## Програма ModelComp

1. **Установка програми.**

Для роботи з програмою ModelComp необхідно, щоб на машині була встановлена версії Java не нижче 7.

Потрібно в одному каталозі розмістити наступні файли:

* ModelComp.jar – містить необхідну Java–програму.
* Model.db – база даних, в якій зберігаються створені моделі обчислень.

Запуск програми:

* З командного рядка – ввести в каталозі, де розміщена програма ModelComp.jar, наступну команду
  + *Java –jar ModelComp.jar*
* З провідника (файлового менеджера) – подвійна фіксація (дворазове натискання кнопки мишки) на файлі ModelComp.jar

1. **Призначення програми.**

Головне призначення Java-програма ModelComp - підтримка роботу з наступними моделями обчислення (пункт меню *Модель обчислень*):

* Машини з необмеженими регістрами *(Модель обчислень/Машини з необмеженими регістрами)*
* Машини Тюрінга *(Модель обчислень/Машини Тюринга)*
* Нормальні алгоритми Маркова *(Модель обчислень/Нормальні алгоритми Маркова)*
* Системи Поста *(Модель обчислень/Системи Поста)*
* Рекурсивні функція. *(Модель обчислень/Частково-рекурсивні функції)*
* Лямбда-вирази. *(Модель обчислень/Лямбда-вирази*

Робота з кожним типом моделі ведеться через графічний інтерфейс, котрий включає:

* Опис моделі
* Програму(елементи) моделі.
* Кнопки, що дозволяють вибрати конкретну команду (елемент) програми
  + Надпис k:n вказує номер вибраної команди ( елементу) моделі k і загальну кількість команд (елементів) n в моделі
  + Якщо команда (елемент) ще НЕ вибрана, то надпис буде 0:n.
* Кнопки для роботи з окремою командою (елементом) моделі
* Кнопки, що дозволяють вибрати модель з бази даних. Всі моделі одного виду впорядковані за своїми іменами
  + Надпис l:m вказує номер вибраної l і загальну кількість моделей m даного виду в базі даних.
  + Якщо модель ще НЕ вибрана, то надпис буде 0: m а області, що містять інформацію про модель та її програму – порожні.
* Кнопки для роботи з моделлю в цілому

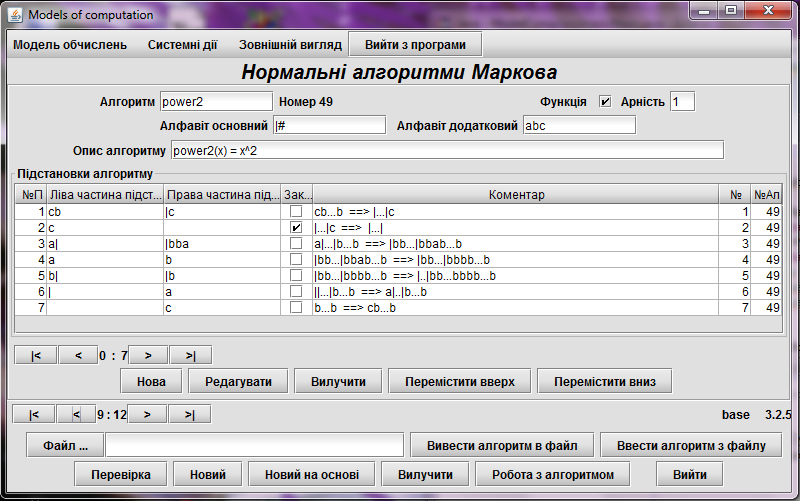
Програма дозволяє:

* Створювати, редагувати і зберігати в БД моделі.
* Тестувати вибрану модель в режимі інтерпретації.
* Працювати з текстовим форматом моделі – введення з текстового файлу / виведення в текстовий файл.

