_2 = 3{,}7 \text{ В}; \ \gamma_{m1} = 0{,}742 \text{ Дж/м}^2. \quad (4.31)$$

Тут індекс ”1” відповідає сталі, яка контактує з повітрям, а “2” – сталі, яка контактує з корозійним середовищем (розчином *NACE*); $\Delta\Psi$ – стрибок потенціалу напруженості електричного поля на границі метал – агресивне середовище; γ_{mo} – міжфазна енергія ненавантаженого зразка.

Проведено аналіз процес довготривалої корозійної міцності (циклічної тріщиностійкості) сталей у розчині *NACE* протягом 720 годин від початкового напруження σ_{yo} до порогового σ_{yz} [22]. Порогове напруження σ_{yz} – граничне напруження, за якого метал не руйнується протягом 720 годин.

Методика визначення циклічної тріщиностійкості матеріалів передбачає [61,62,64] використання таких приладів та матеріалів: машина МО-041 для циклічних випробувань, EUS-20, шлейфовий осцилограф, блок живлення ВИП-009, катетометр В-630, призматичний зразок (сталі) з концентратором та виведеною тріщиною [22]. Пристрій EUS-20 – універсальна машина для випробування на міцність; служить для випробування матеріалів, зокрема, для випробування матеріалів всякого роду на міцність при розтягуванні, стискуванні і згині.

На основі системи рівнянь (1.1)–(1.10), (4.1)–(4.6) для даних [22] отримано конкретні числові значення змін міжфазної енергії γ_m [100]. Зокрема, для різних марок сталей:

сталь 20: σ_{yo} = 330 МПа; σ_{yz} = 195 МПа;

$$\gamma_m = [0{,}767\ Дж/м^2;\ 0{,}797\ Дж/м^2];\quad \Delta\Psi_2 = 3{,}650\ В; \tag{4.32}$$

сталь 12 Х1МФ – σ_{yo} = 510 МПа; σ_{yz} = 170 МПа;

$$\gamma_m = [0{,}77\ Дж/м^2;\ 0{,}827\ Дж/м^2];\quad \Delta\Psi_2 = 3{,}839\ В; \tag{4.33}$$

сталь 20ЮЧ – σ_{yo} = 305 МПа; σ_{yz} = 195 МПа;

$$\gamma_m = [0{,}774\ Дж/м^2;\ 0{,}792\ Дж/м^2];\quad \Delta\Psi_2 = 3{,}616\ В; \tag{4.34}$$

сталь 17ГС – σ_{yo} = 310 МПа; σ_{yz} = 210 МПа;

$$\gamma_m = [0{,}777\ Дж/м^2;\ 0{,}793\ Дж/м^2];\quad \Delta\Psi_2 = \Delta\Psi = 3{,}622\ В; \tag{4.35}$$

сталь 14ГБ – σ_{yo} = 390 МПа; σ_{yz} = 250 МПа;

$$\gamma_m = [0{,}783\ Дж/м^2;\ 0{,}807\ Дж/м^2];\quad \Delta\Psi_2 = \Delta\Psi = 3{,}686\ В; \tag{4.36}$$

сталь 14ГБ (зварне з’єднання) – σ_{yo} = 380 МПа; σ_{yz} = 210 МПа;

$$\gamma_m = [0{,}777\ Дж/м^2;\ 0{,}805\ Дж/м^2];\quad \Delta\Psi_2 = \Delta\Psi = 3{,}70\ В; \tag{4.37}$$

сталь 02 Х20Н6М3 – σ_{yo} = 420 МПа; σ_{yz} = 290 МПа;

$$\gamma_m = [0{,}79\ Дж/м^2;\ 0{,}812\ Дж/м^2];\quad \Delta\Psi_2 = \Delta\Psi = 3{,}744\ В. \tag{4.38}$$

На першому етапі обмежимось розглядом трьох визначальних параметрів: σ_{yz}, $(\gamma_m)_{max}$, і $\Delta\Psi_2$. Перші два параметри характеризують міцність матеріалу і вони повинні бути максимальними, третій параметр $\Delta\Psi_2$ ($\Delta\Psi_2=\Delta\Psi_r$) характеризує електричний бар’єр на границі «метал – агресивне середовище» і він також повинен бути максимальним. Частинки сірководню адсорбуються на поверхні зразка сталі, зменшують міцність поверхні (ефект

Ребіндера), а також значення $\Delta\Psi_2$ і процес корозії підсилюється (стає більш інтенсивним приблизно в 3 рази як свідчать експерименти [23]).

Всі три параметри мають різну розмірність і тому їх не можна порівнювати. А порівнювати потрібно, Тому їх зробимо безрозмірними. Для цього серед кожного набору даних визначаємо максимальне значення:

γ_{mm} = 0,827 Дж/м^2 (12 Х1МФ); $\Delta\Psi_{2m} = \Delta\Psi_m$ = 3,616 В (20ЮЧ);

$$\sigma_{yzm} = 290 \text{ МПа (02 Х20Н6М3).} \tag{4.39}$$

На ці значення (4.39) ділимо інші параметри і заносимо в таблицю 4.1.

$$\gamma_{mv} = \frac{\gamma_m}{\gamma_{mm}}; \quad \sigma_{yzv} = \frac{\sigma_{yz}}{\sigma_{yzm}}; \quad \Delta\Psi_v = \frac{\Delta\Psi}{\Delta\Psi_m}. \tag{4.40}$$

Відповідна ілюстрація подана на рис. 4.6.

Таблиця 4.1.

Відносні значення параметрів γ_{mv}, σ_{yzv}, $\Delta\Psi_v$, M_j

No	Марка сталі	γ_{mv}	σ_{yzv}	$\Delta\Psi_v$	M_1	M_2
1	20	0,963	0,672	0,951	0,8703	0,8703
2	12 Х1МФ	1,0	0,586	1,0	0,8758	0,9172
3	20ЮЧ	0,958	0,672	0,942	0,8658	0,8912
4	17ГС	0,959	0,724	0,9435	0,8823	0,9027
5	14ГБ	0,976	0,862	0,9601	0,9354	0,9437
6	14ГБ (зварне з’єднання)	0,973	0,724	0,9638	0,8946	0,9177
7	02 Х20Н6М3	0,972	1,0	0,9753	0,9847	0,9815

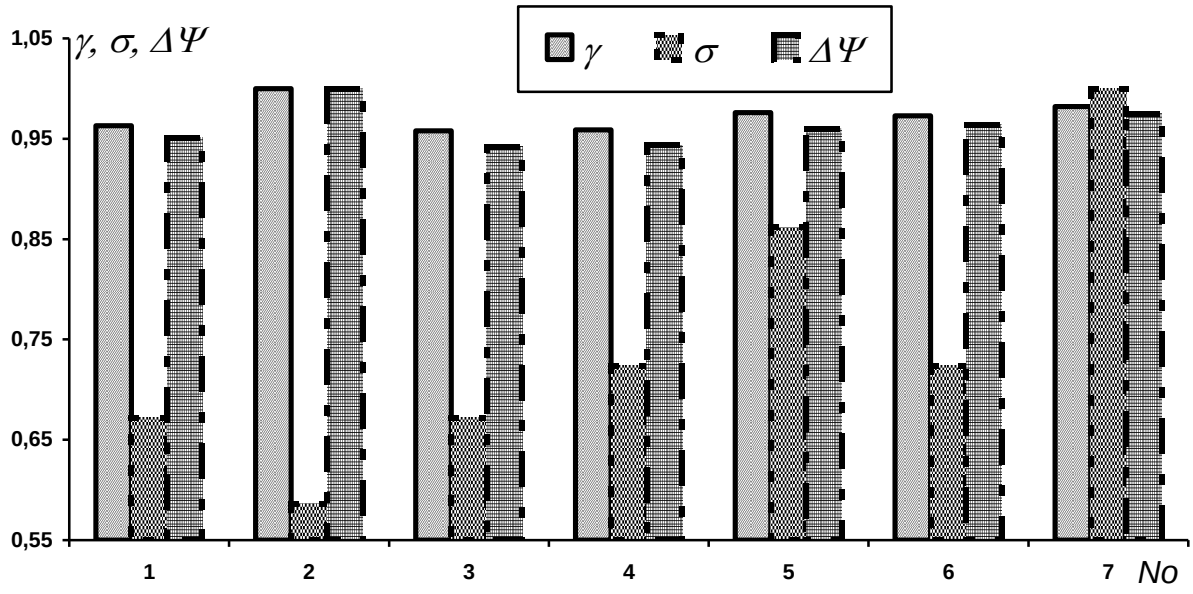


Рис. 4.6. Параметри $\sigma \equiv \sigma_{yzv}$, $\gamma \equiv \gamma_{mv}$ які характеризують міцність матеріалу, і зміна потенціалу $\Delta\Psi \equiv \Delta\Psi_{2v}$ для оцінювання електричного бар'єру на границі «метал – агресивне середовище» (за даними табл.. 4. 1).

Формулюємо критерій оптимальності для параметрів γ_{mv} , σ_{yzv}, $\Delta\Psi_v$:

$$\alpha_1 \cdot \gamma_{mv} + \alpha_2 \cdot \sigma_{yzv} + \alpha_3 \cdot \Delta\Psi_v = M_j \Rightarrow \max, \qquad (4.41)$$

де α_1, α_2, α_3 – коефіцієнти вагомості, які визначають експертним методом ($\alpha_1+\alpha_2+\alpha_3=1$); j – індекс відповідної комбінації факторного варіанту (j =1,2,3,…).

Розрахунок M_j проводимо для різних можливих комбінацій коефіцієнтів вагомості, зокрема, в табл. 4.1 наведено результати розрахунку для двох варіантів:

$$M_1 \Rightarrow \alpha_1 = 0{,}3; \alpha_2 = 0{,}3; \alpha_3 = 0{,}4; \quad M_2 \Rightarrow \alpha_1 = 0{,}2; \alpha_2 = 0{,}2; \alpha_3 = 0{,}6. \qquad (4.42)$$

Максимальні значення параметрів γ_{mv}, σ_{yzv}, $\Delta\Psi_v$ відповідають двом маркам сталі 12 Х1МФ (*No* 2) і 02 Х20Н6М3 (*No* 7). З допомогою співвідношень (4.41) і (4.42) встановлюємо остаточний оптимальний варіант 02 Х20Н6М3 (*No* 7).

Ускладнюємо інформаційну задачу, ввівши в розгляд четвертий параметр – ціну P_s (price, економічний показник) (P_m – максимальна ціна).

Якщо параметри γ_{mv} , σ_{yzv}, $\Delta\Psi_v$ в (4.41) бажано вибирати максимальними, то ціну P_s слід вибрати мінімальною. В цьому випадку критерій типу (4.41) для 4-х параметрів γ_{mv}, σ_{yzv}, $\Delta\Psi_v$, P_s повинен мати вигляд критерію компромісу:

$$\alpha_1 \cdot \gamma_{mv} + \alpha_2 \cdot \sigma_{yzv} + \alpha_3 \cdot \Delta\Psi_v = M_{rj} \Rightarrow \max, \quad P \Rightarrow \min. \tag{4.43}$$

Щоб спростити ситуацію, відносний параметр для ціни вибираємо оберненим (обернена ціна), тобто максимальну ціну P_m ділимо на P_s:

$$\frac{P_s}{P_m} \Rightarrow \min; \qquad P = \frac{P_m}{P_s} \Rightarrow \max. \tag{4.44}$$

Аналогічно як (4.41) формулюємо критерій оптимальності для параметрів γ_{mv} , σ_{yzv}, $\Delta\Psi_v$, P:

$$\alpha_1 \cdot \gamma_{mv} + \alpha_2 \cdot \sigma_{yzv} + \alpha_3 \cdot \Delta\Psi_v + \alpha_4 \cdot P = M_{rj} \Rightarrow \max, \tag{4.45}$$

де α_1, α_2, α_3, α_4 – коефіцієнти вагомості, які визначають експертним методом ($\alpha_1+\alpha_2+\alpha_3+\alpha_4=1$).

Відповідні дані та ілюстрація подані в табл. 4.2 і на рис. 4.7.

Розрахунок M_{rj} проведено для різних можливих комбінацій коефіцієнтів вагомості, зокрема, в табл. 4.2 наведено результати розрахунку для п’яти варіантів ($M_{r3} \div M_{r7}$):

$$M_{r3} \Rightarrow \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0{,}25;$$

$$M_{r4} \Rightarrow \alpha_1 = 0{,}2;\ \alpha_2 = 0{,}2;\ \alpha_3 = 0{,}3;\ \alpha_4 = 0{,}3;$$

$$M_{r5} \Rightarrow \alpha_1 = 0{,}15;\ \alpha_2 = 0{,}15;\ \alpha_3 = 0{,}35;\ \alpha_4 = 0{,}35;$$

$$M_{r6} \Rightarrow \alpha_1 = 0{,}35;\ \alpha_2 = 0{,}35;\ \alpha_3 = 0{,}15;\ \alpha_4 = 0{,}15;$$

$$M_{r7} \Rightarrow \alpha_1 = 0{,}35;\ \alpha_2 = 0{,}3;\ \alpha_3 = 0{,}2;\ \alpha_4 = 0{,}15. \qquad (4.44)$$

Таблиця 4.2.

