**УДК 004.5**

# МОВНА МОДЕЛЬ ЗДІЙСНЮВАНИХ КОДІВ

**Гнидюк Едуард Миколайович,**
здобувач вищої освіти спеціальності 122 «Комп'ютерні науки»
Миколаївський національний аграрний університет
м. Миколаїв, Україна

**Анотація:** *в даний час існує безліч завдань, при вирішенні яких необхідно застосовувати методи аналізу та трансформації програм. Це завдання безпеки програм, генерації і верифікації вихідних кодів, і навіть зворотної інженерії*

**Ключові слова:** *машинний код, Марківські ланцюги, класифікація.*

Будь-яку послідовність байт довжини більше 16, згідно специфікації Intel IA-32, можна дизасемблювати як послідовність команд, причому єдиним чином. Якщо дизасемблювання проводити з різних позицій, то з'явиться ефект накладання команд, у якому той самий байт може входити до кількох команд.

Отже, та сама послідовність байт може формувати різні послідовності здійснюваних команд залежно від позиції початку дизасемблювання. Зазначений факт є основною проблемою завдання поділу невизначених ділянок програм на код і дані, так як будь-який фрагмент даних можна інтерпретувати як код, хоча він не є таким, і при його виконанні на центральному процесорі відбудеться помилка виконання.

Так як код програми є логічно завершеною послідовністю команд, що реалізує конкретний алгоритм мовою високого рівня, то порядок команд суворо фіксований в ланцюжку і логічно впорядкований. Зазначена семантична особливість команд коду, що здійснюється, є їх специфічною відмінністю від команд, отриманих при дизасемблюванні випадкового фрагмента даних. Аналогічно у природних мовах слова у реченнях, як правило, мають упорядкованість на основі семантики та синтаксису мови. Деякі комбінації слів та літер відповідають різним мовам.

У лінгвістиці одним із найуспішніших методів визначення мови за текстом є Марківські моделі. У цій роботі з урахуванням аналогії з методами порівняння природних мов перевіряється гіпотеза, що однорідні Марківські моделі першого порядку зможуть ефективно моделювати здійсненний код завдання поділу невизначених ділянок програм на код і дані.

За послідовність дискретних випадкових величин $\{X_n\}, n > 0$ приймемо послідовність команд (без параметрів), у якій:

$$P(X_{n+1} = i_n + 1 \setminus X_n = i_n, X_{n-1} = i_n - 1, \dots, X_0 = i_0) = P(X_{n+1} = i_n + 1 \setminus X_n = i_n)$$

тобто на ймовірність появи наступної команди у ланцюжку впливає лише поточна команда. Тоді $\{X_n\}$ утворює однорідний Марківський ланцюг першого порядку, в якому $n$ – номер команди в послідовності. На початковому етапі перевірки гіпотези матрицю перехідних ймовірностей

$$P_j(n) = P(X_{n+1} = j \setminus X_n = i)$$

побудуємо на основі серії експериментів. Тоді, прийнявши за $\{O_j\}$, $j = 0..L$ – послідовність команд, що тестується, яку необхідно класифікувати, ймовірність приналежності її певної моделі $M$ буде виражатися за формулою:

$$P_m(O) = P_m(O_1)P_m(O_2)P_m(O_3) \dots P_m(O_j)$$

Зазначимо, що якщо $\{O_j\}$ являє собою коректну послідовність команд довжиною $L$, то існує тільки одна послідовність байт $\{B_j\}$, що представляє цю послідовність.

Грунтуючись на тому факті, що в тілі програми код функції може починатися в будь-якій позиції файлу, можна отримати кілька інтерпретацій у вигляді послідовностей команд. Відмінність в інтерпретації визначатиметься позицією дизасемблювання першої команди послідовності. У роботі вивчено і доведено той факт, що дизасемблювання послідовності команд, розпочаті з різних позицій файлу, зійдуться з великою ймовірністю менш ніж через 32 байти в одну послідовність. Отже, для будь-якого фрагмента програми більше 64 байт можна отримати послідовність команд $\{O_j\}$ довжиною $L$, з яких перші $K$ будуть командами «сходження» (різними командами на однакових позиціях послідовностей), а $L - K$ команд, що залишаться істинними командами послідовності при дизасемблюванні. Отже, при повному дизасемблюванні фрагмента довжиною понад 64 байт можна з великою ймовірністю отримати справжню послідовність команд, яку вона представляє, або побудувати помилкову послідовність, не характерну для файлів, що здійснюються.

Побудуємо моделі здійсненних та нездійсненних послідовностей команд. Матриці перехідних ймовірностей моделей спочатку пропонується будувати навчанням з урахуванням вибірок файлів. Команди обох видів моделей на першому кроці приймемо рівноймовірними. Навчальну вибірку файлів (для побудови матриць перехідних ймовірностей) будуть представляти набір з файлів формату Windows PE і різні файли нездійсненних форматів. Відповідно до проведеного в роботі порівняння природних мов і машинної, за аналогією з розбиттям текстів природних мов, розділимо здійсненні та нездійсненні файли вибірки на класи і для кожного класу на основі описаного нижче алгоритму побудуємо мовну модель, що представляє даний клас. Для побудови матриці перехідних ймовірностей моделі здійснюваних файлів необхідно отримати послідовності команд здійснюваних файлів. Для цього необхідно побудувати граф потоку управління та команд, починаючи з точки входу в програму, та виділити всі шляхи як коректні послідовності команд моделі здійснюваних файлів.

Імовірність відповідності побудованої моделі рівна добутку відповідних ймовірностей для кожного паросполучення. Можна дійти висновку, що модель, побудована за алгоритмом, здатна класифікувати невизначені ділянки програм на код і дані. З метою перевірки даного висновку та моделі було проведено серію експериментів.

Відповідно до алгоритму пошуку підпослідовності команд з максимальною ймовірністю знаходяться випадкові команди у файлі, здійснене представлення яких у вигляді команд дає великі значення на моделях, що здійснюються, і при дуже великих розмірах файлу може призводити до невірних результатів.

**Список використаних джерел**

1. Leslie Pack Kaelblinga, Michael L. Littman, Anthony R. Cassandrac1 Planning and acting in partially observable stochastic domains // Artificial Intelligence. Volume 101. Issues 112. May 1998. P. 99-134.

2. Карташов М. В. Імовірність, процеси, статистика. Київ. ВПЦ Київський університет, 2007. 504 с.

***Abstract:*** *Currently, there are many tasks that must be solved using the methods of program analysis and transformation. This is a task of program security, generation and verification of source codes, and even reverse engineering..*

***Key words:*** *machine code, Markov chains, classification.*

**Науковий керівник:**
**Пархоменко О.Ю.,**
*канд.ф-м .наук, доцент,*
*доцент кафедри економічної кібернетики і*
*математичного моделювання,*
*Миколаївський національний аграрний університет*

**УДК 338.12**

## УПРАВЛІННЯ ЕКОНОМІЧНИМ РОЗВИТКОМ ПІДПРИЄМСТВА ПІД ЧАС ВІЙНИ

**Гончар Єва Альбертівна,**
здобувач вищої освіти спеціальності 072
«Фінанси, банківська справа та страхування»
Миколаївський національний аграрний університет
м. Миколаїв, Україна

**Анотація:** *встановлено, що ініціативи, спрямовані на розвиток підприємців, можуть сприяти зростанню та стабільності малого та середнього бізнесу. Проаналізовано темпи росту довоєнної економіки. Запропоновано напрями державної підтримки підприємств в умовах воєнного стану.*

**Ключові слова:** *управління, економічний розвиток, війна, підприємництво, корупція.*