# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет комп’ютерних наук та кібернетики Кафедра теорії та технології програмування

## Звіт до проектної роботи №5 на тему: «рекурсивні програми

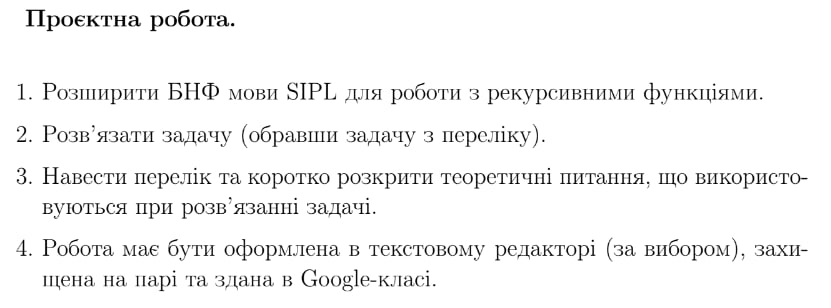
**з дисципліни «Теорія програмування»**

Виконав студент 3-го курсу Групи ТТП-31

Козурак Денис

Київ – 2024

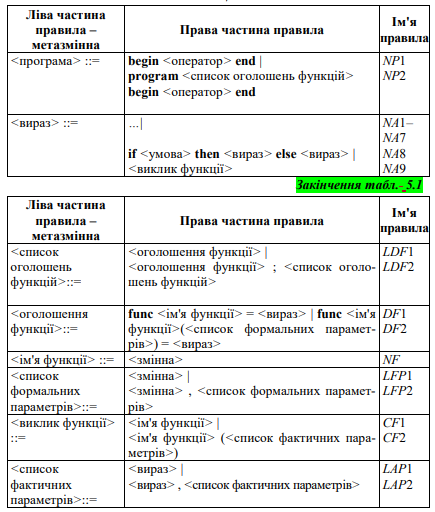
**Завдання проектної роботи:**

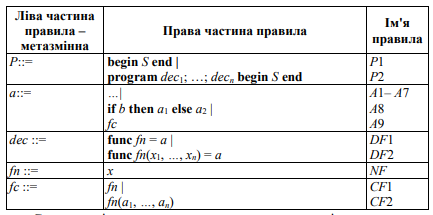


**Мій варіант(номер 9):**



**Розширення БНФ мови SIPL для роботи з рекурсивними функціми:**





**Розв’язання задачі – (2x)!!**

1. **SIPL-програма**: (2x)!! = 2\*4\*6\*…\*(2(x-1)) \* 2x. Добуток всіх множників, окрім останнього, рівний (2(x-1))!!, адже це добуток усіх парних чисел від 2 до 2(x-2).

Отже, (2x)!! = (2(x-1))!! \* 2x, тому рекурсивна функція матиме наступний вигляд:

func f(X) = if X>1 then f(X-1)\*X\*2 else 2

1. **Семантичний терм**: ф(f) = IF(S2(gr,X=>, 1̅), S2(mult,SX(f, S2(sub,X=>,1̅)), S2(mult,X=>, 2̅)), 2̅)

Тут S2(gr,X=>, 1̅) відповідає умові x>1, S2(mult,SX(f, S2(sub,X=>,1̅)), S2(mult,X=>, 2̅)) – добутку

(2(x-1))!! \* 2x(значення виразу в разі істинності умови), 2 – значення виразу в разі хибності умови

1. **Побудова 3 апроксимацій**:

Апроксимації будуємо за наступною схемою: f0 = ⊥, fk=ф(fk-1) для k>=1, тобто для побудови наступної апроксимації ми маємо підставити попередню апроксимацію замість f у семантичний терм.

1. *Перша апроксимація*: f0 = ⊥
2. *Друга апроксимація*: f1 = ф(f0) =

IF(S2(gr,X=>, 1̅), S2(mult,SX(f0, S2(sub,X=>,1̅)), S2(mult,X=>, 2̅)), 2̅) =

IF(S2(gr,X=>, 1̅), S2(mult,SX(⊥, S2(sub,X=>,1̅)), S2(mult,X=>, 2̅)), 2̅) =

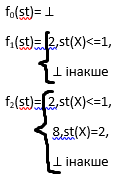
IF(S2(gr,X=>, 1̅), S2(mult,⊥, S2(mult,X=>, 2̅)), 2̅) = IF(S2(gr,X=>, 1̅), ⊥, 2̅)