Відносні значення параметрів P , $M_{r3} \div M_{r7}$

No	Марка сталі	*P*	M_{r3}	M_{r4}	M_{r5}	M_{r6}	M_{r7}
1	20	1,488	1,018	1,058	1,099	0,937	0,951
2	12 Х1МФ	1,707	1,073	1,129	1,185	0,961	0,982
3	20ЮЧ	1,488	1,015	1,055	1,095	0,935	0,949
4	17ГС	1,381	1,002	1,034	1,066	0,938	0,949
5	14ГБ	1,160	0,990	1,036	1,018	0,961	0,966
6	14ГБ (зварне з’єднання)	1,381	1,011	1,043	1,075	0,946	0,958
7	02 Х20Н6М3	1,00	0,989	0,989	0,989	0,990	0,989

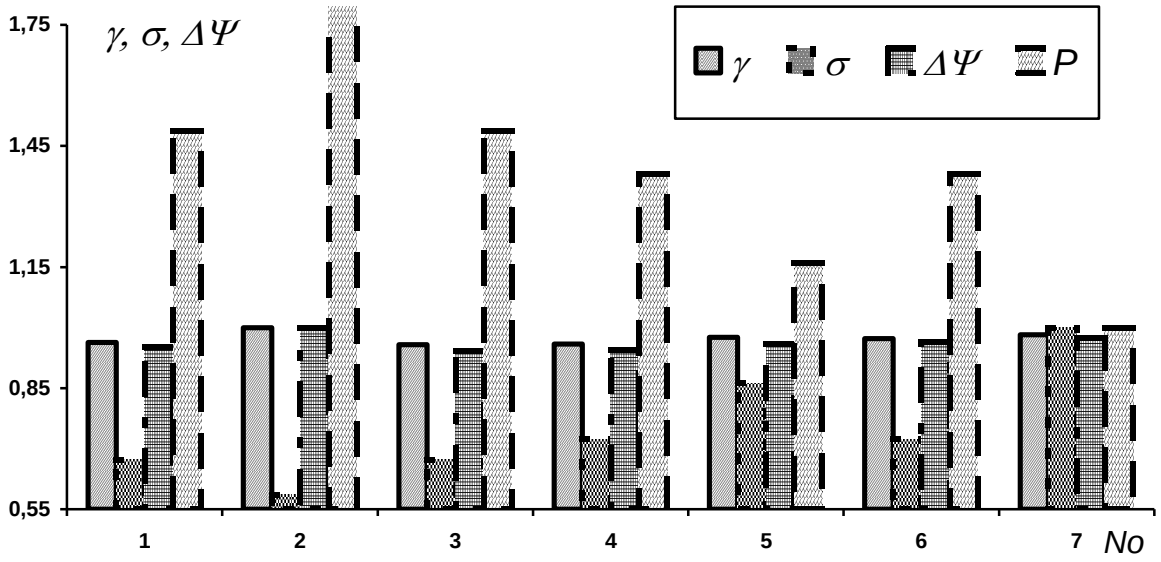


Рис. 4.7. Параметри $\sigma \equiv \sigma_{yzv}$, $\gamma \equiv \gamma_{mv}$, $\Delta\Psi \equiv \Delta\Psi_v$, що характеризують міцність і електричний бар’єр для матеріалу, й обернена ціна *P*, яка відповідає конкретній марці сталі (за даними табл. 4.2).

З допомогою співвідношень (4.43) і (4.44) встановлюємо оптимальний варіант параметра M_j і відповідну марку сталі в кожному із п’яти варіантів:

$$12\ X1M\Phi\left(No2\right)\Rightarrow M_{r3}=1{,}073;\ M_{r4}=1{,}129;\ M_{r5}=1{,}185;$$

$$02\ X20H6M3\left(No7\right)\Rightarrow M_{r6}=0{,}99;\ M_{r7}=0{,}989. \qquad (4.45)$$

Оптимальному варіанту марки сталі відповідає максимальна міцність, мінімальна ціна і максимальний бар'єр потенціалу на границі «метал – електроліт, насичений сірководнем».

Критерію (4.45) і числовим даним, які подано на рис. 4.6 і 4.7, відповідає алгоритм на рис. 4.8.

алгоритм
змінні
дійсні $\alpha_1, \alpha_2, \alpha_3, \alpha_4, M_{rj}, \gamma_{mv}, \sigma_{yzv}, \Delta\Psi_v, P$ *,a1,b1,b2,b3,b4,b5;*
дійсні aaa,bbbb,cccc;
початок
ввести $\alpha_1, \alpha_2, \alpha_3, \alpha_4$*;*
M_{rj} *=1.4;*
$\alpha_1 \cdot \gamma_{mv} + \alpha_2 \cdot \sigma_{yzv} + \alpha_3 \cdot \Delta\Psi_v + \alpha_4 \cdot P = M_{rj}$*;*
вивести M_{rj}*;*
вивести &Перевірка_процедури_DoIt, M_{rj}*;*
|- $M_{rj} = \alpha_1$, $r = \alpha_2$, $\alpha_1 > \alpha_2$ -|*;*
вивести &Перевірка_оператора_|-.....-|, M_{rj}*;*
M_{rj} *=1;*
$ d $M_{rj} = M_{rj}$ **a , a=a+1;*
вивести &Перевірка_оператора_$_x....., , M_{rj}*;*
кінець

Рис. 4.8. Застосування ІТ для корозіометрії стосовно визначення марки сталі з урахуванням ціни

Для встановлення значень коефіцієнтів вагомості (4.42), (4.44) використовували експертні оцінки.

Для оцінки узгодженості [100-108] думок експертів найчастіше використовують коефіцієнт конкордації, величина якого вказує на достовірність оцінок. Коефіцієнт конкордації W визначають за формулою [100]:

$$W = \frac{D_f}{D_{\max}} = \frac{\sum_{k=1}^{n_{ob}} \{a_i - 0{,}5 \cdot n \cdot (m+1)\}^2}{n^2 \cdot m \cdot (m^2 - 1)/12}, \qquad (4.46)$$

де D_f – фактична дисперсія (середньоквадратичне відхилення) підсумкових (впорядкованих, проранжованих) оцінок, які надані експертами; $D_{\max}$ – дисперсія підсумкових (впорядкованих) оцінок за умови, що думки експертів повністю збігаються; a_i – сумарна оцінка, одержана для i-го об'єкту; m – кількість коефіцієнтів вагомості; n – кількість досліджуваних об'єктів (експертів досліджуваних об'єктів.

Величина коефіцієнта конкордації W коливається в межах від 0 до 1. При нульовому значенні коефіцієнта зв'язок між оцінками різних експертів відсутній, тобто узгодженості немає. Думки експертів збігаються при значенні W =1.

Перевірка істинності коефіцієнта W реалізується з допомогою критерію згоди Пірсона (χ^2). Такий підхід використовується у тих випадках, коли необхідно встановити відповідність двох множин (розподілів) - емпіричного і теоретичного, або двох емпіричних. При цьому порівнюються частоти відповідних наборів даних, встановлюються невідповідності між ними і визначаються ймовірності таких невідповідностей.

Статистична характеристика для перевірки істинності коефіцієнта W визначається за критерієм Пірсона з допомогою співвідношення [100]:

$$\chi^2 = W \cdot n \cdot (m-1) \qquad (4.47)$$

і порівнюється з табличними даними.

Погодженість думок експертів щодо важливості критеріїв (4.41), (4.43), (4.45), які розглядались в даній праці, оцінювали за допомогою коефіцієнта конкордації Кендела [100]:

$$W_K = \frac{12 \cdot S}{n^2 \cdot (m_k^3 - m_k)}, \tag{4.48}$$

де – m_k – число аналізованих змінних (коефіцієнтів вагомості); n – кількість об’єктів (експертів); S – сума квадратів рангів –

$$S = \sum_{i=1}^{n} r_{ij}^2 - \frac{1}{n} \cdot \left(\sum_{i=1}^{n} r_{ij}^2 \right)^2, \tag{4.49}$$

r_{ij} – розставлені ранги суджень групи експертів.

Коригування фізико-хімічної та експертної інформації проводилось на основі терії опрацювання результатів вимірювань серій експериментальних досліджень [22,23].

Після обчислення значення коефіцієнтів W, W_K їх перевіряють на значимість, тобто перевіряється гіпотеза про випадковість одержання даного конкретного значення W чи W_K за процедурою статистичної перевірки гіпотез. Відповідні гіпотези можна інтерпретувати або як незалежність оцінок експертів, або як випадковість протиставлення експертами своїх оцінок.

Підсумкову схему (алгоритм) інформаційного забезпечення [62, 65] задачі прийняття рішення щодо вибору марки сталі, якій відповідає максимальна міцність матеріалу, мінімальна ціна і максимальний електричний бар’єр на границі «метал – корозійне (електролітичне) середовище» подано на рис. 4.6.

Максимальний електричний бар’єр на границі «метал – корозійне (електролітичне) середовище» забезпечить мінімальний струм у вершині тріщини на поверхні елемента конструкції.

Алгоритм на рис. 4.9 використовувався для задач підрозділу 4.3.

Рис. 4.9. Структура інформаційно-аналітичної моделі відбору, опрацювання і використання інформації про фізико-хімічні параметри контактуючих середовищ в задачі прийняття рішень щодо вибору оптимального варіанту марки сталі, якому відповідає максимальна міцність, мінімальна ціна і максимальний бар'єр потенціалу на границі «метал – електроліт, насичений сірководнем».

Розроблена низка фізико-хімічних параметрів і співвідношень, які характеризують метал і середовище поблизу вершини корозійної тріщини дозволяє проводити оцінювання енергетичних характеристик поверхневих шарів для циліндрів друкарських апаратів поліграфічних машин, а також вузлів машин для електрохімічного нарощування і травлення металів у процесі виготовлення форм глибокого друку.

4.4. Висновки до четвертого розділу

1. Удосконалено функціональну структуру ІТ для протикорозійного захисту з використанням даних про енергетичні характеристики поверхневих і міжфазних шарів та активаційних процесів, які характеризують метал, середовище і динаміку корозійних процесів поблизу вершини тріщини в корозійному середовищі.

2. На конкретних прикладах (сталь – алюміній, сталь – срібло, різні марки сталі – розчин електроліту з домішкою сірководню) проведено дослідження адекватності розроблених математичних моделей та відповідного інформаційного забезпечення.

3. Розроблена інформаційна технологія для протикорозійного захисту реалізована у вигляді програмного комплексу, що дає можливість знизити витрати на проведення важливих корозійних обстежень за рахунок адекватної математичної моделі та алгоритму обробки даних і дозволяє зменшити обсяги експериментальних вимірювань.

4. В запропонованій інформаційній технології для корозіометрії поєднано математичну модель визначення енергетичних характеристик поверхневих шарів металів в корозійних середовищах з методами теорії сенквенційних алгоритмів, які забезпечують формалізований опис і тотожні перетворення алгоритмів з метою мінімізації виконання операцій. Таке поєднання забезпечує раціональну основу створення та використання інформаційної технології.

5. Реалізовано відбір, опрацювання і використання інформації про фізико-хімічні параметри контактуючих середовищ в задачі прийняття рішень

щодо оптимального варіанту марки сталі, якому відповідає максимальна міцність, мінімальна ціна і максимальний бар'єр потенціалу на границі «метал – електроліт, насичений сірководнем»

6. Результати дослідження і комп'ютерну програму впровадженно в інформаційну технологію, що дозволяє на основі аналізу енергетичних характеристик поверхневих та міжфазних шарів металів приймати рішення щодо покращення умов протикорозійного захисту металів в агресивному електролітичному середовищі. Для розглянутих в роботі марок сталі (сталь 20, 12 Х1МФ, 20ЮЧ, 17ГС, 14ГБ (основний метал), 14ГБ (зварне з'єднання), 02Х20Н6М3) отримано результати і встановлено, що в процесі дії зовнішнього навантаження міжфазна енергія γ_m зростає і цьому зростанню відповідає діапазон відповідних їй приростів від 4 % до 11,8 %. Розраховані при цьому максимальні значення міжфазної енергії γ_m можна вважати критеріальними щодо міцності елемента конструкції і адгезійного покриття.

ОСНОВНІ РЕЗУЛЬТАТИ ТА ВИСНОВКИ

У дисертаційній роботі розв’язано науково-практичне завдання удосконалення інформаційної технології для визначення енергетичних характеристик поверхневих шарів металів з використанням алгебри алгоритмів. Інформаційна технологія призначена для розроблення рекомендацій щодо підвищення ефективності функціонування системи протикорозійного захисту.

При цьому отримано такі результати:

1. Проведено аналіз масиву даних, відповідних технології протикорозійного захисту, і обґрунтована необхідність створення інформаційної технології щодо визначення енергетичних характеристик поверхневих та міжфазних шарів металів з використанням алгебри алгоритмів; проаналізовано існуючі фізико-математичні моделі корозіометрії і виявлено недоліки щодо практичного їх застосування, оскільки вони мали частковий характер. Сформульовано принципи побудови узагальненої математичної моделі корозіометрії, яка акумулює великий обсяг інформації, і запропоновано відповідну концептуальну схему розробки інформаційної технології для аналізу інформаційних потоків в сфері електрохімії та термодинаміки нерівноважних процесів.