Для кожної моделі обчислень програма дозволяє:

* Переглянути всі моделі, що знаходяться в базі даних (*Вибір моделі*).
* Створити нову модель («Нова»)
  + Вибрати конкретну модель з бази даних і на її основі створити нову модель повністю ідентична вибраній за винятком імені, котре встановлюється за замовчуванням і може бути змінено(«Нова на основі»).
* Ввести модель з текстового файлу («Ввести модель з файлу»).
* Вибрати конкретну модель з бази даних і виконати з нею наступні дії:
  + Вилучити модель («Вилучити»).
  + Вивести модель в окремий текстовий файл («Вивести модель в файл»).
  + Відредагувати модель (*Опис моделі, Програма (елементи ) моделі*).

***Опис моделі***



***Програма (елементи) моделі***

***Вибір команди (елемента) моделі***

***Робота з окремою командою (елементом) моделі***

***Вибір моделі***

***Робота з моделями***

* + Виконати модель в режимі інтерпретації («Робота з моделлю»).

Програма дозволяє виведення / введення декількох моделей (можливо, різного виду) в один текстовий файл / з одного текстового файлу (Пункти меню *Системні дії/Введення моделей з файлу* та *Системні дії/Виведення моделей в файл*).

1. **Робота з машиною з необмеженими регістрами.**

Машина з необмеженими регістрами має програму, котра задає числову функцію з декількома аргументами (арність функції).

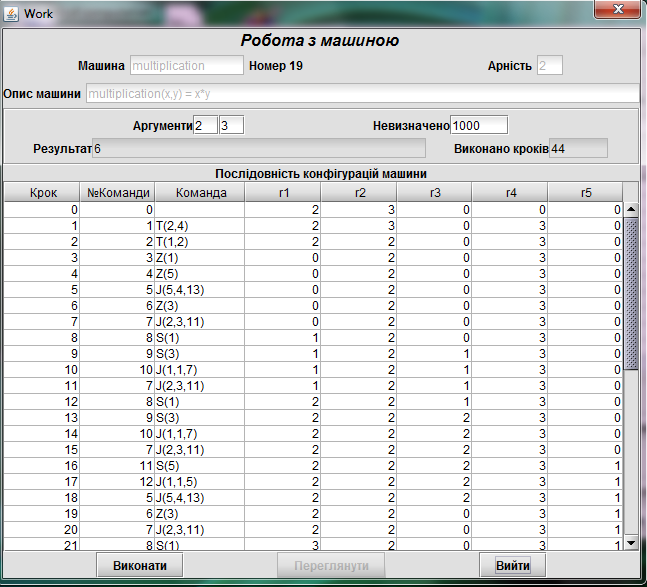
В вибраній машині можна редагувати назву, кількість аргументів (арність), опис машини та працювати з її програмою.

Для модифікації програми доступні наступні дії:

* «Нова» - відкриває форму Command, що містить всі компоненти нової команди, котрі можна редагувати.
  + Всі компоненти нової команди співпадають з компонентами виділеної, а номер на одиницю більше.
  + Дозволяється редагувати номер команди.
* «Редагувати» - відкриває форму Command, що містить всі компоненти виділеної команди, котрі можна редагувати, окрім номера команди.
* «Вилучити» - вилучає виділену команду.
* «Перемістити вверх» / «Перемістити вниз» - міняє виділену команду місцями з сусідньою зверху (номер команди на 1 менше) / знизу (номер команди на 1 більше).
* «Вставити програму» - вставляє всі команди програми за виділеною командою, коригуючи команди переходу обох програм.

Команди переходу програми вибраної машини можуть коригуватися при виконанні наступних дій:

* Вилучити команду.
* Додати нову команду за командою, котра не являється останньою.
* Вставити програму за командою, котра не являється останньою.

Всі інші дії з програмою не змінюють команд переходу. 

Для інтерпретації (виконання) вибраної програми необхідно перейти на форму Work (кнопка «Робота з машиною»).