1. *Третя апроксимація*: f2 = ф(f1) =

IF(S2(gr,X=>, 1̅), S2(mult,SX(f1, S2(sub,X=>,1̅)), S2(mult,X=>, 2̅)), 2̅) =

IF(S2(gr,X=>, 1̅), S2(mult,SX(IF(S2(gr,X=>, 1̅), ⊥, 2̅), S2(sub,X=>,1̅)), S2(mult,X=>, 2̅)), 2̅)

Аналізуючи отримані апроксимації, можна описати отримані функції наступним чином:



1. **Тестування**

Проведемо тестування побудованих апроксимацій на вхідних даних n=2, для цього застосуємо відповідні семантичні терми до стану st=[X ↦ 2]

1. *перша апроксимація*: f0(st) = ⊥(st) ↑

Апроксимація невизначена на вказаних вхідних даних, потрібно розглянути наступну апроксимацію

1. *друга апроксимація*: f1(st) = IF(S2(gr,X=>, 1̅), ⊥, 2̅)

Обчислимо значення умови S2(gr,X=>, 1̅)(st) = gr(X=>(st), 1̅)(st)) = gr(2,1) = true – умова істинна, тому f1(st) = ⊥(st) ↑

Апроксимація невизначена на вказаних вхідних даних, потрібно розглянути наступну апроксимацію

1. *третя апроксимація*: f2(st) =

IF(S2(gr,N=>, 1̅), S2(mult,SX(IF(S2(gr,X=>, 1̅), ⊥, 2̅), S2(sub,X=>,1̅)), S2(mult,X=>, 2̅)), 2̅)(st)

Умова оператора IF співпадає з умовою в попередній апроксимації, тому маємо:

f2(st) = S2(mult,SX(IF(S2(gr,X=>, 1̅), ⊥, 2̅), S2(sub,X=>,1̅)), S2(mult,X=>, 2̅))(st) = mult(SX(IF(S2(gr,X=>, 1̅), ⊥, 2̅), S2(sub,X=>,1̅))(st), S2(mult,X=>, 2̅)(st))

* перший аргумент множення: SX(IF(S2(gr,X=>, 1̅), ⊥, 2̅), S2(sub,X=>,1̅))(st) =

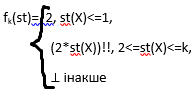
IF(S2(gr,X=>, 1̅), ⊥, 2̅)(st ∇[X↦ S2(sub,X=>,1̅)(st)] = IF(S2(gr,X=>, 1̅), ⊥, 2̅)(st ∇[X↦ sub(2,1)]) = IF(S2(gr,X=>, 1̅), ⊥, 2̅)(st ∇[X↦ 1]). У такому випадку умова рівна S2(gr,X=>, 1̅) = gr(1,1) = false, отже, IF(S2(gr,X=>, 1̅), ⊥, 2̅) = 2

- другий аргумент множення: S2(mult,X=>, 2̅) = mult(2,2)=4

Підставляючи це, отримуємо: f2(st) = mult(2,4) = 8 – апроксимація визначена, результат співпадає з (2\*2)!! – тестування успішне

1. **Доведення правильності**:

За наведеними вище апроксимаціями можна припустити, що а загальному випадку k-та апроксимація буде мати наступний вигляд:



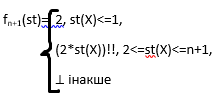
Доведемо це ММІ:

* *База індукції*: для k=1 (2\*1)!!=2, а отже, побудована в пункті 3 апроксимація:

f1(st)= 2,st(X)=1,

⊥ інакше співпадає з припущенням, база виконується

* *Крок індукції*: припустимо, що твердження виконується для k<=n, доведемо для k=n+1. Потрібно показати, що:



Враховуючи неперервність і монотонність оператора ф маємо, що якщо fn(st) = ↓, то fn+1(st) ↓ = fn(st), що доводить твердження для випадку st(X)<=n, лишається довести для випадків st(X)=n+1 I st(X)>n+1

Нехай st=[X ↦ x], обчислимо апроксимацію fn+1 :

fn+1(st) = ф(fn)(st) = IF(S2(gr,X=>, 1̅), S2(mult,SX(fn, S2(sub,X=>,1̅)), S2(mult,X=>, 2̅)), 2̅)