2. Вперше при розгляді інформаційної технології корозіометрії поєднано математичну модель для аналізу параметрів стану та процесів, що характеризують умови розчинення поверхонь зразків сталей різних марок, з моделлю розрахунку характеристик адгезії покрить елементів конструкцій.

3. Удосконалено функціональну структуру інформаційних потоків ІТ для протикорозійного захисту з використанням даних про енергетичні характеристики поверхневих і міжфазних шарів та активаційних процесів, які характеризують метал, середовище, динаміку корозійних процесів поблизу вершини тріщини в корозійному середовищі, дають можливість керувати масивами інформації і на основі алгебри алгоритмів приймати оптимальні рішення щодо проведення експериментальних досліджень.

4. На конкретних прикладах (сталь – алюміній, різні марки сталі – електроліт) репрезентативно проведено дослідження адекватності розроблених математичних моделей та відповідного інформаційного забезпечення і встановлено межі зміни параметрів та інваріантних співвідношень, які характеризують інформаційні потоки і забезпечують фільтрацію даних.

5. Розроблена інформаційна технологія для протикорозійного захисту реалізована у вигляді модуля програмного комплексу та пакету процедур. Програмний комплекс, в основі якого метод розкладу параметрів стану за малим параметром, дає можливість знизити витрати на проведення важливих корозійних обстежень за рахунок адекватної математичної моделі та алгоритмів обробки даних, що дозволяють зменшити обсяги експериментальних вимірювань.

6. В запропонованій інформаційній технології поєднано математичну модель визначення енергетичних характеристик поверхневих шарів металів в корозійних середовищах з методами теорії сенквенційних алгоритмів, які забезпечують формалізований опис і тотожні перетворення алгоритмів з метою мінімізації виконання операцій. Таке поєднання забезпечує основу створення програмних комплексів нового типу та більш ефективне використання результатів інформаційної технології. Синтезовано та мінімізовано за кількістю термів структуру системи генерації програм з операцій теорії секвенційних алгоритмів, яка включає операції секвентування, елімінування, циклів і це дає можливість зменшити непевності (похибки), а також точніше проводити операції обчислень енергетичних характеристик поверхневих шарів металів, які знаходяться в агресивному середовищі.

7. Розроблено концепцію і спосіб побудови автоматизованого програмного комплексу, в якому на відміну від відомих поєднано математичні засоби опису фізико-механічних полів, еквівалентних перетворень інформаційних потоків та дослідження математичних моделей корозіометрії з відповідними алгоритмами.

8. Запропоновано математичну модель (систему співвідношень) для оцінювання параметрів приграничних шарів елементів конструкцій, поверхня яких покрита захисними плівками і в цих плівках можуть бути дефекти типу тріщин. Оскільки числові розрахунки в цьому випадку складні й громіздкі, то для їх вдосконалення використано методи і засоби автоматизації математичних обчислень на основі алгебри алгоритмів. Модель з комплексом алгоритмів за рахунок використання розширеного набору параметрів стану забезпечує зниження витрат на проведення експериментальних досліджень.

9. Використовуючи можливості алгебри алгоритмів та єдиного підходу інформаційній технології щодо написання алгоритмів в конкретних задачах корозіометрії, на основі бібліотеки програм символьної математики побудовано структури алгоритмів, їх моделей, проведено моделювання та дослідження вірогідності отриманих результатів. З допомогою засобів об’єктно-орієнтованого програмування розроблено допоміжну комп’ютерну програму, що дає можливість ефективніше реалізовувати математичні обчислення, які забезпечують достовірні результати аналізу інформаційних потоків в сфері корозіометрії. Програмною реалізацією і апробацією підтверджено достовірність моделі системи генерації програм з операцій теорії секвенційних алгоритмів та коректне функціонування відповідної програми в конкретних задачах.

10. Розроблено в інформаційній технології і доведено до позитивного варіанту модель відбору, опрацювання і використання інформації про фізико-хімічні параметри контактуючих середовищ в задачі прийняття рішень щодо оптимального варіанту. Результати дослідження і програму системи генерації програм з операцій теорії секвенційних алгоритмів впроваджено у виробництво і вони використовуються в інформаційній технології, що дозволяє на основі аналізу енергетичних характеристик поверхневих та міжфазних шарів металів приймати рішення щодо покращення умов протикорозійного захисту металів в агресивному електролітичному середовищі.

11. Для розглянутих в роботі марок сталі (сталь 20, 12 Х1МФ, 20ЮЧ, 17ГС, 14ГБ (основний метал), 14ГБ (зварне з’єднання), 02Х20Н6М3) отримано результати оцінювання енергетичних характеристик поверхневих шарів і встановлено, що в процесі дії зовнішнього навантаження міжфазна енергія γ_m зростає і цьому зростанню відповідає діапазон відповідних їй приростів від 4 % до 11,8 %. Розраховані при цьому максимальні значення міжфазної енергії γ_m можна вважати критеріальними щодо міцності елемента конструкції. Такого типу результати дозволяють вибрати для практичного застосування марку сталі, якій відповідає максимальна міцність, мінімальна ціна і максимальний бар’єр захисного потенціалу на границі «метал – електроліт».

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Автоматизація математичних обчислень для оцінки параметрів поверхневих шарів / П. М. Сопрунюк, В. М. Юзевич, О. І. Огірко, П. В. Луговий // Відбір і обробка інформації. – 2000. – №14 (90). – С. 151–156.
2. Афанасьєва О. Ю. Інформаційна технологія формування стеганографічних систем для цифрових графічних середовищ на основі використання багатопараметричної адаптації: автореф. дис. на здобуття наук. ступеня доктора техн. наук: спец. 05.13.06 “Інформаційні технології” / О. Ю. Афанасьєва. – Львів, 2011. – 36 с.
3. Афанасьєва О. Ю. Метод комп'ютерної візуалізації результатів експериментальних досліджень з використанням принципів когнітивної графіки: автореф. дис. на здобуття наук. ступеня канд. техн. наук: спец. 05.13.06 “ Автоматизовані системи управління і прогресивні інформаційні технології” / О. Ю. Афанасьєва. – К. 2003. – 22 с.
4. Афанасьєва О. Ю. Методи утаємлення інформації, що укривається в цифровому середовищі / О. Ю. Афанасьєва, Ю. М. Коростіль // Науково-методична конференція «Сучасні проблеми телекомунікації і підготовки фахівців в галузі телекомунікацій» (Національний університет «Львівська політехніка»). – Львів, 2008. – С. 23–24.
5. Антропов Л. И. Теоретическая электрохимия / Л. И. Антропов. – М.: Высшая школа, 1975. – 560 с.
6. Бешелев С. Интенсификация науки: пути и особенности / С. Бешелев, Ф. Гурвич // Наука и жизнь. – 1986. – №10. – С. 62–66.
7. Бритковський В. М. Редактор формул алгоритмів і аналіз синтаксису і семантики алгебри алгоритмів-секвенцій / В. М. Бритковський // Матеріали 7 Всеукраїнської наукової конференції “Сучасні проблеми прикладної математики та інформатики”. – Львів: НУ ім. І. Франка, 2000. – С. 17–18.
8. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++ / Г. Буч: 2-е изд. – М.: Изд–во Бином, 1998. – 560 с.

9. Возна М. Аналіз і синтез алгоритмів системи керування електропроводом друкарської машини / М. Возна, В. Овсяк // Палітра друку. – 1999. – №6. – С. 31–32.

10. Возна М. Математична модель алгоритму "поштовх вперед - поштовх назад" / М. Возна // Поліграфія і видавнича справа. – Львів: УАД, 1999. – № 34. – С. 212–216.

11. Гавенко С. Ф. Логістика в поліграфічному виробництві: Навч. посіб. / С. Ф. Гавенко, Б. В. Дурняк, Р. С. Зацерковна. – Л.: Укр. акад. друкарства, 2006. — 144 с.

12. Гавенко С.Ф., Мельников О. В. Оцінка якості поліграфічної продукції: Навч. посіб. / С. Ф. Гавенко, О. В. Мельников, Ред.: Е.Т. Лазаренко; Укр. акад. друкарства. – Л.: Афіша, 2000. – 120 с.

13. Глушков В. М. Алгебра. Языки. Программирование / В. М. Глушков, Г. Е. Цейтлин, Е. Л. Ющенко. – К.: Наук. Думка, 1978. – 320 с.

14. Глущак Ю. І. Використання прогресивних інформаційних технологій в державному управлінні / Ю. І. Глущак, О. І. Огірко // Ефективність державного управління в контексті становлення громадянського суспільства. – Львів: УАДУ при Президенті України. – Львів, 2002. – С. 64 – 67.

15. Гофман В. Работа с базами данных в Delphi / В. Гофман, А. Хоменко. – СПб.: БХВ – Петербург, 2000. – 281 с.

16. Грис Д. Наука программирования / Д. Грис: пер. с англ. под ред. А. П. Ершова. – М.: Мир, 1984. – 416 с.

17. Грицык В. В. Распараллеливание алгоритмов обработки информации в системах реального времени / В. В Грицык. – К.: Наук. думка, 1981. – 261 с.

18. Грищенко В. Н. Методы и средства компонентного программирования / В. Н. Грищенко, Е. М. Лаврищева // Кибернетика и системный анализ. – 2003. – № 1. – С. 39–55.

19. Гуревич Р. С. Інформаційно-телекомунікаційні технології в навчальному процесі та наукових дослідженнях : навчальний посібник для студентів

педагогічних ВНЗ і слухачів інститутів післядипломної педагогічної освіти / Р. С. Гуревич, М. Ю. Кадемія. – Київ: Освіта України, 2006. – 366 с.

20. Дантеманн Дж. Программирование в среде Delphi / Дж. Дантеманн, Дж. Мишел, Д. Тейлор: пер. с англ. – К.: НИПФ “Диа Софт Лтд.”, 1995. – 608 с.
21. Денисенко В. В. Інформаційно-комунікаційні технології і навчально-виховний процес / В. В. Денисенко // Інформатика в школі. – Харків: ТОВ «Видавнича група «Основа»», 2009. – №4. – С. 2-4.
22. Джала Р. Визначення параметрів подвійного електричного шару для моделювання корозії на межі метал-електроліт / Р. Джала, В. Юзевич, М. Мельник, О. Чабан // Комп’ютерні науки та інформаційні технології. Вісник Національного університету “Львівська політехніка”. – Львів: Вид. НУ “Львівська політехніка”. – 2011. – № 694. – С. 370-376.
23. Джала Р. М. Основи обстеження і контролю корозійного стану підземних трубопроводів. Механіка руйнування та міцність матеріалів: Довідн. посібник: Під ред. В. В. Панасюка. Том 11: Міцність і довговічність нафтогазових трубопроводів і резервуарів / Р. М. Джала. – Львів: Сполом, 2009. – С. 143–184.
24. Джеймс Д. Мюррей. Энциклопедия форматов графических файлов / Джеймс Д. Мюррей, Уильям ван Райнер. – Киев: BHV, 1997. – 672 с.
25. Дмитрах І. М. Вплив корозійних середовищ на локальне руйнування металів біля концентраторів напружень / І. М. Дмитрах, В. В. Панасюк. – Львів: Ред. журналу «Фізико-хімічна механіка матеріалів», 1999. – 342 с.
26. Дорошенко А. Ю. Алгеброалгоритмічні основи програмування / А. Ю. Дорошенко, Г. С. Фінін, Г. О. Цейтлін. – К.: Наук.думка, 2004. – 457с.
27. Дурняк, Б. В. Системний аналіз та оптимізація параметрів книжкових видань: Моногр. / Б. В. Дурняк, І. В. Піх, В. М. Сеньківський. — Л.: Укр. акад. друкарства, 2006. — 197 с.
28. Дьяконов В. П. Математическая система Maple V R3/R4 /R-5 / В. П. Дьяконов. – М.: Солон, 1998. – 400с.

29. Дьяконов В. П. MATLAB 5.0/5.3. Система символьной математики / В. П. Дьяконов, И. В. Абраменкова. – М.: Полидж, 1999. – 640с.

30. Дьяконов В. П. Системы символьной математики Mathematica 2 и Mathematica 3. — М.: «СК-ПРЕСС», 1998. — С. 320.

31. Дьяконов В. П. Справочник по системе символьной математики Derive / В. П. Дьяконов. – М.: СК Пресе, 1998. – 256 с.

32. Дьяконов В. П. Техника визуализации учебных й научных задач с применением системы класса MathCAD. Информационные технологии / В. П. Дьяконов, И. В. Абраменкова. – М.: Нолидж, 1998. – 39 с.

33. Енергія активації деформованого металу поблизу вершини корозійної каверни / Джала Р. М., Юзевич В. М., Мельник М. І., Семенюк О. М. // Міжвузівський збірник «Наукові нотатки». Луцький Національний технічний університет. – Луцьк: Вид. ЛНТУ. – 2011. – № 31. – С. 114-119.

34. Захарова И. Г. Информационные технологии в образовании: Учеб. пособие для студ. высш. учеб. заведений / Ирина Гелиевна Захарова. – 2-е изд., стер. – М.: Издательский центр «Академия», 2005. – 192 с.