* Для виконати необхідно ввести аргументи і нажати кнопку «Виконати».
* Виконання програми зупиняється, коли:
  + Необхідно виконати команду, номер якої не входить в програму, по завершенню виводиться результат – зміст регістра 1.
  + Виконано команд (кроків) більше ніж вказано, то в цьому випадку вважається, що результат невизначений і виводиться: "Невизначено"
* Кнопка «Переглянути» виводить послідовність конфігурацій машини при виконанні програми на введених аргументах

1. **Робота з машиною Тюрінга.**

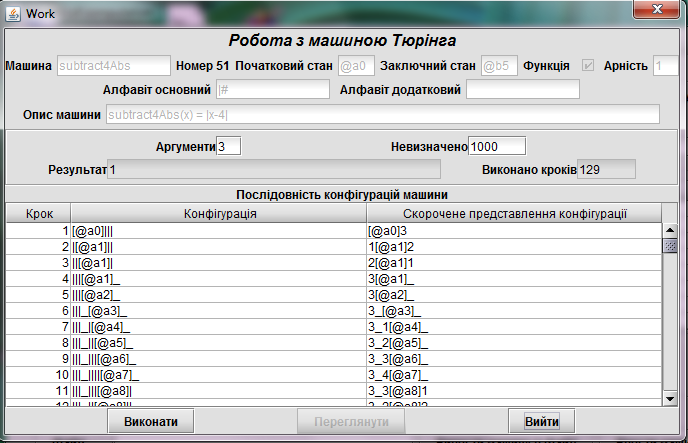
В вибраній машині Тюрінга можна редагувати назву, початковий і заключний стан, вказівку, про те що машина Тюрінга обчислює числову функцію, основний і додатковий алфавіти, опис машини та програму машини (таблицю переходів), кожний елемент (рядок) якої описує поведінку машини Тюрінга в одному з станів.

Зауваження. 1. Більшість елементів графічного інтерфейсу, що описують модель-це текстові поля. Наприклад, для машини Тюрінга це – назва, початковий і заключний стан, основний і додатковий алфавіт, опис машини. Результат редагування тестового поля обробляється програмою (включаючи зміни в базі даних) лише ПІСЛЯ введення символу Enter в текстовому полі. 2. В текстових полях опису моделі та коментаря команди неможна використовувати символ '.

Для модифікації таблиці переходів доступні наступні дії:

* «Перевірка» - перевіряє коректність таблиці переходів машин, видаючи відповідне повідомлення при наявності помилки.
* «Новий» - дозволяє створити новий стан, відкриваючи форму Command, що містить всі компоненти нового стану, котрі можна редагувати.
  + Всі переходи нового стану співпадають з переходами виділеного.
  + Ім»я нового стан, котре НЕ співпадає з жодним іменем стану таблиці переходів, модна редагувати
* «Редагувати» - дозволяє відредагувати переходи виділеного стану, відкриваючи форму Command.
* «Вилучити» - вилучає виділений стан з таблиці переходів.
* «Переіменувати» - змінює назву визначеного стану, переходи якого виділені, на іншу в усіх елементах таблиці переходів.
* «Вставити програму» - вставляє програму іншої машини Тюрінга, перейменовуючи всі іі стани.
  + Імена станів таблиці переходів, котра вставляється, перейменовуються так, щоб вони були більшими за імена всіх станів, переходи яких вже задані.
  + Тобто програма, що вставляється, розміщується в кінці таблиці переходів.

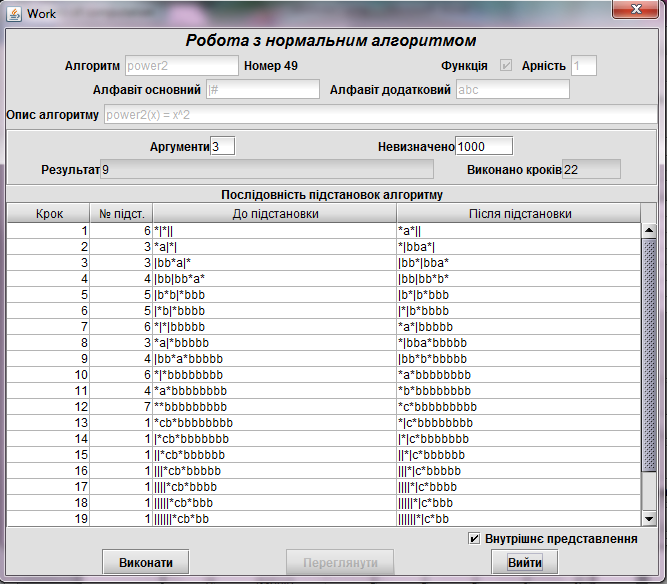
Для інтерпретації (виконання) вибраної машини Тюрінга необхідно перейти на форму Work (кнопка «Робота з машиною»).

