

ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова праця  
на правах рукопису

**ПОЛТОРАЦЬКИЙ МАКСИМ ЮРІЙОВИЧ**

УДК 004.942:519.7(043.3)

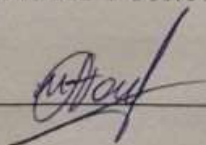
ДИСЕРТАЦІЯ  
**АЛГЕБРАЇЧНИЙ ПІДХІД У ФОРМАЛІЗАЦІЇ ТА АНАЛІЗІ  
ЕКОНОМІЧНИХ ТА ПРАВОВИХ МОДЕЛЕЙ**

121 Інженерія програмного забезпечення  
(шифр і назва спеціальності)

12 Інформаційні технології  
(галузь знань)

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей,  
результатів і текстів інших авторів мають посилання на відповідне джерело



М. Ю. Полторацький

Науковий керівник - Летичевський Олександр Олександрович,  
доктор фізико-математичних наук, старший науковий співробітник

Херсон – 2021

## АНОТАЦІЯ

*Полторацький М. Ю.* Алгебраїчний підхід у формалізації та аналізі економічних та правових моделей. - Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 121 Інженерія програмного забезпечення (12 Інформаційні технології). – Херсонський державний університет, Херсон, 2021.

Загальна проблема дослідження полягає у вивченні та розробленні методів аналізування й моделювання правових та економічних моделей за допомогою алгебраїчного підходу, а саме – за допомогою використання технології інсерційного моделювання.

**Метою роботи** є розроблення загальних принципів аналізування й верифікації правових та економічних моделей засобами інсерційного моделювання.

### **Основні завдання дослідження:**

- аналізування сучасних методів і технологій моделювання й верифікації правових моделей;
- дослідження сучасних методів та технологій аналізування економічних моделей;
- застосування методів інсерційного моделювання для аналізування формальної моделі Податкового кодексу України;
- застосування методів інсерційного моделювання для доведення повноти й відсутності протиріч у нормативно-правовій базі України;
- розроблення програмного засобу для трансляції текстів в алгебраїчний вид;
- застосування методів інсерційного моделювання для доведення властивостей економічних моделей, зокрема властивості рівноваги в моделі Вальраса;
- застосування методів інсерційного моделювання для аналізування моделей токеноміки.

Перший розділ дисертації «*Алгебраїчний підхід у моделюванні правових моделей та аналізуванні їхніх властивостей*» присвячено вивченню методів і систем моделювання для верифікації правових моделей. Розглянуто типові підходи до верифікації правових вимог: використання онтологій (*Symboleo*), *UML/OCL*, стандартів *XML* для кодифікації нормативно-правових баз, застосування технології *eFLINT* і фреймворку *Legal-URN*. Проаналізовано слабкі й сильні сторони кожного з описаних у дисертаційному дослідженні підходів. Описано системи моделювання правових вимог.

У другому розділі «*Інсерційна семантика до аналізу правових моделей*» розкрито основні поняття інсерційного моделювання. Розроблено модель реєстрації платника податку за допомогою інсерційного моделювання. Запропоновано підхід до аналізування й перевіряння правильності юридичних прецедентів. У результаті верифікації моделі знайдено низку структурно-логічних помилок у законодавчій базі. Запропоновано новий підхід до перевіряння законодавчих актів щодо протиріч, повноти й цілісності.

У третьому розділі «*Синтаксичний аналіз юридичних текстів і розпізнавання трас*» схарактеризовано системи синтаксичного аналізу та системи автоматичного реферування, які можуть бути використані під час дослідження правових документів. Для роботи зі слабо структурованим текстом обрано *SDK Pullenti Python* як одну з найпотужніших систем.

За допомогою *SDK Pullenti Python* розроблено програмний засіб для трансляції текстів в алгебраїчний вид і запропоновано підходи до аналізування цієї моделі з використанням інсерційного моделювання. Описаний підхід надасть змогу перевіряти правильність вироків у судових постановках і юридичних прецедентах.

Четвертий розділ «*Алгебраїчний підхід до моделювання економічних моделей та аналізування їхніх властивостей*» містить опис сучасних методів і технологій, які використовуються в економічному моделюванні. Розглянуто системи економічного й економіко-математичного моделювання. Розроблено конкретну та символічну модель Вальраса. Запропоновано підхід для доведення наявності

економічного балансу за допомогою символного моделювання. Схарактеризовано поняття токеноміки й токеномічних законів, формалізовано комерційну модель *SKILLONOMY*. У результаті верифікації моделі токеноміки *SKILLONOMY* виявлено низку структурно-логічних помилок. Запропоновано підхід до аналізування токеномічних моделей із застосуванням символного моделювання.

### **Практичне значення наукових результатів.**

Описані в дисертаційному дослідженні підходи до формалізації та верифікації нормативно-правових актів можуть бути використані для аналізування нормативно-правової бази не тільки України, а й інших країн Європи. Крім того, запропоновані підходи вможливають перевіряння гомогенності нормативно-правової бази України, що суттєво вплине на її якість. Запропонований підхід до аналізування юридичних прецедентів може бути реалізований під час розроблення програмного забезпечення для юристів, що допомагатиме в прийнятті рішень .

Результати застосування алгебраїчного підходу до формалізації та верифікації економічних моделей було використано під час аналізування комерційного проєкту *SKILLONOMY*. Крім того, описаними в дисертаційному дослідженні підходами можливо послуговуватися, аналізуючи й інші токеномічні моделі.

Одержані результати також використовуються з навчальною метою в Херсонському державному університеті під час викладання дисциплін для ІТ-спеціальностей («*Технології символних перетворень*», «*Моделювання та аналіз програмного забезпечення*», «*Системи штучного інтелекту*», «*Спеціальні мови програмування*») і для спеціальності «*Прикладна лінгвістика*» («*Теорія розподіленої концептуальної моделі*»).

**Ключові слова:** верифікація, моделювання правових вимог, алгебраїчний підхід в економіці, інсерційне моделювання, токеноміка, моделювання токеноміки.

## ABSTRACT

**Poltorackiyi Y. Maksym. Algebraic approach in formalization and analysis of economic and legal models. - Qualification scientific paper, manuscript.**

Thesis of a Doctor of Philosophy in specialty 121 Software engineering (12 Information technologies). - Kherson State University, Kherson, 2021.

The general problem of the research is to study and develop methods of analysis and modeling of legal and economic models using an algebraic approach, namely, with using of insertion modeling technology.

The **purpose of the work** is to develop the general principles of analysis and verification of legal and economic models by means of insertion modeling.

### **The main objectives of the study:**

- an analysis of the modern modeling and verification methods and technologies of legal models.
- an analysis of modern methods and technologies that are used for the economic models analysis.
- an application of insertion modeling methods for analysis of the Ukrainian Tax Code formal model.
- an application of insertion modeling methods to prove the completeness and absence of contradictions in the legal framework of Ukraine.
- the development of software for translating texts into algebraic form.
- an application of insertion modeling methods to prove the properties of economic models using the properties of equilibrium in the Walras model.
- an application of insertion modeling methods for analysis of tokenomics models.

In the first section of the dissertation (Algebraic approach in the modeling of legal models and analysing of their properties) an analysis of existing methods and modeling systems for verification of legal models was presented. Typical approaches to legal requirements verification such as using ontologies (Symboleo), UML / OCL, XML standards for codification of legal bases, use of eFLINT technology and Legal-URN framework were considered. Weaknesses and strengths of each of the approaches

that were described in the dissertation research were analyzed. Systems of legal requirements modeling are considered.

In the second section (Insertion semantics for the analysis of legal models) the basic concepts of insertion modeling were presented. A model of taxpayer registration using insertion modeling has been developed. An approach to the analysis and verification of legal precedents was proposed. As the result of model verification, the number of structural and logical errors were found in the legal framework. A new approach to the verification of legislation for inconsistencies, completeness and integrity was proposed.

In the third section (Parsing of legal texts and recognition of traces) the systems of parsing and automatic abstracting systems that can be used to study legal documents were considered. The Pullenti Python SDK was chosen as one of the most powerful systems for working with poorly structured text.

The software tool for translating texts into algebraic form was developed (using the Pullenti Python SDK) and approaches to the analysis of this model (using insertion modeling) were proposed. The described approach will make it possible to check the correctness of sentences in court rulings and legal precedents.

In the fourth section (Algebraic approach in modeling of economic models and analysis of their properties) the modern methods and technologies that are used in economic modeling were considered. Systems of economic and economic-mathematical modeling are considered. A concrete and symbolic model of Walras has been developed. An approach for proving the existence of economic balance using symbolic modeling was proposed. The concepts of tokenomics and tokenomic laws were considered, the commercial model SKILLONOMY was formalized. As a result of verification of the SKILLONOMY tokenomics model, a number of structural and logical errors were found. An approach to the analysis of tokenomic models using symbolic modeling was proposed.

### **The practical significance of scientific results.**

The approaches to formalization and verification of normative legal acts that were described in the dissertation research can be used for the analysis of normative legal

base not only of Ukraine, but also of the European countries. In addition, the proposed approaches make it possible to verify the homogeneity of the regulatory framework of Ukraine, which will significantly affect its quality. The proposed approach for the analysis of legal precedents can be used to develop software to support decision-making for lawyers.

The obtained results of applying the algebraic approach to the formalization and verification of economic models were used to analyze the commercial project SKILLONOMY. In addition, the approaches that were described in the dissertation research can be used to analyze other tokenomic models.

The obtained results are also used for educational purposes at Kherson State University, as a part of methodical materials for courses for IT specialties ("Symbolic transformation technologies", "Software modeling and analysis", "Artificial intelligence systems", "Special programming languages") and course "Distributed Conceptual Model Theory» for applied linguistics.

**Keywords:** verification, modeling of legal requirements, algebraic approach in economics, insertion modeling, tokenomics, modeling of tokenomics.

## СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

### *Основні наукові результати дисертації*

1. V. Kobets, and M. Poltoratskiy "Forming an evolutionarily stable firm strategy under Cournot competition using social preferences." *Communications in Computer and Information Science*. Springer, Cham, 2014, 2014, (CCIS, vol. 469), Juny 2014, pp. 343-361, DOI: 10.1007/978-3-319-13206-8\_17.
2. V. Kobets, V. Yatsenko, and M. Poltoratskiy, "Dynamic Model of Double Electronic Vickrey Auction", in *Proc. 11-th Int. Conf. ICTERI 2015, CEUR-WS*, May 2015, vol. 1356, pp. 236-251. DOI: 10.13140/RG.2.1.4066.7040.
3. V. Kobets, and M. Poltoratskiy, "Using an evolutionary algorithm to improve investment strategies for industries in an economic system", in *Proc. 5-th Int. Workshop ITER 2016, CEUR-WS*, June 2016, vol. 1614, pp. 485-501.
4. Godlevskiy et al., "Formalization and Algebraic Verification of Legal Requirements", in *Proc. 13-th Int. Conf. ICTERI 2017, CEUR-WS*, May 2017, vol. 1844, pp. 524-535.
5. Letichevsky, O. Letychevskiy, V. Peschanenko, and M. Poltorackij, "An Algebraic Approach for Analyzing of Legal Requirements", in *2017 IEEE 25th Int. Requirements Eng. Conf. Workshops (REW'17)*, Sept. 2017, pp. 209-212. DOI: [10.1109/REW.2017.51](https://doi.org/10.1109/REW.2017.51).
6. V. Peschanenko, M. Poltorackiy, "Use of methods of algebraic programming for the formal verification of legal acts", *Проблеми програмування*, № 2–3, с. 109-114, 2018. DOI: 10.15407/pp2018.02.109.
7. O. Letychevsky, V. Peschanenko, V. Radchenko, M. Poltoratzkyi, P. Kovalenko, and S. Mogylo, "Formal Verification of Token Economy Models", in *Proc. 2019 IEEE Int. Conf. Blockchain Cryptocurrency (ICBC'19)*, May 2019, pp. 201-204. DOI: 10.1109/BLOC.2019.8751318.
8. O. Letychevskiy, V. Peschanenko, V. Radchenko, M. Poltoratskiy, and Y. Tarasich, "Formalization and algebraic modeling of tokenomics projects", in *Proc. 15th Int. Conf. ICTERI 2019: Workshops (RMSE'19), CEUR-WS*, June 2019, vol. 2393, pp. 577-584.



9. V. Peschanenko, M. Poltoratskyi, and K. Pryimak, "Formalization and algebraic modeling of university economics", in *Proc. 15th Int. Conf. ICTERI 2019: Workshops (RMSE'19)*, CEUR–WS, June 2019, vol. 2393, pp. 585-594.
10. O. Letychevskyi, V. Peschanenko, M. Poltoratskyi, and Y. Tarasich, "Our Approach to Formal Verification of Token Economy Models", in *Communications in Computer and Information Science*, vol. 1175, pp. 348-363. DOI: 10.1007/978-3-030-39459-2\_16.
11. O. Letychevskyi, V. Peschanenko, M. Poltoratskyi, and Y. Tarasich, "Platform for Modeling of Algebraic Behavior: Experience and Conclusions", in *Proc. 16th Int. Conf. ICTERI 2020: Workshops (RMSEBT'20)*, CEUR–WS, Oct. 2020, vol. 2732, pp.42-57.

.

## ЗМІСТ

ВСТУП .....	12
<b>РОЗДІЛ 1. АЛГЕБРАЇЧНИЙ ПІДХІД У МОДЕЛЮВАННІ ПРАВОВИХ МОДЕЛЕЙ ТА АНАЛІЗУВАННІ ЇХНІХ ВЛАСТИВОСТЕЙ</b>	<b>18</b>
1.1. Бази знань OpenLaws	19
1.2. Формальні мови для моделювання правових вимог	22
1.2.1. XML як стандарт представлення юридичних знань	22
1.2.2. Онтології як стандарт представлення й аналізування юридичних знань	27
1.2.3. UML як стандарт представлення та аналізування юридичних знань.	34
1.2.4. Використання Legal-URN Framework для формалізації правових вимог	37
1.2.5. Використання eFLINT для формалізації правових вимог	39
1.2.6. Деонтична логіка	40
1.3. Машинне навчання й класифікація правових норм	41
1.4. Системи правового моделювання	45
Висновки до першого розділу	47
Список джерел до Розділу 1	48
<b>РОЗДІЛ 2. ІНСЕРЦІЙНА СЕМАНТИКА ДО АНАЛІЗУ ПРАВОВИХ МОДЕЛЕЙ</b>	<b>56</b>
2.1. Теоретичні відомості	56
2.1.1. Агенти й середовища	56
2.1.2. Алгебра поведінки	57
2.1.3. Мова дій	59
2.1.4. Система IMS	61
2.1.5. Символьне моделювання	64
2.2. Використання інсерційного моделювання для аналізування правових моделей	66
2.2.1. Агенти й середовища в правовій моделі	66
2.2.2. Формалізація нормативно-правових актів за допомогою мови дій	70
2.2.3. Алгебра поведінки	72
2.2.4. Результати моделювання	76

	10
2.2.5. Використання інсерційного моделювання для аналізу нормативно-правових прецедентів	77
2.2.6. Методи доведення властивостей	80
Висновки до другого розділу	81
Список джерел до Розділу 2	83
<b>РОЗДІЛ 3. СИНТАКСИЧНИЙ АНАЛІЗ ЮРИДИЧНИХ ТЕКСТІВ І РОЗПІЗНАВАННЯ ТРАС</b>	<b>86</b>
3.1. Огляд систем морфологічного аналізу	86
3.2. Функціональні можливості Pullenti SDK	89
3.2.1. Pullenti SDK як інструмент аналізу слабоструктурованого тексту	89
3.2.2. Модель токена й іменованої сутності. Специфіка роботи з юридичним текстом	90
3.3. Використання Pullenty SDK для розпізнавання трас	93
Висновки до третього розділу	97
Список джерел до Розділу 3	99
<b>РОЗДІЛ 4. АЛГЕБРАЇЧНИЙ ПІДХІД У МОДЕЛЮВАННІ ЕКОНОМІЧНИХ МОДЕЛЕЙ ТА АНАЛІЗУВАННІ ЇХНІХ ВЛАСТИВОСТЕЙ</b>	<b>102</b>
4.1. Формальні методи в економічних дослідженнях	103
4.1.1. Диференціальні рівняння	106
4.1.2. Business process model and notation	109
4.1.3. Скінченні автомати	111
4.1.4. Нечітка логіка	113
4.1.5. Машинне навчання	114
4.2. Системи економіко-математичного моделювання	115
4.3. Інсерційна семантика в економічних дослідженнях .Агенти в економічних процесах	120
4.4. Задача Вальраса й аналіз умови рівноваги	123
4.4.1. Результати моделювання	130
4.5. Токеноміка	134
4.5.1. Результати моделювання. Конкретна модель	142
4.5.2. Результати моделювання. Символьна модель	145
4.6. Пошук вразливих поведінок	147

	11
Висновки до четвертого розділу	148
Список джерел до Розділу 4	150
ВИСНОВКИ	156
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	159
ДОДАТКИ	178
Модель реєстрації платника податку ПДВ	181
Модель SKILLONOMY	190

## ВСТУП

*Актуальність теми.* Усебічна комп'ютеризація спричинила певну трансформацію досліджень формальних моделей. Зокрема, важливим аспектом таких досліджень у наш час є побудова й аналіз формальних правових та економічних моделей.

Використання формальних методів дослідження в правовому аналізі є актуальним питанням, адже це дає змогу дослідити нормативно-правову базу й виявити недоліки та структурно-логічні помилки, а також перевірити цілість і повноту законодавчої бази. Одна з вимог інтеграції України до Європейського союзу (ЄС) – запровадження процесу діджиталізації – зумовлює необхідність кодифікації нормативно-правової бази країни. Аналіз нормативно-правових актів, у свою чергу, допоможе залучити закордонні інвестиції, оскільки компанії – нерезиденти України, упевнені у відсутності юридичних колізій і прецедентів, а також протиріч у законодавчих актах, скоріше погодяться на співпрацю.

Досвід європейських учених доводить, що процес кодифікації та комп'ютерного аналізу нормативних документів є ефективним підходом до правотворчого процесу; про це свідчить поява форматів кодифікації та обміну нормативно-правових документів країн Європи (Італії, Швейцарії, Нідерландів).

Комп'ютерний аналіз правових моделей досліджують Т. Бенч-Капон, О. Боєр, Т. Енгерс, К. Ешл, П. Казановас, Р. Хекстра. Крім того, на сьогодні створено потужні організації, які займаються комп'ютерним аналізом нормативно-правової бази, проектуванням і розробленням систем підтримки прийняття рішень для спеціалістів правової галузі.

Математичні методи доведення властивостей формальних моделей законодавчих та інших нормативно-правових актів здатні істотно знизити ризики виникнення протиріч, що трапляються в юридичній практиці України. Прикладом такого явища є “засуджений” ПДВ, або прецедент на користь платника податку [1], який висвітлив проблему неоднозначного трактування Закону про ПДВ і Закону про оподаткування прибутку. Тобто в законах допускається множинна інтерпретація процесу формування валових витрат, зокрема в разі придбання

підприємством легкового автомобіля. Цей прецедент доводить доцільність використання формальних методів у дослідженні правових моделей.

Розвиток комп'ютерних технологій також вплинув на вивчення й розроблення економічних моделей. Так, використання кінцевих автоматів, машинного навчання та теорії нечіткої логіки знайшли своє місце в наукових працях економічного напрямку, причому більшість теорій, застосованих для формалізації бізнес-процесів, базується на теорії формальних мов. Важливе місце в практиці комп'ютерного моделювання посідає робота з моделями ринкової рівноваги: моделлю Вальраса – Маршала, моделлю міжгалузевого балансу Леонтьєва, моделлю Кейнса тощо. Крім того, упровадження технології блокчейну зумовило зародження нового напрямку економічних досліджень – токеноміки (криптоеконіміки). Створення ефективної моделі токеноміки потребує аналізу поведінки такої моделі при дотриманні токеномічних законів.

Для вивчення правових та економічних моделей дійовими методами можуть бути формальні. Доведення ефективності формальних моделей програм спирається на загальновідомі математичні проблеми: адекватні способи представлення моделі в алгебраїчній формі (В. Глушков, Ю. Капітонова, В. Клименко, О. Летичевський, М. Нікітченко; Й. Баєтен, Л. Карделлі, Р. Мілнер, Дж. Хіллстон, Т. Хоар), експоненціальний вибух кількості станів моделі (О. Летичевський; Е. Кларк, К. Макмілан; ). Дослідженню моделей токеноміки присвячено праці К. Малінової; Н. Ванга, У. Конга, А. Парка, Ч. Хана та ін.

**Зв'язок роботи з науковими планами й темами.** Держбюджетна науково-дослідна робота № 4/16-18 “Методи комп'ютерної алгебри та інсерційного моделювання в системах статичного аналізу і верифікації програмного забезпечення” – 0115U001128. Термін виконання – 2016–2018 рр.

**Мета і завдання дослідження.** Мета дослідження полягає в розробленні загальних принципів аналізування й верифікації правових та економічних моделей засобами інсерційного моделювання.

Для досягнення цієї мети в дисертації розв'язано такі основні **завдання**:

- аналізування сучасних методів і технологій моделювання й верифікації правових моделей;
- аналізування сучасних методів та технологій аналізування економічних моделей;
- застосування методів інсерційного моделювання для аналізування формальної моделі Податкового кодексу України;
- застосування методів інсерційного моделювання для доведення повноти й відсутності протиріч у нормативно-правовій базі України;
- розроблення програмного засобу для трансляції текстів в алгебраїчний вид;
- застосування методів інсерційного моделювання для доведення властивостей економічних моделей, зокрема властивості рівноваги в моделі Вальраса;
- застосування методів інсерційного моделювання для аналізування моделей токеноміки.

*Об'єкт дослідження* – алгебраїчні методи й інструменти аналізу та верифікації моделей, методи інсерційного моделювання.

*Предмет дослідження* – формальні моделі Податкового кодексу України, юридичних документів та економічних моделей (на прикладі моделі економічної рівноваги Вальраса й моделей токеноміки), їхні властивості та методи аналізу.

*Методи дослідження:* при написанні дисертаційного дослідження застосовано такі методи наукового дослідження:

- методи систематизації, порівняння й узагальнення – аналіз систем і методів правового та економічного моделювання, а також систем обробки текстового масиву інформації (розділ 1, підрозділ 3.1, підрозділи 4.1 і 4.2);
- методи теорії алгебраїчного програмування, методи формалізації, теорію трансформації моделей, теорії інсерційного моделювання, теорії формальних мов, теорію логіки, теорії предикатних трансформерів – реалізації моделі реєстрації платника податку ПДВ, використання

методів інсерційного моделювання для аналізу, нормативно-правових прецедентів, моделі економічного балансу Вальраса, реалізації моделі токеноміки (розділ 2, підрозділи 4.3, 4.4, 4.5);

- методи теорії онтологій, теорію трансформації моделей, теорії обробки природного тексту – розроблення програмного засобу для трансляції текстів в алгебраїчний вид (підрозділ 3.3.);
- методи теорії перевірки виконуваності формули – доведення властивостей правових актів (підрозділ 2.7).

**Наукова новизна отриманих результатів.** Наукова новизна й теоретична значущість дослідження полягає в тому, що:

1. Уперше запропоновано використання методів інсерційного моделювання для аналізування та верифікації правових актів, зокрема для зіставлення на відповідність із законами.

Цей підхід дозволить покращити нормативно-правову бази країни, усунути структурні й логічні помилки, перевірити законодавчу базу на відсутність протиріч і цілісність.

2. Уперше формалізовано Податковий кодекс України шляхом створення технологічної лінії від тексту до формального представлення.

Запропоновано підхід до трансляції юридичних текстів в алгебраїчний вид.

Такий підхід дозволяє розв'язати низку проблем, насамперед – створити стандарт кодифікації нормативно-правової бази для країни, а також розробити кодифіковану базу для подальшого аналізу. Така кодифікація допоможе вирішити питання документної коректності, причому не тільки для чинних документів, а й для нещодавно розроблених, і правильності винесених вироків у юридичних прецедентах.

3. Набуло подальшого розвитку дослідження проблеми формалізації економічних моделей, зокрема моделей токеноміки. Представлено новий підхід до аналізу моделей, який полягає в застосуванні методів інсерційного моделювання, запропоновано використання методів алгебраїчного програмування для доведення задачі економічної рівноваги.



Цей підхід сприяє ефективності процесу верифікації економічних і токеномічних моделей, вивченню надійності та стабільності, дослідженню множини значень атрибутів системи, досягненню більш повного покриття станів.

4. Удосконалено процес верифікації законодавчої бази країни методами інсерційного моделювання.

Використання алгебраїчного підходу вможливить покращення стану нормативно-правової бази України, а також проведення аналізу цілісності такої бази для різних країн.

**Практичне значення отриманих результатів.** Застосування інсерційного моделювання під час аналізування правових моделей дозволяє не лише опрацювати наявну законодавчу базу, а й перевірити законопроекти на суперечності з чинними законами. Запропоновані підходи можуть бути ефективними під час кодифікації нормативно-правових актів із метою її подальшого аналізування для пошуку протиріч із законодавчою базою країн Європейського союзу. Крім того, подані методи будуть дієвими в разі зіставлення судових постанов і юридичних прецедентів із законами, з'ясування правильності винесених вироків.

Описані підходи реалізовано під час перевірки комерційного проєкту *SKILLONOMY* [2], що підтверджує можливість їхнього застосування для аналізу токеномічних моделей.

Одержані результати також використовуються з навчальною метою в Херсонському державному університеті під час викладання дисциплін для ІТ-спеціальностей («*Технології символічних перетворень*», «*Моделювання та аналіз програмного забезпечення*», «*Системи штучного інтелекту*», «*Спеціальні мови програмування*») і для спеціальності «*Прикладна лінгвістика*» («*Теорія розподіленої концептуальної моделі*»).

#### **Особистий внесок здобувача.**

У процесі дисертаційного дослідження розроблено формальну модель реєстрації платника податку ПДВ засобами інсерційного моделювання, розроблено конкретну й символічні моделі економічного балансу Вальраса,

розроблено конкретну та символні моделі токеноміки SKILLONOMY, запропоновано підхід до побудови моделі траси із судового рішення для подальшої верифікації правильності винесеного вироку, методи перевірки нормативно-правових прецедентів за допомогою алгебраїчного програмування й інсерційного моделювання.

**Апробація результатів дисертації.** Результати дослідження повідомлялися та обговорювалися на таких конференціях і воркшопах: «ICTERI-2017: International Conference On Ict In Research, Education And Industrial Application». (ICTERI, 2017, м. Київ), Двадцять п'ятій міжнародній конференції «IEEE 25th International Requirements Engineering Conference Workshops (REW)» (2017, м. Лісабон), Міжнародній конференції «ICTERI -2019: International Conference On Ict In Research, Education And Industrial Application» (ICTERI, 2019, м. Херсон), Міжнародній конференції IEEE International Conference on Blockchain and Cryptocurrency (ICBC) «ICTERI-2020: International Conference On Ict In Research, Education And Industrial Application» (ICTERI, 2019, м. Харків).

Дисертаційна робота складається зі вступу, основної частини, що містить чотири розділи, висновків, списку використаних джерел, додатків. Повний обсяг дисертаційної роботи – 191 сторінка, із яких 127 сторінок – основний текст (таблиць – 3, рис. – 31, додатків -4).

**Публікації.** Основні результати дисертаційної роботи викладено в 11 працях. Серед них у виданнях України та зарубіжжя, що включені до міжнародних наукометричних баз, – 11, авторських свідоцтв – 1. Крім того, 8 праць опубліковано в матеріалах міжнародних наукових конференцій. Опубліковані праці повно висвітлюють основний зміст дисертації.

## РОЗДІЛ 1

### АЛГЕБРАЇЧНИЙ ПІДХІД У МОДЕЛЮВАННІ ПРАВОВИХ МОДЕЛЕЙ ТА АНАЛІЗУВАННІ ЇХНІХ ВЛАСТИВОСТЕЙ

У законодавчій базі як апріорі змінному продукті часто трапляється нечітке розмежування юридичних понять при формулюванні та трактуванні питань, а ухвалені нещодавно закони й інші правові чи юридичні документи подеколи суперечать чинним. Звісно, усе це може призвести до неправильних вироків та інших судових прецедентів. Причиною таких невідповідностей нерідко є прихована або незрозуміла взаємозалежність тих чи тих правових актів. Неправильне трактування нормативно-правових актів може призвести до неправильних вироків та інших судових прецедентів. Використовуючи моделювання, можна досить ефективно проаналізувати нормативно-правові документи для подальшого редагування й тим самим покращити стан нормативної бази. Крім того, можливо аналізувати юридичні прецеденти, а також правильність винесених вироків.

Проблемі правового моделювання й гармонізації нормативно-правових актів присвячено багато наукових праць закордонних учених. У Фінляндії та Нідерландах створено низку організацій, які займаються дослідженнями в галузі права й правового регулювання. Досвід зарубіжних науковців (із Нідерландів, Данії, Фінляндії, Італії) доводить, що використання методів і технологій моделювання саме в царині правового моделювання є досить ефективним підходом.

Наразі проблема правового регулювання й гармонізації нормативно-правової бази досить актуальна для багатьох учених Європи, оскільки однією з вимог вступу до різних торговельних союзів і палат є аналіз чинних документів щодо можливих протиріч. Основна концепція взаємодії ґрунтується на гарантії цілісності правової системи, що внеможливіє кардинальні зміни складників нормативно-правової бази в процесі співробітництва з іншими суб'єктами міжнародного права.

Створення торговельних союзів, економічні взаємовідносини між різними країнами спричинили потребу в обміні юридичною інформацією та в запровадженні уніфікованих форматів обміну нормативно-правовими актами, у створенні юридичних консультаційних платформ.

Наразі найпопулярнішими підходами до аналізу й моделювання правових вимог є підходи з використанням *XML*-стандартів, онтологій, представлення нормативно-правових документів у вигляді *uml* тощо. Надалі розглянемо більш детально найбільш поширені підходи до моделювання правових вимог.

### **1.1. Бази знань OpenLaws.**

За останнє десятиліття активно розвивається концепція відкритих даних, представлено досвід учених Нідерландів і Данії у створенні платформи, яка дозволить користувачам швидко знайти потрібну юридичну інформацію. Автори проєкту зазначають, що метою є створення не тільки своєрідного юридичного порталу, а й мережі прецедентного права.

Яскравим представником систем пошукового типу є система *Openlaws.eu* [3], розробники якої ставили перед собою завдання насамперед спроектувати та розробити платформу для зберігання великих масивів юридичних знань, що, у свою чергу, дає змогу користувачам взаємодіяти з різними видами нормативно-правових документів. Юридичні дані в цій системі складаються з чотирьох основних типів спеціалізованих об'єктів:

- документи – структуровані тексти, які можна ієрархічно розкласти на пов'язані між собою списки документів;
- метадані про документи та фрагменти документів;
- посилання між фрагментами документа, які можна використовувати для створення складних ознак, що описують документи або фрагменти документів;
- папки й буфер обміну, які містять документи, фрагменти документів і метадані.

Ця система є системою відкритих юридичних даних, а її основна мета – полегшення для користувачів доступу й використання різної юридичної та правової інформації.

*OpenLaws* [4]. Автори проєкту прагнуть розробити платформу, яка дозволяє користувачам оптимізувати процес пошуку правової інформації. Особливістю цього проєкту є спроба створити мережу прецедентного права.

Розробники вказують, що до створення системи залучено фахівців із правової лінгвістики та юристів; для аналізу нормативно-правових документів система застосовує алгоритми напіваавтоматичного зв'язку джерел права.

Юридичні дані в *OpenLaws* складаються з двох типів об'єктів:

1. Структуровані тексти, ієрархічно розкладені на ланцюжки текстових фрагментів.

2. Метадані про тексти та фрагменти тексту:

2.1. Позначені посилання між фрагментами текстів.

2.2. Опис за допомогою метаданих фрагментів тексту.

Юридичні дані зберігаються в RDF-форматі [5] на сервері *Metalex Document Servers* (на січень 2015 р. кількість документів складала 41 581). RDF-модель представлення даних розроблена консорціумом W3C [6] як технологія семантичної павутини, що включає в себе середовище й опис ресурсів. Тобто фактично RDF [5] – триплет: об'єкт, суб'єкт і предикат, який визначає відношення між об'єктом та суб'єктом.

Для роботи з RDF-моделлю використовують спеціалізовану мову запитів SPARQL [7; 8] або DQL [9], N3QL [10], RDQL [11]. Застосування цих технологій дає змогу розробляти більш складні пошукові запити в правовій системі.

Для розміщення посилань на законодавчі акти послуговуються регулярними виразами, які містять офіційний ідентифікатор закону (номер BWB), що можна використовувати для аналізу перехресних посилань.

Важливою проблемою, яку вирішує описана система, є проблема точного посилання між версією закону та юридичною справою. Нормативно-правовий документ представляється у вигляді XML-документа.

У статті [12] автори представляють досвід побудови графів RDF для представлення моделі відносин між правовою інформацією. Дані зберігаються в XML-документі, що відповідно складається з таких ключових частин:

- ключові слова документа,
- концепція таксономії документа та фрагментів тексту.

Кожен фрагмент має типи, судові факти, представлені за допомогою різних тегів елементів XML-документа.

Обрані користувачем документи або фрагменти тексту можуть бути анотовані поняттями певної юридичної таксономії. Використання концепцій таксономії дає можливість визначити конкретні юридичні питання й процеси.

Спільне вживання таких понять у документах відбиває відношення між правовою інформацією. Ця характеристика не є явною, оскільки кожен документ представлений окремо у вигляді XML-файлу. Для відображення цієї властивості досить зручно використовувати візуалізовані графи знань. Вирази XQuery застосовують для доступу до даних XML, а запити SPARQL CONSTRUCT – для перетворення й формування запитів до XML-даних.

*Justeus* [13] – пошукова система правових знань, у якій представлено нормативно-правові документи, правові акти та юридичні прецеденти у вигляді XML та HTML DOM-структури.

Користувач може здійснювати пошуковий запит, за результатом якого отримує документ у вигляді простої текстової форми або HTML DOM-структури.

Основною відмінністю цієї пошукової системи від інших є аналіз і моделювання перехресних посилань, що трапляються в правовому тексті. Вигляд юридичних посилань показано на Рис. 1.1.



**Рис. 1.1.** Представлення перехресних посилань у системі *Justeus*.

Центральним вузлом виступає документ, який вивчає користувач, а всі інші вузли – це документи, на які посилається поточний центральний документ. Такий підхід дає змогу не тільки представити й візуалізувати перехресні посилання в нормативному документі, а й оцінити складність цього документа. Крім того, розробники системи хочуть працюють над розробленням механізму експортування різних форматів представлення юридичних даних, які використовуються в країнах Європейського Союзу.