35. Кучеров Д. П. Алгоритм адаптивного термінального управління потрійним інтегратором / Кучеров Д.П., Василенко О.В., Іванов Б.П. // Вісник Вінницького політехнічного інституту. – 2009. – № 2. – С. 22-27.

36. Кучеров Д. П. Синтез адаптивных систем терминального управления: монография / Д. П. Кучеров, В. В. Камишин // АПН Украины, Центр. НИИ вооружения и воен. техники Вооруж. Сил Украины. – К.: Інфосистем, 2010. – 232 с.

37. Кнут Д. Э. Искусство программирования. Основные алгоритмы / Д. Э. Кнут. – М.: Вильямс, 2000. – Том 1. – 720 с.

38. Кормен Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест. – М.: МЦНМО, 2000. – 960 с.

39. Корн Г. Справочник по математике / Г. Корн, Т. Корн. – М.: Наука, 1972. – 831 с.

40. Лаврищева Е. М. Методы программирования. Теория, инженерия, практика / Е. М. Лаврищева. – Киев: Наукова думка, 2006. – 451с.

41. Лукович В. В. Розрахунок параметрів катодного захисту магістрального трубопроводу / В. В. Лукович. – Київ, 2011. – 47 с. – (Препр. / НАН України, Ін-т проблем матеріалознавства ім. І. М. Францевича).

42. Мальцев А. И. Алгоритмы и рекурсивные функции. 2-е Изд. / А. И. Мальцев. – М.: Наука, 1986. – 368с.

43. Марка Д. А. Методология структурного анализа и проектирования / Д. А. Марка, К. МакГруэн. – М.: МетаТехнология, 1997. – 346 с.

44. Марков А. А. Теория алгоритмов / А. А. Марков. – Тр. матем. ин-та АН СССР им. В. А. Стеклова, 38. – М.: Изд. АН СССР, 1951. – С. 176–189.

45. Морзе Н. В. Основи інформаційно-комунікаційних технологій / Н. В. Морзе. – К.: Видавнича група BHV, 2006. – 298 с.

46. Назаркевич М. А. Методи підвищення ефективності поліграфічного захисту засобами Ateb-функцій / М. А. Назаркевич. – Львів: Вид-во НУ "Львів. політехніка", 2011. – 186 с.

47. Назаркевич М. А. Автоматизовані видавничо-поліграфічні процеси: навч. посіб. / М. А. Назаркевич. – Львів. 2010. – 98 с.

48. Овсяк В. К. Синтез, оптимізація і дослідження математичних моделей алгоритмів керування електроприводом друкарської машини / В. К. Овсяк А. М. Назаркевич // Вісник Державного університету “Львівська політехніка”. Прикладна математика. – 1999.– № 364. – С. 105 – 110.

49. Овсяк В. К. Засоби еквівалентних перетворень алгоритмів інформаційно-технологічних систем / В. К. Овсяк // Доповіді національної Академії наук України. – 1996. – № 9. – С. 83–89.

50. Овсяк В. К. Модели и моделирование алгоритмов / В. К. Овсяк, Я. П. Драган, Л. С. Сикора // Материалы 2-ой международной конференции “Теория и техника передачи, приема и обработки информации. Часть 2”. – Харьков-Туапсе: 1996. – С. 303–304.

51. Овсяк В. Алгебра алгоритмів-секвенцій / В. Овсяк // Збірник наукових праць «Компютерні технології друкарства». – 1999. – №3. – С. 3–13.

52. Овсяк В. Алгоритми: методи побудови, оптимізації, дослідження вірогідності / В. Овсяк. – Львів: Світ, 2001. – 160 с.

53. Овсяк В. Теорія секвенційних алгоритмів і проектування комп’ютерних систем: Навчальний посібник / В. Овсяк, В. Бритковський, О. Овсяк, Ю. Овсяк. – Львів: УАД, 2001. – 141 с.

54. Огірко О. І. Автоматизація математичних обчислень для оцінки параметрів поверхневих шарів / П. М. Сопрунюк, В. М. Юзевич, О. І. Огірко, П. В. Луговий // Відбір і обробка інформації. – 2000. – Вип. 14(90) – С. 151-156.

55. Огірко О. Автоматизовані способи розпізнавання для алгебри алгоритмів / О. Огірко // Автоматика. – 2000. – Т.6. – Львів: Державний НДІ інформаційної інфраструктури, 2000. – С. 138 – 140.

56. Огірко О. І. Алгебра алгоритмів та інформаційні технології / О. І. Огірко. – [Електронний ресурс]. – Режим доступу: http://www.ohirko2008.narod.ru/ С.7. навч. посіб. для студ. вузів. – 80 Min / 700 MB. –Львів. УАД. 2012

57. Огірко О. І. Інформаційна технологія корозіометрії оцінювання об’єктів із сталі / О. І. Огірко // Поліграфія і видавнича справа . – 2012. – № 4 (60). – С. 140–152.

58. Огірко О. І. Інформаційні системи в оперативній поліграфії : навчальний посібник / Т. Й. Гаранько, О. І. Огірко. – Львів: Університет «Львівський Ставропігіон». – 2011. – 174 с.

59. Огірко О. І. Інформаційні технології: [Електронний ресурс]. – Режим доступу: https://sites.google.com/site/inftechnolog/home Навч. посіб. для студентів вузів.– 80 Min / 700 MB. –Львів. УАД. 2012.

60. Огірко О. Комп’ютерна реалізація алгебри алгоритмів / О. Огірко // Комп'ютерні технології друкарства. – 2000. – № 5. – С. 200–205.

61. Огірко О. І. Свідоцтво про реєстрацію авторського права на твір №.48276. Україна.Інформаційна технологія визначення енергетичних характеристик поверхневих шарів металів з використанням алгебри алгоритмів.-17 С.

Авторське майнове право належить Огірко О. І. Зареєстровано в м. Київ, 14. 03. 2013.

62. Огірко О. І. Корозіометрія: [Електронний ресурс]. – Режим доступу: http://www.youtube.com/watch?v=cprWSweLEiA Навч. посіб. для студ. вузів.– 80 Min / 700 MB. –Львів. УАД. 2012.

63. Огірко О. Модель комп’ютерної системи генерації програм з формул алгоритмів / О. Огірко // Комп'ютерні технології друкарства. – 2001. – № 6. – С. 93–97.

64. Огірко О. Модель системи генерації програм СКАНЕР / О. Огірко // Комп'ютерні технології друкарства. – 2001. – № 6. – С. 42–48.

65. Огірко О. Оцінка поверхневої енергії сталей у сірководневих середовищах / П. Сопрунюк, В. Юзевич, О. Огірко // Фіз.-хім. механіка матеріалів. Проблеми корозії та протикорозійного захисту матеріалів. – 2000. – Т. 2, № 1. – С. 726-730.

66. Огірко О. І. Алгебра алгоритмів та інформаційні технології . : [Електронний ресурс]. – Режим доступу: http://ohirko.io.ua/ Львів. УАД. 2012.

67. Огірко О. Синтаксис оптимізації моделі та моделювання синтаксису механізму розпізнавання символіки алгебри алгоритмів секвенції / О. Огірко // Комп’ютерні технології друкарства. – 2000. – № 5. – С. 296–303.

68. Огірко О. І. Мультимедійно інформаційні технології корозіометрії / О. І. Огірко / Науково-практична конференція «Мультимедійні технології в освіті та інших сферах діяльності»: Тези доповідей. – К.: НАУ, 2012. – С. 57.

69. Огірко О. І. Комп’ютерне та програмне забезпечення математичного моделювання: навчальний посібник. / О. І. Огірко, Л. М. Ясінська, М. Ф. Ясінський. – Львів: Університет «Львівський Ставропігіон», 2004 – Т. 1. – С. 1-22.

70. Огірко О. І. Математичне моделювання та інформаційних технології електронних таблиць / О.І. Огірко, Л. М. Ясінська, М. Ф. Ясінський – Львів: Університет «Львівський Ставропігіон», 2004 – Т. 1. – С. 23-34.

71. Основи проектування цифрових логічних пристроїв: навч. посіб. / Б. В. Дурняк, І.Т. Стрепко, Г.Н. Тітов, О.В. Тимченко. – Л.: УАД, 2006. – 272 с.

72. Плескач В. Л. Агентные технологии / В. Л. Плескач, Ю. В. Рогушина. – К.: КНТЕУ, 2005. – 337с.

73. Поляков С. Г. Система корозійного моніторингу трубопроводів / С. Г. Поляков, А. В. Клименко, С. Ю. Коваленко // Наука та інновації. – 2010. – Т. 6. № 5. – С. 25-28.

74. Похмурський В. І. Адсорбційно-електрохімічна теорія корозійної втоми металів / В. І. Похмурський // Фізико-хімічна механіка матеріалів. – 2010. – № 2. – С. 21-30.

75. Рамбо Дж. UML. Специальный справочник / Дж. Рамбо, А. Джекобсон, Г. Буч. – СПб.: Питер. – 2002. – 656 с.

76. Русин Б. П. Структурно-лингвистические методы распознавания изображений в реальном времени / Б. П. Русин. – К: Наук. думка, 1986. – 128 с.

77. Русын Б. П. Матричные пульсирующиеся информационные решетки / Б. П. Русын, В. И. Шмойлов, А. И. Адамацкий, М. Н. Кузьо. – Львов: Меркатор, 2003. – 338 с.

78. Русин Б. П. Системи синтезу, обробки та розпізнавання складноструктурованих зображень / Б. П. Русин. – Львів: Вертикаль, 1997. – 264 с.

79. Сван Т. Delphi 4. Библия разработчика / Т. Сван; пер. с англ. – К.; СПб.: Диалектика, 1998. – 672 с.

80. Седжвик Р. Фундаментальние алгоритмы на С++. Анализ. Структури данных. Сортировка. Поиск / Р. Седжвик. – К.: ДиаСорт, 2001. – 688 с.

81. Сікора Л. С. Перспективні інформаційні технології в системах автоматичного управління енергоактивними об'єктами виробничих структур: монографія / Л. С. Сікора, М. О. Медиковський, В. В. Грицик; НАН України. Держком зв'язку та інформ. України. Держ. НДІ інформ. інфраструктури. – Львів, 2002. – 416 с.

82. Скляров С. А. Алгоритм оптимизации режимов работы системы противокоррозионной защиты магистральных трубопроводов / С. А. Скляров // АСУ и приборы автоматики – 1999 г. – № 110. – С. 45 – 50.

83. Скляров С. А. Математическая модель идентификации распределения защитного потенциала вдоль магистрального трубопровода при заданных режимах работы установок катодной защиты / С. А. Скляров // АСУ и приборы автоматики 1998. – №108 – С. 77 – 85.

84. Скляров С. А. Математическая модель распределения разности потенциалов труба – земля с учетом влияния электрических полей анодных заземлителей / С. А. Скляров, З. В. Дударь // Новые решения в современных технологиях. – 2000. – № 80 – С. 30 – 33.

85. Скляров С. О. Математичні моделі та інформаційні технології автоматизованого управління системами протикорозійного захисту магістральних трубопроводів: автореф. дис. на здобуття наук. ступеня канд. техн. наук: спец. 05.13.06 “Автоматизовані системи управління та прогресивні інформаційні технології” / С. О. Скляров. – Харків, 2002. – 19 с.

86. Скляров С. А. Передумови побудови АСУ системи електрохімзахисту магістральних трубопроводів / С. А. Скляров // Радиоэлектроника и информатика – 1998. – №4 – С. 41 – 43.

87. Стащук М. Розрахунок зміщення електродного потенціалу, зініційованого пружним полем, на межі еліптичного отвору із середовищем / М. Стащук, Л. Журавчак, М. Дорош // Проблеми корозії та протикорозійного захисту матеріалів: в 2-х т. / Спецвипуск журналу «Фізико-хімічна механіка матеріалів». – Львів: Фізико-механічний інститут НАН України, 2010. – Т. 1, № 8. – С. 49-54.

88. Терендій О. До проблеми розробки інтерфейсних систем / О.Терендій // Комп’ютерні технології друкарства. Збірник наукових праць. – Львів: УАД. – 1998. – С. 182-183.

89. Тимченко О. В. Дослідження особливостей функціонування сенсорних мереж / М. Ю. Зеляновський, О. В. Тимченко // Моделювання та

інформаційні технології: Зб. наук. пр. – К.: ІПМЕ ім. Г.Є.Пухова НАН України, 2010. – Вип. 58. – С. 172-180.