* Для виконати необхідно ввести початкове слово або аргументи, якщо машина обчислює числову функцію, і нажати кнопку «Виконати».
* Машина закінчить виконання програми і зупиниться, якщо:
  + Машина попаде в заключний стан і в цьому випадку виводить результат – заключне слово.
  + Машина зустріне стан @tt, котрого немає в таблиці переходів, результат вважєається невизначеним і виводиться повідомлення "Невизначений стан: @tt".
  + Машина попаде в конфігурацію стан @tt i символ S, для якої не описана поведінка машини (в таблиці переходів є стан @tt але для символа S не визначена дія). Результат вважається невизначеним і виводиться повідомлення "Невизначений перехід: @tt+’S’ ".
  + Машина виконає кроків більше ніж вказано і не попале в заключний стан. Вважається, що результат невизначений і виводиться: "Невитзначено (вичерпана кількість переходів)"
* Послідовність конфігурацій машини, що виникла при виконанні програми, можна переглянути, нажавши кнопку «Переглянути».

1. **Робота з нормальним алгоритмом Маркова.**

В вибраному алгоритмі можна редагувати назву, вказівку, про те що алгоритм обчислює числову функцію, основний і додатковий алфавіти, опис алгоритму та працювати зі списком підстановок.

Для роботи зі списком підстановок доступні наступні дії:

* «Перевірка» - перевіряє коректність списку підстановок, видаючи відповідне повідомлення при наявності помилки.
* «Нова» - дозволяє створити нову підстановку, відкриваючи форму Command, що містить всі компоненти нової підстановки, котрі можна редагувати.
  + Компоненти нової підстановки співпадають з компонентами виділеної.
  + Дозволяється редагувати номер підстановки.
* «Редагувати» - дозволяє відредагувати компоненти виділеної підстановки, відкриваючи форму Command
* «Вилучити» - вилучає виділену підстановку.
* «Перемістити вверх» / «Перемістити вниз» - міняє виділену підстановку місцями з сусідньою зверху / знизу.

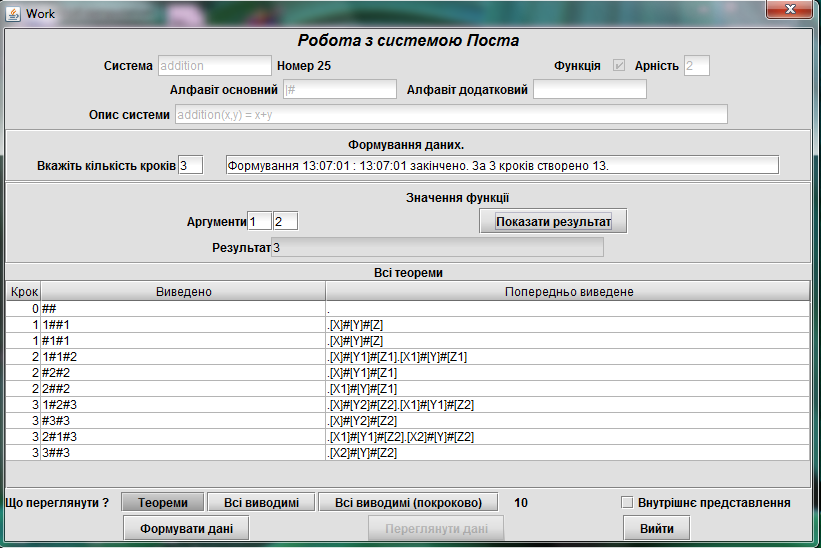
Для інтерпретації (виконання) вибраного алгоритму необхідно перейти на форму Work (кнопка «Робота з алгоритмом»).