## **1.2. Формальні мови для моделювання правових вимог**

### **1.2.1. XML як стандарт представлення юридичних знань.**

Протягом останнього десятиліття проводилися дослідження в галузі юридичного оформлення документів у Східній Європі. Слід відзначити деякі спроби представити нормативно-правові акти у вигляді XML-моделі. Застосування мови XQuery дає змогу вирішити проблеми обробки даних. Зокрема, проєкт такого типу був упроваджений у Болгарії [14; 15].

Зауважимо, що створення різноманітних торговельних палат і союзів спричинило потребу появи уніфікованих форматів обміну юридичними знаннями між різними країнами Європи та світу. Так, в Італії використовується формат кодифікації нормативно-правових знань NormeinRete, який базується на XML-стандарті представлення юридичних документів, DTD та XML-схемах. Крім того, XML-стандарті застосовано в:

- 1) LexDania [16] (Данія);
- 2) CHLexML [17] (Швейцарія);
- 3) eLaw [18] (Австрія);
- 4) SDU BWB [19] (Голландія).

Усі згадані стандарти кодифікації нормативно-правових знань реалізують XML-підхід, можливі лише деякі відмінності: наприклад, в Австрії використовують макроси для перевірки правильності побудови XML-моделі.

У статтях [20, 21] описано відкритий стандарт для розмітки правових документів – METALex [20], який є розширенням стандартної мови XML. Цей стандарт, розроблений групами POWER і E-POWER для DTCA7 у Нідерландах, створено з метою полегшення підтримки програмного забезпечення для прийняття рішень. Нижче представляємо фрагмент юридичного документа з використанням METALex:

```
<xsd:element name="regeling" type="Regulation"
substitutionGroup="Regulation"/>
<CitationDesignation id="statute">
  <TextVersion xml:lang="en">Rome Statute of the International Criminal
Court</TextVersion>
  <TextVersion xml:lang="nl">Statuut van Rome inzake het Internationaal
Strafhof</TextVersion>
</CitationDesignation>
```

Розглянемо деякі особливості у формалізації.

У вузлі *CitationDesignation* зазначаються ідентифікатори нормативно-правового акту. Далі, у вузлі *TextVersion xml:lang="en"*, користувач вказує текст нормативно-правового акту. METALex пропонує більш-менш традиційні функціональні можливості, наявні в інших пошукових системах.

METALex відповідає відкритим стандартам:

1. Консорціум World Wide Web (W3C).
2. Стандарти й пропозиції W3C використовуються в такому контексті:
  - специфікація елементів METALex у мові схем XML та схемі RDF;
  - підтримка просторів імен;
  - XSL (eXtensible Stylesheet Language) для перетворення між специфічними для мови розширення METALex;
  - підтримка статичних імен URL та URN для нормативних актів, осіб і державних органів;
  - підключення XML та підтримка XPointer для посилань між ресурсами.



Схема XML METALex покликана стандартизувати формат обміну юридичними документами та юридичними знаннями, оптимізувати стандартні пошукові запити, описати відношення між законодавчими документами для полегшення пошуку й фільтрації на різних рівнях деталізації. У статті [21], де описано різні стандарти для обміну правовою інформацією, констатовано, що METALex є досить зручним форматом представлення юридичних знань і форматом обміну юридичними документами між різними програмними засобами та платформами відкритих баз юридичних даних.

*LegalRuleML*[22]. Для розширення *RuleML* до *LegalRuleML* було розроблено дві XML-схеми – *LegalMeta.xsd* і модуль *Legal\_operators.xsd*. *LegalMeta.xsd* застосовують для моделювання всіх правових метаданих та правових норм, *Legal\_operators.xsd* визначає деонтичні оператори й поведінку, що необхідна для подальшого використання правил виводу.

Основним тегом є *metaInfo*. Нижче представлено метадані, які містять цей блок.

Блок *identification* використовується для тлумачення правил, таким чином ідентифікація авторів дозволяє розробити політику довіри до розробленої нотації. Політика довіри заснована на авторитетності автора. Отже, якщо конституційний суддя тлумачить норму, довіра до нього вища ніж до іншого користувача. Для визначення ролі в анотації використовують наступний атрибут:

```
<identifications>
```

```
  <identification id="aut1"
```

```
    uri="http://www.cirsfid.unibo.it/monica.palmirani.owl"
```

```
    as="author"/>
```

```
  ...
```

```
</identifications>
```

Блок *sources* забезпечує ідентифікацію текстових фрагментів, що використовується для моделювання правил, а блок *references* – зв'язки між блоками моделі, текстовими фрагментами та правилами. Блоки *sources* і *references* вирішують проблему ізоморфізму юридичних вимог. Такий підхід дав

зможу авторам розробити можливість підключення одного правила до декількох частин тексту або, навпаки, кількох правил до одного й того самого фрагмента тексту. Це реалізовано наступним чином:

```
<references>
  <reference id="customerContract" uri="http://text1#art1"/>
  ...
</references>
<sources>
  <source element="#rule1" refersTo="#customerContract"/>
  ...
</sources>
```

Блок *events* забезпечує визначення певної часової події, *timesInfo* додає семантичну інформацію події, а *rulesInfo* описує метаінформацію, що стосується правил. Оскільки правова подія може мати досить багато часових характеристик (час коли положення або норма набуває чинності; час коли це положення застосовується та інші часові умови), використання такого підходу сприяє уникненню будь-якої надмірності даних і збереженню компактної нотації та високої виразності конструкції. Далі представлено фрагмент опису часових обмежень.

```
<events>
  <event id="e1" value="2011-08-25T01:01:00.0Z"/>
  ...
</events>
<timesInfo>
  <times id="t1">
    <time start="#e1" timeType="efficacy"/>
    <time start="#e1" timeType="inforce"/>
    ...
  </times>
  ...
```

</timesInfo>

Блок *Hierarchy* визначає ієрархію правил, а також взаємозв'язок між правилами – це двійковий оператор, який пов'язує метаправила серед наявних у документі правил. Блок представляється таким чином:

<hierarchy>

```
<range id="rng1" function="superior" from="#rule1"
to="#rule2" timesBlock="#t1" author="#aut2"/>
```

</hierarchy>

Твердження, що клієнт має статус «преміум», якщо його витрати в попередньому році становили не менше за 5000 доларів, за допомогою описаного вище підходу, автори статті [22] формалізують наступним чином:

<Assert mapClosure="universal">

```
<Implies times Block="#t2" ruleType="defeasible" id="rule1">
```

```
<the timesBlock="#t1">
```

```
<Atom id="atm1">
```

```
<Rel>premium</Rel>
```

```
<Var>customer</Var>
```

```
</Atom>
```

```
</then>
```

```
<if timesBlock="#t1">
```

```
<Atom id="atm2" timesBlock="#t3">
```

```
<Rel>previous year spending</Rel>
```

```
<Var>customer</Var>
```

```
<Var>x</Var>
```

```
<Data>= 5000$ </Data>
```

```
</Atom>
```

```
</if>
```

```
</Implies>
```

</Assert>

Як бачимо, використання данного підходу сприяє досить ефективному досягненню високої виразності конструкції, що формалізується, причому особливості технології *LegalRuleML* дають змогу уникати надмірності в даних.

### **1.2.2. Онтології як стандарт представлення й аналізування юридичних знань.**

Онтологія – це формалізоване представлення певного знання про деяку предметну область [23]. Формально поняття онтології можна представити за допомогою триплету:

$O = \langle X, R, F \rangle$ , де

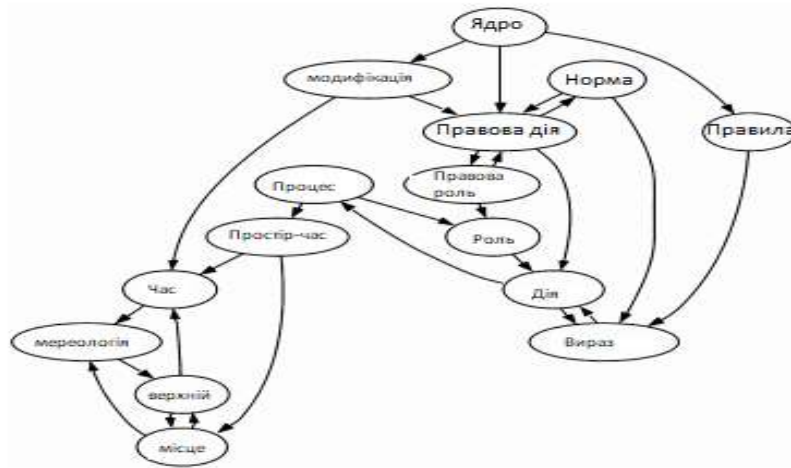
- $X$  – скінченна множина понять предметної області
- $R$  – скінченна множина відношень між цими поняттями
- $F$  – скінченна множина функцій інтерпретації [23].

Фактично онтологія – це набір класів певної предметної області та відношень між цими класами. Наразі технологія онтологій широко застосовується для моделювання різних предметних областей. Найпопулярнішими серед онтологій для моделювання правових вимог є:

- FOLaw [24];
- LRI-Core [25];
- Lexical [26]
- LKIF-Core [27, 28];
- UFO-L [29].

*LRI-Core* є основною та найпотужнішою онтологією для правового моделювання.

На Рис. 1.2 представлено структуру онтології *LKIF-Core*.

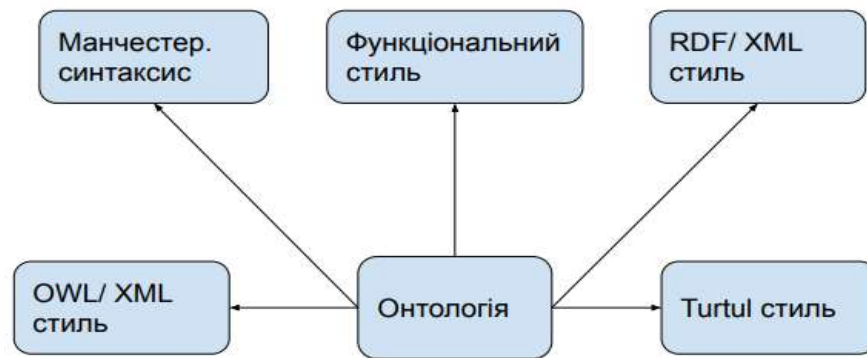


**Рис. 1.2.** Структура онтології LKIF-Core

Цю онтологію створено в рамках розробки формату представлення юридичних знань (LKIF, Legal Knowledge Interchange Format) [30]. Вона містить такі категорії, як «Правова норма», «Судження», «Зобов'язання», «Документ» та інші, що, зі свого боку, дозволяє представляти й структурувати правові документи та знання. Онтологія LKIF-Core складається з 15 модулів, кожен із яких описує набір взаємозв'язаних юридичних понять, а також деякі категорії. Існує ряд підходів до представлення онтологічного знання:

- OWL/XML [31];
- Functional Syntax [32];
- Turtle [33];
- Manchester Syntax [34].

На рис.1.3 показано структуру онтології та її види представлення. За останні п'ять років широкого вжитку здобула технологія Semantic Web, актуалізувалося застосування теорії онтологій у юриспруденції. Інструменти на основі Semantic Web й теорії онтологій загалом призначені не тільки для оптимізації пошуку правової інформації, а також для кластеризації, класифікації та управління юридичним знанням. [35]



**Рис.1.3.** Види представлення онтологій.

У статті [36] розкрито досвід автоматичного створення формальної моделі нормативно-правового акту (змодельовано в owl-файлі). Запропонована модель призначена для семантичного пошуку правових актів та автоматичної генерації законопроектів із набору правових норм, що містяться в ній.

Ефективність такого підходу полягає в раціональному моделюванні різних елементів правових норм та елементів правових відносин, які використовуються для визначення інших понять. Крім того, представлення нормативно-правового документа у вигляді owl-файлів дає змогу послуговуватися мовою SWRL [37]. Самі правила SWRL можуть зберігатися безпосередньо в OWL-онтології і застосовуватися машиною логічного висновку, наприклад, Pellet [38].

Правила, написані SWRL, подають у вигляді диз'юнктивів Хорна, які можуть складатися тільки з одномісних або двомісних предикатів. Кожне правило формується з заголовка й тіла:

$$(has-parent (? X, ? Y) \ \&\& \ has-father (? Y, ? Z) ) \ -> \ has-uncle (? x, ? z)$$

Онтологія LKIF-Core базується на розроблених раніше LKIF-правилах [39].

Наведемо приклад LKIF-правила:

$$\begin{aligned}
 & (rule \ \S-1601-BGB ( \\
 & \quad if (direct-lineage \ ?x \ ?y) \\
 & \quad (obligated-to-support \ ?x \ ?y)))
 \end{aligned}$$

Правило представлено як *if then* і є диз'юнктом Хорна. Воно буде істинним, коли істинною є умова (*direct-lineage ?x ?y*), адже тоді й предикат (*obligated-to-support ?x ?y*) також буде істинним.

Положення нормативних документів в онтології LKIF-Core представляються у вигляді пропозицій (Proposition) або суджень (Qualification) деяких агентів (Agent), які мають певний стосунок (Propositional attitude) до пропозицій. Агенти поділяються на два основних типи:

- Персони (Person) – позначають окремих людей;
- Організації (Organization) – позначають різні інститути, фірми, корпорації, підприємства тощо.

Агенти переважно визначають певні ролі (Role), які, зі свого боку, визначають їхню множину дій. Пропозиції (Proposition) і судження (Qualification) агентів утілюються через певну основу (Medium). Зазвичай її роль відіграє деякий домен (Document). На основі LKIF-правил розроблено онтологію LKIF-Core [40]. Нижче представлено приклад онтології LKIF-Core:

```
<? Xml version = "1.0" encoding = "UTF-8" >
  <lkif>
    <rule id = "s11601-BGB" >
      <body > <s > (direct-lineage ?x ?y) </s > </body >
      <head > <s > (obligated-to-support ?x ?y) </s > </head >
    </rule >
  </lkif >
```

Оскільки LKIF-Core ґрунтується на LKIF-правилах, то формалізація є досить схожою. Основна відмінність полягає у використанні XML-представлення, адже так у *rule id* вказується ідентифікатор нормативно-правового акту. Далі у вузлі *body* зазначають умову (*direct-lineage ?x ?y*), а в *head* – предикат (*obligated-to-support ?x ?y*), тобто умову, яка повинна бути виконана, якщо предикат (*direct-lineage ?x ?y*) істинний.

LKIF-Core та мова OWL уможливають більш повне уявлення про положення документа, також пропонується використовувати мову правил семантичного вебу (Semantic Web Rule Language – SWRL). SWRL сформовано завдяки об'єднанню мов OWL і RuleML [41]. Натомість основою RuleML є мова Datalog, яка, зі свого боку, є синтаксичною підмножиною мови логічного

програмування Prolog.

У статті [42] описано механізм аналізу нормативно-правових документів із використанням онтологій. Автори статті представляють онтологію, що визначає еквівалентність поняття інвестиційного проєкту та формалізації проєкту; потрібно перевірити, чи підпадає він під модель поняття стратегічного інвестиційного проєкту. Далі подано онтологію понять інвестиційного проєкту (у манчестерському стилі):

***Class:*** *Strategy-invest-project*

***Annotations:*** *Стратегічний інвестиційний проєкт*

***EquivalentTo:***

*Strategy-invest-project and (*

*has-invest-sum some (*

*Money-sum and (*

*more-Than value 100-million)))*

*and (member of value register-of-strategic-invest-project) and (apply-for value)*

***SubClassOf:*** *medium value law-about-government-supplyof-investment-activity.*

Онтологію представлено у вигляді класу, вона містить блок для визначення (назви) та *EquivalentTo* для формалізації визначень у вигляді онтологічних класів, наприклад, сумарного обсягу інвестицій, що не менше за 100 мільярдів умовних одиниць.

Застосування онтологій для аналізу нормативно-правових документів дає можливість проаналізувати дві формалізовані онтології:

Нехай А, В – дві онтології, які будуть аналізуватися. Тоді:

1. А і В – еквівалентні положення. А і В позначають одне й те саме, визначення не містять значущих відмінностей.

2. Клас А є суперкласом для В, водночас рівень юридичної сили документа В вищий, ніж документа А. У такому випадку ймовірно, що положення першого надлишкове або визначене недостатньо докладно.



3. Клас А не визначений підкласом класу В – визначення не відповідає необхідним і достатнім умовам заявленого користувачем визначення, потребує уточнення.

За допомогою мови SWRL можливо створювати більш складні конструкції, які, наприклад, будуть сприяти контролюванню наявності протиріч завдяки вбудованим відношенням. Нижче представлено фрагмент SWRL-правил:

$$\begin{aligned} & \textit{Strategy-invest-project}(?act) \quad \wedge \quad \textit{part}(?act, ?part-act) \quad \wedge \quad \textit{Strategy-invest-} \\ & \textit{project}(?part-act) \quad \wedge \quad \textit{has-invest-sum}(?act, ?sum) \quad \wedge \quad \textit{has-invest-sum}(?part-act, \\ & ?sum-part) \quad \wedge \quad \textit{swrlb: greaterThan}(?sum-part, ?sum) \quad \textit{!Wrong-} \\ & \textit{definition}(?actpart); \end{aligned}$$

Крім того, використовуючи *LinkData* [43], можливо розв'язати проблему перехресних посилань у нормативно-правових документах. Якщо представити нормативно-правовий акт у вигляді OWL-файлів, пошукові запити можна оптимізувати – зробити їх більш складними.

*Symboleo* [44, 45] . Онтологія *Symboleo* ґрунтується на правовій онтології UFO-L, розробленій для формалізації правових смарт-контрактів. Розвиток технології блокчейну спричинив появу інтересу до технології представлення юридичних знань у вигляді смарт-контрактів. Розробники запевняють, що ця технологія може бути ефективно використана для представлення юридичних знань мовою Solidity – базовою мовою програмування смарт-контрактів.

Автори технології визнають, що юридичні тонкощі дуже складно формалізувати за допомогою Solidity, тому доцільно застосовувати цю технологію як основну мову представлення й верифікації нормативно-правового знання в смарт-контрактах.

Основною метою *Symboleo* є сприяння у визначенні параметризованих шаблонів контрактів (для розробників контрактів), що в подальшому дасть змогу використовувати формальну семантику *Symboleo* для перевірки контракту щодо порушень властивості безпеки та властивості життєдіяльності контракту. Нижче подано формалізацію смарт-контракту за допомогою технології *Symboleo*:

**Domain salesD**

*Goods* isA *Asset* with *goodsID*: *Integer*;

*Delivered* isA *Event* with *delAddress*: *String*,

*delDueDate*: *Date*;

**endDomain****Contract salesC**

*seller*: *Seller*, *buyer*: *Buyer*, *ID*: *Integer*, *amnt*: *Integer*,

*curr*: *Currency*, *delAdd*, *delDd*: *String*

**Declarations**

*goods* : *Goods* with *goodsID* := *ID*;

*delivered* : *Delivered* with *delAddress* := *delAdd*, *delDueDate* := *delDd*;

**Preconditions**

*isOwner*(*seller*, *goods*) **AND NOT** *isOwner*(*buyer*, *goods*);

**Postconditions**

*isOwner*(*buyer*, *goods*) **AND NOT** *isOwner*(*seller*, *goods*);

**Obligations**

*O1* : *O*(*Seller*, *Buyer*, *true*, *happensBefore*(*delivered*, *delivered.delDueD*));

*O2* : *O*(*Buyer*, *Seller*, *true*, *happensBefore*(*paid*, *paid.payDueD*));

**Powers**

*P1* : *violates*(*O2*, )  $\rightarrow$  *P*(*Seller*, *Buyer*, *true*, *terminates*(*salesC*));

*SurvivingObl*

**Constraints**

*not*(*isEqual*(*buyer*, *seller*));

**endContract**

*Symboleo* ґрунтується на поняттях зобов'язання й повноважень для ролей, заявлених у контракті. Ролі визначаються як класи, що представляються за допомогою *Symboleo*. У тексті формалізації є ролі *Seller*, *Buyer*, які мають відповідні допоміжні ідентифікатори *ID*. Оскільки кожна роль може мати зобов'язання та повноваження, у тексті формалізації є два зобов'язання:

- продавець повинен доставити товар за адресою доставки до дати доставки (*O1*);

- покупець повинен сплатити кошти за товар (*O2*).

Крім того, зауважимо, що в тексті є формалізація умови *P1*, яка визначає можливість розірвання контракту за умови порушення повноваження *O2*.

Застосування програмного засобу jREC [46] дає змогу перевірити й змодельовати шаблон формалізованого смарт-контракту, оскільки цей засіб використовує вхідні дані – набір сценаріїв, кожен із яких складається з послідовності подій з очікуваними кінцевими станами контракту для кожного сценарію.

### **1.2.3. UML як стандарт представлення та аналізування юридичних знань.**

У Нідерландах податкова митна адміністрація очолює дослідницький проєкт POWER [47], мета програми – розробити методи й інструменти для аналізування нормативно-правової бази, що в подальшому може поліпшити якість законодавства.

Так, у статті [47] зосереджено увагу на методі, що базується на UML і його розширенні UML/OCL [48], як основному інструменті моделювання й верифікації нормативно-правових вимог. Автори виділяють такі етапи моделювання правових вимог із використанням UML/OCL підходу:

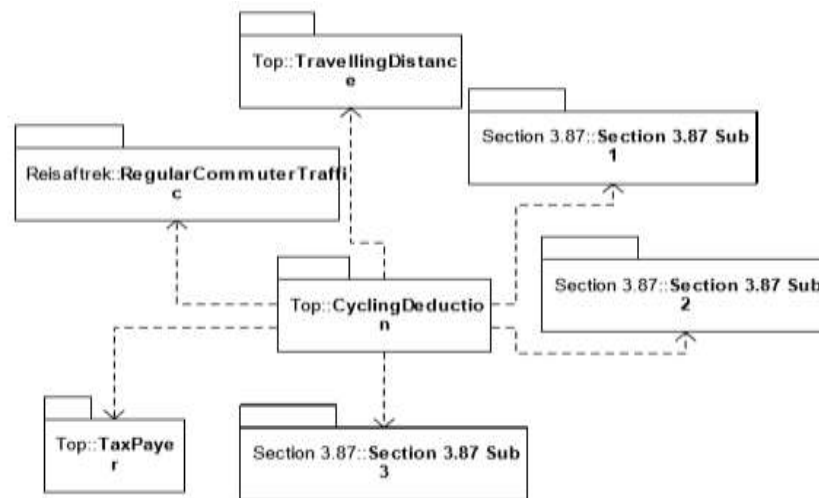
- Процес формалізації нормативно-правових актів.
- Формалізація обмежень OCL
- Верифікація концептуальних моделей, включно з виявленням неповноти та виявленням відсутності законодавчих і нормативних положень (неповноти).

- Тестування й перевіряння компонентів знань із залученням експертів для сертифікації компонентів знань.

На основі розробленої інтегрованої моделі, за допомогою розробленого програмного інструменту FORCE генерується база знань, що надалі

використовується як джерело юридичних знань для подальшої правової аргументації.

Після встановлення структури закону ми визначаємо поняття, властивості понять і взаємозв'язків між поняттями в аналізованому правовому тексті. Поняття представлено у вигляді UML, формалізацію правил – у вигляді інваріантів, записаних мовою Object Constraint Language (OCL). Автори статті подають модель закону у вигляді діаграми класів UML-моделі, із чим можна ознайомитися на рис. 1.4.



**Рис.1.4.** Представлення поняття нормативно-правового документа у вигляді діаграми класів[46].

Для розробленої діаграми класів нормативно-правової моделі формалізуються обмеження за допомогою OCL-конструкцій. Нижче проілюстровано фрагмент OCL-обмеження:

```

<<type>> CyclingDeduction -amount:: real
«attributeExceptionInvariant»
{
    amount = amount@pre -
    TravelingDistance.allInstances->
    select(travelledByBicycle = true)->
    collect(travellingExpense.amount->sum
}
  
```

Далі формалізовану правову модель та OCL-обмеження аналізують за допомогою OCL-вирішувачів.

У статті [49] опубліковано аналіз нормативно-податкової бази Люксембургу із використанням UML-підходу для моделювання процесуальних правових норм. Основу моделі складає підхід Domain-Specific Modeling (DSM).

Автори розрізняють два етапи:

- Розроблення й моделювання простору імен.
- Моделювання правових вимог.

Важливим чинником є узгодженість між правовим доменом (простором імен) і правовими вимогами. Формування домену спрямовано на точне уявлення про юридичне поняття та взаємозв'язки у вхідних юридичних текстах.

Для формування правових вимог використовують UML; після того як модель розроблено, її автоматично транслюють у модель із використанням OCL-виразів; отримана правова модель з OCL-обмеженнями може бути використана для подальшого автоматичного аналізу за допомогою OCLsolvers [50].

Статтю [51] присвячено розробленню програмного модуля– інструменту підтримки прийняття рішень для прогнозування можливих наслідків правових реформ.

Основним завданням PolicySim є підтримка моделювання процесуальної правової політики. До особливостей цього модуля зараховано таке:

- моделювання простору імен, який виражений за допомогою діаграми класів UML. Моделювання простору імен (домену) використовуються для представлення основних понять та їхніх взаємозв'язків у тексті правового контенту;
- для кожної правової стратегії розробляється діаграма діяльності UML, що описує детальну формалізацію певної юридичної процедури.

Наприклад, модель домену анотується текстовою інформацією (з агрегованих даних: переписів або експертних оцінок для забезпечення реалістичних даних для моделювання).

Автори статті [52] наводять приклад формалізації тексту закону із застосуванням OCL-обмеження для UML-моделей. Нижче подано фрагмент OCL-формалізації для визначення родинних зв'язків:

*self <> Parent AND NOT(Parent.isMarried)*

*AND Parent.isAdult*

*AND NOT(self.isMarried)*

*AND self.isAdult*

*AND sharesAHouseholdForMoreThanSixMonthsContinuouslyWith (Parent, ContributionYear)*

*AND isRegisteredAtTheSameAdressDuringThePeriodOfTheJointHouseholdWith (Parent, ContributionYear) ==> self: Partner*

Розглянемо деякі фрагменти цієї формалізації:

- *AND NOT(self.isMarried)* – формалізація, яка визначає, що поточний екземпляр класу не одружений;

- *isRegisteredAtTheSameAdressDuringThePeriodOfTheJointHouseholdWith (Parent, ContributionYear)* – формалізація визначає, що поточний екземпляр класу зареєстрований за однією адресою протягом періоду *ContributionYear* спільного проживання.

Такий підхід є досить ефективним у пошуку правових помилок і логічних колізій, але його основним недоліком є громіздка конструкція, оскільки спочатку потрібно побудувати модель законодавчого акту засобами мови UML, після чого формалізувати обмеження до цього правила мовою OCL-обмежень.

#### **1.2.4. Використання Legal-URN Framework для формалізації правових вимог.**

Під час розроблення експертних систем і систем підтримки прийняття рішень основною проблемою є забезпечення відповідності між нормативними вимогами й базою даних, яку опрацьовують, у розробці та формалізації бази знань. Моделювання юридичних вимог є важким завданням для інженерів баз знань, оскільки правові тексти є досить складними для розуміння.

У статті [53] автори представляють спеціалізований для юридичних текстів Legal-URN Framework, який може полегшити тлумачення складних правових конструкцій.

Цей підхід складається з чотирьох рівнів абстракції, пов'язаних між собою певним чином. Основою фреймворку є мова нотації вимог користувача (URN), яку застосовують для опису й поєднання правових моделей. Моделі диференціюють на чотири класи Хофельда [54], а потім створюють моделі Хофельда. Для цього автори коментують кожне юридичне твердження в кожному юридичному документі за допомогою одного з корелятивних класів Хофельда. Текстову класифікацію Хофельда використовують для аналізу юридичних норм. Після набору уточнень класифікації Хофельда ця модель перетворюється на модель формату GRL [55].

Для перевірки сформованості моделей і виявлення випадків їхньої невідповідності застосовують мову обмежень об'єктів (OCL). Завдяки алгоритмові розставлення пріоритетів та оцінювання відповідності такий підхід уможливорює виявлення подібностей і конфліктів між нормативними актами та (за потреби) виключення їх із моделі.

Автор праці стверджує, що інструмент jUCMNav [56], інструмент Eclipse для моделювання та аналізування URN [57] було розширено для підтримання описуваного фреймворку.

Плагін jUCMNav використовують як основний інструмент графічного моделювання й аналізування правового процесу. Цей програмний засіб дозволяє визначати метадані, траси URN, які відповідають формалізації OCL-обмежень. У статті зауважено, що jUCMNav підтримує інтерфейси для URN та GRL, і наголошено на таких результатах:

1. Розширення інструменту jUCMNav для підтримання нових налаштувань URN у правовому моделюванні й використання конструкцій OCL-виразів.
2. Правила перевірки відповідності та нові алгоритми аналізу GRL.

### 1.2.5. Використання eFLINT для формалізації правових вимог.

eFLINT [58] – це мова для формалізації правових вимог, розроблена на основі юридичної концепції моделі Хофельда.

Мова може бути використана:

- для формалізації нормативно- правових вимог;
- для формування міркувань;
- для розроблення програмного забезпечення моніторингу нормативно-правових документів.

Автори *eFLINT* указують, що вона призначена для опису найрізноманітніших видів нормативно-правових документів: законів, нормативних актів, політичних і юридичних контрактів. На рис.1.5 наведено абстрактний синтаксис цієї мови.

```

x ∈ type_ids      ::= ...
s ∈ strings       ::= ...
z ∈ Z             ::= {0, 1, ..., -1, -2, ...}
i ∈ instance_exprs ::= ...
b ∈ boolean_exprs ::= ...

δ ∈ domains      ::= strings
                  | string_set(s1, ..., sn)
                  | Z
                  | int_set(z1, ..., zn)
                  | product(x1, ..., xn)
                  | ...

fdc ∈ fact_decls  ::= fdecl(d, δ, b02)
adc ∈ act_decls   ::= adecl(d0, d1, d2, d*, b*, c*, b02)
edc ∈ event_decls ::= edecl(d0, b2, c*, b02)
ddc ∈ duty_decls  ::= ddecl(d0, d1, d2, d*, b2, b02)
c ∈ posts         ::= create(i) | terminate(i)
dc ∈ decls        ::= fdc | adc | edc | ddc | ...
specifications    ::= dc*

```

Рис. 1.5. Абстрактний синтаксис технології eFLINT

Крім того, мова eFLINT допускає використання кванторів *exists* та *forall*. Нижче представлено фрагмент використання кванторів в *eFLINT*:

( *Exists* candidate : vote ( voter ( Alice ) , candidate ) )

( *Forall* citizen : vote ( voter ( citizen ) , Chloe ) )



Мову eFLINT можливо застосовувати для розроблення різних додатків для аналізу нормативно-правових документів і моніторингу відповідності чинної правової системи.

Прагматичні дизайнерські рішення були прийняті, щоб дозволити повторне використання специфікацій для різних стратегій міркувань. Наведемо фрагмент-приклад формалізації засобами технології eFLINT:

*Fact* *hass* voted **Identified by** voter

**Holds when** ( *Exists* candidate : vote ( voter , candidate ) )

**Predicate** vote concluded **When** ( *Exists* candidate : winner( candidate))

**Predicate** voters done **When** ( *Forall* citizen : ! voter( ) || has voted ( voter( ) ) )

*Fact* number of votes **Identified by** Int

**Derived from Count** ( *Foreach* vote : vote **When Holds** ( vote ) )\

*Act* cast vote

*Actor* voter

*Recipient* administrator

*Related to* candidate

**Conditioned by** voter && ! has voted( )

*Creates* vote ( )

**Terminates** cast vote duty ( )

Описана формалізація – це правила для голосування. Так, зокрема *Exists candidate : winner( candidate)* формалізує твердження, що повинен існувати принаймні один переможець серед кандидатів. Ця мова оперує фактами.

Автори технології констатують, що eFLINT буде ефективною під час розроблення інтелектуальних, правових контрактів у DLT-проектах, оскільки є більш гнучкою, ніж мова Solidity (базова мова розроблення смарт-контрактів) і може більш точно передавати правові нюанси, на відміну від Solidity.

### 1.2.6. Деонтична логіка.

*LLD (Language for Legal Discourse)* [59] представляє текст законів у термінах числення, масових умовах, станах, подіях, діях, дозволах, забов'язаннях.

Єдина відмінність від логічного програмування в запропонованій реалізації LLD полягає в наявності нечітких категорій. Припустимо, нам потрібно формалізувати вимогу, що *A* не може бути одночасно *Person* та *Corporation*, тоді в термінах LLD це можна представити таким чином:

$$FALSE \leq (Person\ A) \text{ AND } (Corporation\ A)$$

Умову, коли *Person* та *Corporation* взаємовичерпні, у термінах LLD чином реалізовано так:

$$(Person\ A)$$

$$\leq (Actor\ A) \text{ AND NOT } (Corporation\ A)$$

$$\text{ and } (Corporation\ A)$$

$$\leq (Actor\ A) \text{ AND NOT}$$

$$(Person\ A)$$

*LLD* є досить виразною мовою й може досить ефективно відтворювати складні зв'язки в нормативно-правових документах. У юридичних експертних системах природа права як нормативної системи породила усвідомлену необхідність включення деонтичної логіки. Оскільки традиційна та класична логіка надає певні форми для міркування (форми, що приймають істини), то деонтична логіка дає стандарти для міркувань із твердженнями, які не можуть приймати значення істини в тому сенсі, що вони описують норми або імперативи.

Ці форми не можуть бути перевірені на істинність чи хибність. Багато вчених визнають, що саме некласична логіка, зокрема деонтична, повинна бути засадничою для теорій щодо представлення юридичних знань і концепцій. Розроблена мова LLD має певні переваги в представленні юридичних знань.

### **1.3. Машинне навчання й класифікація правових норм .**

Досить популярним та результативним підходом до аналізу нормативно-правових документів і юридичних прецедентів є використання машинного навчання. Так, у статті [60] описано підхід до прогнозування судових рішень із використанням машинного навчання. Автори статті стверджують, що їхній підхід дає досить високу точність і залежить від судових атрибутів. Основна ідея полягає

в тому, що машинне навчання може бути застосовано для автоматичного виявлення «байдужості» суддів, тобто випадків, коли суддя ігнорує важливі обставини справи під час прийняття рішень. Реалізація такого підходу сприятиме усуненню упередженості суддів і покращенню судової процедури загалом, крім того, його можна розглядати як інструмент підтримки прийняття рішень для юристів.

Автори статті [61] представляють досвід використання машинного навчання для класифікації нормативно-правових документів, наводячи дані про 90 % точності класифікації. Наведений приклад стосується виявлення тенденції думок Бразильських суддів, проте вказано, що підхід із застосованим машинним навчанням дасть змогу будь-яким юридичним фірмам отримувати повну інформацію для пошуку більш ефективних правових стратегій.