90. Тимченко О. В. Дослідження якості багатоадресної передачі мультимедійного трафіку / Тимченко О.В., Верхола Б.М., Кирик М.І., Самі Аскар // Зб. наук. пр. ІПМЕ НАН України. –К.: 2008. – Вип. 48. – С. 161-169.
91. Тимченко О. В. Ефективність передачі відеотрафіку MPEG в мережі MPLS / О. В. Тимченко, Б. М. Верхола, А. Т. Ратич // Моделювання та інформаційні технології: Зб. наук. пр. — К.: ІПМЕ ім. Г. Є. Пухова НАН України, 2010. – Вип. 55. – С. 206-213.
92. Тимченко О. В. Математичне моделювання і реалізація систем керування стрічкопровідними системами / О. В. Тимченко, Б. В. Дурняк. – К., Видавн. центр "Просвіта", 2003. – 232с.
93. Туфанов Д. Г. Коррозионная стойкость нержавеющих сталей, сплавов и чистых металлов (справочник) // Д. Г. Туфанов. – М.: Металлургия, 1982. – 352 с.
94. Урский В. Методология программирования / В. Урский; пер. с англ. под ред. А. П. Ершова. – М.: Мир, 1981. – 264 с.
95. Успенский В. А. Теория алгоритмов: Основные открытия и приложения / В. А Успенский, Л. Семенов. – М.: Наука, Гл. ред. физ. - мат. лит., 1987. – 288 с.
96. Устройство для натяжения декеля на офсетном цилиндре печатной машины / (В. Н. Юзевич, В. Т. Сенкусь, Я. И. Чехман, В. Е. Босак, И. Н. Кравчук) // А.с. № 1730794. От 03.01.1992 г. По заявке № 4221233. Гос. Реестр Изобретений СССР. – 4 с.
97. Цейтлин Г. Е. Введение в алгоритмику / Г. Е. Цейтлин. – К.: Сфера, 1998. – 310 с.
98. Чвірук В. П. Електрохімічний моніторинг техногенних середовищ / В. П. Чвірук, С. Г. Поляков, Ю. С. Герасименко. – Київ: Академперіодика, 2007. – 223 с.

99. Чернецки К. Порождающее программирование. Методы, инструменты, применение / К. Чернецки, У. Айзенекер. – Издательский дом Питер. – М.: СПб. – Харьков. – Минск, 2005. – 730 с.

100. Юзевич В. М. Діагностика матеріалів і середовищ. Енергетичні характеристики поверхневих шарів / В. М. Юзевич, П. М. Сопрунюк. – Львів: ФМІ ім. Г. В. Карпенка НАН України, вид-во “СПОЛОМ”. – 2005. – 292 с.

101. Юзевич В. М. Енергія адгезійних зв’язків у системі мідь – тверде тіло / В. М. Юзевич, П. М. Сопрунюк, Б. П. Коман, П. В. Луговий // Український фізичний журнал. – Київ, 2005. – Т. 50, № 6. – С. 575 – 581.

102. Юзевич В. М. Інформаційна технологія оцінювання станів об’єктів із сталі в морському середовищі із сірководнем з використанням алгебри алгоритмів / В. М. Юзевич, О. І. Огірко // Наукові записки. – 2012. – № 4 (41). – С. 160–172.

103. Юзевич В. М. Критерії міцності твердого тіла з урахуванням розмірного ефекту і впливу середовища / В. М. Юзевич // Фіз.-хім. механіка матеріалів. – 1999. – № 2. – С. 80–85.

104. Юзевич В. Моделювання туристичних потоків з використанням мурашиних алгоритмів / В. Юзевич, Н. Крап // Комп’ютерні науки та інформаційні технології. Вісник Національного університету “Львівська політехніка”. – Львів: Вид. НУ “Львівська політехніка”. – 2011. – № 710. – Р. 98-105.

105. Юзевич В. Моделювання корозійних процесів у системі «метал-електроліт» з урахуванням дифузійного імпедансу / В. Юзевич, І. Огірко, Р. Джала // Фізико-математичне моделювання та інформаційні технології. – 2011. – Вип. № 13. – С. 173-181.

106. Юзевич В. М. Роль масштабного фактора при випробуванні поліуретанового зразка різної твердості / В. М. Юзевич, Я. І. Чехман // Вища школа. Поліграфія і видавнича справа. – Львів, 1988. – № 24. – С. 47-49.

107. Юзевич В. М. Розрахунок впливу кількісного фактора на деформаційну характеристику поліуретанового зразка / В. М. Юзевич, Я. І. Чехман // Вища школа. Поліграфія і видавнича справа. – Львів, 1987. – № 23. – С. 46–50.

108. Юзевич В. М. Термодинамічний підхід до контролю корозії підземних трубопроводів / Р. М. Джала, В. М. Юзевич, О. М. Семенюк // Матеріали 16-ї міжнародної науково-технічної конференції “Електромагнітні та акустичні методи неруйнівного контролю матеріалів та виробів ЛЕОТЕСТ-2011”. – Славське Львівської обл., 21-26.02.2011. – С. 152–155.

109. Юзевич В. М. Факторний аналіз як засіб управління конфігураціями проектів туристичних потоків / В. М. Юзевич, Н. П. Крап // Матеріали III Всеукраїнської науково-практичної конференції “Сучасні інформаційні технології в економіці, менеджменті та освіті”. 21 листопада 2012 р., Львів. – С. 15-19.

110. Юзевич В. М. Інформаційна технологія оцінювання станів об’єктів із сталі в корозійному середовищі із сірководнем на основі термодинамічної моделі / В. М. Юзевич, О. І. Огірко // Матеріали III Всеукраїнської науково-практичної конференції “Сучасні інформаційні технології в економіці, менеджменті та освіті”. 21 листопада 2012 р., Львів. – С. 71-76.

111. Crnkovik I. Component–Based Software Engineering: building systems from Components at 9th Conference and Workshops on Engineering of Computer-Based Systems / I. Crnkovik, S. Larsson, J. Stafford // Software Engineering Notes. – 2002.– 27.– N 3 .– P. 47–50.

112. Demark D. A. SADT: Structured Analysis and Design Technique / D. A Demark, R.L. McGowan. – New York: McCray Hill, 1988. – 378 p.

113. Jacobson I. Software Reuse: Architecture, Process and organization for Business Success / I. Jacobson, M. Griss, P. Johnson. – Addison Wesley, Reading , MA, May 1997. – 501 p.

114. Yuzevych V. Modelling adhesion bonds in copper condensates on monocrystalline silicon / V. Yuzevych B. Koman // Electronics and information technologies. – 2011. – Issue 1. – P. 89-97.

115. Katz S. Logical Analysis of Programs / S. Katz, Z. Manna // Comm. ACM. – 1976. – Vol.19, №4. – P.188-206.

116. Kiselev I. Aspect–Oriented Programming with AspectJ / I. Kiselev. – Indianapolis: USA: SAMS Publishing, 2002. – 164 p.

117. London R. L. Programm Verification / R. L. London // Research Directions in Software Technology (Wegner P. ed.). – MIT Press, 1980. – P. 302-315.

118. Post E. L. Finite combinatory proceses – formulation 1 / E. L.Post // JSL. – V. 1, №3, 1836. – P. 103-105.

119. Schonhage A. Univetselle Turing Speicherung / A. Schonhage // Automaten. Theorie und formale Sprachen. Eds. J. Dorr, Ghots. – Manuheim, 1970. – P. 369.-389.

120. Shoham Y. Agent–oriented programming / Y. Shoham // Artif. Intell. – 1993 – 60, №1. – P. 51–92.

121. Turing A. M. On compurable numbers, wirs an application to the Entscheidungs problem / A. M. Turing // Proceedings of LMS. Set.2. – V. 42, №3, 4, 1936. – P. 230-265.

122. Zanevskyy I. Mathematics computer systems and modeling of complex systems in sports / I. Zanevskyy // Theory and methods of the physical education. 2007. – № 8. – S. 17-23

123. Zanevskyy I. Modeling and computer simulation of bow stabilization in the vertical plane / I. Zanevskyy // International Journal of Sports Science and Engineering. – 2008. – Vol. 2, No. 1 – P. 3-14.

124. Zanevskyy I. Mechanical and mathematical modeling of sport archery arrow ballistics / I. Zanevskyy // International Journal of Computer Science in Sport. – 2008. – Vol. 7, No. 1. – P. 37-46.

Д О Д А Т К И

Додаток 1

Роздрук програм системи генерації програм з формул теорії секвенційних алгоритмів стосовно інформаційної технології визначення енергетичних характеристик поверхневих шарів металів.

{Процедура перевірки наявності всіх розділів алгоритму}

```
procedure TEditForm.Testing1Click(Sender: TObject);
var
i,poch:integer;
g1,g2,g3,g4,gg:boolean;
begin
FrameForm.Memo1.Lines.Clear;
FrameForm.Memo2.Lines.Clear;
FrameForm.Memo3.Lines.Clear;
FrameForm.Memo2.Lines.Add('Результати:');
FrameForm.Memo1.Lines.Add('Помилки і підказки:');
FrameForm.Memo3.Lines.Add('Змінні:');
g1:=false;
g2:=false;
g3:=false;
g4:=false;
for i:=0 to editor.Lines.Count do
begin
if editor.Lines.Strings[i]='алгоритм' then g1:=true;
if editor.Lines.Strings[i]='змінні' then g2:=true;
if editor.Lines.Strings[i]='початок' then
begin
g3:=true;
```

```
poch:=i;
end;
if editor.Lines.Strings[i]='кінець' then g4:=true;
end;

if (g1=false) then FrameForm.Memo1.Lines.Add('Відсутній заголовок');
if (g2=false) then FrameForm.Memo1.Lines.Add('Відсутній розділ змінних');
if (g3=false) then FrameForm.Memo1.Lines.Add('Відсутній початок
алгоритму');
if (g4=false) then FrameForm.Memo1.Lines.Add('Відсутній кінець
алгоритму');
if ((g1=true)and(g2=true)and(g3=true)and(g4=true)) then
FrameForm.Memo1.Lines.Add('Всі розділи алгоритму присутні');

end;
//////////////////////////////////////////
```

{Процедура яка отримує як параметр кожен рядок алгоритму і в залежності від типу рядка виконує певні дії}

```
procedure TEditForm.Scaner(s:string;var bll:boolean);
var ps,iv:string;
   i,ii,cv,i11:integer;
   sdo,op1,op2,umova,op3,s1,ps1:string;
   bool,bl,bol:boolean;
begin
sdo:=s;
ps:=NextMegaWord(s);
ps1:=s;
bl:=false;
```

{Опрацювання радка опису змінних }

```
if ps='дійсні' then
    begin
     bl:=true;
     ps:=NextMegaWord(s);
      while (ps<>';')and(ps<>'') do
         begin
          if ps<>',' then
           begin
           bol:=true;
             for i:=1 to varCount do
                  begin
                    if varName[i]=ps then
                        begin
                           ShowMessage('Помилка.Подвійне використання імені
змінної:'+ps);
                           bll:=false;
                           bol:=False;
                           Exit;
                        end;
                  end;
           if bol=true then
              begin
               varCount:=varCount+1;
               varName[varCount]:=ps;
              end;
            end;
         ps:=NextMegaWord(s);
         end;
    end;
```

{Опрацювання оперетора введення }

```
if ps='ввести' then
    begin
      bl:=true;
      ps:=NextMegaWord(s);
      while (ps<>';')and(ps<>'') do
         begin
         if ps<>',' then
         begin
         iv:=InputBox('Введення даних','Введіть '+ps,'');
              for i:=1 to varCount do
                 begin
                  if varName[i]=ps then
                      begin
                      varVal[i]:=StrToFloat(iv);
                      end;
                 end;
             end;

         ps:=NextMegaWord(s);
         end;
    end;
```

{Опрацювання рядка з оператором умови}

```
 if ps='|-' then
      begin
           bl:=true;
      ps:=NextMegaWord(s);
          op1:=ps;
          ps:=NextMegaWord(s);
          ps:=NextMegaWord(s);
          op2:=ps;
```

```
        ps:=NextMegaWord(s);
        ps:=NextMegaWord(s);
        umova:=ps;
        s1:=s;
        op3:=NextMegaWord(s1);
        umova:=umova+' '+op3;
        op3:=NextMegaWord(s1);
        umova:=umova+' '+op3;
         bool:=IfThen(umova);
          if bool=true then
                begin
                DoIt(op1);
                end
                else
                begin
                DoIt(op1);

                end;

    end;
```

{Опрацювання рядка з оперетором циклу }

```
 if ps='$' then
    begin
    bl:=true;
    ps:=NextMegaWord(s);
    while (ps<>';')and(ps<>'') do
      begin
        i11:=0;
        for i:=1 to VarCount do
            begin
```

```
            if varName[i]=ps then
                begin
                  cv:=Round(varVal[i]);
                  i11:=i;
                end;
           end;
        if i11=0 then cv:=StrToInt(ps);
         if cv<1 then
              begin
               ShowMessage('Не може бути виконана процедура циклу через
невірний показник кількості повторень циклу');
               bll:=false;
               Exit;
              end;
         sdo:=s;
          for i:=1 to cv do
          begin
         s:=sdo;
         ps:=NextMegaWord(s);
          while (ps<>';')and(ps<>'') do
           begin
           if ps<>',' then
              begin
                DoIt(ps);
              end;
             ps:=NextMegaWord(s);
          end;

         end;
        end;
```

```
      end;
{Опрацювання радка з оперетором виведення }
if ps='вивести' then
    begin
         bl:=true;
     ps:=NextMegaWord(s);
      while (ps<>';')and(ps<>'') do
         begin
           if ps<>',' then
             begin
               if ps[1]='&' then
                 begin
                 op1:=ps;
                 op1[1]:=' ';
                 FrameForm.Memo2.Lines.Add(op1);
                 end
               else
               begin
               for i:=1 to varCount do
                 begin
                 if varName[i]=ps then
                      begin
                      FrameForm.Memo2.Lines.Add(FloatToStr(varVal[i]));
                      end;
                 end;
               end;
             end;
         ps:=NextMegaWord(s);
         end;
```

```
    end;
{Опрацювання рядка з оперетором присвоєння}
   if (bl=false)and(StandId(ps)=0)and(sdo<>'') then DoIt(sdo);

end;
{Функція, яка розбиває рядок на окремі слова і розділювачі}
function TEditform.NextMegaWord(var S: String): String;
 var
   i,Len : Integer;
   TempS1 : String;
   NotDel: Boolean;
begin
 TempS1:='';
 Len:=0;

 NotDel:=True;
  for i:=1 to Length(S) do
    if (S[i]<>#9) and (S[i]<>' ') and (S[i]<>',') and (S[i]<>';') then begin
      TempS1:=TempS1+UpCase(S[i]);
      Len:=Len+1;
    end else
          if (TempS1<>'') then begin
            Delete(S,1,i-1);
            NotDel:=False;
            Break ;
          end else
                if (S[i]<>#9) and(S[i]<>' ') then begin
                  TempS1:=TempS1+UpCase(S[i]);
                  Delete(S,1,i);
```

```
                NotDel:=False;
                Break ;
              end;
  if NotDel then Delete(S,1,length(s));
 NextMegaWord:=TempS1;
end;
```