* Для виконати необхідно ввести початкове слово або аргументи, якщо алгоритм обчислює числову функцію, і нажати кнопку «Виконати».
* По закінченню виконання виводиться результат.
  + Якщо алгоритм виконав більше підстановок (кроків) ніж вказано і не зупинився, то вважається, що результат невизначений і виводиться: "Невизначено"
* Послідовність підстановок, котрі виконав алгоритм, можна переглянути, нажавши кнопку «Переглянути».

1. **Робота з системою Поста.**

В вибраній системі можна редагувати назву, вказівку, про те що система задає числову функцію, основний і додатковий алфавіти, опис системи, та список аксіом та / правил виводу.

Для модифікації списку аксіом та правил виводу системи доступні наступні дії:

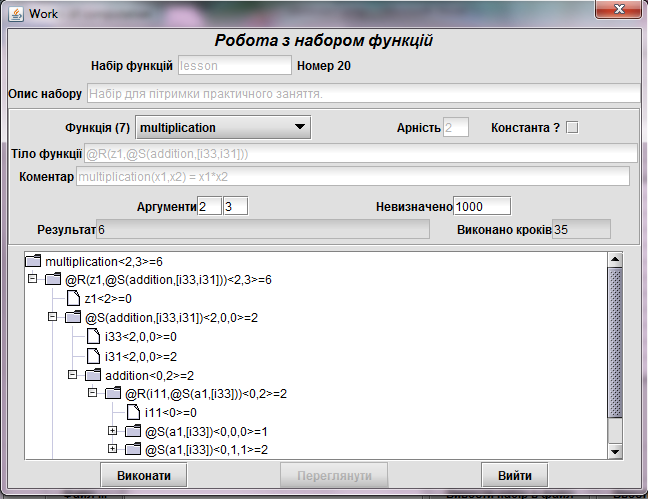
* «Перевірка» - перевіряє коректність списку аксіом та правил виводу, видаючи відповідне повідомлення при наявності помилки.
* «Нове» - відкриває форму Command, що містить всі компоненти нового правила виводу чи аксіоми, котрі можна редагувати
  + Всі компоненти нової правила виводу співпадають з компонентами виділеного, а номер на одиницю більше.
  + Дозволяється редагувати номер аксіоми / правила виводу.
* «Редагувати» - - дозволяє відредагувати компоненти виділеної аксіоми / правила виводу, відкриваючи форму Command
* «Вилучити» - вилучає виділену аксіому / правило виведення.
* «Перемістити вверх» / «Перемістити вниз» - міняє виділену аксіому / правило виводу місцями з сусіднім зверху / знизу.

Для тестування (виконання) вибраної системи необхідно перейти на форму Work (кнопка «Робота з системою»).

* Програма дозволяє побудувати (сформувати) частину слів (теорем), котрі можна вивести за певну кількість кроків, згідно аксіом і правил виводу системи. Для цього необхідно вказати кількість кроків k і нажати кнопку «Формувати дані».
  + Програма формує всі слова які можна вивести з аксіом системи застосувавши не більше ніж k разів правила виводу системи.
* Кнопка «Перегляти дані» надає можливість переглянути всі сформовані дані певної категорії:
  + «Теореми» - показує ВСІ виведені з аксіом теореми.
  + «Всі виводимі» - показує ВСІ виведені з аксіом слова.
  + «Покроково виводимі» - показує виведення з аксіом ВСІХ слів покроково.
  + Якщо деяке слово (теорема) має декілька виводів, то в при останньому перегляді будуть показані ВСІ виводи на відміну від перших двох де це слово (теорема) зустрінеться лише один раз.
* Якщо система задає числову функцію, то можна отримати значення функції для конкретних аргументів (кнопка «Показати результат»).
* Якщо за вказану кількість кроків НЕ сформовано слово, котре описує значення функції на вказаних аргументах, то результата «Невизначено».

1. **Робота з частково-рекурсивними функціями.**

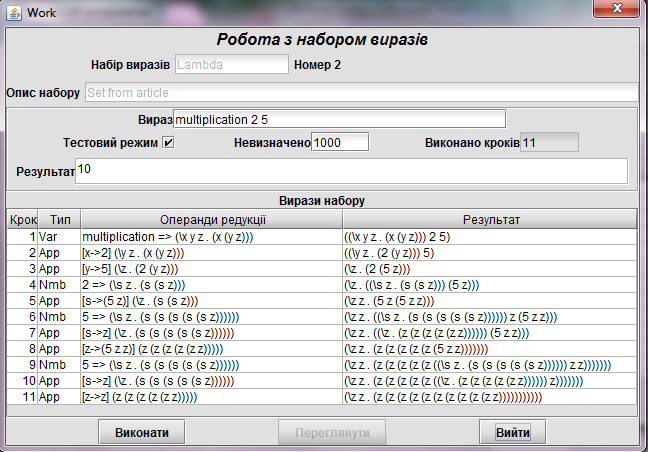
Кожна модель частково-рекурсивних функцій – це набір (множина) примітивно-рекурсивних та частково-рекурсивних функції, визначених користувачем. Якщо в означенні функції f2 використовується функція f1, котра не являється базовою, то ці функції повинні входити в один набір.

 В вибраному наборі можна редагувати назву, опис набору і працювати з усіма функціями набору використовуючи наступні дії:

* «Перевірка» - перевіряє набір функцій на коректність і видає повідомлення про помилки, при їх наявності.
* «Нова» - відкриває форму Command, що містить назву, тіло і коментар нової функції, котрі можна редагувати.
  + Початкові значення тіла та коментаря нової функції співпадають з тілом та коментарем виділеної, а початкове значення імені - name01, якщо ім.»я виділеної функції name.
  + Кнопка «Тестувати» - перевіряє коректність функції введеної в текстовому полі «Тіло функції».
  + Кнопка «Структура» - показує структуру коректної введеної функції у вигляді дерева.
* «Редагувати» - - відкриває форму Command, котра дозволяє відредагувати тіло і коментар виділеної функції.
* «Вилучити функцію» - вилучає виділену функцію.
* «Виконати» - відкриває форму Work, котра дозволяє обрахувати значення вибраної функції для конкретних аргументів.
  + Необхідно ввести аргументи і нажати кнопку «Вичислити». Виводиться результат обчислення функції або невизначене значення - «Невизначено (вичерпана загальна кількість кроків)».
  + Кнопка «Переглянути» дозволяє переглянути послідовність обчислення функції у вигляді дерева обчислення.

1. **Робота з лямбда-виразами.**

Кожна модель лямбда-виразів – це набір (множина) лямбда-виразів, визначених користувачем. Якщо в означенні лямбда-виразу e2 використовується лямбда-вираз e1, то ці вирази повинні входити в один набір. Зауважимо, що означення виразу e1 повинно розміщатися раніше означення лямбда-виразу e2, котрий його використовує. Кожний вираз набору можна поміняти місцями з виразом, що розташований над ним, або з виразом, що розташований під ним, використовуючи кнопки «Перемістити вверх» або «Перемістити вниз»

В вибраному наборі можна редагувати назву, опис набору і працювати з усіма лямбда-виразами набору використовуючи наступні дії:

* «Перевірка» - перевіряє набір виразів на коректність і видає повідомлення про помилки, при їх наявності.
* «Новий» - відкриває форму Command, що містить назву, тіло і коментар нового лямбда-виразу, котрі можна редагувати.
  + Початкові значення тіла та коментаря нового виразу співпадають з тілом та коментарем виділеноного, а початкове значення імені - name01, якщо ім»я виділеного виразу name.
  + Кнопка «Тестувати» - перевіряє коректність лямбда-виразу введеного в текстовому полі «Тіло виразу».
* «Редагувати» - - відкриває форму Command, котра дозволяє відредагувати тіло і коментар виділеного лямбда-виразу.
* «Вилучити» - вилучає виділений лямбда-вираз.
* «Вставити вираз» - дозволяє вибрати довільний вираз з довільного набору виразів.
  + Якщо в наборі вже є вираз, з таким же іменем, то ім»я виразу, що вставляється на name01, name02 ….
  + Вставляємий вираз додається нижче виділеного виразу
* «Робота з набором» - відкриває форму Work, котра дозволяє працювати з довільними лямбда-виразами, що використовують вирази набору.
  + Необхідно ввести вираз в полі «Вираз» і нажати кнопку «Виконати». Виводиться результат обчислення виразу або невизначене значення - «Невизначено (вичерпана загальна кількість кроків)».
  + Кнопка «Переглянути» дозволяє переглянути послідовність обрахування виразу – послідовність виконаних редукцій.
  + Послідовність виконаних редукцій формується, якщо встановлено «Тестовий режим», і становиться доступною кнопка «Переглянути»

1. **Представлення моделей в текстовому файлі.**

Кожну модель можна описати в текстовому файлі, формат якого залежить від виду моделі. В наступному описі текстових форматів моделей будуть вживатися терміни:

* **Ім»я *(Ідентифікатор*) моделі або функції – послідовність букв *латинського* алфавіту і цифр, що починається з букви.** Позначається –Id.
* Коментар – послідовність символів, що починається символом ‘ і закінчується кінцем рядка (символ \n). Послідовність не містить в собі символів ‘ та “, позначається Comm.
* Рядок – послідовність символів, котра обмежена з обох кінців символом “ і не містить його в середині, позначається – St, St1, Al, .. Наприклад: “” –рядок без жодного символа (порожній рядок), “a” – рядок, що скдадається з одного символа a.
* Число – послідовність цифр, позначається – n, r, r1,…
* Виділені слова (наприклад – **Machine**, **end**, …) - обов»язкові (службові) слова.
* При описі формату інколи вживаються символи [, ] – це мета символи. Конструкція, котра знаходиться між ними, може бути відсутня.

*Нормальний алгоритм Маркова*

[Comm]

**Algorithm** Id

**Alphabet** Al1, Al2; [**Numerical** n;]

RuleAlg1

……

RuleAlgk

**end** Id

* Comm – загальний коментар моделі
* Id – ім»я моделі.
* Al1, Al2 – рядки, котрі задають, відповідно, основний і додатковий алфавіти.
* RuleAlg1…. RuleAlgk - список правил алгоритма, кожний з яких розташований в окремому рядку і має вид:
  + St1 -> St2 ; [Comm]
  + St1 ->. St2; [Comm]
* Якщо присутня конструкція **Numerical** n;, то алгоритм обчислює часткову функцію з n аргументами і основний алфавіт має вид Al1 = “|#”.

*Машина з необмеженими регістрами*

[Comm]

**Computer** Id : t;

Command1

……

Commandk

**end** Id

* Comm – загальний коментар програми
* Id – ім.»я програми.
* t - число, що задає кількість вхідний даних.
* Command1…. Commandk - список команд, котрі утворюють програму. Кожна команда розташована в окремому рядку і має вид:
  + i: Z(n) [Comm]
  + i: S(n) [Comm]
  + i: T(n,m) [Comm]
  + i: J(n,m,q) [Comm]
  + i , n, m, q – числа, що задають номер команди (i) і її аргументи (n,m,q).

*Машина Тюрінга*

[Comm]

**Machine** Id

**Alphabet** Al1, Al2; [**Numerical** n;]

**Initial** ini; **Final** fin;

MoveState1

……

MoveStatek

**end** Id

* Comm – загальний коментар моделі
* Id – ім»я моделі.
* Al1, Al2 – рядки, котрі задають, відповідно, основний і додатковий алфавіти.
* ini, fin – рядки, котрі задають, відповідно, початковий і заключний стан машини.
  + Стан машини Тьюрінга – це рядок виду «@XY”, X,Y –довільні символи.
* MoveState1,…. MoveStatek - список правил машини, котрі задають таблицю переходів. Кожне з правил описує поведінку машини в одному зі станів, розташовується окремому рядку і має вид:
  + State : St1 -> Move1: …: Stn -> Moven ; [Comm]
  + State – рядок вида , задає стан машини.
  + St1,…, Stn – рядки, кожний з яких є символ алфавиту (основного або додаткового).
  + Move1, …, Moven - рядки вида “@XYSM”, що описують поведінку машини в стані State з символом St1, …, Stn.
    - @XY – наступний стан машини,
    - S - символ основного або додаткового алфавіту,
    - M - символ <,> або ., що описує рух головки машини.
  + Comm – можливий, коментар поведінки машини в стані State.
* Якщо присутня конструкція **Numerical** n;, то машина обчислює часткову функцію з n аргументами і основний алфавіт має вид Al1 = “|#”.

*Система Поста*

[Comm]

**System** Id

**Alphabet** Al1, Al2; [**Numerical** n;]

RuleSyst1

……

RuleSystk

**end** Id

* Comm – загальний коментар моделі
* Id – ім»я моделі.
* Al1, Al2 – рядки, котрі задають, відповідно, основний і додатковий алфавіти.
* RuleSyst1,…., RuleSystk - список аксіом і правил виведення системи, кожний з яких розташований в окремому рядку і має вид:
  + St1 ; [Comm]
    - аксіома
  + St1 -> St2; [Comm]
    - правило виведення
  + St1, St2 – рядки в яких можуть вживатися символи основного та додаткового алфавіту і змінні. Кожна змінна – це пара символів @R, @S, @T, @U, @V, @W, @X, @Y або @Z.
* Якщо присутня конструкція **Numerical** n;, то система породжує часткову функцію з n аргументами і основний алфавіт має вид Al1 = “|#”.

*Набір частково-рекурсивних функцій*

[Comm]

**Recursive** Is;

Function1

……

Functionk

**end** Is

* Модель з іменем Is і, можливим, коментарем Comm складається з множини функцій Function1, …, Functionk.
* Кожна функція описується в одному рядку і має вигляд
  + Id : n = Expr; [Comm]
  + Id – ім”я функції.
  + n – число, кількість аргументів функції.
  + Expr – вираз, котрий задає тіло функції (визначає як вирахувати функцію), і має вид:
    - a1 – базова функція s(x).
    - z1 – базова функція o(x).
    - iNM – (NM-десяткові цифри) базова функція селектор M аргумента з N аргументів.
    - @S(e0,[e1,…,en]) – операція суперпозиції в n-арну функцію, котра задається виразом e0, n функцій, що задаються виразами e1, …, en.
    - @R(eg,eh) – операція примітивної рекурсії за функціями, що задаються виразами eg і eh.
    - @M(eg) – операція мінімізації за функцією, що задається виразом eg
    - Id1 – ідентифікатор раніше визначеної функції.
  + Comm – коментар функції.

*Набір лямбда виразів*

[Comm]

**Calculus** Lid;

Lambda1

……

Lambdak

**end** Lid

* Модель з іменем Lid і, можливим, коментарем Comm складається з послідовності лямбда виразів з іменами Lambda1, …,Lambdak.
* Кожний вираз описується або двома рядками
  + ‘ Comment lambda
  + Idl = \ x1 x2 … xk . Expr
* Або одним рядком (вираз без коментаря)
  + Idl = \ x1 x2 … xk . Expr
* Кожний вираз описується в одному рядку і має вигляд
  + Idl = \ x1 x2 … xk . Expr
  + Idl – ім”я лямбда виразу.
  + x1, x2, .., xk – ідентифікатори, що задають змінні лямбда виразу.
  + Expr – вираз, котрий має вид:
    - xi – змінна .
    - idP – ідентифікатор раніше визначеного лямбда виразу
    - n – десятковий запис цілого числа без знаку.
    - (e) – лямбда вираз e в дужках.
    - ef ea – операція застосування – два лямбда вирази ef і ea.
    - \y.eb – операція абстракції: y – змінна–ідентифікатор, а – лямбда-вираз