Статтю [62] також присвячено підходові з використанням машинного навчання, але для правової аргументації. Автори цієї праці визначають особливість свого підходу – реалізацію Argument-Based Machine Learning (далі – ABML) як алгоритму, що вивчає правила типу *if-then* із бази аргументованих прикладів; машинне навчання застосовано як розширення алгоритму Карла Ніблета [63]. Проілюструємо фрагмент правил, ужитих для аргументації:

R1: *IF HairColor = blond THEN CreditApproved = yes*

R2: *IF PaysRegularly = no AND HairColor = blond THEN CreditApproved = yes*

R3: *IF PaysRegularly = no AND Rich = yes THEN CreditApproved = yes*

R4: *IF HairColor = blond AND Rich = yes THEN CreditApproved = yes*

Наведені приклади – формалізовані вимоги, що репрезентують умови, за яких потрібно надати клієнтові кредитні послуги. Далі – приклад значень атрибутів для затвердження кредиту:

*((PaysRegularly = no, Rich = yes, HairColor = blond),*

*CreditApproved = yes,*

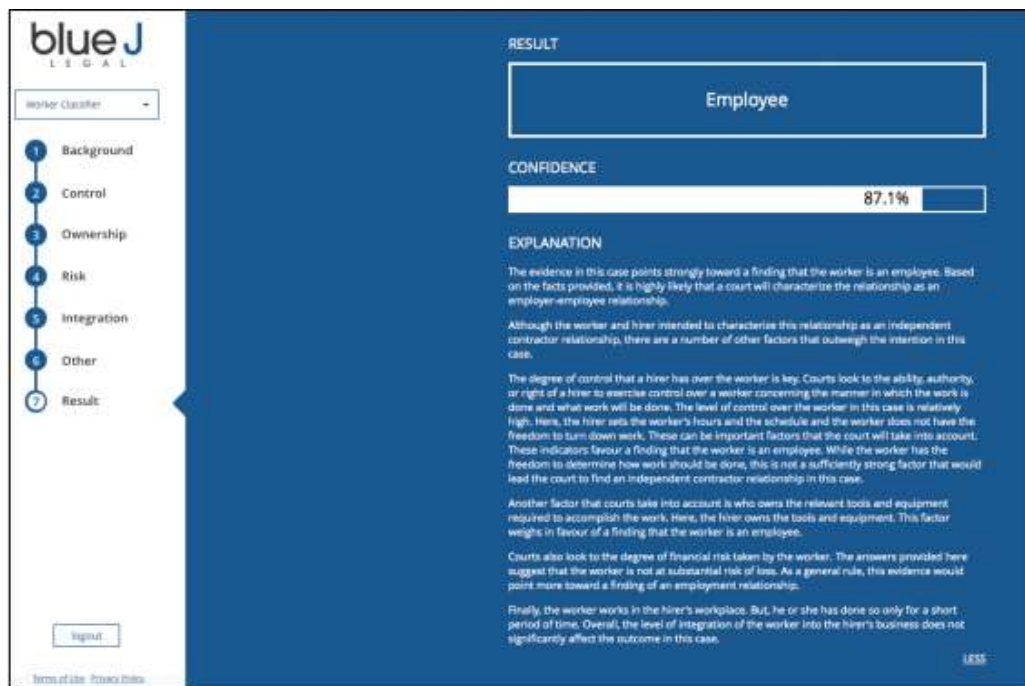
*{CreditApproved = yes because Rich = yes,*

*CreditApproved = yes despite PaysRegularly = no}).*

У цьому фрагменті є і негативний аргумент *CreditApproved = yes despite PaysRegularly = no*, і позитивний *CreditApproved = yes because Rich = yes*. Усі чотири правила охоплюють приклад для значень атрибутів, але зазначимо, що перше правило не узгоджується з позитивним аргументом. Друге та третє правила узгоджуються з негативним аргументом *PaysRegularly = no*, тому ці правила не можемо застосувати. Тільки четверте правило охоплює початковий стан атрибутів, тому його ми можемо застосувати. Використання такого підходу дає можливість зменшити простір для пошуку імовірних гіпотез, тим самим знизивши складність пошукового процесу.

Система Blue J Legal [64], спрямованим на дослідження технологій машинного навчання для прогнозування правових результатів податкового законодавства. Цей програмний засіб базується на аналізі прецедентного права, результати його роботи представлено на рис. 1.6.

Система Blue J, налізує базу судових постанов, використовуючи прецедентний аналіз, крім результатів аналізу користувач може отримати доступ до схожих справ. Автори даної системи зазначають, що використання машинного навчання дає можливість з'ясувати найкращий спосіб присвоєння ваг для змінних та з'ясувати, які змінні взаємодіють між собою. Крім використання методів та алгоритмів машинного навчання користувачеві надається можливість заповнити декілька запитань, які спрощують роботу алгоритмів машинного навчання. Нейронні мережі знаходять приховані зв'язки між інформацією у справах, та роблять аналіз, що досить складно уявити при традиційному підході до прецедентного аналізу навіть із необмеженим використанням часу та ресурсів.



**Рис.1.6.** Результат роботи системи Blue J

Система Blue J аналізує базу судових постанов, використовуючи прецедентний аналіз; крім результатів аналізу, користувач може отримати доступ до схожих справ. Автори системи зазначають, що застосування машинного навчання вможливілює знайдення найкращого способу присвоєння ваг для змінних і з'ясування того, які змінні взаємодіють між собою. Крім використання методів та алгоритмів машинного навчання, користувачеві пропонують відповісти на декілька запитань для спрощення роботи алгоритмів машинного навчання. Нейронні мережі знаходять приховані зв'язки між інформацією в справах і здійснюють аналіз, чого не буває при традиційному підході до прецедентного аналізу, навіть із необмеженим використанням часу й ресурсів.

На думку авторів, використання машинного підходу допоможе якісно вдосконалити процес аналізування не тільки податкового кодексу, а й інших розділів права. За допомогою цього програмного модуля юрист може досить швидко та якісно проаналізувати й дослідити юридичний прецедент, спираючись на факти з минулих справ і прецедентів.

#### 1.4. Системи правового моделювання.

Починаючи з 1970 року, було розроблено понад 25 дослідницьких проєктів щодо використання інформаційних технологій у правовій діяльності. Для ЄС проблема застосування інформаційних технологій до правового регулювання не є новою, у Союзі існує кілька систем моделювання правової політики: EUROMOD [65, 66], POLIMOD [67], Shyster [68, 69] та ін. У Фінляндії та Нідерландах створено організацією дослідників у галузі права й інформатики, які займаються комп'ютерним аналізом юридичних текстів і документів – JURIX [70, 71].

*EUROMOD* [67] являє собою програмну платформу, що складається з мови програмування, специфічної (статичної) для обчислення податкових пільг. Правила податкової пільги зберігаються й відображаються в дуже гнучкому модульному дизайні системи, стандартизованому для кожної країни окремо; Ці правила виражені або як грошові, або як негрошові еквіваленти, що використовуються як параметри функцій *EUROMOD*; функції виступають складниками мови моделювання податкової пільги *EUROMOD*. Код проєкту написаний мовою C ++. Під час запуску платформи код C++ зчитує правила податкової пільги, що зберігаються в інтерфейсі користувача.

*Shyster* [68, 69] – правова система, яка основі ґрунтується на досвіді прецедентного права. Функціональні можливості експертної системи можна розділити на дві частини: 1) юридична пошукова система; 2) система юридичного аналізу.

Важливою особливістю цієї системи є можливість надавати певні юридичні висновки й аргументацію для адвоката, користувача системи. Закони представлено у вигляді спеціалізованої мови специфікації. Основна ідея базується на прецедентному описі юридичних випадків, тобто в системі існує база знань, і система розглядає наявні випадки й на основі старих суджень пробує передбачити результат нової справи. Як бачимо, механізм виводу базується на аргументації наявних результатів справ.

*TAXADVISER* [72] є експертною системою, призначеною для процесу оподаткування. Ця система ґрунтується на певних математичних функціях і

стратегіях, які вже сформулював експерт-юрист на основі юридичних матеріалів. Комп'ютер здійснює умовиводи, але, по суті, розраховує результати на основі стратегій, уже сформульованих експертом, який сам інтерпретував юридичні матеріали. Система розраховує стратегії податкового планування для великих земельних ділянок, засновані на стратегіях, отриманих від юристів, які мають досвід податкової консультації. Цю систему можна використовувати як інструмент моделювання й аналізування у фінансовому плануванні, наприклад, у прогнозуванні пенсій на основі вислуги років.

Система *C-clips* [73], базуючись на принципі представлення знань у вигляді продукційної моделі, є системою логічного виводу на основі фактів, представлених у вигляді *if – «умова» і then – «дія»*.

*TAXAMAN-I* і *TAXAMAN-II* [74], створені на основі логічного програмування, є одними з найпотужніших систем такого типу. В архітектурі цих експертних систем реалізовано механізм логічного виводу. Спеціалізуються вони виключно на податковому праві Великобританії.

Правова система *SARA* [75] слугує юристам для аналізування деталей злочинів, ґрунтується на фреймовій моделі знань. Принцип роботи наступному цієї системи можна описати так: юрист указує фактор і його значення, після чого система приписує ваги кожному факторові, а далі представляє фактори й рішення у вигляді фреймів.

*LEXIS* [75-76], *SCALE* [76], *INFO* [76] зазвичай використовують функції пошуку ключових слів, але такий підхід має очевидний недолік. Вважається, що пошук за змістом документів, індексованих термінами юридичних понять, був би ефективнішим, ніж пошук за ключовими словами.

Використання різноманітних методів і підходів для індексації текстової правової інформації може бути більш ефективним. У пошукових правових системах широко застосовуються дерева дискримінації та методи узагальнення на основі понять.

## Висновки до першого розділу

Найпопулярнішим підходом до моделювання правових вимог є теорія онтологій, що надає змогу моделювати структуру юридичного документа, часткового аналізувати документи щодо відсутності протиріч і цілісності. До недоліків зараховуємо відсутність сценарного підходу та складність управління правовою онтологією.

Іншим досить ефективним підходом до моделювання правових вимог є використання формальної мови *UML* та *OCL*-розширення. Проте й він має недоліки: відсутність сценарного підходу й перевірка тільки статичних вимог.

У фреймворці *Legal-URN Framework* відсутня можливість автоматичного аналізу нормативно-правового документа.

*XML*-підхід використовується як стандарт обміну юридичними даними. Крім того, він уможлиблює ефективне аналізування перехресних посилань у юридичному документі.

Описані вище підходи певною мірою вирішують проблему моделювання нормативно-правових актів, проте ефективно перевірити цілісність законодавчої бази за їхньою допомогою неможливо, що свідчить про відкритість цього підходу.



## Список джерел до Розділу 1

- [1] “«Засуджений» ПДВ або Прецедент на користь платника податку”, *Консультант бухгалтера*, № 16 (400), Квіт. 16, 2007. [Електронний ресурс].  
Доступно: <http://cons.parus.ua/d.asp?r=03WMGd7e3c4816c93692e4d5ba8e0b8234dec>.  
Дата звернення: Січ. 10, 2018.
- [2] *Skillonomy* (сайт). [Электронный ресурс]. Доступно: <https://skillonomy.org/ru/>. Дата обращения: Янв. 10, 2018.
- [3] C. Wass et al., “OpenLaws. eu”, in *Proc. 16th Int. Legal Informatics Symposium (IRIS '13)*, Feb. 2013, pp. 209-211.
- [4] R. Winkels, "The openlaws project: Big open legal data", *Proc. 18th Int. Legal Informatics Symposium (IRIS '15)*, Feb. 2015, pp. 189-196.
- [5] RDF: Resource Description Framework, in *W3C: Semantic Web (website)*. [Online]. Available: <https://www.w3.org/RDF/>. Accessed on: May 19, 2018.
- [6] Standarts, in *W3C: Semantic Web (website)*. [Online]. – Available: <https://www.w3.org/standards/>. Accessed on: May 19, 2018.
- [7] J. Pérez, M. Arenas, and C. Gutierrez, “Semantics and complexity of SPARQL”, *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 16, pp. 1-45, Sept. 2009. Doi: 10.1145/1567274.1567278.
- [8] D. F. Barbieri, D. Braga, S. Ceri, E. D. Valle, and M. Grossniklaus, “C-SPARQL: SPARQL for continuous querying”, in *Proc. 18th Int. World Wide Web Conf. (WWW'09)*, Apr. 2009, pp. 1061-1062. Doi: 10.1145/1526709.1526856.
- [9] R. Fikes, P. Hayes, and I. Horrocks, "DQL-a query language for the semantic web", Knowledge Systems Laboratory, Tech. Rep. KSL-02-05, Nov. 2002.
- [10] N3QL – RDF Data Query Language, in *W3C: Semantic Web (website)*. [Online]. Available: <https://www.w3.org/DesignIssues/N3QL>. Accessed on: May 19, 2018.
- [11] RDQL – A Query Language for RDF, in *W3C: Semantic Web (website)*. [Online]. Available: <https://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>. Accessed on: May 19, 2018.
- [12] K. Hailbronner, *EU Immigration and Asylum Law: A commentary*. C. H. Beck/Hart, 2010.

[13] G. Görög, and P. Weisz, “Legal Entity Recognition in an Agglutinating Language and Document Connection Network for EU Legislation and EU/Hungarian Case Law”, *Preprints*, 2019070310, 2019. doi: 10.20944/preprints201907.0310.v1. [Online]. – Available: <https://www.preprints.org/manuscript/201907.0310/v1>. Accessed on: Nov. 12, 2019.

[14] V. Peychev, “XML model for legal documents”, *Problems of Eng. Cybern. and Robot.*, vol. 54, pp. 86–91, 2004.

[15] V. Peychev, “Legal document – a formal model”, *Problems of Eng. Cybern. and Robot.*, vol. 55, pp. 64–70, 2005.

[16] K. E. Petersen, “Experiences with «Lex Dania Live»”, in *From Information to Knowledge: Online Access to Legal Information: Methodologies, Trends and Perspectives*, M. A. Biasiotti, S. Faro, Eds. (Frontiers in Artificial Intelligence and Applications series (FAIA), vol. 236), 2011, p. 69-76.

[17] A. D. Maurer, and U. P. Hostenstein, “La publication électronique des actes législatifs suisses. Elaboration d’un modèle structuré pour les textes de loi fédéraux, cantonaux et communaux: le schéma XML – CHLexML”, in *Bundesamt für Justiz, Mitarbeiterinnen und Mitarbeiter*, B. Schindler, Ed. Basel: Helbing & Lichtenhahn, 2006, ss. 391-401.

[18] eLAW, in *LAWi: World Encyclopedia of Law (website)*. [Online]. Available: <https://lawin.org/eLaw/>. Accessed on: May 19, 2018.

[19] E. Francesconi, “A review of Systems and Projects: Management of Legislative Resources”, in *Legislative XML for the semantic Web*, G. Sartor, M. Palmirani, E. Francesconi, M. A. Biasiotti, Eds. (Law, Governance and Technology Series (LGTS), vol. 4), 2011, pp. 173-188. Doi: 10.1007/97894-007\_1887-6\_10.

[20] A. Boer, R. Hoekstra, and R. Winkels, “<sup>META</sup>Lex: Legislation in XML”, in *Legal Knowl. and Inf. Syst.: JURIX 2002*, T. Bench-Capon, A. Daskalopulu, R. Winkels, Eds. (FAIA, vol. 89), Amsterdam: IOS Press, 2002, pp. 1-10.

[21] A. Boer, R. Winkels, and F. Vitali, “MetaLex XML and the Legal Knowledge Interchange Format”, in *Computable Models of the Law: Languages, Dialogues, Games, Ontologies*, P. Casanovas, G. Sartor, N. Casellas, R. Rubino, Eds. (Lecture Notes in Computer Science series (LNCS), vol. 4884), 2008, pp. 21-41. DOI: 10.1007/978-3-540-85569-9\_2.

[22] Palmirani, M., Governatori, G., Rotolo, A., Tabet, S., Boley, H., & Paschke, A. (2011, November). LegalRuleML: XML-based rules and norms. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web* (pp. 298-312). Springer, Berlin, Heidelberg.

[23] А. А. Никоненко, "Обзор баз знаний онтологического типа", *Искусств. интеллект*, № 4, с. 208-219, 2009.

[24] J. Breuker, and R. J. Hoekstra, "Epistemology and ontology in core ontologies: FOLaw and LRI-Core, two core ontologies for law", in *CEUR Workshop Proc.*, vol. 118, Oct. 2004. [Online]. Available: <http://ceur-ws.org/Vol-118/paper2.pdf>. Accessed on: May 19, 2018.

[25] A. Gangemi, A. Prisco, M.-T. Sagri, G. Steve, and D. Tiscornia, "Some Ontological Tools to Support Legal Regulatory Compliance, with a Case Study", in *Confeder. Int. Conf. "On the Move to Meaningful Internet Systems" (OTM'03 Workshops)*, (LNCS, vol. 2889), Nov. 2003, pp. 607-620. Doi: 10.1007/978-3-540-39962-9\_64.

[26] D. Tiscornia, "The LOIS project: Lexical Ontologies for Legal Information Sharing", in *Proc. V Legislative XML Workshop*, 2007, pp. 189-204.

[27] R. Hoekstra, J. Breuker, M. Bello, and A. Boer, "The LKIF Core Ontology of Basic Legal Concepts", in *Proc. 2nd Workshop LOAIT'07*, June 2007, pp. 43-63.

[28] R. Hoekstra, J. Breuker, M. Bello, and A. Boer, "LKIF Core: Principled Ontology Development for the Legal Domain", *Law, Ontologies and the Semantic Web: Channelling the Legal Inf.Flood*, vol. 188, pp. 21-52, 2009. Doi: 10.3233/978-1-58603-942-4-21.

[29] C. Griffo, "Ufo-l: A Core Ontology of Legal Concepts Built from a Legal Relations Perspective", in *Doctoral Consortium on Knowledge Discovery, Knowledge Engineering and Knowledge Management (DC3K)*, 2015, pp. 13-20.

[30] T. F. Gordon, "Constructing Legal Arguments with Rules in the Legal Knowledge Interchange Format (LKIF)", in *Computable Models of the Law: Languages, Dialogues, Games, Ontologies*, P. Casanovas, G. Sartor, N. Casellas, R. Rubino, Eds. (LNCS, vol. 4884), 2008, pp. 162-184. Doi: 10.1007 / 978-3-540-85569-9\_11.

[31] I. Horrocks, P. F. Patel-Schneider, S. Bechhofer, and D. Tsarkov, “OWL Rules: A Proposal and Prototype Implementation”, *Journal of Web Semantics*, vol. 3, no. 1, pp. 23-40, 2005. Doi: 10.1016/j.websem.2005.05.003.

[32] OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (Second Edition), B. Motik, and B. Parsia, Eds., in *W3C: Semantic Web (website)*, Dec. 2012. [Online]. Available: <https://www.w3.org/2012/pdf/REC-owl2-syntax-20121211.pdf>. Accessed on: May 19, 2018.

[33] M. Horridge, and S. Bechhofer, “The OWL API: a Java API for Working with OWL 2 Ontologies”, in *Proc. 6th Int. Workshop OWL: Experiences and Directions (OWLED’09)*, Oct. 2009.

[34] M. Horridge, and P. F. Patel-Schneider, “Manchester Syntax for OWL 1.1”, in *OWLED’08, 4th Int. Workshop*, 2008. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.159.1433>. Accessed on: May 23, 2019.

[35] N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson, and M. A. Musen, “Creating Semantic Web Contents with Protege-2000”, *IEEE intelligent systems*, vol. 16, iss. 2, pp. 60-71, 2001. Doi: 10.1109/5254.920601.

[36] P. Casanovas, M. Palmirani, S. Peroni, T. van Engers, and F. Vitali, “Semantic Web for the Legal Domain: The next step”, *Semantic Web*, vol. 7, no. 3, pp. 213-227, 2016. DOI: 10.3233/SW-160224.

[37] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, and M. Dean, “SWRL: A Semantic Web Rule Language Combining OWL and RuleML”, in *W3C: Semantic Web (website): Member submission*, May 2004. [Online]. Available: <https://www.w3.org/Submission/SWRL/>. Accessed on: May 23, 2019.

[38] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A practical OWL-DL reasoner”, *Journal of Web Semantics*, vol. 5, iss. 2, pp. 51-53, 2007. Doi: 10.1016/j.websem.2007.03.004.

[39] T. F. Gordon, “Constructing Legal Arguments with Rules in the Legal Knowledge Interchange Format (LKIF)”, in *Computable Models of the Law*. P. Casanovas, G. Sartor, N. Casellas, R. Rubino, Eds. (LNCS, vol. 4884), 2008, pp. 162-184. DOI: 10.1007/978-3-540-85569-9\_11.

[40] A. Boer, R. Winkels, and F. Vitali, “Proposed XML standards for law: Metalex and LKIF”, in *JURIX 2007*, A. R. Lodder, L. Mommers, Eds. (FAIA, vol. 165), Vol. 165, 2007, pp. 19-28.

[41] H. Boley, A. Paschke, and O. Shafiq, “RuleML 1.0: The Overarching Specification of Web Rules”, in *Int. Workshop Rules and Rule Markup Languages for the Semantic Web*, (LNCS, vol. 6403), 2010, pp. 162-178. Doi: 10.1007/978-3-642-16289-3\_15.

[42] П. А. Ломов, и А. Г. Олейник, “Разработка технологии проверки и согласования нормативно-правовой базы на основе онтологий”, *Труды Института системного анализа Российской академии наук*, Т. 63, №. 2, С. 62-69, 2013.

[43] O. Corcho, M. Poveda-Villalón, and A. Gómez-Pérez, “Ontology engineering in the era of linked data”, *Bulletin of the Association for Inf.Science and Technology*, vol. 41, nú. 4, pp. 13-17, 2015. Doi: 10.1002/bult.2015.1720410407.

[44] S. Sharifi, A. Parvizimosaed, D. Amyot, L. Logrippo, and J. Mylopoulos, “Symboleo: Towards a Specification Language for Legal Contracts”, in *28th IEEE Int. Requirements Eng. Conf. (RE)*, 2020, pp. 364-369. Doi: [10.1109/RE48521.2020.00049](https://doi.org/10.1109/RE48521.2020.00049).

[45] A. Parvizimosaed, S. Sharifi, D. Amyot, L. Logrippo, and J. Mylopoulos, “Subcontracting, Assignment, and Substitution for Legal Contracts in Symboleo”, in *9th Conceptual Modeling Int. Conf. (ER'20)*, LNCS, vol. 12400, Nov. 2020, pp. 271-285. Doi: 10.1007/978-3-030-62522-1\_20.

[46] M. Montali, “jREC”, in *marcomontali on Free University of Bozen-Bolzano (website)*, 2016. [Online]. Available: <https://www.inf.unibz.it/~montali/tools.html>. Accessed on: May 23, 2019.

[47] T. M. Van Engers, R. Gerrits, M. Boekenoogen, E. Glassée, and P. Kordelaar, “POWER: Using UML/OCL for Modeling Legislation: an application report”, in *Proc. 8th Int. Conf. on Artificial intelligence and law (ICAAIL'01)*, 2001, pp. 157-167. Doi: 10.1145/383535.383554.

[48] J. Cabot, R. Clarisó, and D. Riera, “Verification of UML/OCL Class Diagrams using Constraint Programming”, in *2008 IEEE Int. Conf. Software Testing Verification and Validation Workshop (ICSTW'08)*, 2008, pp. 73-80. Doi: 10.1109/ICSTW.2008.54.

[49] G. Soltana, E. Fourneret, M. Adedjouma, M. Sabetzadeh, and L. Briand, “Using UML for modeling procedural legal rules: Approach and a study of Luxembourg’s Tax Law”, in *Int. Conf. Model Driven Engineering Languages and Systems (MODELS’14)*, (LNCS, vol. 8767), 2014, pp. 450-466. Doi: 10.1007/978-3-319-11653-2\_28.

[50] S. Ali, L. C. Briand, A. Arcuri, and S. Walawege, “An Industrial Application of Robustness Testing Using Aspect-Oriented Modeling, UML/MARTE, and Search Algorithms”, in *MODELS’11*, (LNCS, vol. 6981), 2011, pp. 108-122. Doi: 10.1007/978-3-642-24485-8\_9.

[51] G. Soltana, N. Sannier, M. Sabetzadeh, and L. C. Briand, “Model-Based Simulation of Legal Policies: Framework, Tool Support, and Validation”, *Softw Syst Model*, vol. 17, pp. 851–883, 2018. Doi: 10.1007/s10270-016-0542-0.

[52] G. Sileno, A. Boer, and T. van Engers, “Legal Knowledge Conveyed by Narratives: Towards a Representational Model”, in *Proc. Int. Workshop on Computational Models of Narrative (CMN’14)*, 2014, vol. 41, pp. 182-191. Doi: 10.4230/OASICS.CMN.2014.182.

[53] S. Ghanavati, “Legal-URN framework for legal compliance of business processes”, PhD Thesis, University of Ottawa, Ottawa, Canada, 2013.

[54] *Основные юридические понятия Уэсли Н. Хофельда*, М. В. Антонов, Науч. ред., А. А. Гайдамакина, Сост. и пер. с англ. СПб, Россия: Алеф-Пресс, 2016, с. 58-91.

[55] S. Ghanavati, D. Amyot, and A. Rifaut, “Legal Goal-Oriented Requirement Language (Legal GRL) for Modeling Regulations”, in *Proc. 6th Int. Workshop on Modeling in Software Engineering (MiSE’14)*, June 2014, pp. 1-6. Doi: 10.1145/2593770.2593780.

[56] D. Amyot et al., “Towards Advanced Goal Model Analysis with jUCMNav”, in *Int. Conf. Advances in Conceptual Modeling (ER’12)*, (LNCS, vol. 7518), 2012, pp. 201-210. Doi: 10.1007/978-3-642-33999-8\_25.

[57] D. Amyot, and G. Mussbacher, “User Requirements Notation: The First Ten Years, The Next Ten Years”, *JSW*, vol. 6, no. 5, pp. 747-768, 2011. Doi: 10.4304/jsw.6.5.747-768.

[58] L. T. van Binsbergen, L.-Ch. Liu, R. van Doesburg, and T. van Engers, “eFLINT: a domain-specific language for executable norm specifications”, in *Proc. 19th ACM SIGPLAN Int. Conf. Generative Programming: Concepts and Experiences (GPCE’20)*, 2020, pp. 124-136. Doi: 10.1145/3425898.3426958.

[59] L.T. McCarty, “A language for legal Discourse I. basic features”, in *Proc. 2nd Int. conf. ICAIL ’89*, May 1989, pp. 180-189. Doi: 10.1145/74014.74037.

[60] D. L. Chen, "Machine learning and the rule of law", in *Law as Data*, M. Livermore and D. Rockmore, Eds. Santa Fe Institute Press, 2019, pp. 433-441. DOI: 10.37911/9781947864085.16.

[61] R. Barros, A. Peres, F. Lorenzi, L. K. Wives, E. Hubert da S. Jaccottet, "Case Law Analysis with Machine Learning in Brazilian Court", in *Recent Trends and Future Technology in Applied Intelligence (IEA/AIE’18)*, (LNCS, vol. 10868), pp. 857-868, 2018. Doi: 10.1007/978-3-319-92058-0\_82.

[62] Možina, M., Žabkar, J., Bench-Capon, T., & Bratko, I. (2005). Argument based machine learning applied to law. *Artificial Intelligence and Law*, 13(1), 53-73.

[63] P. Clark, T. Niblett, The CN2 induction algorithm, *Machine Learning Journal* 4 (3) (1989) 261–283

[64] Blue J Tax Analytics for tax positions [Электронный ресурс]. Доступно <https://www.bluej.com/ca/blue-j-tax>. Дата обращения: 2021.

[65] H. Sutherland, and F. Figari, “EUROMOD: the European Union tax-benefit microsimulation model”, *International journal of microsimulation*, vol. 6 (1), pp. 4-26, Apr. 2013.

[66] C. O'donoghue, M. Baldini, and D. Mantovani, *Modelling the Redistributive Impact of Indirect Taxes in Europe: An Application of EUROMOD*. EUROMOD Working Paper, no. EM7/01, 2004.

[67] R. Taylor, H. Sutherland, and J. Gomulka, “Using POLIMOD to evaluate alternative methods of expenditure imputation”, *Microsimulation Unit Research Note*, MU/RN38, 2001.

[68] J. D. Popple, “SHYSTER: A Pragmatic Legal Expert System”, PhD Thesis, Australian National Univ., Australia, 1993.



[69] T. A. O'callaghan, J. Popple, and E. McCreath, "SHYSTER-MYCIN: a hybrid legal expert system", in *Proc. 9th int. conf. on Artificial intelligence and law*, 2003, pp. 103-104.

[70] *Legal Knowledge and Inf.Systems: JURIX 2007: The Twentieth Annual Conference*, A. R. Lodder and L. Mommers, Eds. IOS Press, Dec. 2007.

[71] A. Paliwala, "An Intellectual Celebration: A Review of the Jurix Legal Knowledge Based Systems Scholarship", in *ICAIL 2000*, vol. 8, no. 4, pp. 317-335, Dec. 2000. Doi: 10.1023/A:1011242531937.

[72] R. H. Michaelsen, "An expert system for federal tax planning", *Expert Systems*, vol. 1, no. 2, pp. 149-167, Oct. 1984. Doi: 10.1111 / j.1468-0394.1984.tb00440.x.

[73] R. M. Wygant, "CLIPS – a powerful development and delivery expert system tool", *Computers & industrial engineering*, vol. 17, iss. 1-4, pp. 546-549, 1989. Doi: 10.1016/0360-8352(89)90121-6.

[74] L. T. McCarty, "Reflections on TAXMAN: An experiment in artificial intelligence and legal reasoning", *Harv. L. Rev.*, vol. 90, no. 5, pp. 837-893, Mar. 1977. Doi: 10.2307/1340132.

[75] R. E. Susskind, "Expert systems in law: A jurisprudential approach to artificial intelligence and legal reasoning", *The modern law review*, vol. 49, no. 2, pp. 168-194, 1986. Doi: 10.1111/j.1468-2230.1986.tb01683.x.

[76] A. Greinke, "Legal expert systems: A humanistic critique of mechanical legal inference", *E Law*, n. 16, Nov. 1994. [Online]. Available: <http://www.murdoch.edu.au/elaw/issues/v1n4/greinke14.txt>. Accessed on: Nov. 25, 2019.



## РОЗДІЛ 2

### ІНСЕРЦІЙНА СЕМАНТИКА ДО АНАЛІЗУ ПРАВОВИХ МОДЕЛЕЙ

#### 2.1. Теоретичні відомості

##### 2.1.1. Агенти й середовища.

Інсерційне моделювання являє собою напрям, що розвивається протягом останнього десятиліття як підхід до побудови загальної теорії взаємодії агентів і середовищ у складних розподілених багатоагентних системах [77]. Основні поняття інсерційного моделювання (середовище, агенти, функція занурення) уведено в працях, опублікованих у 90-х роках попереднього століття [77–80].

Слід зазначити, що в роботах В. Глушкова ще в 60-х роках описано моделі взаємодіючих керуючих та операційних автоматів для опису структур обчислювальних машин, які можна вважати засадничими дослідженнями для інсерційного моделювання.

Іншим джерелом для напряму інсерційного моделювання є загальна теорія взаємодіючих інформаційних процесів, що сформувалась у 70-х роках і слугує основою для сучасних досліджень. Вона охоплює *CCS* (Calculus of Communicated Processes) [81; 82] та  $\pi$ -числення Р. Мілнера [83], *CSP* (Communicated Sequential Processes) Т. Хоара [84], *ACP* (Algebra of Communicated Processes) [85] і багато інших різноманітних гілок цих базових теорій.

Інсерційне програмування – це програмування на базі моделі поведінки агентів у середовищах [86]. Модель ґрунтується на поняттях розміщеної транзиційної системи й відношеннях бісимуляційної еквівалентності.

Агент – це транзиційна система, стан якої визначають із точністю до бісимуляційної еквівалентності [86].

Середовище – це агент, який має функцію занурення. За більш точним визначенням, середовище – це набір  $\langle E, C, A, Ins \rangle$ , де  $E$  – це множина станів середовища,  $C$  – множина дій середовища,  $A$  – множина дій агентів, що занурюються в середовище,  $Ins: E \times F(A) \rightarrow E$  – функція занурення. Тут  $F(A)$  –

повна алгебра поведінки агентів із множиною дій  $A$ . Таким чином, будь-яке середовище  $E$  допускає занурення будь-якого агента з множиною дій  $A$ .

Після занурення одного агента середовище може прийняти іншого; середовище розглядається як агент, що може бути самостійно зануреним в інше середовище вищого рівня. Такі багатоагентні й багаторівневі середовища можуть бути створені за допомогою функцій занурення. Функція занурення сприймає поведінку агента та поведінку середовища як аргументи й повертає нову поведінку цього середовища [87].

### 2.1.2. Алгебра поведінки.

В інсерційному моделюванні вихідним поняттям є поняття атрибутної традиційної системи [88], що формально визначається як п'ятірка:

$\langle S, A, U, T, \varphi \rangle$ , де  $S$  – множина станів,  $A$  – множина дій, що використовуються для розмітки переходів,  $U$  – множина атрибутних розміток, які використовуються для розмітки станів, відношення переходів.

Транзиційні системи можуть характеризуватися виділенням деяких специфічних множин станів із множини станів  $S$ . Слід зазначити, що найважливішими є множини початкових станів,  $S_{\Delta}$  – заключних і  $S_{\perp}$  – невизначених станів.

В інсерційному моделюванні для опису поведінки використовують вирази алгебри поведінки. Алгебра поведінки являє собою двохосновну алгебру

$\langle U, A \rangle$ , першою компонентою є множина  $U$  поведінок, а друга представляє множину дій  $A$ . Сигнатура алгебри поведінки складається з чотирьох операцій, одного відношення та трьох постійних:

1. Першою операцією є префіксинг  $A$ . Її аргументами є дія  $a$  та поведінка  $u$ . Результатом є нова поведінка.
2. Друга операція є операцією недетермінованого вибору  $u+v$ . Ця операція є бінарною, вона визначена на множині поведінок, комутативна, асоціативна й ідемпотентна,  $u, v$  – поведінка.
3. Третя операція – паралельна композиція  $u // v$ , де  $u, v$  – поведінка.

4. Четверта операція – послідовна композиція  $u;v$ , де  $u, v$  – поведінка.
5. Константи алгебри поведінок. До константи алгебри поведінки слід віднести успішне завершення  $\perp$ , невизначена поведінка  $\perp$  та тупикова поведінка  $0$ , яка є нейтральним елементом недетермінованого вибору.

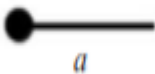
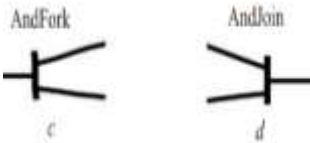
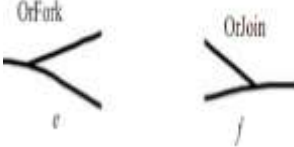
Наголошуємо, що важливу роль частини теорії алгебри поведінки відіграє повна алгебра поведінок  $F(A)$ , у якій певне місце посідає теорема про найменшу нерухому точку. Більш детально інформацію про теорію алгебри поведінки викладено в [89].