Для випадків st(X)=n+1 I st(X)>n+1 умова S2(gr,X=>, 1̅) істинна, тому fn+1(st) =

S2(mult,SX(fn, S2(sub,X=>,1̅)), S2(mult,X=>, 2̅))(st) =

mult(SX(fn, S2(sub,X=>,1̅))(st), S2(mult,X=>, 2̅)(st)) =

mult(fn(st∇[X↦ S2(sub,X=>,1̅)(st)]), S2(mult,X=>, 2̅)(st)) =

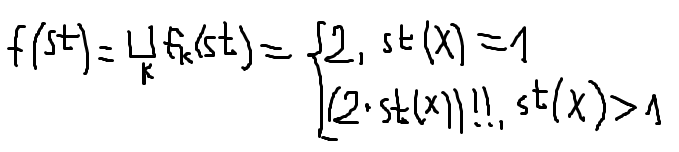
mult(fn(st∇[X↦ sub(X=>(st),1̅(st)]), mult(X=>(st), 2̅(st))) =

mult(fn(st∇[X↦ sub(x,1)(st)]), mult(x,2)) = mult(fn(st∇[X↦ x-1]), 2x)

Якщо st(X) = n+1, то x+1 = n+1-1 = n <= n, апроксимація fn на відповідних даних визначена і mult(fn(st∇[X↦ x-1]), 2x) = mult((2n)!! ,2\*(n+1)) = (2n)!!\*2\*(n+1)=(2\*(n+1))!! – що і треба було довести

Якщо st(X)>n+1, то x+1 > n, апроксимація fn на відповідних даних невизначена, тому невизначеними є множення і результат fn+1 – твердження виконується

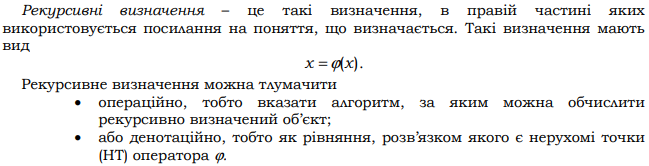
Було розглянуто всі варіанти вхідних даних – твердження доведено. Розглядаючи ланцюг апроксимацій, функція, що задається рекурсивним визначенням, співпадає з границею ланцюга:



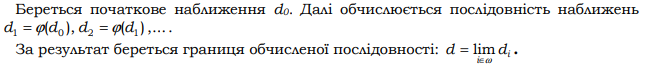
Дана функція співпадає з (2x)!! Для x>0 – правильність рекурсивної програми доведено

**Теоретичні питання, використані при розв’язанні задачі:**

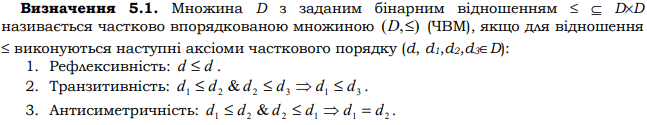
* Означення **рекурсивних визначень**(які було використано для побудови функції, що розв’язує задачу)



* Суть **методу послідовних наближень**, який було використано для побудови апроксимацій:

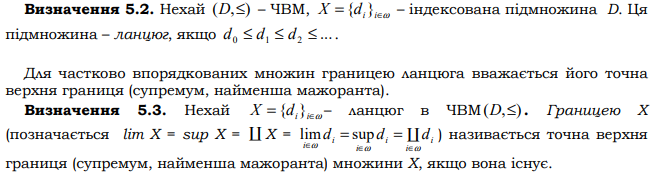


* Означення **ЧВМ**, необхідне для визначення поняття ланцюга і його границі:

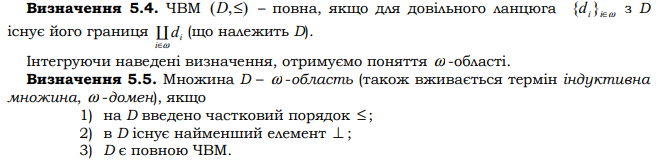


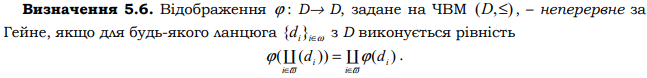
При застосуванні методу послідовних наближень таким відношенням є «наближеність» одного наближення до іншого

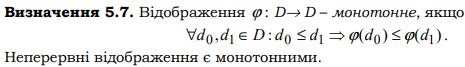
* Означення **ланцюга** і його **границі**, використані в останній частині доведення правильності програми:



* Означення **повної ЧВМ**, **w-області**, **неперервних** і **монотонних відображень**, потрібні для формулювання теореми Кнастера-Тарського-Кліні:





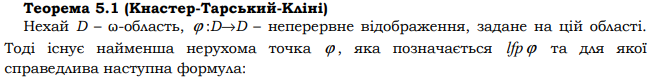


Повнота необхідна для можливості застосування методу послідовних наближень

* Лема, необхідна для доведення теореми Кнастера-Тарського-Кліні



* **Теорема Кнастера-Тарського-Кліні**, що доводить наявність найменшого розв’язку рекурсивного рівняння, який ми розглядаємо для доведення коректності нашої програми:





* Властивості оператора ННТ:





