

МОДЕЛЬ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЗАДАННЯ ПАРАМЕТРІВ ОПЕРАЦІЇ СЕКВЕНТУВАННЯ

Описана декопозиція інформаційної технології на функційну і графічну складові. Вони декомпозовані на функційні унітерми вибору і задання орієнтації та розділювача унітермів операції секвентування.

Described decomposition information technology for functionality and graphic elements. They decomposition uniterms functionality to select and task orientation and separator uniterms sequention operation.

1. ВСТУП І ФОРМУЛЮВАННЯ ЗАДАЧІ

Алгебри алгоритмів [1, 2] є засобом математичного опису інформаційних технологій і систем. Операції алгебри алгоритмів мають специфічну символіку. Наприклад, знак операції секвентування є складної графічної конфігурації у вигляді дуги. Знак елімінування утворено з трьох прямолінійних відрізків. Також особливими є знаки операцій паралелення, циклічних секвентування, елімінування і паралелення. З метою автоматизації процесів набору і редагування формул алгоритмів інформаційних технологій і систем створено спеціалізовані інформаційні системи – редактори Модал [3] і Абстрактал [4]. У Модалі необхідним є задання початку і кінця знаку операції. Побудова елементарних формул алгоритмів не викликає особливих труднощів. Коли ж синтезуються формули більшої складності то у більшості випадків приходиться декілька раз перезадавати початок і кінець формули. У системі Абстрактал цей недолік усунено, але у ній відсутні засоби оперативного редагування формул алгоритмів. Наприклад, заміну горизонтальної на вертикальну орієнтацію формули, наявність спеціального формату збереження формул алгоритмів у пам'яті комп'ютера, наявність збереження координат знаків операцій. З метою усунення цих недоліків у роботі створено нову інформаційну технологію задання орієнтації і властивості комутативності та не комутативності операції секвентування.

2. ДЕКОМПОЗИЦІЯ НА ПІДСИСТЕМИ

Щоб виконувати вибір орієнтації і розділювача унітермів операцій алгебри алгоритмів необхідно ввести підсистему графічної візуалізації

⁸ Львівська філія Київського національного університету культури і мистецтв

з відповідними засобами, а також підсистеми опису і опрацювання вибраних властивостей.

Заголовок підсистеми графічної візуалізації утворюємо публічним методом доступу, з властивістю часткової підсистеми і наслідуванням стандартної підсистеми Window [5, 6], що запишемо як $pu \text{ \%par Te.@Sf:@Win}$, де pu – ідентифікатор публічного методу доступу до підсистеми, $\%$ - ознака графічної підсистеми, par – ідентифікатор частини підсистеми, Te – назва системи, $@$ - ідентифікатор підсистеми, Sf – назва підсистеми, $:$ - ознака наслідування, Win - назва стандартної підсистеми.

Нехай заголовок підсистеми яка описує можливості орієнтації і розділювача унітермів має публічний метод доступу, назву S і наслідує підсистему T , що опишемо так: $pu \text{ Te.@S:@T}$.

Підсистема опрацювання має бути доступною із інших підсистем, тому для неї задаємо публічний метод доступу і нехай вона утворює функційну частину підсистеми графічної візуалізації: $pu \text{ par Te.@Sf:@Win}$.

Розташування опису підсистем одна відносно другої немає значення але вони є обов'язковими складовими, що опишемо так:

$pu \text{ \%par Te.@Sf:@Win, pu \text{ Te.@S:@T, pu \text{ par Te.@Sf:@Win}}$

3. ДЕКОМПОЗИЦІЯ ПІДСИСТЕМ

Кожну із трьох підсистем декомпузуємо на функційні унітерми і властивості.