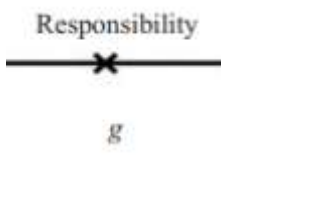
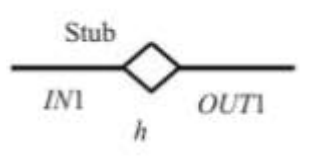
Алгебру поведінки можна задати за допомогою *UCM*-нотацій. Мову *UCM Live* [90] використовують для специфікації багатопоточних програм, заснованій на теорії моделювання бізнес-процесів.

Розглянемо семантику *UCM*-вершин, які застосовуються для опису поведінки в інсерційних моделях:

Таблиця 2.1

### UCM-семантика

№	Назва	UCM-нотація	Алгебра поведінки
1	<i>StartPoint</i> – початок шляху		$StartPoint = b1 \dots$
2	<i>AndFork</i> – паралельне розгалуження, <i>AndJoin</i> – з'єднання альтернатив зі синхронізацією		$StartPoint = b1.S1$ $// b2.S2$
3	<i>OrFork</i> – альтернативне розгалуження <i>OrJoin</i> – з'єднання альтернатив без синхронізації		$StartPoint = b1.S1$ $+ b2.S2$

4	<i>Responsibility</i> – дія		<i>StartPoint = b1.S1</i>
5	<i>Стаб</i>		

- *WaitingPlace* – очікування;
- *EmptyPoint* – порожній конструктор;
- *Timer* – таймер.

Кожен шлях має початкову вершину, їх може бути декілька, причому з початкової вершини шляхи виходять, а в кінцеву входить тільки один шлях. Шляхи можуть розгалужуватися й зливатися; існує два види розгалуження: паралельне та альтернативне. У паралельному розгалуженні є один вхідний шлях і кілька вихідних. Паралельне злиття може мати кілька вхідних шляхів, але тільки один вихідний. В альтернативних розгалуженнях (OrFork) та злиттях (OrJoin) така сама структура, як і в паралельних (розгалуження має один вхідний шлях та кілька вихідних, злиття – кілька вхідних і один вихідний шлях). Альтернативні розгалуження й злиття представлені точками на шляхах .

Крім розгалужень і злиттів, на шляхах є два види символів: зобов'язання, позначене хрестиком (*Responsibility*), і стаб, позначений ромбом (*Stub*). Символи зобов'язань мають по одному вхідному й вихідному шляху, символ стабу може мати декілька вхідних і вихідних шляхів. Із кожним зобов'язанням асоційований деякий локальний опис системи локальних описів, які зазвичай відносяться до базової системи, хоча в деяких випадках – і до управління.

### 2.1.3. Мова дій.

Мову дій застосовують для представлення семантики інсерційних моделей. Загальна теорія мови дій представлено в роботі [91]. На основі методу мови дій

розроблено методи верифікації вимог і специфікацій розподілених систем у галузі телекомунікацій, вбудованих систем та систем реального часу (див. праці [92, 83]). Специфікована система є середовищем із зануреними в неї агентами. Тому атрибутна розмітка станів середовища охоплює не тільки інформацію про стан середовища, але також інформацію про стани агентів.

Кожна дія являє собою вираз виду  $\forall x(a \rightarrow \langle u \rangle \beta)$  де  $x$  – список (типізованих) параметрів,  $a$  і  $\beta$  – формули базової мови,  $u$  – процес дії (цілком визначена поведінка). Формула  $a$  називається передумовою, а формула  $\beta$  – постумовою дії. Базовою мовою, яку застосовують у перед- і постумовах, є мова числення предикатів першого порядку [92]. Дія може розглядатися як формула темпоральної логіки, що виражає такий факт: якщо (для відповідних значень параметрів) стан системи має розмітку, яка задовольняє умову  $a$ , то процес  $u$  може бути ініційований, а після його завершення розмітка буде задовольняти умову  $\beta$ . Систему, яка задовольняє цю умову, називають реалізацією.

Специфікація системи на основі мови дій складається з опису середовища та множини дій, що виражають певні властивості поведінки системи в певних станах, які задовольняються передумовами. Для кожної системи  $A$  дій як її реалізація буде визначена система  $S(P)$ , породжена цими діями. Система є атрибутною системою, що використовує формули мови дій як розмітки станів.

Поведінка системи  $S(P)$  визначається за допомогою спеціальної композиції поведінок системи. Цю композицію називають частково послідовною композицією і застосовують рекурсивно до процесів мови дій. Для визначення частково-послідовної композиції застосовують відношення перестановки на множині дій системи  $S(P)$ . У разі перестановки всіх дій їхня композиція вироджується в паралельну, а в разі, коли жодна дія не перестановочна, ідеться про послідовну композицію процесів [93].

Для обчислення функції атрибувної розмітки станів використовують предикатний трансформер – перетворення, визначене на формулах базової мови. Умову, що характеризує розмітку стану до застосованої дії, і його постумову предикатний трансформер перетворює на нову розмітку (трансформовану

постумовою).

#### 2.1.4. Система IMS.

У перших роботах, присвячених інсерційному моделюванню, було запропоновано модель взаємодії агентів і середовищ, засновану на понятті функції занурення й алгебрі поведінки [78–80], теорію взаємодії агентів і середовищ розвинено в праці О. Летичевського й Д. Гілберта [78]. Базові принципи інверсійного моделювання представлено на рис. 2.1.

Основним компонентом є *model driver* – компонент, що контролює рух уздовж дерева поведінки моделі. Стан моделі представляється у вигляді входної мови інсерційного моделювання й розглядається як алгебраїчний вираз.

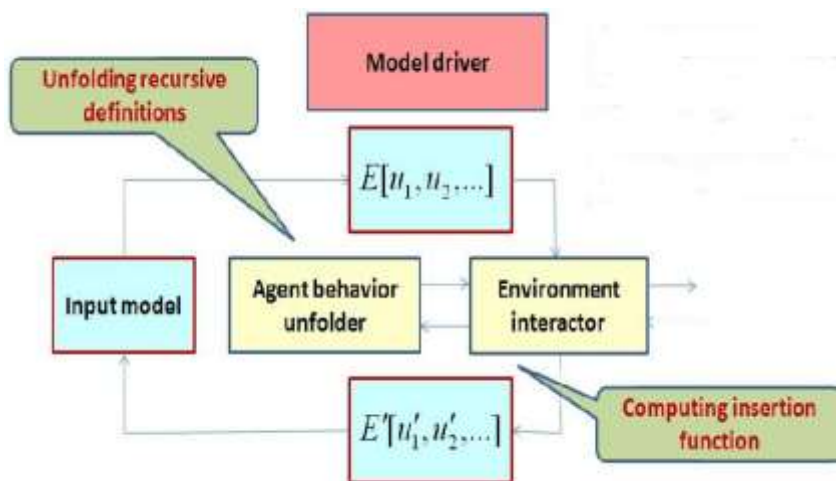


Рис. 2.1. Базові принципи інсерційного моделювання.

Перед застосуванням функції занурення система повинна бути зведена до стану  $E[u_1, u_2, \dots, u_n]$ , функція зведення виконується модулем *agent behavior unfold*. Для здійснення руху стан середовища повинен бути зведений до нормальної форми  $\sum_{i \in I} a_i * E_i + \varepsilon$ , де  $E_i$  – стан середовища,  $\varepsilon$  – термінальна константа, дію виконує модуль *environment interactor* [93].

Поняття середовища формалізується у вигляді агента, який, крім своєї поведінки, має також функцію занурення. Ця функція визначає зміну поведінки середовища внаслідок занурення агента. Після занурення одного агента середовище готове до занурення інших агентів, але й саме може розглядатися як агент і може бути зануреним у середовище більш високого рівня. Оскільки

базовий синтаксис інсерційного моделювання змінився (наразі це *JAVA*-подібний синтаксис), розглянемо його особливості на прикладі реалізації моделі.

Для опису агентів і середовища використовують глобальні атрибути середовища та атрибути агентів. Глобальні атрибути середовища можуть приймати значення таких типів даних: *bool*, *real*, *int*, перелічного типу, типу агента та неінтерпретовані функції (далі за текстом – функціональний тип).

Розглянемо приклад реалізації глобальних атрибутів середовища в системі інсерційного моделювання:

*currentStage:int* – змінна цілочислового типу.

*stageAVG:(int)->int* – зміна функціонального типу, яка приймає параметри цілочислового типу й повертає значення типу *int*.

Крім глобальних атрибутів середовища, є атрибути агентів, які описують поведінку агентів. Атрибути агентів можуть приймати значення таких типів: *bool*, *real*, *int*, перелічного типу, неінтерпретованих функцій (функціональний тип), а також змінну типу агент. Розглянемо фрагмент реалізації агента засобами інсерційного моделювання:

```
Subject:obj(
    taxPayer:bool,
    sale:(int)->real,
    ...
    commercialActivity:obj (
        status:TENSE,
        ....
    )
)
```

- *taxPayer* – змінна типу *bool*,
- *sale* – функціональний атрибут, що приймає як параметр змінну типу *int* та повертає змінну типу *real*.

- *commercialActivity* – атрибут типу агент, який має атрибут *status* перелічного типу *TENSE*.

Для визначення типу *TENSE* потрібно в розділі *types* визначити елементи перелічного типу:

```
types: obj(
    TENSE:(GONNA_BE, DOES, DID, DOES_NOT),
    PRESENT:(PERSON, DIRECTOR, REPRESENTER, OTHER)
);
```

Початковий стан задається у вигляді логічної формули, яка в процесі моделювання може змінювати свої значення. Розглянемо приклад специфікації такого початкового стану:

```
currentStage == 1 &&
subject.taxPayer == false &&
subject.commercialActivity.status == DOES_NOT &&
Forall (i:int)( (0<i<=10) -> stageAVG(i) == 0)
```

Оскільки початковий стан задається за допомогою логічної формули, кожна змінна задається через операцію кон'юнкції. Зауважимо, що змінна функціонального типу може повертати значення типів агентів (нижче наведено приклад):

```
ListOfAgents:(int)->Subject,
```

Використовуючи квантор *Forall*, можливо задати значення атрибутів для декількох агентів:

```
Forall(i:int)( (0<=i<=10) ->
    (ListOfAgents(i).taxPayer == false) &&
    ListOfAgents(i).commercialActivity.status == DOES_NOT)
)
```

На відміну від формули початкового стану, аксіоми не змінюються в процесі моделювання, для реалізації аксіом ми використовуємо логіку мови *IMS*. Наведемо приклад такої аксіоми:

```
VIOLATION_CONDITION_3 == (
    !((x.registrationVAT.presentApplication == PERSON) //
```



```

(x.registrationVAT.presentApplication == DIRECTOR) //
(x.registrationVAT.presentApplication == REPRESENTER))
)

```

Розглянемо детальніше реалізацію цієї аксіоми. Агент  $x$  типу *Subject* має атрибут типу агента *registrationVAT*, який, зі свого боку, має атрибут *presentApplication* перелічного типу *PRESENT*. Аксіома буде істинна, коли буде істинною умова цієї аксіоми.

### 2.1.5. Символьне моделювання.

Символьне моделювання ґрунтується на двох теоретичних складниках: теорії інсерційного моделювання [92–94] і теорії предикатних перетворювачів [95; 96], яка, власне, є основою символічного моделювання системи інсерційного моделювання. Перша теорія визначає методи взаємодії агентів і середовища, де агент розглядається як розмічена транзиційна система з певною множиною дій. Поведінка агента визначається функцією занурення в середовище, яка представляє зміну символічних станів агента. Основною задачею предикатного перетворювача є побудова транзиційної системи для мови дій. Предикатний перетворювач є функцією, аргументами й значенням якої виступають формули в певній теорії. Сформульовано теорію предикатних перетворювачів для цілочислових, перелічних типів даних і логіки першого порядку з універсальним квантором [98]. Узагалі, предикатний перетворювач – це формула, що визначає новий символічний стан середовища й агента. Кожна формула базової мови відповідає безлічі конкретних станів, за яких вона істинна. Тому функція занурення для символічного середовища повинна бути узгодженою з функцією занурення відповідної функції для конкретної середовища. Ця узгодженість визначається в термінах предикатного перетворювача, який сам по собі визначає переходи як у конкретному, так і в символічно-атрибутному середовищі. Більш детально з теорією предикатних перетворювачів можна ознайомитися у працях О.Летичевського[95,97], О. Летичевського (мол) [95,97] В. Песчаненка[95, 97], О. Годлевського[95, 96, 97]

Узагальнене завдання символного моделювання полягає в тому, щоб досліджувати властивості конкретних систем на їхніх символних моделях. Між станом символних і конкретних систем визначено відношення, яке розглядається як відношення покриття. Символьна модель середовища виступає як певна абстракція конкретної моделі, а її стан – як абстракція конкретних станів[98].

Основна відмінність конкретних моделей від символних полягає в детермінованій поведінці. Це означає, що після застосування дії перетворений стан середовища буде визначатися однозначно. Методи символного моделювання дозволяють працювати з безліччю формул, які покривають множину значень атрибутів системи, а це дозволяє досягти більш повного покриття станів.

Одним із прикладів використання символних методів є перевірка властивостей безпеки в задачах, що критичні до безпеки. Типовими представниками таких задач є «інтерлокінг» залізничних колій. У системі символного моделювання властивість безпеки в цій задачі перевіряється за допомогою оберненого символного моделювання від символного стану, що визначає аварійну ситуацію.

Ще однією класичною задачею символного моделювання є задача про підключення великої кількості абонентів до телефонної мережі. Цей приклад демонструє ефективність застосування символного моделювання в задачах із великою кількістю агентів. Використання символного моделювання, зокрема й у цьому випадку, дає змогу краще проаналізувати систему за рахунок більшої кількості досліджуваних станів системи, поданих у символному вигляді.

На відміну від конкретного моделювання, символне відкриває для нас можливість перевірити надійність і стабільність моделі. Символьне моделювання – це ефективний підхід до аналізу формальних моделей, який дозволяє обійти максимальну кількість станів моделі, на відміну від моделей із конкретними значеннями. Крім того, використання символних методів у тестуванні сприяє ефективності в процесі верифікації систем, критичних до безпеки.

## **2.2. Використання інсерційного моделювання для аналізування правових моделей**

### **2.2.1. Агенти й середовища в правовій моделі.**

Створений 2010 року Податковий кодекс України зазнає суттєвих змін протягом усього свого існування. Реально оцінити цілісність правових актів у цьому кодексі дуже складно, не говорячи вже про зіставлення з іншими нормативно-правовими документами. Податковий кодекс України [99] має суттєві недоліки: невідповідність між статтями та чинні закони, які в різних правових ситуаціях можна трактувати по-різному. Досить непросто ситуацію додатково ускладнює наявність великої кількості перехресних посилань між нормативно-правовими й додатковими актами.

Метою описуваного підходу є застосування методів інсерційного моделювання для доведення властивостей формальних моделей законодавчих актів України, а також проведення формальної експертизи законопроектів України та Європейського Союзу щодо можливих правових колізій у процесі взаємодії. Такий підхід може бути спрямований на усунення й розв'язання методами символічного інсерційного моделювання логічних, правових, математичних задач, що виникають під час верифікації формальних моделей.

Суб'єкти в податковому кодексі можна розглядати як агентів, що взаємодіють між собою в межах нормативно-правових актів. Кожен агент – це певна формальна сутність, яка має атрибути, властивості, представлені формулами базової мови, та поведінку, що визначається нормативно-правовими документами. Усі агенти взаємодіють зі середовищем, у яке вони занурені. У межах системи інсерційного моделювання ми можемо розглянути багаторівневу модель взаємодії, де навколишнє середовище виступає як агент, що взаємодіє з агентом вищого рівня абстракції.

У Податковому кодексі ми маємо два види агентів – платника податків і податкову службу, з якою той взаємодіє. Платник податків функціонує та виконує певні види діяльності: платежі, нарахування податків, отримання фінансів на

рахунок тощо. Податкова служба забезпечує перевірку правильності оподаткування й накладання штрафів відповідно до Податкового кодексу.

На першому етапі слід визначити агентів та їхні атрибути. Платник податків може бути представлений у системі інсерційного моделювання таким чином:

```
TaxPayer : obj(
  commercialActivity:obj(
    status:TENSE,
    totalSalesSumTaxable: (int, int) -> int,
    sale: (int) -> int
  ),
  registrationVAT:obj(
    status:TENSE,
    decisionVoluntaryVAT:bool,
    necessity:bool,
    presentApplication:PRESENT
  )
  ....

  taxVATPayer:bool,
  limitVAT:real
)
```

Розглянемо більш детально деякі атрибути агента *TaxPayer*:

1. *commercialActivity* – атрибут, що визначає комерційну діяльність і, зі свого боку, визначається іншими атрибутами: *status*, тип якого *TENSE* – перелічний тип, визначення якого описано нижче, та атрибут

2. *totalSalesSumTaxable:(int, int) -> int* – функціональний атрибут, що приймає два параметри типу *int* і повертає значення змінної типу *int*, а також функціональний атрибут *sale*, який приймає як параметр змінну типу *int* та повертає також значення типу *int*.

3. *taxVATPayer* – атрибут логічного типу, що приймає істинне значення, коли агент є платником податків.

4. *limitVAT* – атрибут типу *real*, який визначає загальну суму від здійснення операцій із постачання товарів/послуг.

Розглянемо фрагмент закону, що визначає платника податку: «Особа, що здійснює або планує здійснювати господарську діяльність та реєструється за своїм добровільним рішенням як платник цього податку». Із цього фрагмента можна виокремити положення, що особа планує або проводить господарську діяльність; уведемо перелічний тип *TENSE*, який буде відображати статус проведення господарської діяльності особою.

*types: obj(*

*TENSE:(GONNA\_BE, DOES, DID, DOES\_NOT),*

*RESPONSIBILITY:(TAX\_PAYMENT,KEEP\_TAXFREE\_REQUIREMENTS,*  
*OTHER)*

*);*

Агенти взаємодіють між собою в межах правил, які виражені нормативно-правовими актами Податкового кодексу України, і змінюють свої стани, представлені його властивостями. Початковий стан агентів у процесі моделювання може змінюватися. Далі розглянемо, як можна задати початковий стан:

```
limitVAT == 1000000 &&
```

```
TaxPayer.VATpayer == false &&
```

```
TaxPayer.registrationVAT.status == DOES_NOT
```

Як указано вище, в агента *TaxPayer* є атрибут *VATpayer* логічного типу, який визначає, чи є цей агент платником податку. Наразі ми визначаємо в початковому стані, що агент не є платником ПДВ. *limitVAT* – глобальний атрибут середовища, початковий стан якого 1000000; ця величина визначає загальну суму від здійснення операцій із постачання товарів/послуг.

Надалі опрацюємо певні тексти нормативно-правових актів та схарактеризуємо специфіку їхньої формалізації. Деякі закони можуть бути специфіковані у вигляді аксіом, деякі – у вигляді специфікації мови дій.

Розглянемо фрагмент тексту закону (стаття 180.1, пункт 1): «*Будь-яка особа, що провадить або планує провадити господарську діяльність і реєструється за своїм добровільним рішенням як платник податку у порядку, визначеному статтею 183 цього розділу*» [99].

Оскільки в тексті закону немає факту дії, то формалізувати його потрібно у вигляді аксіоми. Спочатку представимо текст як конструкцію *if then*, такий підхід спростить процес формалізації та полегшить розуміння нормативно-правової конструкції.

A180\_1\_1: *If* ((особа проводить господарську діяльність) *або* (особа планує проводити господарську діяльність)) *та* (особа реєструється як платник податку) *і* ( особа реєструється за своїм добровільним рішенням) *then* Платник податку.

Після того як конструкція *if* сформована, слід замінити сполучники “*та*”, “*або*” на еквівалентні логічні оператори. З урахуванням розробленого формалізму текст статті 180, пункту 1, можна представити у вигляді аксіоми таким чином:

```
A180_1_1 <=> (
    ((x.commercialActivity.status == GONNA_BE) //
    (x.commercialActivity.status == DOES)) &&
    (x.registrationVAT.status == DOES) &&
    x.registrationVAT.decisionVoluntaryVAT)
)
```

Розглянемо приклад статті, яку можна представити у вигляді аксіом (стаття 180.1, пункт 4): «Особа, що веде облік результатів діяльності за договором про спільну діяльність без створення юридичної особи»;

```
A180_1_4 <=> (x.otherActivity == JOINT_ACTIVITY_CONTRACT)
```

Стаття 180 складається з підпунктів, які специфікуються у вигляді аксіом, тоді повністю цю статтю можна представити так:

```
(x.taxVATPayer <=> (
```

```

A180_1_1 // A180_1_2 // A180_1_3 & A180_1_3_3
// (A180_1_3 & (A180_1_3_1 // A180_1_3_2) & A180_1_3_3) //
A180_1_4 // A180_1_5))

```

)

*A180\_1\_1*, *A180\_1\_2* .. – це змінні логічного типу, які приймають істинне значення, коли відповідні аксіоми повертають істинне значення. Тоді *x.taxVATPayer* буде приймати істинне значення, коли вся логічна конструкція буде істинною.

### 2.2.2. Формалізація нормативно-правових актів за допомогою мови дій.

Частина законів потрібно специфікувати за допомогою мови дій. Наголошуємо: якщо в тексті нормативно-правового акту є факт певної дії, то такий документ специфікується у вигляді мови дій. Розглянемо фрагмент тексту закону й відповідну специфікацію за допомогою мови дій.

Стаття 181.1: «У разі якщо загальна сума від здійснення операцій з постачання товарів/послуг, що підлягають оподаткуванню згідно з цим розділом, у тому числі з використанням локальної або глобальної комп'ютерної мережі, нарахована (сплачена) такій особі протягом останніх 12 календарних місяців, сукупно перевищує 1000000 гривень (без урахування податку на додану вартість), така особа зобов'язана зареєструватися як платник податку у контролюючому органі за своїм місцезнаходженням (місцем проживання) з дотриманням вимог, передбачених статтею 183 цього Кодексу, крім особи, яка є платником єдиного податку першої-третьої групи»[99]

```

BP181_1=(Forall(i:int)( ((x.totalVendorSumTaxable(timeActual.month - 11 + i,
timeActual.month)  >=  limitVAT)&&(i  >=  0)  &  (i  <=  11)  &&
!(x.singleTaxGroup1)&&  !(x.singleTaxGroup2)  &&  !(x.singleTaxGroup3)  &&
((x.importActivity.status  !=  DOES)  &&
((x.importActivity.totalImportSumTaxable(timeActual.month  -  11  +
i,timeActual.month)>=limitVATimport))) -> ("181_1")
((MOVE_DATE(x.timeVATstart,timeActual);

```

*x.registrationVAT.necessity))),*

Опишемо більше детально алгебраїчне представлення:

1. *totalVendorSumTaxable* – функціональний атрибут, який приймає два параметри типу *int* та повертає значення типу *int*, що визначає загальну суму, яка підлягає оподаткуванню у відповідний період.

Умова *x.totalVendorSumTaxable(timeActual.month - 11 + i, timeActual.month) >= limitVAT)* визначає, що сума, яка підлягає оподаткуванню в поточний період, повинна перевищувати загальну суму від здійснення операцій із постачання товарів або послуг.

2. *totalImportSumTaxable* – функціональний атрибут, що визначає загальну суму імпорту, яка підлягає оподаткуванню в певний період.

Згідно з законом, сума, що підлягає оподаткуванню, повинна перевищувати загальну суму від здійснення операцій із постачання товарів/послуг на митну територію України. Цей факт можна представити наступною специфікацією:

*x.importActivity.totalImportSumTaxable(timeActual.month - 11 + i, timeActual.month) >= limitVATimport*

Крім того, *x.importActivity.status == DOES* відображає той факт, що агент проводить господарську діяльність і не належить до жодної з груп платників єдиного податку: *!(x.singleTaxGroup1) && !(x.singleTaxGroup2) && !(x.singleTaxGroup3)*

Більш детально з особливостями формалізації можна ознайомитися в роботах [100, 101].

Розглянемо приклад статті закону, що можна специфікувати за допомогою мови дій.

Стаття 183.8: «Контролюючий орган відмовляє в реєстрації особи як платника податку, якщо за результатами розгляду реєстраційної заяви та/або поданих документів встановлено, що особа не відповідає вимогам, визначеним статтею 180, пунктом 181.1 статті 181, пунктом 182.1 статті 182 та пунктом 183.7 статті 183 цього Кодексу, або якщо існують обставини, які є підставою для анулювання реєстрації згідно із статтею 184 цього Кодексу, а також якщо при



поданні реєстраційної заяви чи визначенні бажаного (запланованого) дня реєстрації не дотримано порядок та строки (терміни), встановлені пунктами 183.1, 183.3-183.7 цієї статті».

```
BP183_8=((VIOLATION_CONDITION_1 // VIOLATION_CONDITION_2 //
VIOLATION_CONDITION_3 // !(x.taxVATPayer)) ->
("183.8 Cancelation of registration application") I),
```

У передумові цієї специфікації використовують три змінні логічного типу, які приймають істинне значення, коли особа не відповідає вимогам, визначеним відповідним законом – *VIOLATION\_CONDITION*.

*!x.taxVATPayer* – визначає, що агент не є платником податку.

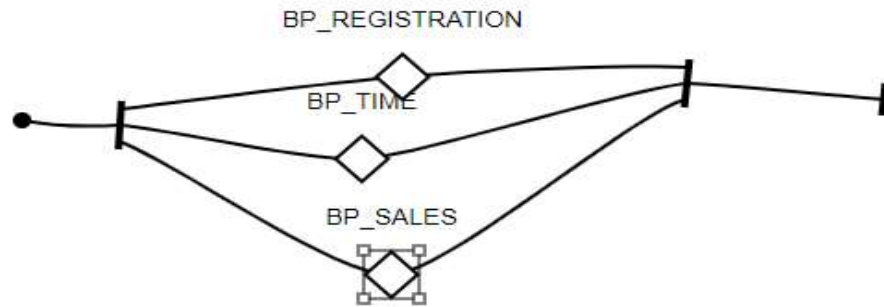
Важливо зазначити, що *VIOLATION\_CONDITION* реалізовано як аксіоми, тоді ці змінні будуть приймати істинне значення, коли аксіоми будуть істинні. Розглянемо одну з таких аксіом, яка описує вимогу, визначену пунктом 183.7 статті 183:

```
VIOLATION_CONDITION_3 == (((x.registrationVAT.presentApplication !=
PERSON) // (x.registrationVAT.presentApplication == DIRECTOR) //
(x.registrationVAT.presentApplication == REPRESENTER)))
```

Атрибут *presentApplication* має перелічний тип *PRESENT:(PERSON, DIRECTOR, REPRESENTER, OTHER)* і специфікує, ким була відправлена заява на реєстрацію платника податку. Зрозуміло, що вимога буде істинною, коли заява була відправлена не фізичною особою, або не керівником, або не представником юридичної особи.

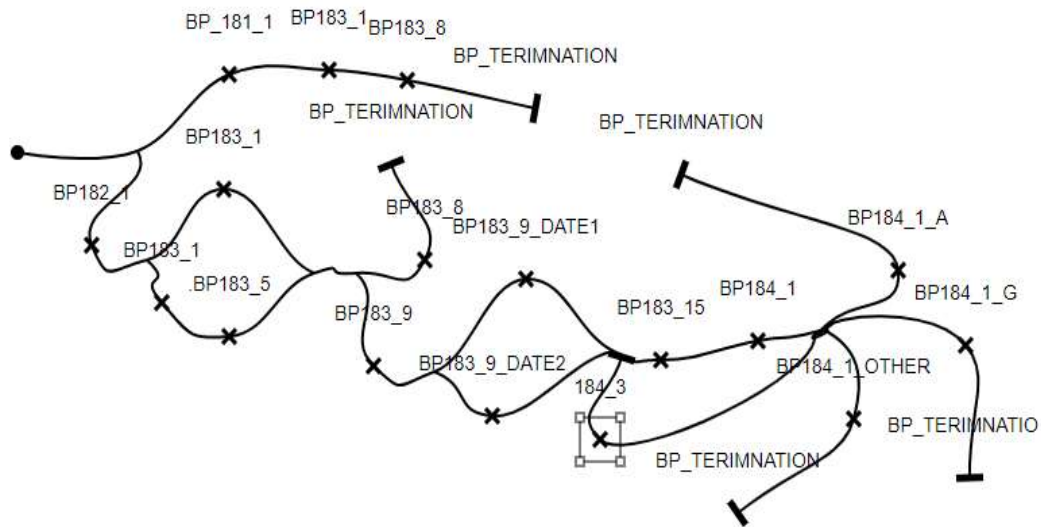
### 2.2.3. Алгебра поведінки.

На найвищому рівні абстракції процедуру реєстрації платника податків ПДВ можна представити за допомогою *UCM*-діаграми у вигляді паралельної композиції стабів, що проілюстровано на рис.2.2. Кожний стаб є певною послідовністю дій, які визначають поведінку агента. Так, стаб *BP\_REGISTRATION* – послідовність дій під час реєстрації агента як платника ПДВ.



**Рис. 2.2.** UCM-реєстрація платника податків

Нижче розглянемо більш детально UCM-стаб процедури реєстрації та анулювання платника податку (див. рис 2.3).



**Рис. 2.3.** Стаб Registration. Реєстрація платника ПДВ та  
Процедура анулювання з Податкового кодексу

Реєстрація платника податків починається з альтернативних гілок *BP\_181\_1.B1\_NEC* + *BP182\_1.B1\_VOL*. Розглянемо більш детально другу гілку альтернативи.

Дія *BP\_181\_1* – вимога щодо реєстрації платника податку, специфікація цього закону за допомогою мови дій була представлена в попередньому пункті або у відповідному розділі додатків. Якщо ця дія буде застосована, то алгебра дій приведе нас до наступної дії *BP183\_1*.

Дія *BP183\_1* – вимога щодо добровільного подання реєстраційної заяви до контролюючого органу, після чого ця дія переводить нас до *BP183\_8* –

специфікації підстав про відмову в реєстрації. Після відмови в реєстрації ми потрапляємо в термінальний стан. Цю гілку можна представити таким чином:

$$BP\_REGISTRATION = BP\_181\_1. BP183\_1. BP183\_8. BP\_TERIMNATION +$$

...

Розглянемо першу гілку альтернативи в реєстрації платника податку.

Дія *BP182\_1* – вимога щодо потреби реєстрації особою, яка відповідає вимогам пункту 181.1 статті 181 цього Кодексу і не є платником податку. Ця дія переводить до дії *BP183\_1*, яка стверджує про обов’язкову реєстрацію. Відповідно до специфікації *BP183\_5* особа повинна вказати дату реєстрації, яка відповідатиме даті початку податкового періоду (календарний місяць), відповідно до *BP183\_8*, якщо є підстави для відмови в реєстрації, потрапляємо знову до термінального стану. Тоді відповідну гілку представляємо так:

$$BP\_REGISTRATION = \dots + BP182\_1.BP183\_1.BP183\_5.B2\_VOL + BP183\_1.B2\_VOL,$$

$$B2\_VOL = BP183\_8.B\_TERMINATION + BP183\_9.B3\_VOL,$$

Схарактеризуємо альтернативний шлях *BP183\_9.B3\_VOL*:

Дія *BP183\_9* специфікує умову, яка стверджує про відсутність підстав для відмови в реєстрації платника податку. Контролюючий орган зобов’язаний протягом трьох робочих днів після надходження реєстраційної заяви внести до реєстру платників податку запис про реєстрацію такої особи як платника податку. Дія *BP183\_9* переводить до  $B3\_VOL = BP183\_9\_DATE1.B2 + BP183\_9\_DATE2.B2$ , що відповідає бажаній даті реєстрації платника податку:

- *BP183\_9\_DATE1* – із бажаного (запланованого) дня реєстрації, зазначеного в реєстраційній заяві, відповідає даті початку податкового періоду (календарного місяця),
- *BP183\_9\_DATE2* – із першого числа місяця, наступного за днем спливу 10 календарних днів після подання реєстраційної заяви до контролюючого органу.

Далі обидві дії переводять до *BP183\_15* – дії, що специфікує змінення платником податку місцезнаходження (місця проживання) або його переведення на обслуговування до іншого контролюючого органу.

Поведінка *B4* представляє анулювання реєстрації платника податку, альтернативні гілки – вибір дати анулювання реєстрації платника податку.

$$B4 = BP184_1\_A.BP\_TERMINATION + BP184_1\_G.BP\_TERMINATION + BP184_1\_OTHER.BP\_TERMINATION + 184\_3,$$

Дія *BP184\_3* специфікує анулювання реєстрацію особи як платника податку, що подала заяву про анулювання реєстрації.

Представимо реєстрацію ПДВ, зображену на рис. 2.2, за допомогою алгебри поведінки:

$$B0 = BP\_TIME // BP\_REGISTRATION // BP\_SALES,$$

$$BP\_TIME = BP\_NEXT\_TIME.BP\_TIME + BP\_TERMINATION,$$

$$BP\_REGISTRATION = BP\_181\_1.B1\_NEC + BP182\_1.B1\_VOL,$$

$$B1\_NEC = BP183\_1.B2\_NEC,$$

$$B2\_NEC = BP183\_8.BP\_TERMINATION,$$

$$B3\_NEC = BP183\_9\_DATE3.B2,$$

$$B1\_VOL = BP183\_1.BP183\_5.B2\_VOL + BP183\_1.B2\_VOL,$$

$$B2\_VOL = BP183\_8.B\_TERMINATION + BP183\_9.B3\_VOL,$$

$$B3\_VOL = BP183\_9\_DATE1.B2 + BP183\_9\_DATE2.B2,$$

$$B2 = BP183\_15.B3,$$

$$B3 = BP184\_1.B4,$$

$$B4 = BP184_1\_A.BP\_TERMINATION + BP184_1\_G.BP\_TERMINATION + BP184_1\_OTHER.BP\_TERMINATION + 184\_3.B2,$$

$$BP\_SALES = BP\_SALES\_1.BP\_SALES + BP\_TERMINATION$$

Під час аналізу алгебри поведінки для реєстрації платника податку було виокремлено та проаналізовано п'ять трас, які ведуть до термінального стану.

#### 2.2.4. Результати моделювання.

Використання системи інсерційного моделювання для аналізу правових моделей дає змогу проаналізувати тексти нормативно-правових актів, а саме:

- знати й проаналізувати перехресні посилання між нормативними документами;
- проаналізувати законодавчі акти щодо наявності протиріч, неповноти, цілісності та інших властивостей.

Аналіз і дослідження такого типу мусять бути пріоритетним напрямом дослідження, оскільки однією з основних вимог з боку Європейського Союзу до України як претендентки на членство в ЄС є вирішення проблеми гармонізації законодавчих баз України та Європейського Союзу. Основна правова концепція ЄС полягає в цілісності правової системи Євросоюзу, а тому необхідним є проведення експертизи українських законопроектів та інших нормативно-правових актів, що за предметом правового регулювання належать до сфер правовідносин, що регулюються правом ЄС. Нормативно-правова база України повинна бути проаналізована на можливі структурно-логічні помилки, протиріччя, циклічність, перехресні посилання.

Використання описаної технології дає змогу також перевіряти не тільки наявні, чинні акти, а також ті, які перебувають у процесі формування. Досвід учених Нідерландів, Данії, Фінляндії, Італії показує, що використання формальних методів є і актуальним підходом, і ефективним. В Україні створено організацію дослідників у галузі права й інформатики, які займаються комп'ютерним аналізом юридичних текстів та документів – JURIX.

Створення технології інсерційного моделювання для доказу формальних моделей правових актів не тільки допоможе усунути можливі структурно-логічні помилки, а й перевірити наявність можливих правових колізій між Україною та ЄС.

Унаслідок використання формальних методів алгебраїчного моделювання в рамках системи APS APLAN[102,103] виявлено неточності, які можуть призвести до неоднозначного трактування закону та судових помилок. Методи статичної

перевірки на недетермінізм визначають двозначності в деяких законах. Наприклад, у статті 183.3 говориться, що особа, яка добровільно реєструється й відповідає вимогам 180.1 п.6, має проходити реєстрацію за процедурою 183.7, але з іншого боку, та сама особа може реєструватись і через пункти 183.2 та 183.1, тому що не вказано, що це – тільки для платників податків, які НЕ реєструються добровільно і відповідають вимогам 180.1 п.6. Це ілюструється наступними формальними висловами:

*TaxPayerRegistration & Condition\_180\_1\_6 ⇔ Registration\_183*

*TaxPayerRegistration & Voluntary & Condition\_180\_1\_6 ⇔ Registration\_187\_1*

Ці умови визначаються виконуваністю кон'юнкції інформаційно залежних аксіом. Неповнота випадків може досліджуватись статичною перевіркою передумов. Наприклад, за добровільної реєстрації, згідно з статтею 183.9 п.2-3, визначається день, із якого нараховується ПДВ. Цей день може бути першим числом наступного місяця після спливання 20 днів із часу подання заяви на добровільну реєстрацію.

Але податкове зобов'язання й умови необхідної реєстрації можуть надійти до визначеної дати (із якої буде нараховуватися ПДВ) – і залишається незрозумілим, як у цьому разі визначати дату початку сплати ПДВ.

### **2.2.5. Використання інсерційного моделювання для аналізу нормативно-правових прецедентів.**

Як приклад роботи з нормативно-правовим документом, розглянемо текст юридичного прецеденту «засуджений» ПДВ, або прецедент на користь платника податку [1]. Зауважимо, що аналіз юридичних прецедентів є важливим складником «здорового» суспільства, адже сприяє уникненню в майбутньому можливих протиріч у вироках.

Обраний приклад (випадок включення до складу податкового кредиту й валових витрат ПДВ платника податку при придбанні автомобіля) ілюструє

випадки суперечливості статей у законі, що ми проаналізуємо за допомогою пошуку недетермінізму в аксіомах.

Протягом року платником податку були придбані автомобілі на суму, що включає певний ПДВ. До податкового кредиту ця сума не відносилась, а була включена до валових витрат на підставі статті 7.4.2 Закону про ПДВ. Але податкова інспекція внаслідок перевірки, згідно зі статтею 4.2.2 (підпункт “б”), статтями 5.2.1, 5.3.3, 5.4.1 Закону про оподаткування прибутку, призначила підприємству податкове зобов'язання за податком на прибуток і штрафні санкції.

Підприємство звернулося до господарського суду з позовом до податкової інспекції про визнання протиправним і часткове скасування податкового повідомлення-рішення. Зазначене податкове повідомлення-рішення оспорується позивачем у частині визначення податкового зобов'язання за податком на прибуток і штрафних санкцій.

Постановою господарського суду позов задоволено повністю. Визнано протиправним і скасовано податкове повідомлення-рішення податкової інспекції в частині донарахування підприємству податкового зобов'язання за податком на прибуток. Формалізуючи дію платника податку, маємо середовище:

*TransactionPurpose == BUY\_FOR\_MANUFACTURE &&*

*TransactionSubject == CAR &&*

*TaxCredit == 0 &&*

*TransactionType == COMMON\_EXPENSES &&*

Діяльність платника позначаємо як

*Activity == TAXI\_SERVICE*

Розглянемо статті закону. Так, у пп. 7.4.2 п. 7.4 ст. 7 Закону про ПДВ ідеться: «Не включається до складу податкового кредиту і відноситься до складу валових витрат сума податку, сплачена платником податку при придбанні легкового автомобіля (крім таксомоторів), що включається до складу основних фондів» [1].

Стаття формалізується такою формулою:

$$A_1: (TransactionPurpose == BUY\_FOR\_MANUFACTURE) \ \&\& \ (TransactionSubject == CAR) \ \&\& \ (Activity != TAXI\_SERVICE) \Rightarrow \\ !((TaxCredit == 0) \ \&\& \ (TransactionType == COMMON\_EXPENSES))$$

Водночас, за Законом про оподаткування прибутку, «5.3. Не включаються до складу валових витрат витрати на: ...5.3.3. ... сплату податку на додану вартість, включеного до ціни товарів (робіт, послуг), що придбаваються платником податку для виробничого або невиробничого використання».

$$A_2: (TransactionPurpose == BUY\_FOR\_MANUFACTURE) \ \parallel \\ (TransactionPurpose == BUY\_NOT\_FOR\_MANUFACTURE) \Rightarrow (TransactionType != COMMON\_EXPENSES)$$

У цьому випадку маємо сумісність середовища з кожною з аксіом  $A_1$  та  $A_2$ , що виражається недетермінізмом, тобто  $A_1 \ \& \ E$  – істинне і водночас  $A_2 \ \& \ E$  також істинне. Тобто одна формула виражає, що все куплене для виробництва не може вважатися валовими витратами, а інша – що не може вважатися валовими витратами все куплене, крім машин для таксі-сервісу.

З іншого боку, маємо неповноту закону для випадку ( $Activity == TAXI\_SERVICE$ ), тобто явно не описано, як у цьому разі співвідносяться витрати й визначається податковий кредит. Недетермінізм аксіом можна визначити, досліджуючи виконуваність  $A_1 \ \&\& \ A_2$ , або доводити виводжуваність протилежних тверджень – як у нашому випадку ( $TransactionType == COMMON\_EXPENSES$ ) та  $!(TransactionType == COMMON\_EXPENSES)$

Якщо це твердження та його заперечення вивідні з аксіом, то маємо несумісність аксіом, що відповідають певному закону. Таким чином ми можемо представити двозначність цих законів. Отже, закон про ПДВ і Закон про оподаткування прибутку припускають неоднозначне множинне трактування порядку формування валових витрат при здійсненні господарських операцій із придбання легкового автомобіля і можливості віднесення сплачених сум податку на додану вартість за цими операціями на валові витрати, тобто виникає конфлікт інтересів.



### 2.2.6. Методи доведення властивостей.

Формалізація юридичних вимог у вигляді мови дій уможливорює проведення їхньої автоматичної верифікації, тобто перевірку виконання коректності вимог в аспекті різних критеріїв. Важливими властивостями формальних вимог, що допускають формальну перевірку, є їхня несуперечність і повнота. Змістове трактування цих понять залежить від конкретних додатків та не завжди добре формалізується. Формулювання необхідних або достатніх критеріїв для несуперечності й повноти дозволить відшукати причини істинних помилок в інженерних рішеннях на рівні вимог. Такий підхід можна застосувати також до нормативно-правових актів, оскільки ми розглядаємо нормативний акт як юридичну вимогу.

Поняття несуперечності часто пов'язується з можливістю однозначного вибору дії для компонента системи, який здійснює управління поведінкою системи, а повнота полягає в тому, що якщо система не перейшла в стан успішного завершення свого функціонування, то повинна бути хоча б одна дія для продовження її функціонування. Вимога несуперечності може бути ослаблена для тих ситуацій, коли агентів дозволено здійснювати недетермінований вибір на своїх діях і тим самим визначати свою поведінку за допомогою різних дій.

Достатніми умовами несуперечності є неможливість одночасного виконання передумов двох різних дій, а достатня умова повноти виражається як тотожна істинність диз'юнкції всіх передумов дій. Формально умова несуперечності для двох дій виглядає так:  $\forall x(a_1 \rightarrow \langle P_1 \rangle \beta_1)$  та  $\forall x(a_2 \rightarrow \langle P_2 \rangle \beta_2)$ , – і реалізується у вигляді заперечення перетину формул, що виражають умови застосовності дій:  $\neg((\exists x a_1) \wedge (\exists y a_2))$  [91, 104].

Дві дії формально суперечливі, якщо ця умова, яка називається умовою транзитивної несуперечності, не може бути доведена. Під час перевірки транзитивної несуперечності необхідно враховувати наступні обставини. По-перше, вона повинна виконуватися тільки на досяжних станах системи. По-друге,

формальна суперечливість може не бути достатньою для змістовного протиріччя, оскільки недетермінізм, викликаний цією суперечливістю, може бути допустимим з інженерних міркувань. Інша ситуація, коли формальна суперечливість може бути припустимою; це полягає в тому, що постумови дій визначають однакові перетворення стану середовища.

### **Висновки до другого розділу**

Використання методів символічного й інсерційного моделювання дають змогу не тільки перевіряти законодавчу базу на суперечливість, повноту та цілісність, а й створювати методи для перевірки судових та інших юридичних постанов. Крім того, на основі методів інсерційного моделювання можуть бути розроблені програмні засоби підтримки прийняття рішень для юристів.

1. Формалізовано Податковий кодекс України засобами інсерційного моделювання.

2. Проаналізовано перевірку цілісності й відсутності протиріч у законодавчій базі України.

За допомогою системи алгебраїчного програмування АПС виявлено низку неточностей, які можуть призвести до неоднозначного трактування закону та судових помилок. Крім того, завдяки методам статичної перевірки на недетермінізм визначено двозначності в законах.

Наприклад, у статті 183.3 Податкового кодексу України говориться, що особа, яка добровільно реєструється й відповідає вимогам 180.1 п.6, має проходити реєстрацію за процедурою 183.7, але з іншого боку, та сама особа може зареєструватися також згідно з пунктами 183.2 та 183.1.

Крім того слід зазначити, що Відповідно до статті 183.9 п.2-3, у процесі добровільної реєстрації визначається день, із якого нараховується ПДВ. Цей день може бути першим числом наступного місяця після спливання 20 днів із часу подання заяви на добровільну реєстрацію. Проте податкове зобов'язання й умови

необхідної реєстрації можуть надійти до визначеної дати – і не зрозуміло, як тоді визначати дату початку сплати ПДВ.

3. Запропоновано підхід до зіставлення юридичних документів із законами на прикладі юридичного прецеденту [1].

Під час аналізу юридичного прецеденту вдалося показати сумісність середовища з кожною з аксіом  $A_1$  та  $A_2$ , що виражається недетермінізмом у юридичному прецеденті.  $A_1 \& E$  – істинне і водночас  $A_2 \& E$ .

З іншого боку, маємо неповноту закону для випадку (*Activity* == *TAXI\_SERVICE*), тобто явно не описано, як у цьому разі співвідносяться витрати й визначається податковий кредит. Недетермінізм аксіом можна визначити, досліджуючи виконуваність  $A_1 \& A_2$ , або доводити виводжуваність протилежних тверджень – як у нашому випадку

Технологія інсерційного моделювання показала себе як ефективний метод для доведення властивостей і аналізу формальних моделей законодавчих та інших нормативно-правових документів. Унікальність цього підходу полягає в можливості створювати сценарні моделі в поєднанні зі символьним моделюванням і статичними методами перевірки формальних моделей.

## Список джерел до Розділу 2

- [1] А. Ад. Летичевский, “Инсерционное моделирование”, *Управляющие системы и машины*, № 6, с. 3-14, 2012.
- [1] A. Letichevsky and D. Gilbert, "A general theory of action languages", *Cybern Syst Anal*, vol. 34 (1), pp. 12-30, Jan. 1998. Doi: 10.1007/BF02911258.
- [2] A. Letichevsky and D. Gilbert, “A Model for Interaction of Agents and Environments”, in *14th Int. Workshop WADT'99, Recent Trends in Algebraic Development Techniques*, (LNCS, vol. 1827), 2000, pp. 311-328.
- [3] A. A. Letichevsky and D. R. Gilbert, “Agents and environments”, in *Proc. 1st Int. scientific and practical conf. on programming*, Sep. 1998, 10 s.
- [4] R. Milner, *A Calculus of Communicating Systems*. Berlin, Heidelberg, Germany: Springer–Verlag, 1980.
- [5] R. Milner, *Communication and Concurrency*. NJ, USA: Prentice Hall, 1989.
- [6] R. Milner, “The polyadic  $\pi$ -calculus: a tutorial”, in *Tech. Rep. ECS–LFCS–91–180, Edinburgh University (1991)*, Reprinted in *Logic and Algebra of Specification*, F. Bauer, W. Brauer and H. Schwichtenberg, Eds. Springer–Verlag, 1993, pp. 203-246.
- [7] C. A. R. Hoare, *Communicating Sequential Processes*. NJ, USA: Prentice Hall, 1985.
- [8] J. A. Bergstra, J. W. Klop, “Process algebra for synchronous communications”, *Inf. and Control*, vol. 60 (1/3), pp. 109-137, 1984.
- [9] A. A. Letichevsky, O. A. Letychevskiy, and V. S. Peschanenko, "Insertion modeling system", in *Perspectives of System Informatics Int. Andrei Ershov Memorial Conf. (PSI'11)*, (LNCS, vol. 7162), 2012, pp. 262-273.
- [10] A. A. Letichevsky et al., “System Specification by Basic Protocols”, *Cybern Syst Anal*, vol. 41, iss. 4, pp. 479-493, July 2005. Doi: 10.1007/s10559-005-0083-y.
- [11] A. Letichevsky, “Algebra of behavior transformations and its applications”, *Structural theory of Automata, Semigroups, and Universal Algebra* (NATO Science Series II: Mathematics, Physics and Chemistry (NAII), vol. 207), pp. 241–272, 2005. Doi: 10.1007/1-4020-3817-8\_10.

[12] А. А. Летичевский, А. Ал. Летичевский, В. С. Песчаненко, и А. А. Губа, “Генерація символічних трас в системі інсерційного моделювання”, *Кибернетика и системный анализ*, Т. 51, № 1, с. 7-19, 2015.

[13] А. А. Губа, и К. И. Шушпанов, "Инсерционная семантика плоских многопоточковых моделей языка UCM", *Управляющие системы и машины*, № 6, с. 15-34, 2012.

[14] A. Letichevsky et al., “Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications”, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, iss. 47, pp. 662-675, Dec. 2005.

[15] A. A.Letichevsky et al., “Insertion modeling in distributed system design”, *Проблеми програмування*, № 4, с. 13-38, 2008.

[16] A. A. Letichevsky, O. A. Letychevskiy, , and V. S. Peschanenko, “Insertion modeling system”, in *PSI'11*, (LNCS, vol. 7162), June 2011, pp. 262-273. Doi: 10.1007/978-3-642-29709-0\_23.

[17] O. Letychevskiy, and T. Weigert, "Symbolic verification of requirements in VRS system", in *2014 IEEE 22nd Int. Conf. RE*, Aug. 2014, pp. 331-332. DOI: 10.1109/RE.2014.6912282

[18] А. А. Летичевский, А. Б. Годлевский, и А. А. Летичевский (мл.), С. В. Потенко, В. С. Песчаненко, “Свойства предикатного трансформера системы VRS”, *Кибернетика и системный анализ*, № 4, с. 3-16, 2010.

[19] А. Б. Годлевский, “Предикатные преобразователи в контексте символьного моделирования транзиторных систем”, *Кибернетика и системный анализ*, № 4, с. 91-99, 2010.

[20] A. Letichevsky, A. Godlevsky, A. Guba, A. Kolchin, O. Letychevskiy, an V. Peschanenko, “Invariants in symbolic modeling and verification of requirements”, in *Ninth Int. Conf. on Computer Science and Inf. Technologies Revised Sel. Papers IEEE (SCIT 2013)*, Sept. 2013, pp. 1-6. Doi: 10.1109/CSITechnol.2013.6710332.

[21] V. Peschanenko, A. Guba, and C. Shushpanov, “Specializations and Symbolic Modeling”, in *Proc. 9-th Int. Conf. Information And Communication*

*Technologies In Education, Research And Industry (ICTERI'13), CEUR-WS*. Jun 2013, vol. 1000, pp. 490-505.

[22] Верховна рада України. (1997, Квіт. 03, зі змінами). *Закон України № 168/97-ВР, Про податок на додану вартість*. [Електронний ресурс]. Доступно: <https://zakon.rada.gov.ua/laws/show/168/97-вр>.

[23] A. Letichevsky, O. Letychevskiy, V. Peschanenko, and M. Poltorackij, "An Algebraic Approach for Analyzing of Legal Requirements", in *2017 IEEE 25th Int. Requirements Eng. Conf. Workshops (REW'17)*, Sept. 2017, pp. 209-212. Doi: 10.1109/REW.2017.51.

[24] A. Godlevskiy et al., "Formalization and Algebraic Verification of Legal Requirements", in *Proc. 13-th Int. Conf. ICTERI 2017, CEUR-WS*. May 2017, vol. 1844, pp. 524-535.

[25] APS APLAN. APS and IMS are best for rewriting and modelling. [Электронный ресурс]. Доступно: <http://apsystems.org.ua>. Дата обращения: Янв. 11, 2010.

[26] A. A. Letichevsky, and J. V. Kapitonova, "Algebraic programming in the APS system", in *Proc. int. symposium on Symbolic and algebraic computation (ISSAC'90)*, July 1990, pp. 68-75. Doi: 10.1145/96877.96896.

[27] A. A. Letichevsky, O. A. Letichevskiy, V. S. Peschanenko, "The Non-Deterministic Strategy of Rewriting", *Управляющие системы и машины*, № 6, с. 53-58, 2013.

## РОЗДІЛ 3

### СИНТАКСИЧНИЙ АНАЛІЗ ЮРИДИЧНИХ ТЕКСТІВ

#### І РОЗПІЗНАВАННЯ ТРАС

##### 3.1. Огляд систем морфологічного аналізу.

Наразі розглянемо сучасні системи семантичного аналізу тексту для автоматизованого вилучення ключової інформації зі слабоструктурованих текстів. Такий підхід уможливить автоматизоване вилучення потрібної інформації для створення формальних правових моделей.

*NLTK* (Natural Language Toolkit) [105; 106] є однією з найпотужніших платформ для створення програмного забезпечення мовою програмування Python для роботи зі слабоструктурованими масивами інформації. *NLTK* надає прості у використанні інтерфейси для взаємодії з багатьма лексичними ресурсами, наприклад, ресурс *WordNet*. Цей програмний модуль може вирішити такі завдання:

1. Обробка тексту для класифікації.
2. Токенізація тексту.
3. Обробка тексту за мітками.
4. Семантичний аналіз тексту.
5. Використання NLP.

У *NLTK* визначено інтерфейси та структури даних, *rechunkparser* забезпечує розбір тексту на основі типів і правил. Крім того, програмний модуль має гнучкий алгоритм розбору тексту, який використовується для запису гіпотез про синтаксичні складники тексту.

*ОРФО* [107] – це морфологічне об'єднання декількох модулів, призначених для морфологічного аналізу слів, дозволяє вирішити такі проблеми:

- Приведення слів до нормальної словникової форми.
- Пошук однієї форми слова за іншою його формою.
- Синтез усіх форм заданого слова.
- Робота зі словником.

ОРФО-модуль видає всі форми заданого слова, якщо воно розташоване в основному словнику або поповнюваному словнику користувача.

Бібліотека *RCO Morphology SDK* [108, 109] призначена для морфологічного аналізу слів російської мови й дозволяє вирішувати такі завдання:

- Визначати всі граматичні характеристики словоформ (частину мови, відмінок, відмінювання тощо) і лексико-семантичні розряди (ім'я, по батькові, прізвище, найменування організації, географічну назву);
- Приводити різні граматичні форми слова до початкової форми (називного відмінка, інфінітива);
- Надавати всі граматичні форми слова;
- Здійснювати точний аналіз відомих слів за словником обсягом понад 115 тисяч слів, що покриває більше 3-х мільйонів словоформ;
- Здійснювати високо достовірний аналіз невідомого слова на основі комплексу правил словотворення й словозміни;
- Здійснювати імовірнісний аналіз за допомогою співвіднесення з моделями словозміни вживаних слів на основі оцінки семи та суфіксальної частини слова.

Модулі машинної морфології *AskNet Morphology* [110] слугують для приведення слів російської та англійської мов до початкової форми, отримання всіх форм аналізованих слів, а також для синтезу лексем незнайомих слів.

Машинна морфологія використовується під час індексації текстів, завдяки чому різні форми слова сприймаються як одне слово. Це надає змогу формувати запит природною мовою, а також забезпечує повноту пошуку за рахунок відбору всіх допустимих варіантів зміни шуканих слів.

Машинний синтез лексем незнайомих слів із визначенням їхніх морфологічних характеристик дозволяє автоматично адаптувати пошукову систему *AskNet* до будь-якої прикладної галузі. Адаптація проводиться на основі автоматичного доповнення словникової бази лінгвістичного процесора *AskNet* в процесі індексації текстів.



Пакет *Stanford Parser* [111] являє собою Java-реалізацію ймовірнісних синтаксичних аналізаторів на природній мові з високооптимізованою PCFG і лексичним аналізатором.

Лексичний ймовірнісний аналізатор реалізує факторизовану модель тексту з окремою структурою фраз PCFG та експертами з лексичної залежності. Цей програмний засіб може бути використано як лексичний стохастичний та синтаксичний аналізатор текстового масиву інформації. Будь-який аналізатор тексту дає хорошу статистичну інформацію для аналізу тексту. GUI надається для перегляду вихідних даних синтаксичного аналізатора структури фраз [2].

*LingPipe* [112] – це набір інструментів для обробки тексту з використанням комп'ютерної лінгвістики, може бути застосований для вирішення таких завдань:

- Пошук імен людей, організацій у текстовій інформації.
- Автоматична класифікація результатів пошуку в Twitter за категоріями.
- Пропозиція правильного написання запитів.

*МетаФраз* [113] дозволяє комбінувати різні компоненти (збірки) автоматизованої системи, завдань має такі особливості:

- повноту необхідного лінгвістичного функціоналу (наприклад, фразеологічний машинний переклад, семантична обробка текстів чи автоматизоване складання словників);
- розрахований на багатьох користувачів або індивідуальний режим використання. Компоненти системи можуть бути клієнт-серверними, тільки серверними, десктоп;
- можлива інтеграція лінгвістичних технологій системи в інтерфейс сторонніх додатків.

Зазначимо, що всі програмні компоненти системи функціують як єдиний інтегрований програмний комплекс із загальними ресурсами (серверна або автономна бібліотека словників та баз даних документів) та з уніфікованим призначенням для користувача інтерфейсом [114].

*Py morphology* [115; 116] – морфологічний аналізатор, розроблений мовою програмування Python. Ця бібліотека володіє таким функціоналом:

- приводити слово до початкової форми.
- поставити слово в потрібну форму.
- поставити слово у форму множини.
- відмінювати слова тощо.
- повертати граматичну інформацію про словесні форми: число, рід, відмінок, частина мови.

Під час роботи система за джерело інформації використовує словник OpenCorpora. Бібліотека досить швидка: зараз швидкість роботи – від декількох тис. слів / сек. до > 100 тис. слів / сек.

*BRAT* [117] – це вебсистема з відкритим вихідним кодом для автоматизованої анотації текстових масивів інформації. Ця система може бути досить успішно застосована як система автоматичного анотування для юридичних документів і прецедентів. Дозволяє анотувати й у подальшому візуалізувати текстовий масив інформації. Так, *BRAT* використовують як систему первинної обробки текстового масиву для виділення класів, підкласів і відношень між ними, а в подальшому анотована інформація потрібна для побудови моделей [117].

Проаналізувавши досить велику кількість бібліотек, серед спеціалізованого програмного забезпечення ми обираємо *Pullenti SDK* [118–121], яка може ефективно працювати зі слабоструктурованим масивом інформації, написаним українською мовою. Далі опишемо підхід до вилучення ключової інформації для генерації моделей трас судових постанов.

## **3.2. Функціональні можливості Pullenti SDK**

### **3.2.1. Pullenti SDK як інструмент аналізу слабоструктурованого тексту.**

*Pullenti SDK* розроблено для проведення лінгвістичного аналізу слабоструктурованих текстів. Користуючись цією бібліотекою, виокремлюємо з нормативно-правового тексту такі сутності: суб'єкти, організації, дати, країни,

укази та інше. Застосування алгоритмів морфології вможливилює встановлення структури семантичних відносин. Алгоритми обробки й аналізу текстів засновано на правилах українського та російського правопису.

Морфологія реалізована в збірці EP.Morphology.dll і може використовуватися незалежно від SDK. Pullenti SDK[121] охоплює такі основні блоки:

- Токенізація. Розбиття семантичних конструкцій на токени.
- Морфологічний аналіз. Визначення для токенів частин мови.
- Нормалізація токенів. Приведення словоформи до потрібного падежу, роду або числа, опрацювання нових слів і режим виправлення помилок. Перетворення текстових конструкцій на трансліт.

- Виділення іменованих сутностей. Аналіз послідовності токенів, суті відповідного типу (персони, організації та ін.).

- Алгоритми для виокремлення з числовими даними, іменними та дієслівними групами, дужками й лапками.

Також відзначимо наявність словника термінів і скорочень, деривативного словника, можливість перевірки еквівалентності (наприклад, еквівалентності рядків, написаних латиницею та кирилицею), визначення спільнокореневих слів із різних частин мови, модель управління групами. Крім морфологічного, здійснимий також семантичний аналіз: токени структуруються у вигляді графа з семантичними зв'язками, що дає змогу аналізувати змістовну суть текстових конструкцій.

### **3.2.2. Модель токена й іменованої сутності. Специфіка роботи з юридичним текстом.**

Поняття токена лежить у центрі семантичного аналізатора SDK Pullenti. Кожен токен посилається на фрагмент вихідного тексту. Спочатку текст розбивається на послідовність текстових токенів, а вже потім, у процесі обробки, відбувається їхнє злиття в метатокени. Метатокен – це токен, на якому задаються правила. Наприклад, слова з великої літери або словесні конструкції з великих

літер за допомогою морфологічного аналізатора буде визначено як ім'я, аббревіатуру або найменування.

Іменована сутність – інше центральне поняття SDK Pullenti. Під іменованою сутністю розуміють інформаційний об'єкт із набором атрибутів і значень, які можна однозначно трактувати й ідентифікувати в межах певної текстової інформації.

Для роботи з тестовою інформацією в Pullenti SDK застосовується ряд досить потужних спеціальних аналізаторів, зокрема:

- *Ner.Date.DateAnalyzer* – знаходить два типи іменованих сутностей: *DATA*, *DATARANGE*, які відносяться до спільної категорії;
- *Ner.Money.MoneyAnalyzer* – знаходить іменовану сутність типу *MONEY*;
- *Ner.Keyword.KeywordAnalyzer* – знаходить іменовану сутність типу *KEYWORD*;
- *Ner.GeoAnalyzer* – знаходить іменовану сутність типу геолокації та інше;
- *Ner.Address.AddressAnalyzer* – знаходить іменовані сутності типу *STREET*, *ADDRESS*;
- *Ner.OrganizationAnalyzer* – знаходить іменовану сутність типу *ORGANIZATION* спільної категорії;
- *Ner.PersonAnalyzer* – знаходить іменовані сутності типу *PERSON*, *PEERSONPROPERTY*, *PERSONIDENTITY* спільної категорії [121].

Оскільки юридичний текст, зокрема текст судового рішення, є досить складним (містить багато непотрібної інформації, багатокomпонентні синонімічні ряди та складні синтаксичні конструкції), нормативно-правові документи важко обробляти за допомогою мов програмування. Розглянемо низку вимог до оформлення судових та інших текстів юридичного напрямку для подальшого ефективного застосування Pullenti SDK:

1. Точність – у змісті документа не допускається подвійного тлумачення слів і висловів.

2. Логічна послідовність – усі частини документа мають бути логічно пов'язані.

3. Стандартність – слід використовувати готові, перевірені практикою словесні форми, що легко сприймаються й точно описують ситуацію, яка неодноразово повторюється.

4. Не використовувати синонімічні конструкції в різних частинах тексту.

5. Необхідно дотримуватися чітких синтаксичних зв'язків між головним і залежним компонентом синтаксичної структури речення (підмет і присудок), словосполучення (головне й залежне слово).

6. Зберігати відповідність граматичної форми нормативному змістові.

7. Особливості регулятивної природи права суперечить використання окличних, питальних речень.

8. Дотримуватися правильного вживання розділових знаків:

- знак оклику, питальний знак і багато крапок наприкінці речення в нормативно-правових актах не вживаються?
- тире використовується, коли опускається присудок, наводиться визначення, необхідно відокремити санкцію норми від інших частин тощо [122];
- широко вживається двокрапка при використанні прийому «винесення за дужки», викладенні переліків;
- крапка з комою вживається при відокремленні частин складного речення та у випадках, коли елементи переліку мають відносно самостійне значення, а зв'язок між ними — послаблений характер.

9. У реченні не повинно бути ланцюжків, складених з однотипних граматичних форм, що слідує одна за одною.

10. Необхідно уникати застосування неозначеної форми дієслова й інфінітивної конструкції, оскільки за допомогою інфінітива зручно виражати прямий і непрямий імператив.

11. Слід правильно вживати сполучники.

12. Нормативно-правові акти України, зокрема, містять багато випадків невдалого вживання сполучників.

13. Уживати різноманітні форми дієслова: доконані й недоконані дієслова, дієприслівники.

1. Послуговуючись указаними формами, законодавець у першому випадку пов'язує настання певних юридичних наслідків не тільки з фактом здійснення певних дій (бездіяльності), але і з фактом настання певного результату.

- У тих випадках, коли використовується недоконана форма дієслова або дієприслівника, настання юридичних наслідків не ставиться в залежність від настання результатів дії, позначених указаними граматичними формами.

14. Складні речення розбити на прості.

Слід обирати таку побудову речення, яка максимально спрощує сприйняття змісту, що ним передається. Застосування мовних засобів має бути максимально економним: при викладенні нормативних правових приписів важливо, щоб їхні формулювання були стислими, містили якомога менше слів, словосполучень, які не несуть необхідної нормативної інформації, а тільки відволікають увагу від основної думки, смислу[122].

15. Не використовувати дієприкметникові та дієприслівникові звороти.

### **3.3. Використання Pullenty SDK для розпізнавання трас.**

Послуговуючись Pullenty SDK, можна розробити програмний засіб для розпізнавання трас із тексту судової постанови. За допомогою описаних аналізаторів даної SDK та з дотриманням описаних вище вимог до побудови тексту судової постанови можна виокремити потрібну нам інформацію для побудови моделі траси. Побудова траси дає можливість перевірити застосовність дій, які представляють собою формалізовані нормативно-правові вимоги, що використовуються для аргументації. Дотримуючись такого підходу, можна розробити методи й засоби для перевірки правильності вироку юридичного прецеденту.

Із повним текстом судової постанови можна ознайомитися в [123].

Так, за допомогою системи вбудованих аналізаторів ми виокремили список агентів та їхні типи, які фігурують у юридичному прецеденті (див. табл. 3.1)

Таблиця 3.1.

### Список агентів у судовій постанові

№	Имя	Тип
1	<i>Окружний адміністративний суд АР Крим.</i>	<i>StateAuthority</i>
2	<i>Державна податкова інспекції у м. Алушті АР Крим Державної податкової служби</i>	<i>StateAuthority</i>
3	<i>Кримпродпоставка</i>	<i>TaxPayer</i>
4	<i>ТОВ БВП «Строитель Плюс»</i>	<i>TaxPayer</i>
5	<i>ПП «Мастер Торг»</i>	<i>TaxPayer</i>
6	<i>ПП «Габарит Груп»</i>	<i>TaxPayer</i>
7	<i>ТОВ «ВТК «Окнополь»</i>	<i>TaxPayer</i>
8	<i>ПП «Сіті буд»</i>	<i>TaxPayer</i>
9	<i>ПП «Скіпер»</i>	<i>TaxPayer</i>
10	<i>ПП «Комманд»</i>	<i>TaxPayer</i>
11	<i>ПП «Мобкримторг»</i>	<i>TaxPayer</i>
12	<i>ПП «Визит Торг»</i>	<i>TaxPayer</i>
13	<i>ПП «Базис Торг»</i>	<i>TaxPayer</i>
14	<i>ПП «Укрторгпартнер»</i>	<i>TaxPayer</i>
15	<i>ПП «Кримспецтехнологджи»</i>	<i>TaxPayer</i>
16	<i>ПП «Ротонда Торг»</i>	<i>TaxPayer</i>
17	<i>ПП «Сіті буд»</i>	<i>TaxPayer</i>
18	<i>Фірма «Капітель Плюс»</i>	<i>TaxPayer</i>

Для видалення списку сутностей було використано такий фрагмент коду, складений мовою програмування Python 3.0:

```

def get_Organization(self):
with
ProcessorService.create_specific_processor(OrganizationAnalyzer.ANALYZER_NAME
) as proc:
    ar = proc.process(SourceOfAnalysis(self.txt), None, None)
    t = ar.first_token
    number = 0
    first_pass = True
    while True:
        if first_pass:
            first_pass = False
        else:
            number += 1
            t = t.next0_
            if (not (t is not None)): break
            if (isinstance(t, ReferentToken)):
                kw = Utils.asObjectOrNull(t.get_referent(), OrganizationReferent)
                if (kw is None):
                    continue
    kwstr = MiscHelper.get_text_value_of_meta_token(Utils.asObjectOrNull(t,
ReferentToken),
Utils.valToEnum((GetTextAttr.FIRSTNOUNGROUPTONOMINATIVESINGLE) | (
GetTextAttr.KEEPREGISTER), GetTextAttr))
        for i in [self.agents]:
            if (kwstr != None):
                if (str(kwstr).lower() not in str(i).lower()):
                    self.agents.append(kwstr)
                    self.agents_number.append(number)
                    self.agents_all.append(kwstr)
                    self.agents_number_all.append(number)

```



```
return [self.agents,self.agents_all,self.agents_number,self.agents_number_all]
[121]
```

За допомогою вбудованих алгоритмів Pullenti SDK ми розділяємо дії. Кожна дія має свою дату, час тривання, як правило, вказується ініціатор цієї дії. Далі в тексті ми намагаємось знайти значення й таким чином створити певний терм. Прикладом терму може бути така формалізація:

**Договір(№298, 01.06.2011, Company\_1, Company\_2):**

*Сума(23.01.2012, №252) = 9840,00*

*податковийКредит(23.01.2012, №252)=1640*

*Сума(23.01.2012, №253) = 3589,56*

*Сума(23.01.2012, №0142) = 3589,56*

**Договір( №186, 01.06.2011, Company\_1, Company\_2):**

*Сума(30.06.2011, №103) = 5200,00*

*Сума(30.06.201 1, №104) =8562,00*

**Договір( №17/33, 24.02.2012, Company\_1, Company\_2):**

*Сума(12.03.2012 , №30) = 11232,00*

*Сума(24.02.2012, №32) = 3744,00*

*ТранспортніНакладні(12.03.2012, №КП-0000004 ) = 11232,00*

*ТранспортніНакладні(24.02.2012, №0230 ) = 3744,00*

Таким чином у термі ми вказуємо всю важливу інформацію (назви компаній, дати укладання договору, суми, на які було укладено договори) для подальшого аналізу.

Оскільки ми можемо визначити порядок дій у тексті, то створюємо певну послідовність дій та їхніх значень, яка представлена трасою MSC. Приклад такої траси представлений нижче:

*Enitial environment:*

*x.taxPayer == true &&*

*x.conected == true &&*

*x.Responsobility.verdict == UNDEFINED &&*

*x.Location.legalAddress== false &&*

```

x.Location.actAddress == false &&
x.Transaction.taxInvoice == true &&
x.Transaction. StausTransaction == false &&
x.Transaction.DataTransaction.day == 23 &&
x.Transaction.DataTransaction.month == 1 &&
x.Transaction.DataTransaction.year == 2012 &&
x.Transaction.cost == 2705.00

```

*article\_12\_1():*

```

x.Location.legalAddress == false &&
x.Location.actAddress == false &&
x.Transaction.stausTransaction == false &&
x.conected == true

```

*articel\_198\_6():*

```

x.taxCredit == 2705.00

```

Крім того, за допомогою розробленого програмного забезпечення ми можемо виокремити перелік законів, які використовуються для аргументації, і порівняти (узгодити) побудовану трасу та закони за допомогою методу зіставлення в APS APLAN.

Таким чином, раніше формалізовані закони податкового кодексу України можна перевіряти на застосовність на цій трасі. Якщо формалізовані дії будуть застосовні й результати зіставляються, то вирок було винесено правильно. Такий підхід дає змогу перевіряти правильність винесених судових рішень і юридичних прецедентів, крім того, методи зіставлення в APS APLAN можна ефективно використовувати під час розроблення програмних засобів для підтримки й прийняття рішень для юристів із практикою.

### **Висновки до третього розділу**

Під час роботи над третім розділом проаналізовано сучасні засоби й підходи до аналізу слабоструктурованого тексту, системи семантичного аналізу, а також

системи автоматичного реферування тексту. На основі проведеного дослідження було обрано *Pullenti SDK* для мови програмування *Python*. Ця система є однією з найпотужніших систем текстової обробки – із розвиненим функціоналом і досить значною кількістю аналізаторів для оброблення тексту.

За допомогою цієї *SDK* представлено підхід до трансляції судової постанови в алгебраїчний вид. Такий підхід уможливив виокремлення з тексту судової постанови траси, яку в подальшому використовуємо для аналізу. За допомогою системи алгебраїчного програмування та вбудованої системи порівняння зіставляємо формалізовані раніше нормативно-правові документи й виокремлені траси. На основі такого підходу можливо в автоматичному режимі аналізувати тексти судових постанов, перевіряти правильність винесених юристами вироків. За цим підходом проаналізовано юридичну справу [123]. У процесі дослідження розроблено програмний засіб автоматичної трансляції тексту в алгебраїчний вид, застосовано методи зіставлення АПС та опрацьовано реальну юридичну справу.

За допомогою системи інсерційного моделювання й алгебраїчного програмування на основі виокремлення трас із судових постанов можливо розробити методи та засоби перевірки правильності винесення вироків судових постанов. Крім того, використовуючи описаний вище підхід, можна розробити програмний продукт підтримки прийняття рішень для юристів із практикою.

### Список джерел до Розділу 3

- [1] E. Loper, and S. Bird, "NLTK: the Natural Language Toolkit", in *arXiv: a free distribution service and an open-access archive*, May 2002. [Online]. Available: <https://arxiv.org/abs/cs/0205028>. Accessed on: Sept. 14, 2019.
- [2] P. Garg, "Sentiment Analysis of Twitter Data using NLTK in Python", PhD Thesis, Thapar University, Patiala – 147004, India, 2016.
- [3] *ОРФО (сайт)*. [Электронный ресурс]. Доступ: <https://orfo.ru/>. Дата обращения: Янв. 11, 2020.
- [4] А. В. Пруцков, "Генерация и определения форм слов естественных языков на основе их последовательных преобразований", *Вестник Рязанского государственного радиотехнического университета*, № 27, с. 51-58, 2009.
- [5] Д. Л. Косов, и В. М. Белов, "Текстовые анализаторы как средство автоматизации правовых экспертиз нормативно-правовых актов и их проектов", *Гражданское общество и правовое государство*, № 1, с. 107-109, 2015.
- [6] А. Л. Огарок, "Метод полного лингвистического анализа неструктурированной текстовой информации", *Информатизация и связь*, № 1, с. 91-99, 2018.
- [7] M.-C. de Marneffe, and C. D. Manning, "Stanford typed dependencies manual", Stanford University, Stanford, USA, Tech. Rep., Sep. 2008.
- [8] B. Carpenter, "Phrasal Queries with LingPipe and Lucene: Ad Hoc Genomics Text Retrieval", in *Proc.13th Text REtrieval Conference (TREC 2004)*, Vol. 1., Oct. 2004. [Online]. Available: <https://trec.nist.gov/pubs/trec13/papers/alias-i.geo.pdf>. Accessed on: Sept. 14, 2019.
- [9] А. А. Хорошилов, А. А. Хорошилов, А. А. Хорошилов, Ю. В. Никитин, М. В. Смирнов, и Д. А. Садовников, "Сервер лингвистического ПО МетаФраз R10 (MF Lingware Server R10)", *Свидетельство о гос. регистрации программы для ЭВМ № 2014662743*, Дек. 08, 2014.
- [10] А. А. Хорошилов, А. А. Хорошилов, А. А. Хорошилов, Ю. В. Никитин, М. В. Смирнов, и Д. А. Садовников, "Лингвистический комплекс МетаФраз R10

(MF Lingware Complex R10)", *Свидетельство о гос. регистрации программы для ЭВМ № 2014663079*, Дек. 15, 2014.

[11] П. В. Паничева, Е. В. Протопопова, А. Р. Мирзагитова, и О. А. Митрофанова, "Разработка лингвистического комплекса для морфологического анализа русскоязычных корпусов текстов на основе Rymorphy и NLTK", в *Труды междунар. конф. "Корпусная лингвистика – 2015"*, СПб, 2015, с. 361-373.

[12] E. Kuzmenko, "Morphological analysis for Russian: Integration and comparison of taggers", in *Int. Conf. on Analysis of Images, Social Networks and Texts (AIST 2016)*, (Communications in Computer and Information Science (CCIS), vol. 661), Apr. 2016, pp. 162-171. Doi: 10.1007/978-3-319-52920-2\_16.

[13] Brat rapid annotation tool manual, in *Brat rapid annotation tool (website)*. [Online]. Available: <http://brat.nlplab.org/manual.html>. Accessed on: Sept. 14, 2019.

[14] D. S. Zuev, A. A. Marchenko, and A. F. Khasiannov, "Text mining tools in legal documents", in *CEUR Workshop Proc.*, vol. 2022, Oct. 2017, pp. 214-218.

[15] А. В. Рубайло, и М. Ю. Косенко, "Программные средства извлечения информации из текстов на естественном языке", *Альманах современной науки и образования*, № 12, с. 87-92, 2016.

[16] E. V. Politsyna, S. A. Politsyn, and A. S. Porechny, "Development of the Cross-platform Library of Morphological Analysis of the Russian Language Text for Industrial Software", in *Proc. 14th Central and Eastern European Software Eng. Conf.*, 2018, Article no. 11, pp. 1-8. Doi: 10.1145/3290621.3290635.

[17] *Pullenti 4.3 (website)*. [Online]. Available: <http://www.pullenti.ru/>. Accessed on: Sept. 14, 2019.

[18] Р. Стефанчук, Д. Монастырский, В. Рогожа, та В. Ватрас, "Вимоги до змісту юридичного документа", у *Правове письмо, навч. посіб.* Київ, Україна: Алерта; ЦУЛ, 2011. [Електронний ресурс]. Режим доступу: [https://pidru4niki.com/1779070751491/dokumentoznavstvo/vimogi\\_zmistu\\_yuridichno\\_go\\_dokumenta](https://pidru4niki.com/1779070751491/dokumentoznavstvo/vimogi_zmistu_yuridichno_go_dokumenta). Дата звернення: Окт. 20, 2019.

[19] Окружний адміністративний суд Автономної республіки Крим (2013, Черв. 04). *Постанова у справі № 801/3685/13-а*. [Електронний ресурс]. Доступно: <https://reyestr.court.gov.ua/Review/32553284>. Дата звернення: Окт. 20, 2019.

## РОЗДІЛ 4

### АЛГЕБРАЇЧНИЙ ПІДХІД У МОДЕЛЮВАННІ ЕКОНОМІЧНИХ МОДЕЛЕЙ ТА АНАЛІЗУВАННІ ЇХНІХ ВЛАСТИВОСТЕЙ

Розвиток комп'ютерних технологій спричинив зміни в дослідженнях економіки загалом та економічних моделей зокрема. Наразі у вивченні економічних процесів важливе місце посідають технології комп'ютерного моделювання економічних процесів і моделей. Експериментальна економіка, що базується на комп'ютерному моделюванні, уможливорює розроблення моделей як на рівні компаній, так і на рівні національної економіки.

Застосування комп'ютерних технологій дає змогу якісно вимірювати ефективність досліджуваних параметрів. Крім того, такий підхід застосовується у вивченні ефективності й реінжинірингу бізнес-процесів, класичної економіки (моделі мікро- та макрорівня), еволюційної економіки тощо.

Вагоме значення в моделюванні має дослідження моделей ринкової рівноваги, таких як модель Вальраса–Маршала, модель міжгалузевого балансу Леонтьєва, модель короткострокової економічної рівноваги Кейнса, модель рівномірної економіки, що розширюється, Неймана.

На сьогодні найпопулярнішими підходами до моделювання економічних процесів і моделей є використання нечіткої логіки, кінцевих автоматів, машинного навчання, формалізації бізнес-процесів: BPMN [124] та IDEF0 [125].

Розвиток і популярність блокчейн-технологій зумовили зародження нового напрямку економічних досліджень – криптоекономіки (токеноекономіки, або токеноміки) [126]. Розроблення нових DLT-додатків (Distributed Ledger Technology) [127], що часто ґрунтуються на токеноміці, спричиняє потребу моделювання й аналізування такого виду економічних моделей. Для реалізації ефективних моделей токеноміки важливо імплементувати в ці моделі закони токеноміки, які так само потрібно досліджувати. Стандартні класичні підходи досить складно імплементувати в дослідження економічних досліджень.

У цьому розділі розглянемо класичні методи й засоби моделювання економічних моделей, використання методів інсерційного моделювання до аналізу моделі рівноваги Вальраса, а також приклад застосування такого підходу до аналізу в токеномічних дослідженнях.

#### 4.1. Формальні методи в економічних дослідженнях.

В статті [128], описано реалізацію моделі адаптивного аукціону Вікрі, розглянемо приклад роботи програмного засобу для моделювання гіпотетичних даних.

Генеруємо випадковим чином від 1 до 100 записів продавців і покупців, де 1 – мінімальна платоспроможність, 100 – максимальна.

Для побудови експерименту в програмному засобі генерується платоспроможність покупців і витрати продавців. Для проведення подальшого експерименту згенерована платоспроможність покупців упорядковується від найбільшої до найменшої величини запису. Собівартість продавців упорядковується від найменшої до найбільшої, як представлено на рис. 4.1.

У програмному засобі порівнюється ідеальний та адаптивний аукціони; основною умовою, що впливає на особливості моделювання, є те, що всі учасники правдиво назвуть свої зазначені ціни.

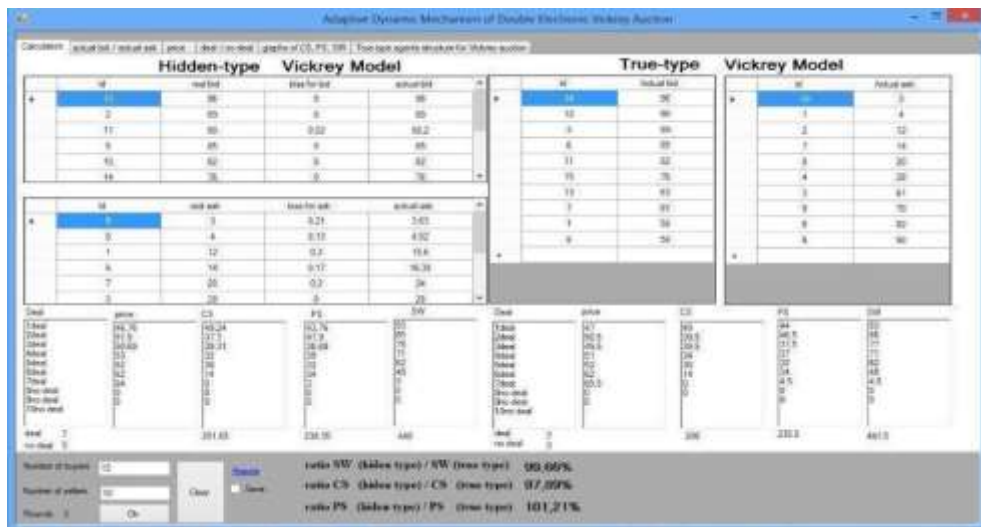


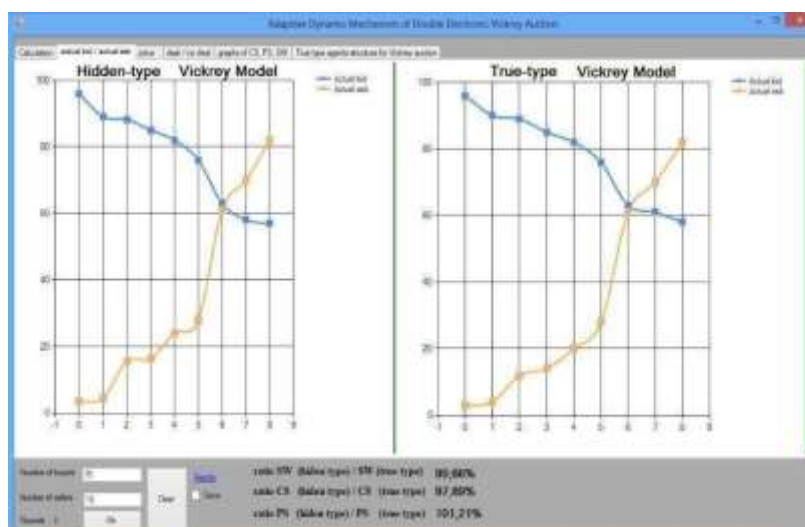
Рис. 4.1. Адаптивний та ідеальний аукціони



Програмний засіб використовує ітераційний підхід, кожна ітерація – це раунд аукціону. Після кожного раунду покупці й продавці можуть переєструвати свої ставки. Особливістю такого типу аукціону є можливість приховати свої ставки для покупців із найвищою платоспроможністю та продавців, які мають найменші витрати.

Ефективність пропонованого аукціону становить 99 % від ефективності ідеального аукціону, коли всі учасники повідомляють свій справжній тип [128].

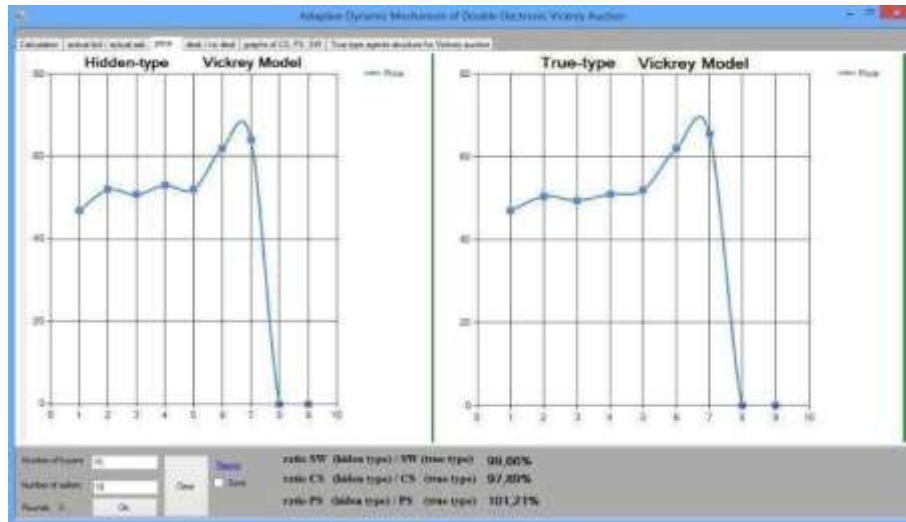
У цьому програмному засобі вбудовано можливість порівняння двох типів моделей аукціонів. Порівняння можливе, оскільки ціни товарів формуються для обох моделей аукціонів. На рис. 4.2 представлено траєкторії ставок. Як бачимо, траєкторії подібні, що свідчить про високу ідентичність порівняних моделей аукціонів.



**Рис. 4.2.** Порівняння цін адаптивного та ідеального аукціонів

Наведений рисунок (рис. 4.2) досить яскраво репрезентує особливості моделей аукціонів. Якщо продавець установлює ціну, вищу за платоспроможність продавця, – угода не відбудеться. Остання транзакція, що відбулася, представляється перетином функцій попиту та пропозиції. Також зауважимо, що обидві сторони зацікавлені в здійсненні угоди, кожен з учасників готовий поступатися, але тільки на певну величину, яка визначається функцією вигоди (ця функція може виявити не вигідну транзакцію).

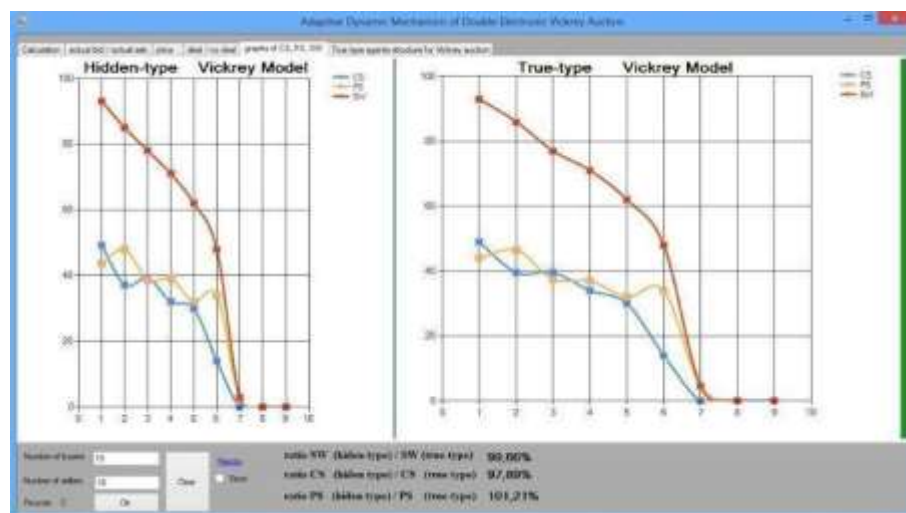
Згідно з вимогами до розробки моделі адаптивного аукціону, угоди спочатку укладають найбільш платоспроможні покупці та продавці, які зареєстрували ці ціни. На рис. 4.3 представлено динаміку цін. Зрозуміло, що коли ціна, запропонована продавцем, вища за платоспроможність покупця, то така угода не відбудеться.



**Рис. 4.3.** Динаміка ціни на адаптивному та ідеальному аукціонах

Таким чином, спочатку відбуваються найвигідніші угоди – коли покупець найбільш платоспроможний, а продавець заявляє найменшу собівартість [128].

На рис 4.4 зображено динаміку траєкторій угод для адаптивної та ідеальної моделей аукціонів. Динаміка є спадною, оскільки кожна наступна угода стає все менш вигідною, максимально наближені до нуля угоди – найменш вигідні.



**Рис. 4.4.** Динаміка вигравів покупців і продавців на адаптивному та ідеальному аукціонах

На основі розробленого програмного засобу для моделювання аукціонів ми зі співавторами дійшли певних важливих висновків:

- покупців відкривають свої зареєстровані ставки за даних умов;
- продавців називають реальну собівартість.

Отже, запропонована форма аукціону максимально наближена до реальних аукціонів, тому ймовірність того, що учасники аукціонів повністю розкривають свій тип, складає близько 70 % для всіх укладених угод [128].

#### 4.1.1. Диференціальні рівняння.

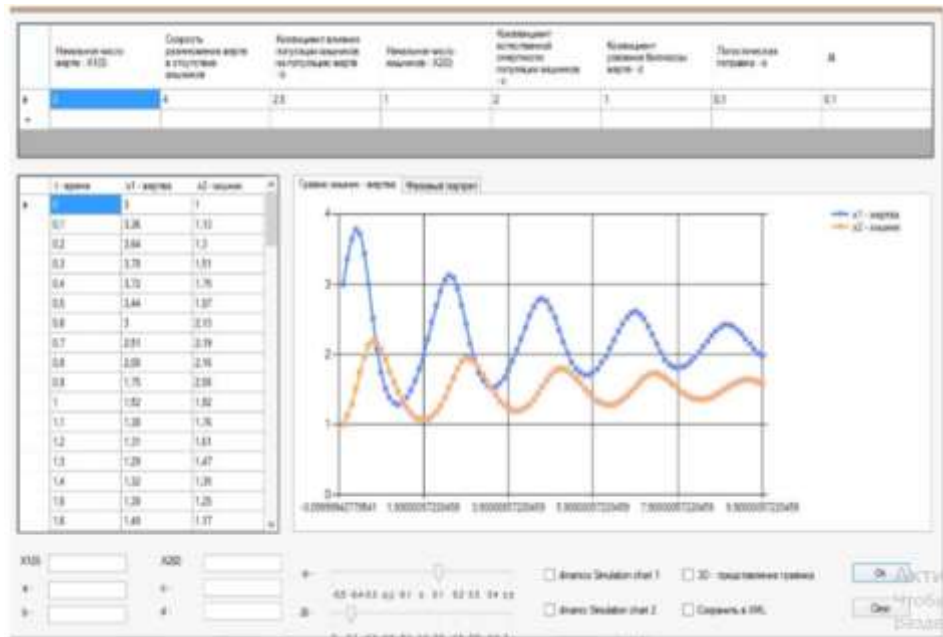
Теорія диференціальних рівнянь стала важливим інструментом економічного аналізу, особливо з моменту, коли комп'ютер зробився загальнодоступним. Сучасну економічну літературу важко уявити без таких базових понять, як біфуркація, хаос або без результатів теорії моделювання диференціальних рівнянь. Диференціальне рівняння виражає швидкість зміни поточного стану як функцію поточного стану; простим прикладом можуть слугувати зміни валового внутрішнього продукту (ВВП) із часом.

Яскравою ілюстрацією моделі, що базується на теорії диференціальних рівнянь, є модель «Хижак–Жертва» [129]. Для опису безперервної поведінки в ній застосовано диференціальне рівняння виду:

$$\frac{dN}{dt} = r * N,$$

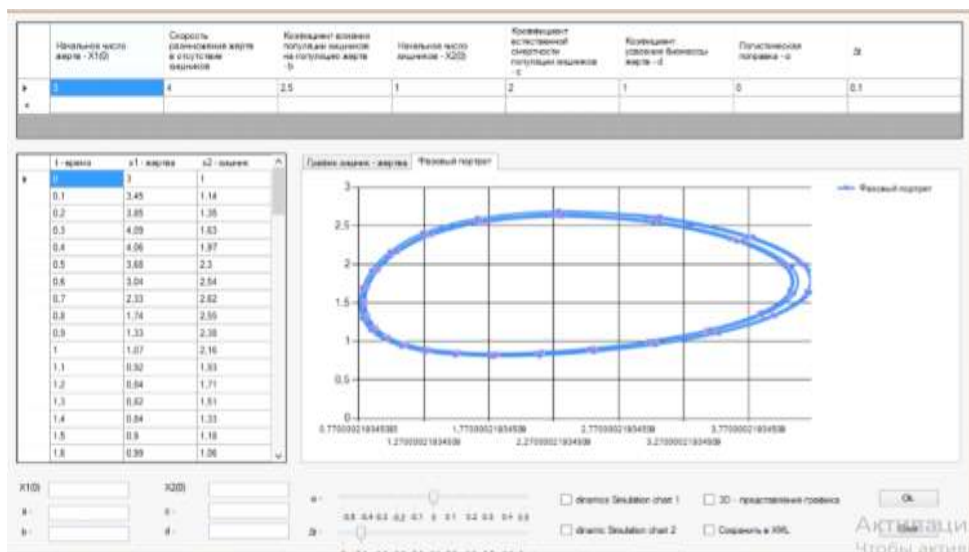
де  $N$  – чисельність популяції,  $r$  – коефіцієнт, що характеризує здатність популяції до відтворення. Розглянемо результати моделювання, якщо задано початкове співвідношення особин обох видів – 2:1. Обидві популяції спочатку зростають; коли кількість хижаків досягає 2.5, популяція жертв не встигає відновлюватися – і число жертв починає зменшуватися. Зменшення кількості популяції жертв призводить до скорочення популяції хижаків. Зменшення числа жертв продовжується до тих пір, поки число хижаків не досягне =1.6.. Із цього моменту популяція жертв починає зростати, що закономірно спричиняє збільшення популяції хижаків. Жертв стає достатньо, щоб забезпечити приріст

хижаків – і цикл повторюється. На рис 4.5. чітко видно періодичний характер цього процесу.



**Рис.4.5.** Динаміка популяції, модель “Хижак–Жертва”

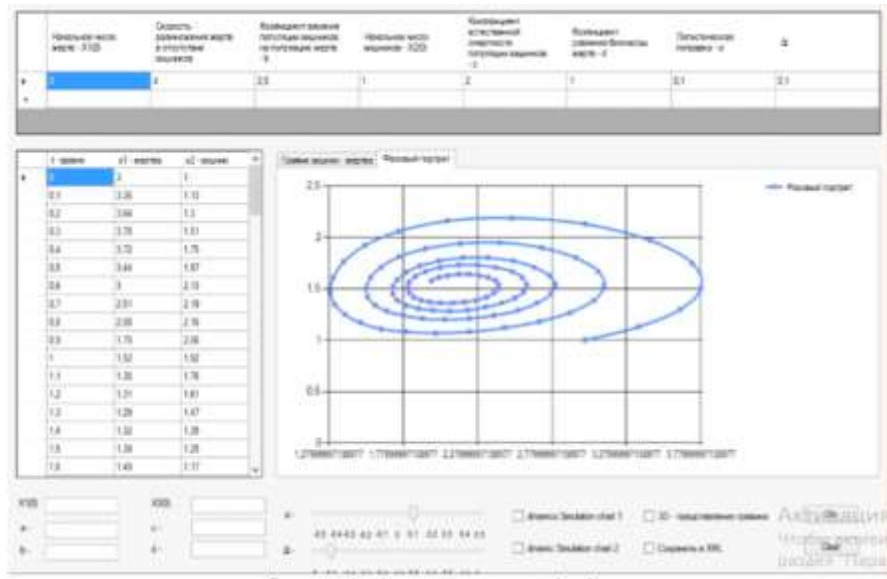
Періодичність процесу також видно й на фазовому портреті (рис. 4.7). Крайня ліва точка площини відповідає за значення, у якому кількість популяції жертв перебуває в мінімумі, крайня права відповідає за пік популяції жертв. Між цими двома точками чисельність хижаків спочатку зменшується до найнижчої точки фазового портрету, а потім зростає до найвищої точки фазового портрету.



**Рис.4.6.** Фазовий портрет, модель “Хижак–Жертва”

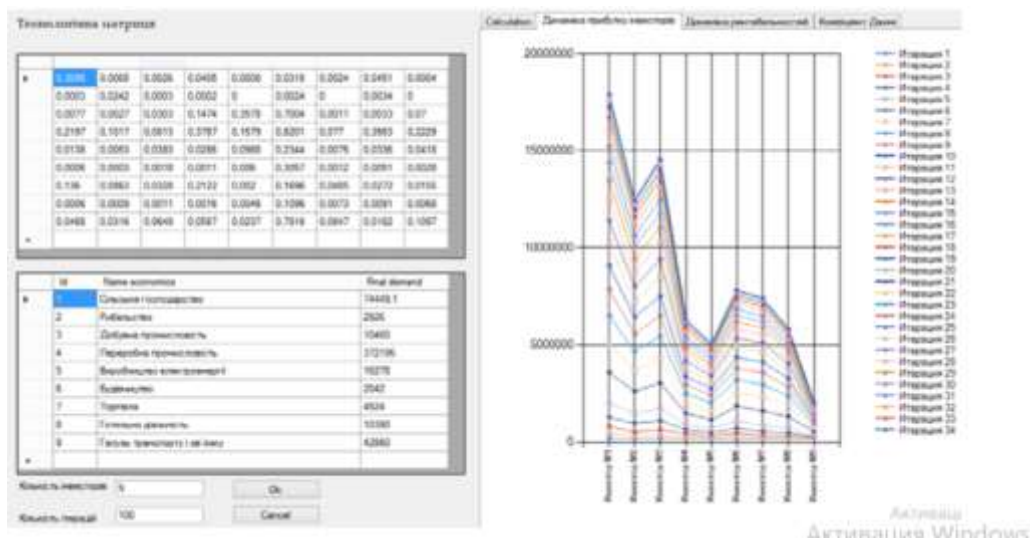
Як бачимо, в такому випадку залежно від значення логістичної поправки система стає нестабільною й коливання чисельності видів зростає. Це означає, що

яким би близьким не був початковий стан до стаціонарного, із плином часу стан системи буде відрізнятися від стаціонарного все помітніше (див. рис. 4.8).



**Рис. 4.7.** Фазовий портрет, модель “Хижак–Жертва”

У статті [130] представляємо експерименти з використанням еволюційної економіки, вхідними параметрами є статистичні дані з офіційних джерел. Кожен інвестор має на меті досягти більш високого рівня особистого добробуту та може вкласти свої гроші в кінцеву кількість галузей. Порядок розподілу коштів інвестором визначено його стратегією вкладання. Після того як кошти будуть вкладені, вони не можуть перемішуватися між секторами всередині одного часового періоду (див. рис. 4.8).



**Рис. 4.8.** Експерименти з використанням еволюційної економіки

Під час експерименту з'ясовано, що еволюційні інвестиційні стратегії адаптують розподіл інвесторів, фонд технологічних змін економічних галузей. У цьому дослідженні ми використовуємо еволюційний алгоритм (EA) для вдосконалення інвестиційних стратегій в економічній системі, порівнюючи частки найбільш прибуткових інвесторів з іншими.

Запропонований алгоритм вибирає оптимальне поєднання частки інвестицій у різних галузях для поліпшення інвестиційних показників. У нашому експерименті ми використовуємо статистичні дані з офіційних джерел, щоб вивчити ефективність запропонованої інвестиційної стратегії.

#### **4.1.2. Business process model and notation.**

В економічних дослідження часто застосовують мови формального опису бізнес-процесів BPMN (Business Process Model and Notation). Із теорію мови формалізації бізнес-процесів можна ознайомитися в працях С. Манкарелли [131], Т. Радемейкера [132].

В. Кобець і В. Яценко [133] пропонують свій підхід до вдосконалення бізнес-процесів і збільшення прибутку за допомогою авторегресійної моделі з використанням мови формальної специфікації бізнес-процесів BPMN 2.0. Науковці стверджують, що такий підхід може бути ефективно використаний для дослідження еволюційної економіки. Вони представляють модель бізнес-процесу, який репрезентує виробництво безалкогольних напоїв, засобами мови BPMN. Використана система для візуалізації та формалізації бізнес процесів Bizagi BPMN [134] дозволяє провести симуляцію процесу, визначивши кількість працівників і деякі інші параметри цього процесу, зокрема витрати (прямі й непрямі) на виконання процесу (див. рис. 4.9).

Після визначення початкових параметрів і запуску симуляції бізнес-процесу дослідник може визначити:

- кількість фактично виконаних замовлень, які постачаються кінцевому замовникові;
- середній час кожної операції;



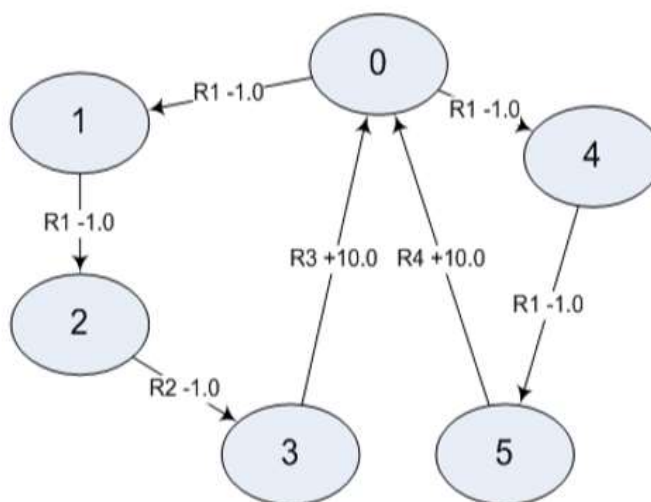
### 4.1.3. Скінченні автомати.

У статті І. Гниломедова [135] описано підхід до побудови агентних економічних моделей за допомогою скінченних автоматів. Автор переконує, що такий підхід може бути використаний для моделювання ринків абсолютної та монополістичної конкуренції.

В описі використання кінцевих автоматів для специфікації та симуляції моделі Вальраса визначено агента як певну пару:  $Agent = \langle FA, Start \rangle$ , де  $F, A$  – скінченні автомати, які специфікують виробництво, а  $Start$  – алгоритм, що керує виробництвом і відповідає за стратегію поведінки на ринку.

Автор статті виділяє три етапи емуляції моделі: реєстрацію пропозицій про продаж, купівлю ресурсів та виробництво. У процесі моделювання акцентовано такі траєкторії, як динаміка цін на товари, динаміка обсягу продажу й динаміка виробничих активів агентів.

Розглянемо скінченний автомат, що моделює виробництво кожного підприємця (рис. 4.10): вершина 0 – стартовий стан автомата, вершини 3, 5 – термінальні.



**Рис. 4.10.** Скінченний автомат, моделювання виробництва

На ребрах знаком «-» позначено витрати ресурсів, а знаком «+» – готову продукцію. Нехай наведений автомат виробництва належить агенту  $A$ . Нехай на ринку є ще й агенти  $\langle U, A \rangle$ . Їхні виробництва не відомі, але відомо, що вони



постачають на ринок ресурси  $R_1, R_2, R_3, R_4$ . Наведемо сценарій роботи цих агентів у двох виробничих циклах.

Реєстрація пропозицій із продажу. Нехай агенти  $B_1 \dots B_n$  в сукупності зареєстрували на ринку такі пропозиції:

$\langle R_1, price = 1.0 \text{ "}, " R_2, price = 2.0 \text{ "}\rangle$ ,

$\langle " U, price = 3.0 \text{ "}, " R_4, price = 10.0 \rangle$ ,

$\langle R_5, price = 4.0 \rangle$ .

Ухвалення рішення щодо роботи. Агент А, проаналізувавши стан ринку, вирішив, що шлях  $\{0, 1, 2, 3\}$  – максимально дешевий і слід обрати його. Автор наголошує, що зареєстровані на ринку пропозиції з продажу є таємницею для агента А, тому той керувався, наприклад, раніше накопиченою статистикою про ціни.

Стверджується, що, незважаючи на простоту поведінки кожного агента, результати емуляції можуть демонструвати цікаву поведінку моделі, але для отримання адекватних результатів необхідна велика кількість агентів.

Описана модель може використовувати ринки досконалої конкуренції та монополістичної конкуренції. Автор акцентує увагу на перспективності цього підходу, адже за успішного навчання моделі на даних із біржових торгів її можна буде імплементувати для прогнозів торгів на біржі.

У роботі Х. Г. Ціммерманна, Р. Нойнейера, Р. Гротмана [136] висвітлено агентно-орієнтовану модель валютного ринку на основі кінцевих автоматів. Процес складається з раундів, у кожному раунді курси валют фіксовані й однакові для всіх учасників.

Агенти ухвалюють рішення: яку валюту вони будуть продавати, яку – купувати й у якому обсязі. Після того, як всі рішення будуть прийняті, обчислюється надлишковий попит чи пропозиція за кожною з валют. Залежно від величини та знаку надлишку визначається зрушення цін на наступний раунд. У своїх рішеннях агенти керуються прогнозами новинних видань, причому кожному джерелу приписують індекс довіри, який ураховують під час власного прогнозування. Стратегію поведінки агентів (побудова довірчих індексів і

прийняття рішення, чого й скільки купувати) реалізовано за допомогою самонавчальної нейронної мережі.

#### 4.1.4. Нечітка логіка.

Р. Дрісеке й Д. Джайлз [137] розкривають поняття нечіткої логіки та її взаємозв'язок з економікою загалом, а також особливості розробки моделей, створених на засадах нечіткої логіки, і можливості застосування такого підходу до економіки.

У процесі розроблення економічних моделей досить складно повністю усунути невизначеність, а отже, абсолютно врахувати вплив усіх факторів. Коли ви працюєте з точними параметрами й системами, усе доволі просто. За принципом несумісності, для отримання певних висновків про поведінку складної системи слід долучитися до її аналізу, що й застосовують у принципах нечіткої логіки. Нечітка логіка – це галузь математики, яка займається складною класичною логікою та теорією нечітких множин. Основна характеристика теорії нечітких множин – маніпуляція компонентою мовної змінної.

Методи, що базуються на теорії нечітких множин, зараховують до методів оцінювання й прийняття рішень у разі невизначеності.

А.-А. Хасан Алі [138] стверджує, що економіка бізнесу – це багатofакторна система, поведінкові характеристики якої передбачити досить проблематично. Крім того, економіка досить чутлива до соціальних тенденцій галузі, і передбачити зміни в економічній діяльності під впливом зовнішніх та внутрішніх факторів у більшості випадків можливо лише в аспекті мовних або нечітких понять. З огляду на цей пріоритет, в економіці послуговуються підходами нечіткої логіки й нечіткого моделювання.

У праці Д. Немченко, В. Кобця та Л. Потравки [139] представлено досвід побудови моделі трансформації українського аграрного сектору з використанням нейронної мережі на нечітких даних. Для розробки застосовано нейромодель, нечітку логіку, оцінку ризику Воронова й Максимова. Під час дослідження побудовано нейромережеву модель «виробництва інфляції», яка використовує

вибірку з 32 квартальних даних за 2009–2016 рр., сформованих на підставі фінансової звітності ПЛК ТГ «Нова Троя» (Україна, Херсонська область, смт Новотроїцьке).

Автори зауважують, що реалізація такого підходу дозволяє здійснити всебічний фінансово-економічний аналіз підприємства з використанням інструментів нечіткої логіки, а отже, сформувати економіко-математичну модель з урахуванням специфіки підприємства. Наголошено, що цей підхід дає змогу розглянути різні типи невизначеності й отримати новий, якісно кращий прогноз розвитку.

#### **4.1.5. Машинне навчання.**

Використання машинного навчання для прогнозування попиту виробника в умовах конкурентного ринку описано в статті Х. Шторма, К. Бейліс і Т. Гекелея [140], основна мета якої – проаналізувати застосовність методів машинного навчання для прогнозування ланцюгових поставок.

Моделювання розширеного ланцюга поставок розроблено в *MATLAB Simulink* для дослідження ефективності прогнозування попиту. Величина попиту пропонується на основі алгоритмів машинного навчання.

Інший автор – А. Сторкі [141] – установлює придатність ринків машинного навчання для представлення за допомогою ринку цін, різних форм композиційної моделі машинного навчання. У його праці висвітлено багато композиційних структур, які зазвичай використовуються в машинному навчанні та інфекційних механізмах, таких як схеми передавання повідомлень.

Деякі ймовірнісні моделі машинного навчання можуть бути перетворені на набори незалежних агентів із певними функціями. Тоді будь-який вибір динаміки ринку може розглядатися як підхід до висновку, а отже, поширення інформації про витрати й інформацію про придбання можна розглядати як повідомлення, що передаються між незалежними агентами.

Перевага цього підходу полягає в тому, що дозволяє створити значно більш універсальні моделі, використовуючи кілька агентів із різними функціями

корисності. Ці агенти можуть функціювати самостійно й не потребують інформації про те, що інші агенти роблять, окрім інформації про ціни, за які ті готові продати ринкові товари.

#### 4.2. Системи економіко-математичного моделювання.

Програмний засіб *MAPLE* [142] – одна з найпотужніших систем, що базуються на теорії символьних перетворень. Це універсальна система символьного числення, яка широко застосовується для математичного моделювання в економіці, а також для моделювання економічної динаміки. Так, у статті З. Чваталової та І. Хребічека [143] представлено досвід використання цієї системи для обчислення фінансових стратегій. Автори доводять, що *MAPLE* може бути потужним інструментом аналізу інвестиційної діяльності й оцінки інвестиційних стратегій.

В іншій праці [144] І. Хребічек описує переваги застосування цієї системи як основного інструмента аналізу економічних явищ. З'ясовано, що програмний засіб має потужний функціонал для побудови різного виду графіків, що, звісно, сприяє кращому зрозумінню різних аспектів економічних явищ.

Робота з цим програмним продуктом проходить інтерактивно, у системі *MAPLE* не потрібна жорстка формалізація всіх змінних та їхніх типів, оскільки вибір відповідних типів змінних забезпечується автоматично й перевіряється коректність виконання операцій. Ядро розроблено мовою програмування *C*.

Для розроблення безперервних моделей широко вживають програмний засіб *POWERSIM*. Зауважимо, що в аспекті створення дискретних моделей ця система менш ефективна. Й. Людике [145] описує її застосування для під час аналізування лінійної регресії та висноує, що система буде ефективною в експериментах із використанням методу Монте-Карло, а також вона підтримує широкий спектр одноваріантних і багатоваріантних коваріантних розподілів.

Пакетом *POWERSIM* можна послуговуватися під час виконання моделей з інструментами анімацій. Такі моделі розробляють мовою *Powersim*, будь-яка з

них складається з безлічі взаємозалежних елементів, що описують змінними. Елементи моделі та зв'язок між ними визначають структуру моделі.

Програмний продукт *iTHINK* [146] застосовується для моделювання системної динаміки. Моделі, розроблені засобами цього програмного продукту, складаються з підрівнів моделей, тобто користувач формалізує перший рівень абстракції, після чого переходить на новий рівень деталізації моделі. Між рівнями абстракції встановлюються зв'язки. Моделі формалізуються засобами вбудованої графічної мови.

*ARENA*, за Д. Келтоном [147], – програмний продукт, що широко використовується для імітаційного моделювання. Основний інструмент розроблення моделей в цій системі – мова моделювання *SIMAN*. К. Пірнау [148] зазначає, що система може бути досить ефективною для побудови й аналізу біоекономічних моделей.

К. Джилча, Е. Берхан і Х. Шериф [149] вивчають моделювання систем масового обслуговування. Автори зазначають, що за допомогою *ARENA* знайдено критично важливі місця, які вможливили оптимізацію і процесу виготовлення товарів кінцевого вжитку, і пропускну здатності; у подальшому це дало змогу ефективно налаштувати бізнес-процеси підприємства. Спочатку користувач крок за кроком будує у візуальному редакторі системи модель, потім система генерує за нею відповідний код на *SIMAN*, після чого автоматично запускає розроблену модель.

Система *GPSS WORLD* [150] – це потужне середовище комп'ютерного моделювання загального призначення, комплексний інструмент моделювання, який охоплює області як дискретного, так і безперервного комп'ютерного моделювання.

С. Власов, Т. Дев'ятков і В. Дев'ятков [151] описують досвід створення віртуальної лабораторії симуляційних досліджень, побудованих на віртуальних машинах. В опублікованій статті вказано, що розроблено модель аналізу технологій суднобудування, модель оптимізації виробництва для заводу феросплавів, модель аналізу транспортної логістики, а також модель оцінки

потужностей заводу та пошук найкращих варіантів виробництва. GPSS World є найбільш сучасною реалізацією мови GPSS, доповненої допоміжною мовою PLUS (Programming Language Under Simulation).

Дві групи авторів [152; 153] досліджують програмний засіб *ABLE* (Agent Building and Learning Environment), який ґрунтується на теорії інтелектуальних агентів і методів машинного навчання. Цей продукт активно використовують для класифікації, кластеризації та прогнозування. Він базується на мові ARL (ABLE Rule Language) – це досить потужна мова, якою можна реалізувати як прості процедурні програми, так і складні сценарії з прямим та зворотним умовиводами (на засадах алгоритмів Rule).

Зразком мультиагентного моделювання є програмний додаток *Adaptive Modeler* [154], досить успішно використовуваний під час створення імітаційних моделей для фінансових ринків. Інструментарій для побудови моделей, розроблений компанією Reticular Systems, Inc., складається з двох компонентів: 1) засобів розробки (development tools); 2) середовища оточення (run-time execution environment). Перший компонент орієнтований на підтримання процесів аналізу предметної області створюваної моделі і проектування агентів із заданою поведінкою. Другий забезпечує ефективне середовище розробки й виконання агентно-орієнтованих програм. Обидва компоненти реалізовані мовою Java, що дозволяє їм працювати на всіх платформах, де встановлено Java-середовище. Агентні програми, проєктовані в рамках AgentBuilder, також є Java-програмами й можуть виконуватися на будь-якому комп'ютері, де встановлено віртуальну Java-машину JVM (Java virtual machine).

Для імітаційного моделювання застосовують *AnyLOGIC* [155] – програмне забезпечення, що має сучасний графічний інтерфейс і дозволяє використовувати мову Java для розроблення моделей. Продукт отримав назву AnyLogic, тому що є першим і єдиним інструментом імітаційного моделювання, який об'єднав методи системної динаміки, «процесного» дискретно-подієвого й агентного моделювання в одній мові та одному середовищі розроблення моделей.

Ц. Рен, Ц. Чжао, Р. Лю, Ч. Чень [156] застосовують імітаційний підхід до моделювання ланцюжка поставок сировини для виробництва. Розробники визначають за таким підходом можливість виявити взаємозв'язок між факторами впливу на процес постачання та спостерігати динаміку змін цих поставок.

Натомість Б. Ду й Х. Ліу [157] послуговуються AnyLogic для аналізування та моделювання регіональної логістичної системи. Автори використовують принципи системної динаміки для створення системної динамічної моделі регіональної логістики. У результаті моделювання виявлено, що місто повинно виділяти 8,5 % свого ВВП на будівництво логістичної інфраструктури та 8 % – на підготовку професійних кадрів.

AnyLogic підтримує безліч різноманітних типів експериментів із моделями: простий запуск, порівняння результатів запуску, варіювання параметрів Монте-Карло, аналіз чутливості, оптимізацію, калібрування, а також довільні експерименти за призначеними для користувачів сценаріями.

Нова Java-версія OptQuest™ від компанії OptTek, Inc., вбудована в AnyLogic, містить засоби аналізу даних і великий набір елементів бізнес-графіки, спроектованих для ефективного обробки й презентації результатів моделювання: статистики, набори даних, графіки, діаграми, гістограми.

*ASCAPE* [158] – це інноваційний інструмент для розроблення та дослідження моделей на основі агентів загального призначення. Він розроблений як гнучкий і потужний, але водночас доступний та простий у використанні інструмент моделювання.

Програмна мова є досить потужною, розроблені моделі займають мінімальну кількість програмного коду, коригувати програмний код досить легко. Система пропонує широкий спектр засобів моделювання й візуалізації, випускається під стандартною ліцензією BSD із відкритим вихідним кодом, вільна для використання.

*MATLAB* [159] – це мова високого рівня, що містить матричні структури даних, власні внутрішні типи даних, великий каталог функцій, середовище для розроблення власних функцій і скриптів, пропонує можливість імпорту й

експорту, можливості об'єктно-орієнтованого програмування та інтерфейси для зовнішніх технологій, таких як COM, Java, програми, написані мовами C і Fortran, і пристрої з послідовним портом (Windows, Linux, Solaris, Macintosh); постійно оновлюється.

Про аналіз нелінійних змінних характеристик економічної системи й реалізацію нелінійної економічної моделі за допомогою MATLAB пишуть Й. Лі та С. Малін [160], зауважуючи, що, крім того, підтримується економетрична модель у структурній формі, функції передачі й загальні лінійні моделі простору станів. Вбудована графіка дозволяє легко візуалізувати інформацію з даних.

Система MATLAB може бути використана для:

- навчання методам симуляційного програмування;
- наукової та інженерної математики;
- обчислювальної техніки;
- аналізу, дослідження й візуалізації даних.

Продукт *NetLOGO* [161] був розроблений У. Віленським у дусі мови програмування Logo для моделювання пов'язаних явищ, наприклад, для навчання великих аудиторій, зокрема дітей в освітньому середовищі або експертів, які не мають досвіду програмування. Ця система навчає концепціям програмування з використанням агентів у вигляді черепах, латок, посилань і спостерігача.

Б. Хед, А. Хьорт, К. Брейді й У. Віленський [162] представляють досвід використання методів імітації агентів на основі складних когнітивних моделей, що розвиваються. Автори наводять як приклад модель «Хижак–Жертва», кожний інтелектуальний агент представлений як незалежний екземпляр моделі нейронної мережі NetLOGO. Крім того, як указує Я. Тіле [163], система дає змогу інтегруватися з мовою програмування R для більш якісного й системного аналізу експериментів.

Інші науковці описують [164] застосування нечіткої логіки в системі агентного моделювання NetLOGO. Реалізація такого підходу вможливує розгляд різних типів поведінки інтелектуальних агентів в умовах невизначеності, а це, зі свого боку, – отримання нових якісно кращих результатів експериментів.



Крім дослідження, система дозволяє створювати нові моделі й модифікувати вже створені. Додамо, що продукт є відкритим кодом і вільно доступний на вебсайті NetLOGO.

*Bizagi Studio* [134] – система для моделювання та формалізації бізнес-процесів, що використовує мову BPMN. Ця система має в своєму арсеналі не тільки можливість формалізації, а й візуалізації та симуляції формалізованого бізнес-процесу. Застосовуючи її, можна також оцінити складність і вартість бізнес-процесу, оскільки передбачено потребу вказати кількість осіб, задіяних у процесі, кількість часу та заробітну плату в контексті певного бізнес-процесу.

Система надає право зберігати експерименти й результати моделювання, що може допомогти розробникові процесу аналізувати та виокремлювати найбільш ефективний бізнес-процес.

Так, А. Копп і Д. Орловський [165] досліджують методи аналізу й удосконалення моделі бізнес-процесів, життєвий цикл управління бізнес-процесами – від моделювання процесу до застосування бізнес-аналітики на прикладі процесу видобутку корисних копалин. Науковці пропонують метод аналізування моделі бізнес-процесів та його подальшого вдосконалення. Цей метод заснований на реалізації найкращих практик формалізації процесів із використанням відповідних метрик і відповідних порогових значень для подальшого виявлення й усунення порушень правильності побудови моделі бізнес-процесів.

#### **4.3. Інсерційна семантика в економічних дослідженнях. Агенти в економічних процесах.**

Агент – це транзиційна система, стан якої визначається з точністю до бісимуляційної еквівалентності [77]. Більш детально дефінування агента в теорії інсерційного моделювання подано у підрозділі 2.1., наразі розглянемо поняття агента з погляду економічного моделювання.

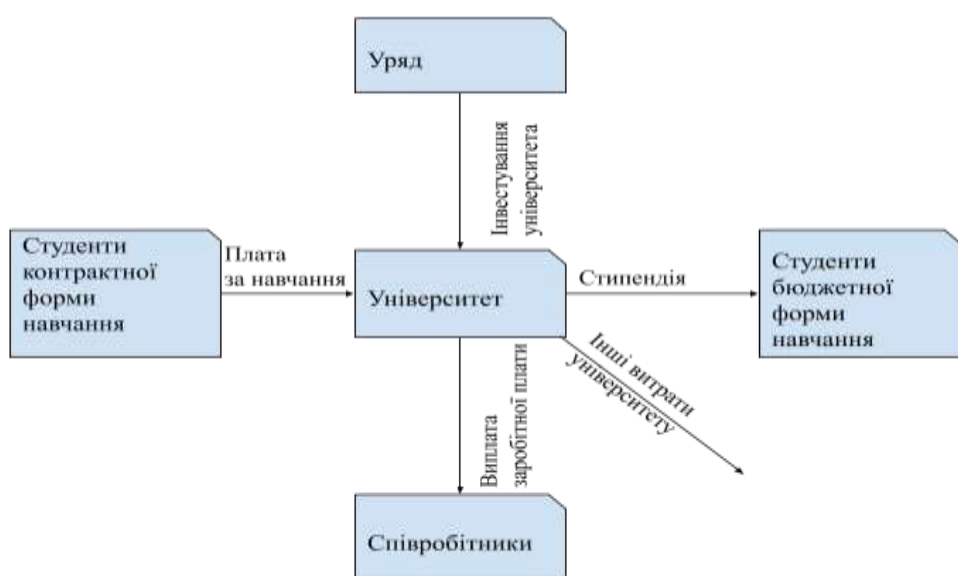
У нашій статті (у співавторстві) [166] описано підхід до моделювання й аналізування економічних процесів університету. Зважаючи на умови моделювання, ми виділяємо 5 типів агентів, які можуть взаємодіяти між собою в процесі симуляції моделі. У моделі можливо виокремити таких агентів:

- Уряд;
- Університет;
- Робітники;
- Студенти (студенти контрактної форми навчання, студенти бюджетної форми навчання).

Кожен агент може взаємодіяти з іншими агентами. Специфіка моделі зумовлює певний список дій:

1. Держава фінансує університет.
2. Університет виплачує заробітну плату викладачам і співробітникам.
3. Університет виплачує стипендію студентам.
4. Студенти контрактної форми навчання платять за навчання.
5. Інші витрати університету: комунальні послуги, незаплановані витрати.

Агентів та взаємозв'язок між ними можна відобразити на діаграмі взаємодії агентів (рис. 4.11).



**Рис. 4.11.** Діаграма взаємодії агентів

Агент типу «університет» можна представити в інсерційному моделюванні таким чином:

```

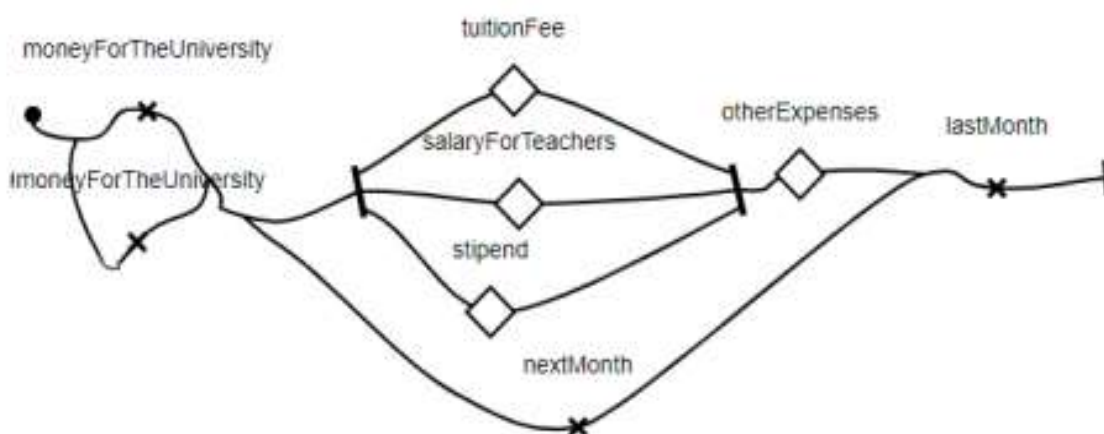
University: obj (
    Score:(ScoreType)→real
    .....
)

```

Зауважимо, що специфіка моделі передбачає для університету декілька типів рахунків (спеціальний, глобальний), тому ми створимо спеціальний перелічувальний тип *ScoreType*:

```
ScoreType:( Total, Special);
```

Важливою особливістю інсерційного моделювання є те, що середовище може бути агентом для середовища вищої ієрархії. Розглянемо цей факт у рамках моделі економіки університету. Для агентів типу «студенти» середовищем може бути агент типу «факультет», але факультет можливо розглядати як агента для середовища «університет». Відповідно до вимог моделі розроблено алгебру поведінки, яка відбиває поведінку агентів у цій моделі. На рис. 4.12 зображено поведінку агентів за допомогою мови *UCM*.



**Рис. 4.12.** UCM-діаграма поведінки агентів  
для моделі економіки університету.

Поведінку можна специфікувати за допомогою алгебри поведінки в алгебраїчному вигляді, що буде еквівалентно до графічного представлення. Нижче наведено приклад поведінки в алгебраїчному вигляді:

$$b0 = (Investment . DistributionMoney).b1 + not\_Investment . b1,$$

$$b1 = ($$

$$(Salary + not\_Salary) //$$

$$(TuitionFees + not\_TuitionFees) //$$

$$(Scholarship + not\_Scholarship)$$

$$).b2,$$

$$b2 = PaymentUtilityBills . b3,$$

.....

$$bn = NextMonth . b1 + LastMont.Delata$$

Розглянемо більш детально елементи алгебри поведінки:

- *Investment, DistributionMoney* – дія, що відбувається тільки один раз протягом першого місяця ( $month == 1$ ). Специфікує інвестування й розподіл коштів в університеті;

- *Salary, TuitionFees, Scholarship* – дії, виконувані паралельно одна до одної, специфікують виплату заробітної плати, оплату за навчання та стипендії. Дія виконується відповідно до визначених періодів.

- *PaymentUtilityBills* – дія, що виконується регулярно кожного місяця моделювання – оплата за комунальні послуги.

#### 4.4. Задача Вальраса й аналіз умови рівноваги.

Вальрас виділяє два суб'єкта взаємодії: фірми та домогосподарства. Фірми виступають на ринку факторів як покупці, на ринку споживчих товарів – як продавці[167].

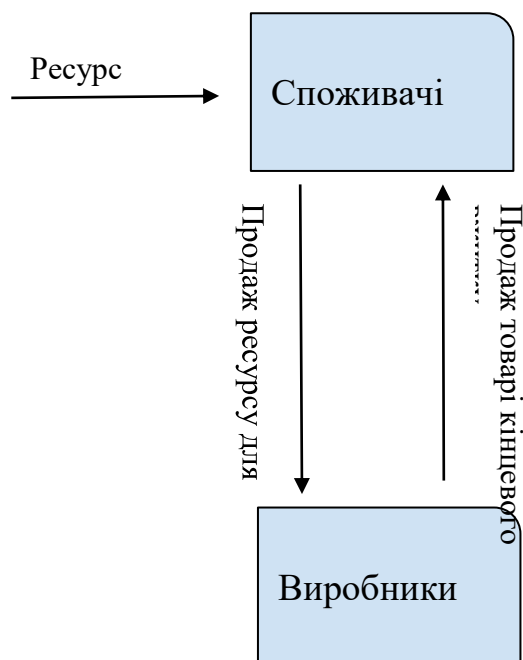
Домашні господарства – власники факторів виробництва, виступають в ролі їхніх продавців і час водночас – покупців споживчих товарів. Ролі продавців і покупців постійно змінюються. У процесі взаємодії витрати виробників товарів перетворюються на доходи домогосподарств – споживачів кінцевого товару, а всі витрати домогосподарств – на доходи виробників (фірм).

Ціни економічних чинників залежать від розмірів виробництва, попиту, а значить, від цін на вироблені товари. У свою чергу, ціни на товари залежать від цін на фактори виробництва. Останні повинні відповідати витратам фірм. Водночас доходи фірм повинні поєднуватися з витратами домогосподарств.

Побудувавши складну систему взаємопов'язаних рівнянь, Вальрас доводить, що система рівноваги може бути досяжна як певний «ідеал», якого прагне конкурентний ринок. Положення, що отримало назву закону Вальраса, стверджує: у стані рівноваги ринкова ціна дорівнює граничним витратам. Таким чином, вартість суспільного продукту дорівнює ринковій вартості використаних для його випуску виробничих факторів; сукупний попит дорівнює сукупній пропозиції; ціна та обсяг виробництва не збільшуються й не зменшуються.

Побудована на основі цієї теоретичної концепції модель Вальраса – це модель загальної економічної рівноваги, свого роду одномоментний знімок національного господарства в «чистому» вигляді. Щодо стану рівноваги, то, на думку автора, він передбачає наявність трьох умов:

Взаємодію агентів більш детально подано на рис. 4.13:



**Рис. 4.13.** Діаграма взаємодії агентів у моделі Вальраса

Розглянемо вимоги до моделі:

1. Нехай ми маємо  $N$  споживачів та  $M$  виробників.

2. Споживачі мають виробничий ресурс, який потрібний для виготовлення товару кінцевого вжитку.

3. Виробники, купуючи виробничий ресурс, виготовляють товар кінцевого вжитку (хліб,...).

4. Для виготовлення товару кінцевого призначення необхідний відповідний ресурс.

5. Ціна товару кінцевого вжитку *Price*.

6. Функція корисності: кількість товарів кінцевого вжитку, необхідних кожному споживачеві в поточний період.

7. Кожний виробник має план підприємства – кількість товарів кінцевого вжитку, яке може виробити виробник за власні гроші.

8. Кожен виробник може коригувати свій план згідно з результатом купівлі товару кінцевого вжитку відповідного поточного періоду.

Також акцентуємо на певних особливостях, які потрібно врахувати під час розроблення моделі:

- Будемо вважати, що на початковому етапі пропозиція товару кінцевого вжитку не перевищує попит на цей товар, тобто сума всього виробленого товару більше від суми необхідних транзакцій із купівлі товару .

- Кожен виробник і споживач мають початкову суму грошей ( $money > 0$ ).

- Існує деякий початковий план виробництва.

Крім того, під час формалізації вимог до моделі ми виокремили низку неточностей, які в класичному, не агентному трактуванні цієї моделі можуть призвести до появи недетермінізмів і безвихідних ситуацій. Розглянемо деякі з них:

- Якщо в споживача немає грошей на заплановані ресурси, скільки товару він купує?

- Якщо в споживача бракує грошей на купівлю запланованої кількості товарів кінцевого вжитку, скільки він купує?

Ми припускаємо, що за умови браку коштів на задоволення власних потреб учасники економічної системи купують товар відповідно до своєї купівельної спроможності.

Зважаючи на вимоги до моделі, виокремлюємо два типи агентів: *Customer* і *Manufacturer*, – та представляємо їх у системі інсерційного моделювання:

*Customer: obj*(

*Money:real*

)

*Money* – атрибут типу *real*, який визначає кількість доступних грошей в агента *Customer*.

*Manufacturer: obj*(

*Plan:real,*

*Money:real*

)

*Plan* – атрибут типу *real*, що визначає кількість товарів, яку може виготовити виробник за свою кількість грошей.

*Money* – атрибут типу *real*, який визначає кількість доступних грошей в агента *Manufacturer*.

Далі розглянемо список глобальних атрибутів середовища:

- *CurrentStage* – поточний період моделювання.
- *LastStage* – крайній період моделювання.
- *DeltaUp* – величина, на яку збільшуємо виробничий план.
- *DeltaDown* – величина, на яку зменшуємо виробничий план.
- *GeneralPlan* – кількість виготовленого товару кінцевого вжитку (ринкова пропозиція).
- *GeneralGoods* – кількість купленого товару кінцевого вжитку (ринковий попит).
- *Resource* – кількість ресурсу.
- *Price* – ціна на товар кінцевого вжитку.

- *Use* – функція корисності, кількість товарів, необхідних кожному споживачеві.

Збільшення або зменшення на фіксовану величину у відсотках від кількості виробленого товару – *DeltaUp* та *DeltaDown*.

Як уже зазначалося раніше, для зручності розроблення специфікації мовою дій перетворюємо текстові конструкції вимог у проміжний варіант, близький до конструкції *if then*:

1. *If (виробникові вистачає грошей на купівлю ресурсу для запланованого плану) then (купувати ресурс на заплановану кількість продукції).*

Наголошуємо, що класичне трактування моделі Вальраса–Маршала є неповним, оскільки не розглядається випадок, коли у виробника немає грошей на купівлю запланованого ресурсу. У цьому варіанті визначимо правило, що виробник у разі браку коштів буде купувати ресурс на стільки, на скільки йому вистачить грошей:

1. *If (виробникові бракує грошей на купівлю ресурсу для запланованого плану) then (купувати ресурсу на доступні гроші).*
2. *If (споживачеві грошей вистачає на купівлю товару відповідно до функції корисності) then (купувати ресурс відповідно до функції корисності).*

Також відмітимо, що випадок, за якого споживачеві бракує коштів на задоволення своєї базової потреби (функції корисності), не розглядається. У нашому випадку варіанті за такої умови споживач купує товар на свої доступні кошти:

1. *If (споживачеві бракує грошей на купівлю товару відповідно до функції корисності) then (купувати ресурс на доступні гроші).*
2. *If (Продана кількість товарів менша від плану в поточний період) then (зменшити обсяг виготовлення на *DeltaDown*).*
3. *If (Продана кількість товарів перевищує план в поточний період) then (збільшити обсяг виготовлення на *DeltaUp*).*

Сформулюємо умову рівноваги в моделі: пропозиція товарів кінцевого вжитку в грошовому вираженні має дорівнювати загальному попиту на них як



сума доходів, що приносять усі фактори виробництва їхнім власникам. Рівновага, за Вальрасом, буде тоді й тільки тоді, коли кількість товарів кінцевого вжитку, виготовлених виробниками, буде дорівнювати кількості товарів, які купили споживачі (рівна функції корисності). Тоді умову рівноваги можна задати за допомогою умови безпеки:

$$GeneralPlan == GeneralGoods$$

Далі нормалізуємо наш проміжний варіант у мову дій. Деякі з цих вимог представлено нижче в таблиці 4.

Таблиця 4.1

### Специфікація мовою дій

№	Специфікація мовою дій	Опис
1	$BuyResources = (Forall(1 \leq i \leq M)$ $(Manufacturer(i).Money >$ $Manufacturer(i).Plan * Resource) \rightarrow$ $(< "Buy Resources" >)$ $(Manufacturer(i).Money =$ $Manufacturer(i).Money -$ $Manufacturer(i).Plan * Resource;$ $GeneralPlan = GeneralPlan +$ $Manufacturer(i).Plan))$	Продаж ресурсів виробникам за умови, що виробникам вистачає грошей на купівлю ресурсу для запланованого плану.
2	$notAllBuyResources = (Forall(1 \leq i \leq M)$ $(Manufacturer(i).Money <$ $Manufacturer(i).Plan * Resource) \rightarrow$ $(< "Buy Resources" >)$ $(Manufacturer(i).Money = 0;$ $GeneralPlan = GeneralPlan +$ $Manufacturer(i).Money / Resource;$ $Manufacturer(i).Plan =$ $Manufacturer(i).Money / Resource))$	Продаж ресурсів виробникам за умови, що виробникам бракує грошей на купівлю ресурсу для запланованого плану.
3	$BuyGoods = (Forall(1 \leq i \leq N)$	Покупка товару відповідно до функції корисності

	$(Consumer(i).Money > Use(i)*Price) \rightarrow$ $(\langle "Buy\ Goods" \rangle)$ $(Consumer(i).Money =$ $Consumer(i).Money - Use(i)*Price;$ $GeneralGoods = GeneralGoods + Use(i))$ $)$	
4	<b><i>AdjustmentPlanUP</i></b> =( $Forall(1 \leq i \leq M)$ $(GeneralGoods*$ $Manufacturer(i).Plan/GeneralPlan >$ $Manufacturer(i).Plan) \rightarrow$ $(\langle "Adjustment\ of\ the\ Plan" \rangle)$ $(Manufacturer(i).Plan = =$ $Manufacturer(i).Plan*DeltaUP))$	Збільшуємо виробничий план на величину <i>DeltaUP</i>
5	<b><i>AdjustmentPlanDOWN</i></b> =( $Forall(1 \leq i \leq M)$ $(GeneralGoods*$ $Manufacturer(i).Plan/GeneralPlan >$ $Manufacturer(i).Plan) \rightarrow$ $(\langle "Adjustment\ of\ the\ Plan" \rangle)$ $(Manufacturer(i).Plan = =$ $Manufacturer(i).Plan*DeltaDown))$	Зменшуємо виробничий план на величину <i>DeltaDOWN</i>

Сценарій поведінки агентів представляємо за допомогою системи рівностей алгебри поведінки:

$$Bo = BuyResources . b1 + not\_BuyResources . b1,$$

$$b1 = BuyGoods . b2 + not\_BuyGoods . b2,$$

$$b2 = Money.b3,$$

$$b3 = PlanAdjustmentDOWN . b4 + PlanAdjustmentUP . b4,$$

$$b4 = NextIteration . Bo + EndIteration . Delta$$

#### 4.4.1.Результати моделювання.

На рис 4.15 представлено *MSC*-діаграму моделі. Такий тип діаграми широко застосовується для представлення вимог, специфікації систем із метою їхнього аналізу.

На відміну від мов програмування, ці мови не призначено для кодування системи, а використовуються для відображення архітектури, системи, що розробляється, вони більш високого рівня абстракції, ніж мови програмування.

*Середовище MSC.* Стандартна семантика діаграм *MSC* визначається як алгебра процесу, *MSC* відображає сукупність слідів процесу над безліччю використовуваних подій у *MSC*.

Цей тип діаграми досить інформативно ілюструє поведінку агентів в економічній системі й поведіку моделі загалом. Надалі більш детально розглянемо отриману *MSC*-діаграму нашої моделі (рис. 4.14).