{Функція, яка опрацьовує рядок умови}

```
function TEditForm.IfThen(s:string):boolean;
var ps,s1,s2:string;
    s1v,s2v:real;
    i:integer;
    b1,b2:boolean;
begin
  s1:=NextMegaWord(s);
  ps:=NextMegaWord(s);
  s2:=NextMegaWord(s);
 b1:=false;
  b2:=false;
  for i:=1 to varCount do
      begin
        if s1=varName[i] then
        begin
        s1v:=varVal[i];
         b1:=true;
        end;
        if s2=varName[i] then
        begin
        s2v:=varVal[i];
```

```
      b2:=true;
     end;
   end;
 if b1=false then  s1v:=StrToFloat(s1);
 if b2=false then  s2v:=StrToFloat(s2);
if ps='<' then
   begin
   if s1v<s2v then IfThen:=true
       else IfThen:=false;
   end;
if ps='>' then
   begin
   if s1v>s2v then IfThen:=true
       else IfThen:=false;
   end;
 if ps='=' then
   begin
   if s1v=s2v then IfThen:=true
       else IfThen:=false;
   end;
 if ps='<=' then
   begin
   if s1v<=s2v then IfThen:=true
       else IfThen:=false;
   end;
 if ps='>=' then
   begin
   if s1v>=s2v then IfThen:=true
       else IfThen:=false;
   end;
```

```
end;
{Процедура яка ініціалізує виконання алгоритму}
procedure TEditForm.Run2Click(Sender: TObject);
var
i,poch:integer;
g1,g2,g3,g4,b:boolean;
begin
FrameForm.Memo1.Lines.Clear;
FrameForm.Memo2.Lines.Clear;
FrameForm.Memo3.Lines.Clear;
FrameForm.Memo2.Lines.Add('Результати:');
FrameForm.Memo3.Lines.Add('Змінні:');
FrameForm.Memo1.Lines.Add('Помилки і підказки:');

varCount:=0;
//////////////////////////////////
g1:=false;
g2:=false;
g3:=false;
g4:=false;
for i:=0 to editor.Lines.Count do
begin
if editor.Lines.Strings[i]='алгоритм' then g1:=true;
if editor.Lines.Strings[i]='змінні' then g2:=true;
if editor.Lines.Strings[i]='початок' then
begin
g3:=true;
poch:=i;
end;
if editor.Lines.Strings[i]='кінець' then g4:=true;
```

```
end;
if ((g1=false)or(g2=false)or(g3=false)or(g4=false)) then
  begin
   ShowMessage('Програма не може бути виконана.');
   Exit;
  end;

//////////////////////////////////
for i:=0 to editor.Lines.Count do
       begin
        b:=true;
       FrameForm.Label1.Font.color:=RGB(255,0,0);
       FrameForm.Label1.Caption:='Виконано до рядка номер '+IntToStr(i);
       Scaner(editor.Lines.Strings[i],b);
        if b=false then Exit;
       end;
   FrameForm.Label1.Font.color:=RGB(0,150,0);
  FrameForm.Label1.Caption:=' Роботу успішно завершено.';
end;
{Функція яка виконує присвоєння і обрахунок виразів}
function TEditForm.DoIt(s:string):boolean;
var sp,sr,spr,s1:string;
  i,oc,i1,i2,i3:integer;
  op1,op2:real;
  b1,b2,minus:boolean;
     begin
      sp:=s;
      spr:=s;
      oc:=0;
      minus:=false;
```

```
for i:=1 to strlen(PChar(sp)) do
  begin
    if sp[i]='=' then
      begin
      sp[i]:=' ';
        if sp[i+1]='-' then
          begin
          minus:=true;
          sp[i+1]:=' ';
          end;
      end;
    if sp[i]='_' then sp[i]:=' ';
    if sp[i]='+' then
        begin
          sp[i]:=' ';
          oc:=1;
        end;
    if sp[i]='-' then
        begin
          sp[i]:=' ';
          oc:=2;
        end;
     if sp[i]='*' then
        begin
          sp[i]:=' ';
          oc:=3;
        end;
     if sp[i]='/' then
        begin
          sp[i]:=' ';
```

```
          oc:=4;
        end;
  end;

sr:=NextMegaWord(sp);
for i:=1 to VarCount do
begin
if sr=varName[i] then i1:=i;
end;
if oc=0 then
   begin
    sr:=NextMegaWord(sp);
    b1:=false;
     for i:=1 to VarCount do
         begin
         if sr=varName[i] then
            begin
            op1:=varVal[i];
            if minus=true then op1:=-op1;
            b1:=true;
            end;
         end;
    if b1=false then
          begin

          if StandId(sr)=0 then
               begin
               sr:=PointToPoint(sr);
               op1:=StrToFloat(sr);
               if minus=true then op1:=-op1;
```

```
                end
                else
                   begin
                   i2:=0;
                    s1:=sr;
                    sr:=NextMegaWord(sp);
                    for i:=1 to VarCount do
                        begin
                         if sr=varName[i] then i2:=i;
                        end;
                   if i2=0 then
                   begin
                   sr:=PointToPoint(sr);
                   op1:=StrToFloat(sr);
                   end
                   else op1:=VarVal[i2];
                   if StandId(s1)=1 then op1:=sin(op1);
                   if StandId(s1)=2 then op1:=cos(op1);
                   if StandId(s1)=3 then op1:=sin(op1)/cos(op1);
                   if StandId(s1)=4 then op1:=cos(op1)/sin(op1);
                   if minus=true then op1:=-op1;
                   end;
            end;
      varVal[i1]:=op1;
    end
  else
   begin
//////////////////////////////
sr:=NextMegaWord(sp);
 if StandId(sr)<>0 then
```

```
begin
  i2:=0;
  s1:=sr;
  sr:=NextMegaWord(sp);
        for i:=1 to VarCount do
              begin
                if sr=varName[i] then i2:=i;
              end;
               if i2=0 then
               begin
                 sr:=PointToPoint(sr);
                 op1:=StrToFloat(sr);
               end
             else  op1:=VarVal[i2];
               if StandId(s1)=1 then op1:=sin(op1);
               if StandId(s1)=2 then op1:=cos(op1);
               if StandId(s1)=3 then op1:=sin(op1)/cos(op1);
               if StandId(s1)=4 then op1:=cos(op1)/sin(op1);
               if minus=true then op1:=-op1;
 end
 else
  begin
    b1:=false;
    for i:=1 to VarCount do
      begin
      if sr=varName[i] then
         begin
         op1:=varVal[i];
         if minus=true then op1:=-op1;
         b1:=true;
```

```
            end;
          end;
        if b1=false then
        begin
        sr:=PointToPoint(sr);
        op1:=StrToFloat(sr);
        if minus=true then op1:=-op1;
        end;
      end;
/////////////

sr:=NextMegaWord(sp);
 if StandId(sr)<>0 then
   begin
      i3:=0;
      s1:=sr;
      sr:=NextMegaWord(sp);
      for i:=1 to VarCount do
         begin
         if sr=varName[i] then i3:=i;
         end;
        if i3=0 then
           begin
           sr:=PointToPoint(sr);
           op2:=StrToFloat(sr)
           end
            else
            op2:=VarVal[i3];
          if StandId(s1)=1 then op1:=sin(op2);
          if StandId(s1)=2 then op1:=cos(op2);
```

```
            if StandId(s1)=3 then op1:=sin(op2)/cos(op2);
            if StandId(s1)=4 then op1:=cos(op2)/sin(op2);
        end
        else
         begin
           b1:=false;
           for i:=1 to VarCount do
             begin
              if sr=varName[i] then
                begin
                op2:=varVal[i];
                b1:=true;
                end;
             end;
            if b1=false then
            begin
            sr:=PointToPoint(sr);
            op2:=StrToFloat(sr);
            end;
          end;

      if oc=1 then varVal[i1]:=op1+op2;
      if oc=2 then varVal[i1]:=op1-op2;
      if oc=3 then varVal[i1]:=op1*op2;
      if oc=4 then varVal[i1]:=op1/op2;
      end;
    end;
```

{Функція, яка розпізнає стандартні ідентифікатори}

```
function TEditForm.StandId(S:string):integer;
  begin
```

```
      StandId:=0;
      if s='SIN' then StandId:=1;
      if s='COS' then StandId:=2;
      if s='TG' then StandId:=3;
      if s='CTG' then StandId:=4;
      if s='алгоритм' then StandId:=5;
      if s='змінні' then StandId:=6;
      if s='дійсні' then StandId:=7;
      if s='початок' then StandId:=8;
      if s='кінець' then StandId:=9;
   end;
```

{Процедура що показує поточні значення змінних у основному вікні}

```
procedure TeditForm.ShowVar;
    var
    i:integer;
    begin
    FrameForm.Memo3.Lines.Clear;
       FrameForm.Memo3.Lines.Add('Змінні:');
       for i:=1 to varCount do
         begin
         FrameForm.Memo3.Lines.Add(varName[i]+'='+FloatToStr(varVal[i]));
         end;

    end;
```

{Процедура, що запускає на виконання попередню функцію}

```
procedure TEditForm.Variables1Click(Sender: TObject);
begin
  ShowVar;
end;
```

{Функція, що перетворює дійсні числа до вигляду, який зрозумілий у серидовищі Delphi}

```
function TeditForm.PointToPoint(S:string): string;
   var i:integer;
       s1:string;
   begin
      s1:=s;
      for i:=1 to StrLen(PChar(s1)) do
      begin
      if s1[i]='.' then s1[i]:=',';
      end;
    PointToPoint:=s1;
   end;
end.
```

Допомога для користувачів програми Алгебра алгоритмів.

Програма алгебра алгоритмів створена для практичного використання на комп'ютері теорії Алгебри алгоритмів Овсяка. Упрограмі реалізовано лише підмножину операцій вказаної теорії, а також для зручності користувача введено деякі допоміжні функції.

Нижче приведвно опис синтаксису алгоритму для використання у програмі :

```
<алгоритм>::=алгоритм змінні <список_змінних>початок<тіло_алгоритму>кінець
<список_змінних>::=дійсні <змінні>
<змінні>::=<змінна>[,<змінна>];
<тіло_алгоритму>::=<оператор>|<оператор>[,<оператор>]
<оператор>::=<оп_вводу>|<оп_виводу>|<оп_присвоєнння>|<оп_циклу>|<оп_умови>
<оп_вводу>::=ввести <змінна>|[,<змінна>];
<оп_виводу>::=вивести <змінна>|[,<змінна>];
<оп_присвоєння>::=<формула>;
<оп_циклу>::=|- <формула>,<формула>,<умова> -|;
<оп_умови>::=$ <змінна>|<ціле>  <формула>[,<формула>];
<формула>::= <змінна>=<оп_1>|<змінна>=<оп_1><операція><оп_2>;
<операція>::=+ | - | * | /
<умова>::=<оп_1> <зн_нерівності> <оп_2>
```

З даного опису випливає що структура алгоритму написаного для виконання у програмі виглядає наступним чином:

```
алгоритм
змінні
```

```
дійсні a,b,c,d;
початок
кінець
```

Попередній фрагмент є прикладом коректного опису структури алгоритму, але ні до яких дії не призведе. Щоб алгоритм виконував певні дії, потрібно дописати рядки з операторами. Наприклад:

```
алгоритм
змінні
дійсні a,b,c,d;
початок
ввести a,b,c,d;
вивести a,b,c,d;
кінець
```

Далі наведено приклад алгоритму з використанням всіх можливих опереторів:

```
алгоритм
змінні
дійсні  a,b,c,d,r,g,h,t,q,w,a1,b1,b2,b3,b4,b5;
дійсні  aaa,bbbb,cccc;
початок
ввести a,b,c,d;
r=1.4;
r=-sin_a+cos_b;
вивести r;
вивести &Перевірка_процедури_Dolt,r;
|- r=a, r=b, a > b -|;
вивести &Перевірка_оператора_|-.....-|,r;
r=1;
$  d  r=r*a , a=a+1;
вивести &Перевірка_оператора_$_x.......,r;
кінець
```

Зауваження до даного фрагменту:

1. **Рядок зприсвоєнням, і взагалі присвоєння які використовуються у інших операторах, не повинен містити пробілів.**
2. **Рядок умови в операторі перевірки умови пишеться через пробіли, тобто спочатку перше значення потім пробіл, потім знак нерівності , знов пробіл і друге значення.**
3. **При використанні тригонометричних функцій параметр цих функцій пишеться через знак "_".**
4. **Дійсні числа пишуться з використвнням крапки а не коми, оскільки кома в програмі використовується як роздільник.**
5. **При виведені на друк рядка який починається з & виводиться сам рядок а не значення змінної.**

Нижче приведені роздруки процедур написаних у середовищі Delphi 4 для реалізації програми Алгебра алгоритмів:

(Зауваження повний роздрук лістінгів програми займає набагато більше місця і тому тут не приводиться)

```
{Процедура перевірки наявності всіх розділів алгоритму}
```

```
procedure TEditForm.Testing1Click(Sender: TObject);
var
i,poch:integer;
g1,g2,g3,g4,gg:boolean;
begin
FrameForm.Memo1.Lines.Clear;
FrameForm.Memo2.Lines.Clear;
FrameForm.Memo3.Lines.Clear;
FrameForm.Memo2.Lines.Add('Результати:');
FrameForm.Memo1.Lines.Add('Помилки і підказки:');
FrameForm.Memo3.Lines.Add('Змінні:');
g1:=false;
g2:=false;
g3:=false;
g4:=false;
for i:=0 to editor.Lines.Count do
begin
if editor.Lines.Strings[i]='алгоритм' then g1:=true;
if editor.Lines.Strings[i]='змінні' then g2:=true;
if editor.Lines.Strings[i]='початок' then
begin
g3:=true;
poch:=i;
end;
if editor.Lines.Strings[i]='кінець' then g4:=true;
end;

if (g1=false) then FrameForm.Memo1.Lines.Add('Відсутній заголовок');
if (g2=false) then FrameForm.Memo1.Lines.Add('Відсутній розділ змінних');
if (g3=false) then FrameForm.Memo1.Lines.Add('Відсутній початок алгоритму');
if (g4=false) then FrameForm.Memo1.Lines.Add('Відсутній кінець алгоритму');
if ((g1=true)and(g2=true)and(g3=true)and(g4=true)) then
FrameForm.Memo1.Lines.Add('Всі розділи алгоритму присутні');

end;
//////////////////////////////////////////
{Процедура яка отримує як параметр кожен рядок алгоритму і в залежності від типу рядка виконує певні дії}
procedure TEditForm.Scaner(s:string;var bll:boolean);
var ps,iv:string;
   i,ii,cv,i11:integer;
   sdo,op1,op2,umova,op3,s1,ps1:string;
   bool,bl,bol:boolean;
begin
sdo:=s;
ps:=NextMegaWord(s);
ps1:=s;
bl:=false;
{Опрацювання радка опису змінних }
if ps='дійсні' then
   begin
    bl:=true;
    ps:=NextMegaWord(s);
     while (ps<>';')and(ps<>'') do
        begin
         if ps<>',' then
          begin
          bol:=true;
            for i:=1 to varCount do
                begin
                  if varName[i]=ps then
                      begin
                        ShowMessage('Помилка.Подвійне використання імені змінної:'+ps);
                        bll:=false;
                        bol:=False;
                        Exit;
                      end;
                end;
          if bol=true then
             begin
              varCount:=varCount+1;
```

```
            varName[varCount]:=ps;
           end;
          end;
       ps:=NextMegaWord(s);
       end;
    end;
{Опрацювання оперетора введення }
if ps='ввести' then
    begin
      bl:=true;
      ps:=NextMegaWord(s);
      while (ps<>';')and(ps<>'') do
        begin
        if ps<>',' then
        begin
        iv:=InputBox('Введення даних','Введіть '+ps,'');
            for i:=1 to varCount do
              begin
               if varName[i]=ps then
                  begin
                  varVal[i]:=StrToFloat(iv);
                  end;
              end;
           end;

        ps:=NextMegaWord(s);
        end;
    end;
{Опрацювання рядка з оператором умови}
 if ps='|-' then
      begin
          bl:=true;
      ps:=NextMegaWord(s);
         op1:=ps;
         ps:=NextMegaWord(s);
         ps:=NextMegaWord(s);
         op2:=ps;
         ps:=NextMegaWord(s);
         ps:=NextMegaWord(s);
         umova:=ps;
         s1:=s;
         op3:=NextMegaWord(s1);
         umova:=umova+' '+op3;
         op3:=NextMegaWord(s1);
         umova:=umova+' '+op3;
          bool:=IfThen(umova);
           if bool=true then
                begin
                DoIt(op1);
                end
                else
                begin
                DoIt(op1);

                end;

      end;
{Опрацювання радка з оперетором циклу }
 if ps='$' then
      begin
      bl:=true;
      ps:=NextMegaWord(s);
      while (ps<>';')and(ps<>'') do
        begin
         i11:=0;
         for i:=1 to VarCount do
             begin
              if varName[i]=ps then
                   begin
                     cv:=Round(varVal[i]);
```

```
                    i11:=i;
                  end;
            end;
         if i11=0 then cv:=StrToInt(ps);
          if cv<1 then
              begin
               ShowMessage('Не може бути виконана процедура циклу через невірний показник кількості повторень
циклу');
               bll:=false;
               Exit;
              end;
          sdo:=s;
           for i:=1 to cv do
           begin
          s:=sdo;
          ps:=NextMegaWord(s);
           while (ps<>';')and(ps<>'') do
            begin
            if ps<>',' then
               begin
                 DoIt(ps);
               end;
              ps:=NextMegaWord(s);
           end;

          end;
        end;

      end;
{Опрацювання радка з оперетором виведення }
if ps='вивести' then
    begin
         bl:=true;
     ps:=NextMegaWord(s);
      while (ps<>';')and(ps<>'') do
        begin
          if ps<>',' then
            begin
              if ps[1]='&' then
                begin
                op1:=ps;
                op1[1]:=' ';
                FrameForm.Memo2.Lines.Add(op1);
                end
              else
              begin
              for i:=1 to varCount do
                begin
                if varName[i]=ps then
                     begin
                     FrameForm.Memo2.Lines.Add(FloatToStr(varVal[i]));
                     end;
                end;
              end;
            end;
        ps:=NextMegaWord(s);
        end;

    end;
{Опрацювання рядка з оперетором присвоєння}
   if (bl=false)and(StandId(ps)=0)and(sdo<>'') then DoIt(sdo);

end;
{Функція яка розбиває рядок на окремі слова і розділювачі}
function TEditform.NextMegaWord(var S: String): String;
 var
   i,Len : Integer;
   TempS1 : String;
   NotDel: Boolean;
begin
```

```
 TempS1:='';
 Len:=0;

 NotDel:=True;
  for i:=1 to Length(S) do
    if (S[i]<>#9) and (S[i]<>' ') and (S[i]<>',') and (S[i]<>';') then begin
      TempS1:=TempS1+UpCase(S[i]);
      Len:=Len+1;
    end else
          if (TempS1<>'') then begin
            Delete(S,1,i-1);
            NotDel:=False;
            Break ;
          end else
                if (S[i]<>#9) and(S[i]<>' ') then begin
                  TempS1:=TempS1+UpCase(S[i]);
                  Delete(S,1,i);
                  NotDel:=False;
                  Break ;
                end;
   if NotDel then Delete(S,1,length(s));
 NextMegaWord:=TempS1;
end;


{Функція яка опрацьовує рядок умови}
function TEditForm.IfThen(s:string):boolean;
var ps,s1,s2:string;
    s1v,s2v:real;
    i:integer;
    b1,b2:boolean;
begin
  s1:=NextMegaWord(s);
  ps:=NextMegaWord(s);
  s2:=NextMegaWord(s);
 b1:=false;
  b2:=false;
  for i:=1 to varCount do
      begin
        if s1=varName[i] then
        begin
        s1v:=varVal[i];
         b1:=true;
        end;
        if s2=varName[i] then
        begin
        s2v:=varVal[i];
         b2:=true;
        end;
      end;
    if b1=false then  s1v:=StrToFloat(s1);
    if b2=false then  s2v:=StrToFloat(s2);
   if ps='<' then
      begin
      if s1v<s2v then IfThen:=true
          else IfThen:=false;
      end;
   if ps='>' then
      begin
      if s1v>s2v then IfThen:=true
          else IfThen:=false;
      end;
    if ps='=' then
      begin
      if s1v=s2v then IfThen:=true
          else IfThen:=false;
      end;
    if ps='<=' then
      begin
      if s1v<=s2v then IfThen:=true
```

```
        else IfThen:=false;
      end;
    if ps='>=' then
      begin
      if s1v>=s2v then IfThen:=true
        else IfThen:=false;
      end;
end;
{Процедура яка ініціалізує виконання алгоритму}
procedure TEditForm.Run2Click(Sender: TObject);
var
i,poch:integer;
g1,g2,g3,g4,b:boolean;
begin
FrameForm.Memo1.Lines.Clear;
FrameForm.Memo2.Lines.Clear;
FrameForm.Memo3.Lines.Clear;
FrameForm.Memo2.Lines.Add('Результати:');
FrameForm.Memo3.Lines.Add('Змінні:');
FrameForm.Memo1.Lines.Add('Помилки і підказки:');

varCount:=0;
//////////////////////////////////
g1:=false;
g2:=false;
g3:=false;
g4:=false;
for i:=0 to editor.Lines.Count do
begin
if editor.Lines.Strings[i]='алгоритм' then g1:=true;
if editor.Lines.Strings[i]='змінні' then g2:=true;
if editor.Lines.Strings[i]='початок' then
begin
g3:=true;
poch:=i;
end;
if editor.Lines.Strings[i]='кінець' then g4:=true;
end;
if ((g1=false)or(g2=false)or(g3=false)or(g4=false)) then
   begin
    ShowMessage('Програма не може бути виконана.');
    Exit;
   end;

//////////////////////////////////
for i:=0 to editor.Lines.Count do
        begin
         b:=true;
        FrameForm.Label1.Font.color:=RGB(255,0,0);
        FrameForm.Label1.Caption:='Виконано до рядка номер '+IntToStr(i);
        Scaner(editor.Lines.Strings[i],b);
         if b=false then Exit;
        end;
    FrameForm.Label1.Font.color:=RGB(0,150,0);
   FrameForm.Label1.Caption:=' Роботу успішно завершено.';
end;
{Функція яка виконує присвоєння і обрахунок виразів}
function TEditForm.DoIt(s:string):boolean;
var sp,sr,spr,s1:string;
   i,oc,i1,i2,i3:integer;
   op1,op2:real;
   b1,b2,minus:boolean;
      begin
       sp:=s;
       spr:=s;
       oc:=0;
       minus:=false;
       for i:=1 to strlen(PChar(sp)) do
         begin
           if sp[i]='=' then
```

```
      begin
      sp[i]:=' ';
        if sp[i+1]='-' then
           begin
           minus:=true;
           sp[i+1]:=' ';
           end;
     end;
    if sp[i]='_' then sp[i]:=' ';
    if sp[i]='+' then
        begin
          sp[i]:=' ';
          oc:=1;
        end;
    if sp[i]='-' then
        begin
          sp[i]:=' ';
          oc:=2;
        end;
     if sp[i]='*' then
        begin
          sp[i]:=' ';
          oc:=3;
        end;
     if sp[i]='/' then
        begin
          sp[i]:=' ';
          oc:=4;
        end;
   end;

sr:=NextMegaWord(sp);
for i:=1 to VarCount do
begin
if sr=varName[i] then i1:=i;
end;
if oc=0 then
   begin
    sr:=NextMegaWord(sp);
    b1:=false;
     for i:=1 to VarCount do
         begin
         if sr=varName[i] then
            begin
            op1:=varVal[i];
            if minus=true then op1:=-op1;
            b1:=true;
            end;
         end;
    if b1=false then
          begin

          if StandId(sr)=0 then
               begin
               sr:=PointToPoint(sr);
               op1:=StrToFloat(sr);
               if minus=true then op1:=-op1;
               end
               else
                  begin
                  i2:=0;
                   s1:=sr;
                   sr:=NextMegaWord(sp);
                   for i:=1 to VarCount do
                       begin
                        if sr=varName[i] then i2:=i;
                       end;
                  if i2=0 then
                  begin
                  sr:=PointToPoint(sr);
```

```
                op1:=StrToFloat(sr);
                end
                else op1:=VarVal[i2];
                if StandId(s1)=1 then op1:=sin(op1);
                if StandId(s1)=2 then op1:=cos(op1);
                if StandId(s1)=3 then op1:=sin(op1)/cos(op1);
                if StandId(s1)=4 then op1:=cos(op1)/sin(op1);
                if minus=true then op1:=-op1;
                end;
        end;
    varVal[i1]:=op1;
   end
 else
  begin
///////////////////////////
sr:=NextMegaWord(sp);
 if StandId(sr)<>0 then
   begin
     i2:=0;
     s1:=sr;
     sr:=NextMegaWord(sp);
          for i:=1 to VarCount do
                begin
                  if sr=varName[i] then i2:=i;
                end;
                 if i2=0 then
                 begin
                   sr:=PointToPoint(sr);
                   op1:=StrToFloat(sr);
                 end
               else  op1:=VarVal[i2];
                 if StandId(s1)=1 then op1:=sin(op1);
                 if StandId(s1)=2 then op1:=cos(op1);
                 if StandId(s1)=3 then op1:=sin(op1)/cos(op1);
                 if StandId(s1)=4 then op1:=cos(op1)/sin(op1);
                 if minus=true then op1:=-op1;
    end
    else
     begin
       b1:=false;
       for i:=1 to VarCount do
         begin
         if sr=varName[i] then
           begin
           op1:=varVal[i];
           if minus=true then op1:=-op1;
           b1:=true;
           end;
         end;
        if b1=false then
        begin
        sr:=PointToPoint(sr);
        op1:=StrToFloat(sr);
        if minus=true then op1:=-op1;
        end;
     end;
//////////////

sr:=NextMegaWord(sp);
 if StandId(sr)<>0 then
   begin
     i3:=0;
     s1:=sr;
     sr:=NextMegaWord(sp);
     for i:=1 to VarCount do
        begin
        if sr=varName[i] then i3:=i;
        end;
       if i3=0 then
          begin
```

```
                sr:=PointToPoint(sr);
                op2:=StrToFloat(sr)
                end
                 else
                 op2:=VarVal[i3];
               if StandId(s1)=1 then op1:=sin(op2);
               if StandId(s1)=2 then op1:=cos(op2);
               if StandId(s1)=3 then op1:=sin(op2)/cos(op2);
               if StandId(s1)=4 then op1:=cos(op2)/sin(op2);
            end
            else
             begin
               b1:=false;
               for i:=1 to VarCount do
                 begin
                  if sr=varName[i] then
                    begin
                    op2:=varVal[i];
                    b1:=true;
                    end;
                 end;
                if b1=false then
                begin
                sr:=PointToPoint(sr);
                op2:=StrToFloat(sr);
                end;
             end;

        if oc=1 then varVal[i1]:=op1+op2;
        if oc=2 then varVal[i1]:=op1-op2;
        if oc=3 then varVal[i1]:=op1*op2;
        if oc=4 then varVal[i1]:=op1/op2;
        end;
      end;
{Функція яка розпізнає стандартні ідентифікатори}
function TEditForm.StandId(S:string):integer;
    begin
       StandId:=0;
       if s='SIN' then StandId:=1;
       if s='COS' then StandId:=2;
       if s='TG' then StandId:=3;
       if s='CTG' then StandId:=4;
       if s='алгоритм' then StandId:=5;
       if s='змінні' then StandId:=6;
       if s='дійсні' then StandId:=7;
       if s='початок' then StandId:=8;
       if s='кінець' then StandId:=9;
    end;
{Процедура що показує поточні значення змінних у основному вікні}
procedure TeditForm.ShowVar;
     var
     i:integer;
     begin
     FrameForm.Memo3.Lines.Clear;
        FrameForm.Memo3.Lines.Add('Змінні:');
        for i:=1 to varCount do
          begin
          FrameForm.Memo3.Lines.Add(varName[i]+'='+FloatToStr(varVal[i]));
          end;

     end;
{Процедура що запускає на виконання попередню функцію}
procedure TEditForm.Variables1Click(Sender: TObject);
begin
   ShowVar;
end;
{Функція що перетворює дійсні числа до вигляду який зрозумілий у серидовищі Delphi}
function TeditForm.PointToPoint(S:string): string;
    var i:integer;
        s1:string;
```

```
  begin
     s1:=s;
     for i:=1 to StrLen(PChar(s1)) do
     begin
     if s1[i]='.' then s1[i]:=',';
     end;
   PointToPoint:=s1;
  end;

end.
```