3.1. Задання складових підсистеми з розділювачом та орієнтацією

Операції алгебри алгоритмів мають орієнтацію (горизонтальну і вертикальну) та розділювач (кома і крапка з комою) унітермів. Орієнтація і розділювач виконують відмінні функції. Тому доцільно виділити окремо опис орієнтації та опис розділювачів. Інформація про орієнтацію буде використовуватись різними частинами системи, що визначає публічний метод доступу до змінної орієнтації Ori . Задаємо залежність до стандартного системного типу $enum$ [5, 6], що запишемо як

$pu \text{ Ori} \in @Enum = \overbrace{Hor, Ver}$

де Hor - ідентифікатор горизонтальної, а Ver – вертикальної орієнтації. Аналогічно описується і змінна - розділювач

$pu \text{ Sep} \in @Enum = \overbrace{Sem, Com}$,

де Sem - ідентифікатор крапки з комою, а Com - коми.

3.2. Модель графічної підсистеми задання орієнтації і розділювача

Перш за все опишемо параметри форми графічного вікна

$$\left(\begin{array}{l} @Win= p1; \\ p2; \\ Tit=Секв; \\ Hei=200; \\ Wid=300; \\ Bg =LiGray; \\ WSL=CS; \\ @Gri, \end{array} \right.$$

де p1 – унітерм підключення до використання засобів платформи, p2 – унітерм задання доступу до стандартних засобів, Tit=Секв – задання заголовку, Hei=200 – опис ширини форми, Wid=300 – довжина форми, Bg =LiGray - колір фону і WSL=CS - розташування у центрі екрана, @Gri - опис сітки.

Графічний елемент сітка описується формулою

$$@Gri= \left(\begin{array}{l} wid=250; \\ lab1 \in @Lab; \\ lab2 \in @Lab; \\ sep \in @ComB; \\ ori \in @ComB; \\ вла @GroB, \end{array} \right.$$

де wid=250 - довжина сітки, lab1 ∈ @Lab – створення елемента стандартного класу Label, lab2, sep, ori і вла - елементи стандартних класів Label, ComboBox і GroupBox, які мають такий опис:

$$lab1 \in @Lab = \left(\begin{array}{l} Hei=23; \\ Va=Top; \\ \left(\begin{array}{l} Орієнтація; \\ Ma=36, 22, 133, 0, \end{array} \right) \end{array} \right.$$

$$lab2 \in @Lab = \left(\begin{array}{l} Wid=95; \\ Ha=Lef; \\ \left(\begin{array}{l} Орієнтація; \\ Ma=36, 57, 0, 50, \end{array} \right) \end{array} \right.$$

$$sep \in @ComB = Ma \overbrace{120; 57; 27; 53,}$$

$$ori \in @ComB = \left(\begin{array}{l} Hei=23; \\ Va=Top; \\ Ma=120, 57, 27, 0, \end{array} \right.$$

```

вла@GroB = (
    Na=gB;
    Hea=Властивості;
    Ma=10, 10,10, 45,
    can∈@But ≠ Hei=95;
    (
        Cli=canCli;
        (
            Скасувати;
            Ma=36,57,0,50;
        )
    )
    ok∈@But=Hei=23;
    (
        Cli=okCli;
        Виконати;
        Ma=0, 0, 30, 10,
    )
)

```

де But - стандартна підсистема Button , Hei, Va, Top, Ma, Cli - стандартні властивості Height, VerticalAlignment, Top, Margin, Click [].

3.3. Модель підсистеми опрацювання

Підсистема опрацювання вибору орієнтації і розділювача описується такою формулою

```

pu Sf() = (
    IniCom();
    (%ori.SelInd=0;
    (%sep.SelInd=0;
    (%sep.+(("Крапка з комою")&("Кома"));
    (%ori.+(("Горизонтальна")&("Вертикальна"));
)

pu Ori∈@S.Ori = (
    Hor ; Ver ; (%ori.SelInd=0)-?
    ;
    %ori.SelInd = 0;1;(value=∈@S.Ori.Hor)-?
)

pu Sep∈@S.Sep = (
    Sem ; Com ; (%sep.SelInd=0)-?
    ;
    %sep.SelInd = 0;1;(value=∈@S.Sep.Sem)-?
)

pr ok(s∈@obj, e∈@REA) = DraRe=t ; Clo(); k
pr can(s∈@obj, e∈@REA) = DraRe=f ; Clo(); k