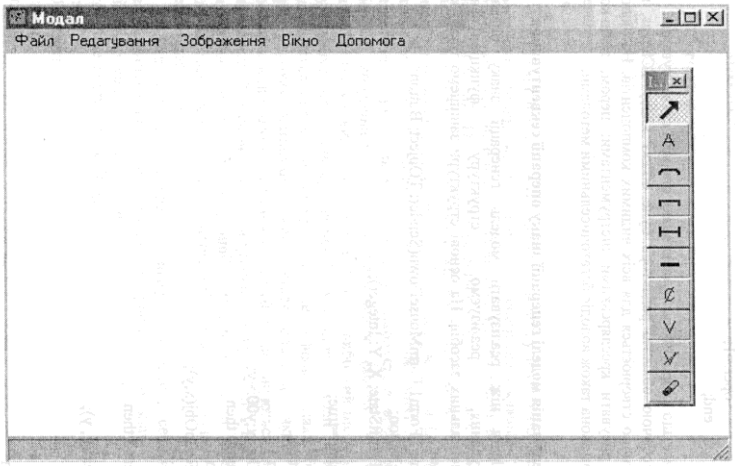


Рис. 2. Інтерфейс користувача редактора формул алгоритмів

Додаток 2

Акти використання результатів дисертаційної роботи

"ЗАТВЕРДЖУЮ"
Проректор з наукової роботи
Української академії друкарства
_____ Мак В.З.
"_____" _____ 2013 р.

Акт використання результатів дисертаційної роботи
Огірко Ольги Ігорівної
у держбюджетній темі
"Модель і програма комп'ютерної системи генерування програм з формул алгоритмів"
з номером державної реєстрації 0109U001214

В Українській академії друкарства з січня 2009 року до грудня 2011 року виконувались фундаментальні дослідження у рамках держбюджетної теми "Модель і програма комп'ютерної системи генерування програм з формул алгоритмів" з номером державної реєстрації 0109U001214.

У рамках цієї держбюджетної теми з фундаментальних досліджень Огірко О.І. розроблений програмний код підсистем опрацювання операції елімінування.

Керівник теми, проф., д.т.н., проф.. кафедри АКТ
Української академії друкарства В.К.Овсяк

Доц., к.т.н., начальник НДЧ, вчений секретар
Вченої ради Української академії друкарства З.М.Сельменська

А К Т

про впровадження та використання комп'ютерної програми Огірко О.І. "Алгебра алгоритмів" та матеріалів дисертації

Підтверджуємо, що отримали в користування комп'ютерну програму "Алгебра алгоритмів" викладача математики Української академії друкарства, аспіранта Огірко О.І.

Програма розроблена в середовищі DELPHI і є автоматизованою системою для побудови алгоритмів. Інтерфейс користувача програми включає панель інструментів конструювання алгоритмів на формалізованому рівні.

Програма Огірко О.І. "Алгебра алгоритмів" розроблена з використанням сучасних прогресивних інформаційних технологій.

Передбачається використання програми для наукових розрахунків та досліджень.

Заступник декана географічного факультету, Львівського національного університету імені Івана Франка, кандидат філософських наук М.Назарук

Асистент кафедри економічної і соціальної географії, Львівського національного університету імені Івана Франка, кандидат географічних наук В.Грицевич

"ЗАТВЕРДЖУЮ"
Заступник директора з навчальної роботи
Львівської філії Української Академії державного
управління при Президентові України,
к.е.н., доцент

..... О.Синицький

"15" серпня 2000р.

АКТ

Комісія у складі заступника завідувача кафедри "Управління трудовими ресурсами" С.Барщика та начальника відділу інформатизації та технічних засобів навчання філії С.Ромашка склала цей акт про те, що філія Академії отримала та впровадила у навчальний процес комп'ютерну програму "Алгебра алгоритмів" викладача математики Української академії друкарства, аспіранта Огірко.

Програма розроблена на основі мови об'єктно-орієнтованого програмування Delphi і забезпечує моделювання алгоритмів на формалізованому рівні. В програмі використано спільний підхід до виконання обчислень та проведення аналізу. Інтерфейс програми включає панель інструментів для побудови алгоритмів.

Програма "Алгебра алгоритмів" є автоматизованою системою управління конструюванням алгоритмів розробника з використанням прогресивних інформаційних технологій.

Програма використана для навчання бакалаврів у дисциплінах "Інформатика та комп'ютерна техніка", "Математика", "Інформаційний менеджмент".

Заст. завідувача кафедри УТР С. Барщик

Начальник відділу ІТЗН С. Ромашко

ЗАТВЕРДЖУЮ

Директор Фізико–механічного інституту

ім. Г.В. Карпенка НАН України

_____ В. В. Панасюк

"30" _____10_____ 2000 р.

А К Т

впровадження результатів роботи Огірко Ольги Ігорівни на тему "Автоматизація математичних обчислень для розрахунку енергетичних характеристик чутливих елементів інформаційно-вимірювальних систем на основі алгебри алгоритмів"

Ми, комісія у складі: старшого наукового співробітника д. ф.–м. н. Юзевича В. М. та наукового співробітника к. т. н. Підгірняка Я. Є. склали даний акт про впровадження наукових і практичних результатів, одержаних в роботі Огірко О. І., при виконанні бюджетної науково-дослідної теми у ФМІ НАН України РБ–26/182 "Розробка принципів побудови високочутливих інформаційно-вимірювальних систем експрес-контролю провідних середовищ" (№ держреєстрації 0198U004615).

Результати роботи реалізовані як програмне забезпечення, що впроваджено в інформаційно-вимірювальних системах контролю морського середовища для попередньої обробки вимірювальної інформації. Розроблений пакет програм призначений для підвищення завадостійкості сигналів.

Впровадження програмного забезпечення, розробленого Огірко О. І., дозволяє скоротити надлишковість даних, що передаються по інформаційних каналах.

Частково результати впровадження опубліковано в науковій праці Сопрунюка П. М., Юзевича В. М., Огірко О. І., Лугового П. В. Автоматизація математичних обчислень для оцінки параметрів поверхневих шарів//Відбір і обробка інформації. – 2000. – № 14 (90). – С. 151–155.

Економічний ефект від впровадження алгоритмів і програм не визначався.

Ст. н. с., д. ф.-м. н. Юзевич В. М.

Н. с., к. т. н. Підгірняк Я. Є.

Погоджено
Проректор з наукової роботи
УАД
доц. доц. канд. техн. наук
Маїк Володимир Зіновійович

11 лютого 2013 року

Затверджую
Директор ТЗОВ НВП "ГАРАС І Л"
Янишевський Любомир Зіновійович

11 лютого 2013 року

Акт

Впровадження пропозицій та розробок кандидатської дисертації Огірко О. І. на тему: «**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВИЗНАЧЕННЯ ЕНЕРГЕТИЧНИХ ХАРАКТЕРИСТИК ПОВЕРХНЕВИХ ШАРІВ МЕТАЛІВ З ВИКОРИСТАННЯМ АЛГЕБРИ**»

В дисертації Огірко О. І. розробила інформаційну технологію, методику та алгоритм оцінки корозії металів. Запропонована інформаційна технологія корозіометрії відбору, впорядкування та опрацювання даних про стан об'єктів із сталі, які знаходяться в корозійному середовищі Запропоновані методи, моделі та інформаційна технологія є теоретичною базою практичних аспектів аналізу корозійних процесів для оцінювання станів конструкційних матеріалів, які знаходяться в агресивному середовищі. Отримані результати дозволяють підвищити достовірність рішень, що приймаються для забезпечення умов протикорозійного захисту металевих конструкцій. Розроблені у дисертаційній роботі методи та моделі були використані в програмно-технічному комплексі відбору та опрацювання даних обстежень корозійного стану, що має відношення до інформаційної системи моніторингу корозійного стану. Розроблена інформаційна технологія протикорозійного захисту дає можливість знизити витрати на моніторинг корозійного стану за рахунок раціонального планування експерименту.

Практично використані публікації:

1. Сопрунюк П.М., Юзевич В.М., Огірко О.І., Луговий П.В. Автоматизація математичних обчислень для оцінки параметрів поверхневих шарів // Відбір і обробка інформації. 2000.-Вип.14(90) с.151-156.

2. Сопрунюк П., Юзевич В., Огірко О. Оцінка поверхневої енергії сталей у сірководневих середовищах // Фіз.-хім. механіка матеріалів. Проблеми корозії та протикорозійного захисту матеріалів. – 2000. – Т.2, випуск № 1. – С. 726-730

3. Огірко О. І. Інформаційна технологія оцінювання станів обєктів із сталі в корозійному середовищі із сірководнем на основі термодинамічної моделі / В. М. Юзевич, О. І. Огірко // Матеріали III Всеукраїнської науково-практичної конференції. Сучасні інформаційні технології в економіці, менеджменті та освіті. 2012 р., Львів.

4.Інтернет сайти О. І. Огірко стосовно інформаційні технології і захисту металів від корозії.

http://www.youtube.com/watch?v=KrCrcNRCeSY&feature=youtu.be О. І. Огірко. Корозіометрія

http://ohirko.blog.net.ua/ О. І. Огірко. Корозіометрія

http://ohirko2008.narod.ru/algebra.html О. І. Огірко. Алгебра алгоритмів та інформаційні технології

http://www.youtube.com/watch?v=1H3JiAO1PWE&feature=plcp О. І. Огірко. Корозіометрія

У перспективі мож на прогнозувати різні аспекти корозійної поведінки металевих платформ та резервуарів із сталі, підданих дії механічного навантаж ення.

Від підприємства
Головний технічний керівник
ТЗОВ НВП "ГАРАС І Л"
Чорний Мирослав Андрійович

(О. І. Огірко)

11.02.2013 р.

11 лютого 2013 року