```

де Sf() - функції ний унітерм задання початкових значень змінним, IniCom() – стандартний функцій ний унітерм ініціалізації графічних елементів системи, %ori.SelInd=0 – приписування значення стандартній властивості SelectedIndex [5, 6] графічного елемента ori, value – назва стандартної системної змінної для збереження значень, REA – стандартна підсистема RoutedEventArgs Close(), DraRe – логічна змінна, Clo() - стандартний функційний унітерм Close()[5, 6].

4. ФРАГМЕНТ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Мовою C# наведено програмну реалізацію інформаційної технології опрацювання вибору орієнтації та задання розділювача:

```
public partial class SequenceForm : Window {
    public Sequence.Orientation Orientation
    {
        get
        {
            if (orientation.SelectedIndex == 0)
                return Sequence.Orientation.Horizontal;
            else
                return Sequence.Orientation.Vertical;
        }
        set
        {
            if (value == Sequence.Orientation.Horizontal)
                orientation.SelectedIndex = 0;
            else
                orientation.SelectedIndex = 1;
        }
    }
}
public Sequence.Separator Separator
{
    get
    {
        if (separator.SelectedIndex == 0)
            return Sequence.Separator.Semicolon;
        else
            return Sequence.Separator.Comma;
    }
    set
    {
        if (value == Sequence.Separator.Semicolon)
            separator.SelectedIndex = 0;
        else
            separator.SelectedIndex = 1;
    }
}
public SequenceForm()
{
    InitializeComponent();
    orientation.SelectedIndex = 0;
    separator.SelectedIndex = 0;
    separator.Items.Add("Крапка з комою");
    separator.Items.Add("Кома");
    orientation.Items.Add("Горизонтальна");
    orientation.Items.Add("Вертикальна");
}
private void cancel_Click(object sender, RoutedEventArgs e)
```

```

{
    DialogResult = false;
    Close();
    return;
}
private void ok_Click(object sender, RoutedEventArgs e)
{
    DialogResult = true;
    Close();
    return;
}

```

5. ВИСНОВКИ

Виконаною декомпозицією зменшена складність інформаційної технології вибору орієнтації і розділювача унітермів операцій алгебри алгоритмів.

Побудовані математичні моделі описують процеси створення графічного вікна, вибору і опрацювання орієнтації і розділювачів операцій формул алгоритмів.

Програмною реалізацією і апробацією верифіковано побудовані моделі.

1. Ovsyak V.K.: *Computation Models and Algebra of Algorithms*. http://www.nbu.gov.ua/Portal/natural/VNULP/ISM/2008_621/01.pdf 2. Owsiak W., Owsiak A. *Rozszerzenie algebry algorytmów /Pomiary, automatyka, kontrola.* – № 2, 2010. – S. 184 – 188. 3. Бритковський В.М. *Моделювання редактора формул секвенційних алгоритмів: автореф. дис. на здобуття наук. ступеня канд. тех. наук: спец. 01.05.02 “Математичне моделювання та обчислювальні методи” / Бритковський В.М. - Львів, 2003. - 18 с.* 4. Василюк А.С. *Підвищення ефективності математичного і програмного забезпечення редактора формул алгоритмів: автореф. дис. на здобуття наук. ступеня канд. тех. наук: спец. 01.05.02 “Математичне та програмне забезпечення обчислювальних машин і систем” / Василюк А.С. - Львів, 2008. - 20 с.* 5. Petzold C. *Programowanie Windows w języku C#.* –Warszawa: „RM”, 2002. – 1161 s. 6. Мэтью Мак-Дональд. *Windows presentation foundation в .NET 3.5 с примерами на C# 2008.* – Москва, Санкт-Петербург, Киев: “Apress”, 2008. – 922 с.