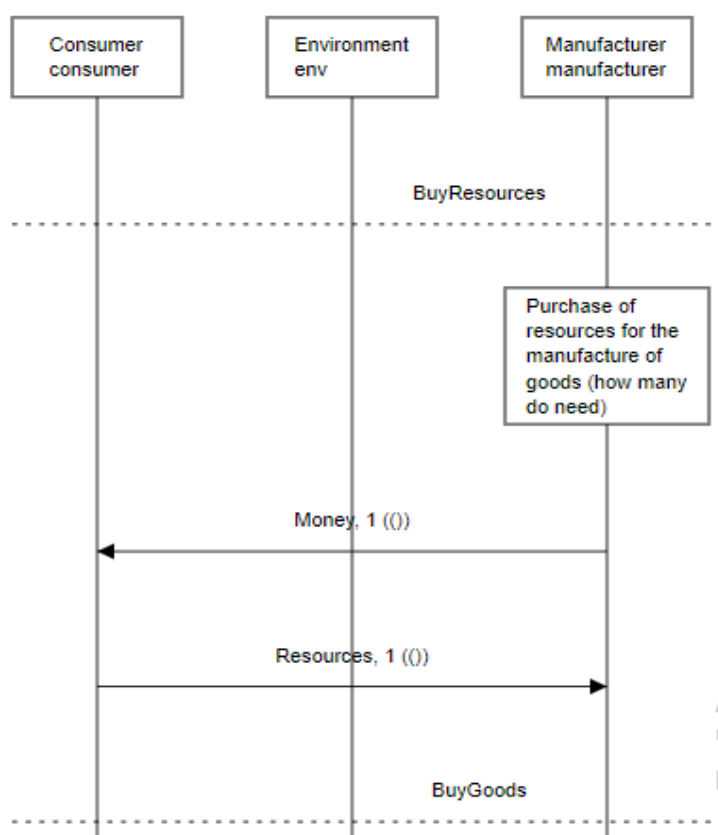


Рис. 4.14. *MSC*-діаграма для моделі Вальраса

Відповідно до розробленої раніше специфікації в моделі взаємодіють 2 типи агентів:

1. *Constomer (Constomer # constomer).*
2. *Manufacturer(Manufacturer # manufacturer).*

Цей факт відображено на *MSC*-діаграмі у вигляді відповідних інстанцій, еквівалентних зі списком агентів у специфікації. Агенти-виробники та споживачі взаємодіють між собою, купуючи ресурс для виготовлення товару – на *MSC*-діаграмі це відбито відповідними стрілками.

```

proc(B)=(
  (Manufacturer # manufacturer:instance).(
  (Manufacturer # manufacturer: Investment
  (Money) to Constomer # constomer))
  (Constomer # constomer:instance).(
  (Constomer # constomer: (money)
  from Manufacturer # manufacturer)))

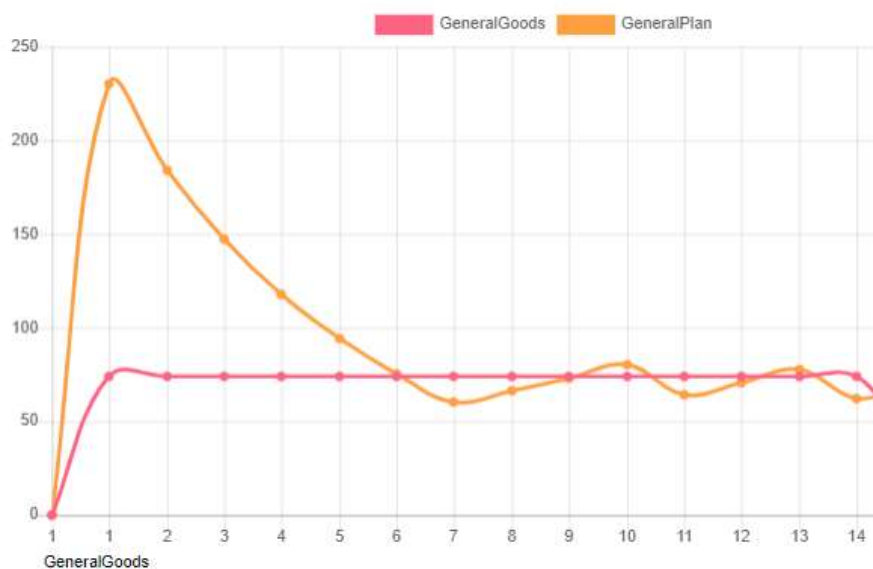
```

Діаграма *MSC* дає змогу досить інформативно візуалізувати поведінку агентів у досліджуваній моделі, такий тип діаграми можливо використовувати для аналізу й формального представлення поведінки економічної моделі, а також використовувати як механізм формального представлення аналізу.

Під час роботи над конкретною моделлю Маршала ми отримали графік (див. рис. 4.15), який репрезентує дві траєкторії:

- траєкторію ринкового попиту;
- траєкторію ринкової пропозиції.

Оскільки графік перетинається, ми можемо говорити про наявність рівноважної кількості продукції. Якщо на першій ітерації (місяці) виробники запланували досить багато продукції, більше ніж потребують споживачі, то після купівлі товару у виробників залишився надлишок товару, тому вони повинні зменшити свій товар.



**Рис. 4.15.** Траєкторія ринкового попиту до ринкової пропозиції

Оскільки на графік на рис. 4.15 перетинається, ми можемо говорити про наявність рівноважної кількості продукції. Оскільки на першій ітерації (місяці) виробники запланували досить багато продукції, більше ніж потребують споживачі, то після купівлі товару у виробників залишився надлишок товару, тому вони повинні зменшити свій товар.

На графіку досить чітко видно, що на шостій ітерації виробники запланували товару рівно стільки, скільки потрібно споживачам. Таким чином, за допомогою конкретної моделі можна показати досяжність рівноважної кількості товару.

Важливою особливістю інсерційного моделювання є можливість створювати символічні моделі. Наприклад, створимо модель, де значення змінних будуть задаватися не конкретними значеннями, а інтервалами, або деякими логічними виразами:

*currentStage* == 1 &&

*lastStage* == 15 &&

*deltaUp* == 1.1 &&

*deltaDown* == 0.8 &&

*generalPlan* == 0 &&

*generalGoods* == 0 &&

```

resource > 0 && price > 0 && price != resource &&
C == 5 && M == 5 &&
Forall(i:int)(
  i >= 1 && i <= C ->
  Consumer(i).money > 0 &&
  Manufacturer(i).money > 0 &&
  0 < Manufacturer(i).plan <= 100 &&
  use(i) == 15
)

```

Задавши такі початкові значення, у процесі моделювання ми отримуємо наступне символічне представлення моделі Вальраса:

```

Forall(i:int)(
  (i >= 1 && i <= M) ->
  attr_Manufacturer_money(i) + resource * 21 + price * (-22) == 100
) &&
Forall(i:int)(
  (i >= 1 && i <= C) ->
  attr_Consumer_money(i) + resource * (-21) + price * (22) == 100 &&
  310 > attr_Consumer_money(i) > (-210) &&
) &&
generalPlan == 55 &&
generalGoods == 55 &&
Forall(i:int)(
  (i >= 1 && i <= M) ->
  attr_Manufacturer_plan(i) == 11 &&
  410 > attr_Manufacturer_money(i) > (-110)
) &&
0 != price + resource * (-1) &&
resource * 21 + price * (-22) > (-100) &&
resource * (-21) + price * 11 > (-100) &&

```

$resource > 0 \ \&\& \ price > 0 \ \&\& \ price < 14.0909 \ \&\&$   
 $resource * (-21) + price * 11 < 155 \ \&\&$   
 $resource < 10$

Під час моделювання можна побачити, що  $generalPlan == generalGoods$

- $generalPlan == 55 \ \&\& \ \forall (1 \leq i \leq M) ($   
 $attr\_Manufacturer\_plan(i) == 11$   
 $)$
- $generalGoods = 55 \ \&\& \ \forall (1 \leq i \leq C) ($   
 $use(i) == 11$   
 $)$

Оскільки  $generalPlan == generalGoods$  – можна говорити про наявність рівноваги в моделі.

#### 4.5. Токеноміка.

Поняття токеноміки неможливо розглядати, не засвоївши попередньо поняття блокчейну. Узагалі, блокчейн – це деяка неперервна послідовність (ланцюжок) блоків, де подано інформацію про певні цифрові угоди, які представлено за допомогою смарт-контрактів. Угоди можуть бути різними – від купівлі до юридичного контракту. Смарт-контракт частіше за все реалізують за допомогою мови програмування *Solidity* [168, 170].

Основна особливість блокчейну полягає в тому, що інформація про здійснені операції одночасно зберігається в усіх користувачів мережі, ця інформація постійно оновлюється, усі оновлення отримують учасники мережі. Якщо хто-небудь намагається неправомірно використовувати інформацію з системи (видалити, перезаписати чи оновити інформацію в ланцюгу блоків), система зможе це ідентифікувати, оскільки цю інформацію має в розпорядженні кожен учасник мережі. Крім того, до кожного блоку додається хеш-сума. Ці блоки розташовані в певному порядку та складаються в ланцюжки. Якщо спробувати

переставити послідовність блоків, то система відкине ланцюг через невідповідність структури й хеш-суми.

Токени – цифрові одиниці обліку, що використовуються для представлення цінності певного цифрового активу[171, 172]. На відміну від децентралізованих криптовалют, токени випускаються на базі вже наявного блокчейна або протоколу, тому вони можуть бути випущені організацією, компанією або приватною особою для того, щоб виконувати певний набір функцій.

Токеноміка – це система утворення цінової політики токенів.

Властивості, що досліджуються:

1. Причини високої спекулятивної діяльності – можливо досліджувати на основі моделі *NVT (Network Value to Transactions Ratio)*.

2. Причини виникнення «мильного пузиря» – також досліджуються на основі *NVT*-моделі.

3. Ціна на токен повинна мати постійний ріст.

Зауважимо: якщо ціна на токен буде опускатися нижче від початкової ціни, це свідчить про колапс токеномічної моделі, при цьому швидкий ріст ціни на токен може призвести до випадку, коли ціна на токен вища за купівельну спроможність учасників токеномічної моделі.

4. Поведінка спекулянтів у системі.

Крім того, існують певні токеномічні закони, які потрібно досліджувати для ефективної побудови токеномічних моделей:

1. *Закон №1*: Після повного запуску токена значення *R (Ratio of Transaction Volume to Transaction Market Cap)* має бути значно нижче за 10. Співвідношення обсягу транзакції в рік до ринкової капіталізації транзакції.

2. *Закон №2*: Значення *HT (Hold Time – час, який учасники токеномічної моделі тримають токени)* повинно бути максимальним.

3. *Закон №3*:  $HT (Hold Time) > TT (Transaction Time)$  – час, який учасники токеномічної моделі тримають токени, має бути більший, ніж час виконання транзакції з токеном.



4. *Закон №4*: Повинна бути причина, щоб купувати токени, утворити дефіцит токенів у моделі.

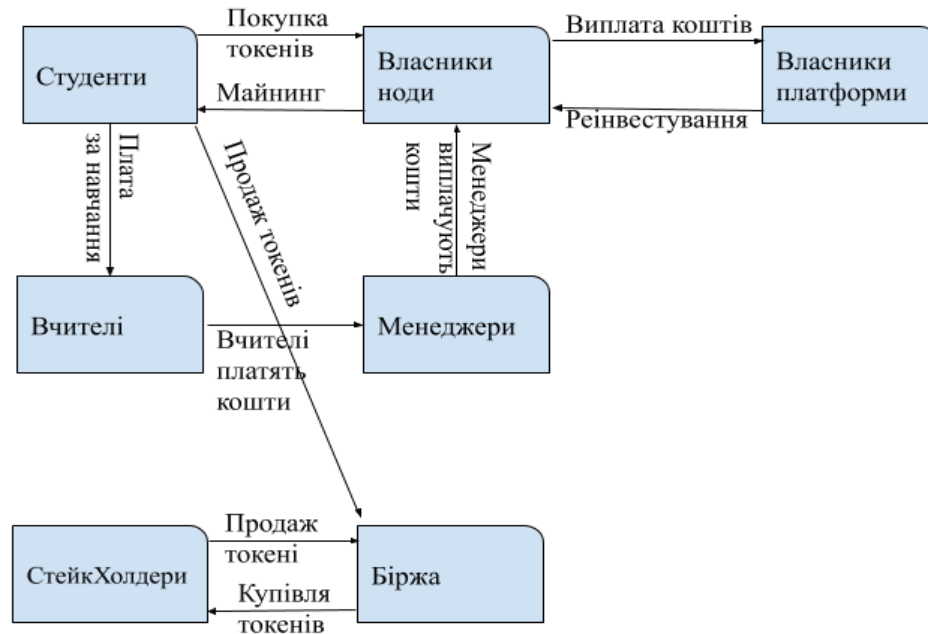
5. *Закон №5*: Наявність постійного рівномірного росту [173].

Розглянемо приклад формалізації та аналізу токеноміки на прикладі моделі Skillonomy[2]. Модель Skillonomy – це освітня платформа, що організовується у формі циклу, який повторюється щомісяця. Ознайомитися зі специфікою моделі можна в нашій праці (у співавторстві) «Formal Verification of Token Economy Models» [174].

Зважаючи на вимоги до моделі, ми виділили одинадцять типів агентів:

- Студенти, середня оцінка яких дорівнює 1.
- Студенти, середня оцінка яких дорівнює 2.
- Студенти, середня оцінка яких дорівнює 3.
- Студенти, середня оцінка яких дорівнює 4.
- Студенти, середня оцінка яких між 4 та 5.
- Студенти, оцінка яких дорівнює 5.
- Учителі.
- Менеджери.
- Власники вузлів.
- Власник освітньої платформи.
- Стейкхолдери.

Кожен з одинадцяти типів агентів виконує певні дії: купівля токенів, емісія, навчання та інше. Детальні взаємозв'язки й процедури взаємодії між агентами проілюстровано на рис. 4.16:



**Рис. 4.16.** Діаграма взаємодії агентів моделі SKILLONOMY

Агент може бути представлений у системі APS (IMS) як об'єкт із деякою кількістю атрибутів. Зважаючи на вимоги до моделі, ми розробляємо багаторівневу структуру. Розглянемо реалізацію агента типу «власник платформи»:

```

PLATFORM_OWNER: obj(
  SMInvestToken: (int) -> real,
  tokenICOSTageEmission: (ICO_STAGE) -> real,
  commonEmission: real,
  RESERVE: real,
  income: real,
  tokenSkillMining: real
  .....
)

```

Агент має такий список атрибутів:

- *commonEmission* – загальна кількість емісії в токенах, зарезервована платформою на весь період проєкту;
- *tokenICOSTageForSale(ICO\_STAGE)* – кількість емітованих токенів для продажу в кожному відповідному періоді;

- $RESERVE$  – кількість токенів, які розташовані в резервному фонді;
- $tokenICOStageEmission(ICO\_STAGE)$  – заплановане інвестування,

кількість токенів у кожен період;

де,  $ICO\_STAGE$  – символічне значення періоду:  $SEED$ ,  $PRE\_TGE$ ,  $TGE\_OPEN\_SALE$ . Тоді задати початкові значення даному атрибуту можна таким чином:

$$tokenICOStageEmission(SEED) = 500$$

$$tokenICOStageEmission(PRE\_TGE) = 1000,$$

$$tokenICOStageEmission(TGE\_OPEN\_SALE) = 1250,$$

- $SMInvestToken(i)$  – заплановані кількості токенів на скілмайнінг від платформи на відповідний період ( $i = 1, 2, 3 \dots$ ):

$$SMInvestToken(1) = 25,$$

$$SMInvestToken(2) = 50,$$

$$SMInvestToken(3) = 28,$$

$$SMInvestToken(4) = 20;$$

- $income$  – грошовий дохід власника в умовних одиницях;

$tokenSkillMining$  – поточна кількість токенів, отримана від скілмайнінгу.

Також акцентуємо, що велике значення для моделі має наявність глобальних атрибутів середовища, наведемо деякі з них:  $pReserve$ ,  $pForSale$ ,  $monthlyPRE\_TGE$ ,  $pOICOcoach$ .

1.  $pReserve$  – атрибути типу  $real$ , процентне відношення токенів, яке віддається в резерв після купівлі ноди.

2.  $pForSale$  – атрибути типу  $real$ , процентне відношення токенів для продажу.

3.  $monthlyPRE\_TGE$  – кількість імітованих токенів на місяць у періоді  $PRE\_TGE$ ,

4.  $pOICOcoach$  – атрибути типу  $real$ , процентне відношення токенів, куплених учителями.

Агенти взаємодіють між собою та змінюють свій стан (представлений властивостями). Наступним кроком у побудові алгебраїчної моделі є побудова

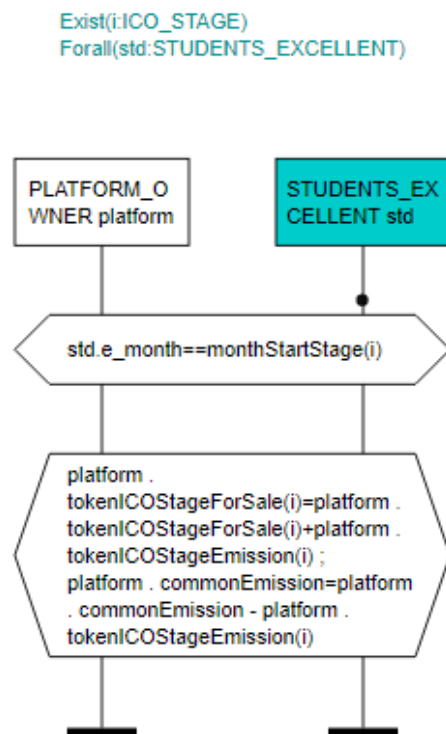
специфікації, яка визначає поведінку агентів. Джерелом для розробки моделі є вимоги до неї.

Для зручності розроблення специфікації потрібно перетворити текстові вимоги на проміжний варіант, білизкий до конструкції *If then*.

Нижче наводимо проміжні вимоги до моделі Skillonomy:

- *вимога 1: If період (SEED, PRE\_TGE, TGE\_OPEN\_SALE) then платформа імітує токени для продажу в поточний період.*
- *вимога 2: If  $6 \leq \text{місяць} \leq 10$  then продаж токенів у відкритий період. Платформа продає щомісячну ставку токенів відповідно до:  $\text{monthlyTGE\_OPEN\_SALE}$   $p\text{Sale}$ ,  $p\text{OICOstd}$  коефіцієнтів.*

Специфікацію першої вимоги засобами мови дій можна розглянути на рис. 4.17.



**Рис.4.17.** Специфікація Вимоги 1 засобами мови дій.

Ця дія емітує токени для продажу в періоди: *SEED*, *PRE\_TGE*, *TGE\_OPEN\_SALE*. Розглянемо специфікації та деталі реалізації другої вимоги в алгебраїчному вигляді *tokenOpenICOSale*:

1. Продаж токенів у відкритий період. Платформа продає щомісячну ставку токенів.
2. Платформа вносить у резервний фонд те, що залишилося після продажу.
3. Платформа отримує дохід за продаж токенів.
4. Студенти (студентські групи) купують свою частину жетонів відповідно до співвідношення  $pOICOXX$ . Усі придбані токени стають недоступними до кінця відкритого продажу.

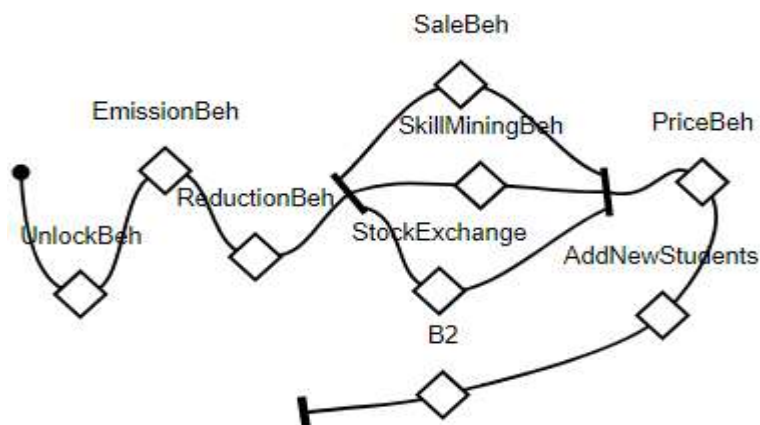
Далі представляємо дію в алгебраїчному виді:

```

tokenOpenICOSale = (Forall(std:Student_EXCELLENT)(
  (std.g_month >= monthStartStage(TGE_OPEN_SALE) &&
  std.g_month <= monthEndStage(TGE_OPEN_SALE))->
  <("Open ICO Sale")>
  (platform . tokenICOSTageForSale(TGE_OPEN_SALE)=platform .
  tokenICOSTageForSale(TGE_OPEN_SALE) - monthlyTGE_OPEN_SALE
  *pSale* pOICOstd_g;
  std.tokenLocked=std.tokenLocked + monthlyTGE_OPEN_SALE * pSale *
  pOICOstd_g;
  platform . income=platform.income + monthlyTGE_OPEN_SALE * pSale *
  pOICOstd_g* tokenPrice))
)

```

Специфікувати поведінку агентів у моделі можна засобами мови *UCM* і *MSC*. Такий підхід сприяє розумінню наборів сценаріїв поведінки в моделі; якщо в специфікації описано не всі варіанти поведінки, то верифікація може виявити неповноту сценаріїв, крім того, ми можемо верифікувати сценарії поведінки моделі на неповноту, недетермінізм і властивості безпеки. Проілюструємо поведінку агентів для моделі Skillonomu засобами *UCM* (рис. 4.18):



**Рис. 4.18.** Алгебра поведінки засобами UCM

На діаграмі UCM зображено послідовну й паралельну композиції. Кожний стаб являє собою множину дій і також є поведінкою.

- *UnlockBeh* – поведінка, яка репрезентує процес розблокування токенів для подальшого використання.
- *EmissionBeh* – процедура емісії токенів.
- *SaleBeh* – процедура продажу токенів.
- *ReductionBeh* – перерахунок кількості студентів після можливого виходу з платформи.
- *StockExchange* – поведінка для репрезентації варіантів стратегій купівлі/продажу токенів на біржі.
- *SkillMiningBeh* – процедура скілмайнінгу.
- *PrieBeh* – зміна ціни відповідно до попиту на токени.
- *B2* – лічильник місяців.
- *AddNewStdBeh* – прихід нових студентів до платформи.

Розглянемо варіант фрагмента поведінки *EmissionBeh* в алгебраїчному вигляді:

$$EmissionBeh = stageEmission . EB1 + not\_stageEmission . EB1,$$

$$EB1 = tokenHolders . EB2 + not\_tokenHolders . EB2,$$

$$EB2 = skillMiningInvest + not\_skillMiningInvest$$

Дії *not\\_stageEmission*, *not\\_tokenHolders*, *not\\_skillMiningInvest* – заперечення відповідних дій, визначення яких описано нижче. Заперечення дії будуть

застосовані, коли умова застосування відповідної дії без “not\_” прийматиме значення *false*.

- *stageEmission* – емісія токенів на продаж у періоди (*SEED, PRE TGE, TGE OPEN SALE*).
- *tokenHolders* – токени надходять до стейкхолдерів відповідно до запланованого плану випуску.
- *skillMiningInvest* – заплановані інвестиції в майнінг із платформи. Токени надходять до загальної кількості, відведеної для майнінгу, до вузлів відповідного атрибута *tokenSkillMining* згідно з початковим планом інвестування токенів по місяцях *SMInvestToken*.

Далі наведемо результати, отримані під час моделювання й аналізу моделі *SKILLONOMY*.

#### **4.5.1. Результати моделювання. Конкретна модель.**

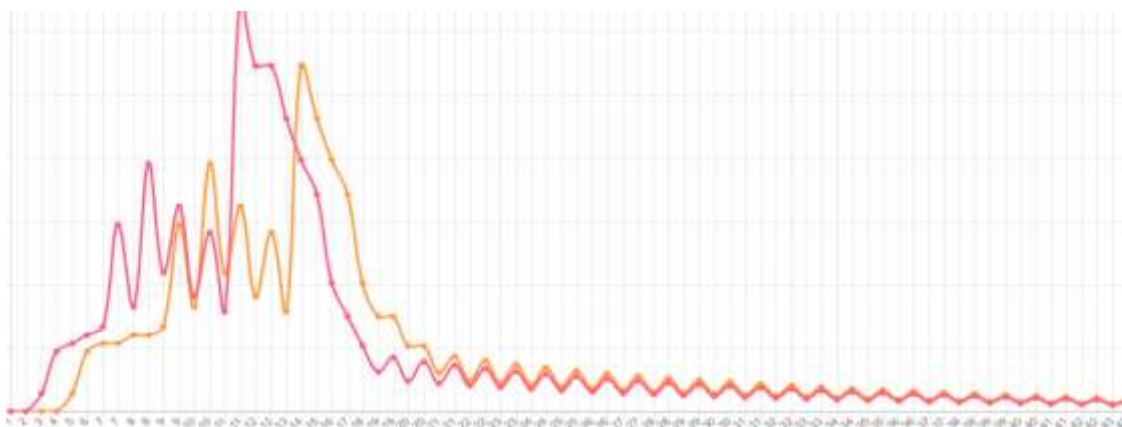
Реалізувавши засобами інсерційного моделювання проєкт *SKILLONOMY*, ми отримали такі результати:

- траєкторії доступних токенів для всіх учасників економічної системи.
- величину прибутку для учасників економічної системи.
- зміну траєкторії ціни токенау, виявили вплив попиту на ціну токенау, вплив біткойна на ціну токенау.
- дослідили життєздатність моделі за різних сценаріїв поведінки спекулянтів в економічній системі.

Більш детально результати досліджень висвітлено в опублікованих працях [174-176].

Далі розглянемо й унаочнимо деякі результати моделювання.

- Кожна траєкторія на графіку рис. 4.19 є траєкторією токенів для студентів-двієчників. Оскільки вони погано вчаться, вони майнять (видобувають) із найменшим коефіцієнтом, це призводить до дефіциту токенів. Студентам недостатньо токенів, для того щоб продовжити навчання, вони починають купувати ту кількість токенів, якої їм бракує.



**Рис. 4.19.** Траєкторія доступних токенів у студентів-двієчників.

Неформальний прибуток студентів 2-ої групи (студентів, які навчаються на «четвірки») складається з деякої кількості жетонів (у період від 2 до 6 місяців). Оскільки (відповідно до вимоги) після 6-го місяця ці студенти отримують кількість жетонів, яка дорівнює їхній базовій потребі, то кількість накопичених жетонів у період від 2 до 6 місяців зберігається до кінця проєкту (рис. 4.20).



**Рис. 4.20.** Траєкторія доступних токенів у студентів-«хорошистів»

- Початкові коефіцієнти видобутку для кожної групи студентів дуже великі. Використовуючи ці коефіцієнти, ми отримуємо кількість жетонів (на період з 2-го по 6-й місяць), яка покриває основну потребу до кінця проєкту. Таким чином, студентам не потрібно купувати жетони, що суперечить вимогам проєкту.

- Під час формалізації вимог усунуто низку неточностей.



У процесі дослідження проекту створено декілька експериментів, які перевіряли життєздатність цієї моделі. Оскільки *SKILLONOMY* – це блокчейновий проєкт, ціна токєну в системі досить сильно залежить від біткоїну (оскільки це криптовалюта з максимальною капіталізацією), цікавим питанням було те, як модель буде реагувати на різні коливання цієї криптовалюти. Визначимо чотири різні тенденції біткоїна:

1. Траєкторія ціни біткоїна рівномірно падає до двадцять третього місяця, а після цього рівномірно збільшується до сорок третього місяця.

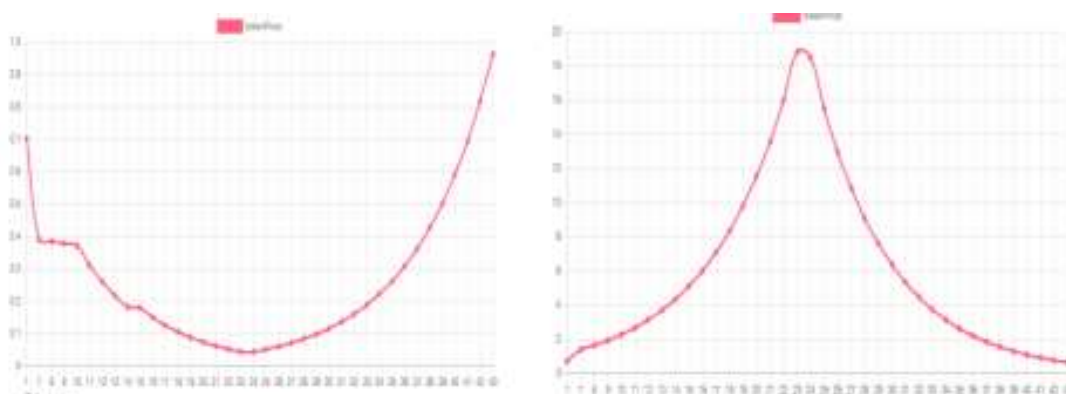
2. Траєкторія ціни біткоїна рівномірно зростає до двадцять третього місяця, а після цього рівномірно падає до сорок третього місяця (див. рис. 4.21).



**Рис. 4.21.** Траєкторія ціни біткоїна (№1,2).

3. Спадаючий тренд ціни біткоїна, після чого зростання до 43 місяця.

4. Зростаючий тренд ціни біткоїна, після чого спадання до 43 місяця (див. рис. 4.22).



**Рис.4.22.** Траєкторія ціни біткоїна (№3,4).

Сценарії поведінки ціни біткоїна суттєво вплинули на поведінку учасників гри й купівельну спроможність студентів. Зазначимо, що такий підхід дає змогу проаналізувати поведінку учасників платформи, коли ціна на токен перевищує їхню купівельну спроможність. Порушення цієї властивості призводить до виходу учасників платформи. Кількість студентів стає настільки малою, що їм вистачає імітованих токенів для покриття своєї базової потреби. Деталі цього дослідження описано в наших роботах (у співавторстві) «Formalization and algebraic modeling of tokenomics projects» [175], «Our Approach to Formal Verification of Token Economy Models» [176].

Такий підхід дає змогу аналізувати гіпотези для оцінення ризиків у разі вибору найгірших умов у процесі токеномічного моделювання.

#### **4.5.2. Результати моделювання. Символьна модель.**

Символьне моделювання, на відміну від конкретного, уможливорює дослідження надійності та стабільності моделі. Зрозуміло, що символічне моделювання дає змогу виразити формули, які характеризують досліджувані в моделі атрибути. Методи символічного моделювання дозволяють працювати з безліччю формул, що покривають множину значень атрибутів системи, а це, зі свого боку, допомагає досягти більш повного покриття станів та ефективно проаналізувати модель, що досліджується.

Так, за допомогою умови надійності (*safety condition*) у системі інсерційного моделювання було досліджено такі параметри:

1. ціну токена;
2. кількість проданих токенів;
3. купівельну спроможність учасників гри.

У системі інсерційного моделювання вони можуть бути виражені так:

$$0.5 < tokenPrice < 6;$$

1.  $0.01 \leq pSale \leq 0.5;$

2.  $10 \leq criticalLimit \leq 100.$

Під час роботи над проектом було розроблено опитування з метою дослідження купівельної спроможності учасників гри. У процесі опитування можна було зрозуміти, скільки коштів у середньому готова витратити людина на купівлю токенів, яких бракує для продовження участі в економічній грі. За результатами опитування висновуємо, що допустима купівельна спроможність – у середньому від 10 до 100 умовних одиниць. У системі інсерційного моделювання цей факт за допомогою *safety condition* можна виразити так:

$$10 \leq \text{criticalLimit} \leq 100.$$

У процесі моделювання для цих прикладів ми можемо отримати такі формули:

$$1. \text{tokenPrice} = F_1(p_1, p_2, \dots, p_n)$$

$$2. p_{\text{Sale}} = F_2(s_1, s_2, \dots, s_n)$$

$$3. \text{criticalLimit} = F_3(l_1, l_2, \dots, l_n)$$

де,  $p_1, \dots, p_n, s_1, \dots, s_n, l_1, \dots, l_n$  - невідомі параметри токеноміки.

Доведення того, що ці параметри входять в описані інтервали, може підтвердити або спростувати властивість безпеки. Цей тип властивості називається властивістю безпеки. Під час моделювання ціна на токен порушує умову безпеки, тобто ціна стає вищою від купівельної спроможності учасників економічної гри. Порушення цієї умови може призвести до втрати учасників, що може негативно вплинути на проєкт.

Крім того, розробка символічної моделі дає змогу більш детально проаналізувати поведінку спекулянтів, виокремити найбільш негативний сценарій поведінки та дослідити критично важливі змінні моделі цього сценарію. Таким сценарієм може бути масова купівля токенів, що призведе до підняття попиту на них.

Таким чином, розроблення символічної моделі вможливило більш якісне й ефективне дослідження моделі, оскільки дозволяє досягти більш повного покриття станів. Також символічне моделювання дає змогу перевірити стабільність і життєздатність моделі в умовах як позитивного, так і негативного сценаріїв. Результати моделювання отримано за допомогою системи LitSoft R&D

IDE, більш детально із функціональними можливостями можна ознайомитися в нашій статті (у співавторстві) [177].

#### 4.6. Пошук вразливих поведінок .

За допомогою моделей алгебри поведінок можна оцінювати здатність алгоритмів витримувати атаки зловмисників; наразі це дуже актуальне питання і для смарт-контрактів (проблема перевірки вразливостей), і для блокчейну загалом. Такий підхід надає можливість знаходити варіанти вразливих поведінок у моделях. Оцінюючи опірності до атак, що призводять до помилок, аналізують здатність системи зберігати конфіденційну інформацію. Уразливу поведінку або поведінку зловмисника моделюють за допомогою шаблонів.

1. Прикладом такого шаблону може бути формалізована поведінка атаки *TheDao* [178,179], *Double Spending* [180], *Reentrancy* [181].

2. Поведінка зловмисників у токеномічних моделях, які заради свого економічного зиску відступають від правил, закладених розробниками моделі.

Шаблон – це вираз в алгебрі поведінок, що містить дії, які представляють уразливості, та невідомі поведінки, що призводять до них. Наприклад, якщо маємо дії читання інформації  $A$  та отримання сигналу, що містить прочитану інформацію, то шаблон має вид такої поведінки:

$S = read(A).X;send(A)$ , де  $X$  – довільна поведінка. Задача полягає в пошуку поведінки  $X$  у моделі, і тих, що призводять до цих дій. Така задача вирішується у два етапи. Перший етап – це розв’язок рівняння алгебри поведінок, що визначає модель. Наприклад, нехай  $B_0$  – система рівнянь алгебри поведінок:

$$B_0 = R(a_1, a_2, \dots, B_1, B_2, \dots, ),$$

$$B_1 = R(a_{11}, a_{12}, \dots, B_{11}, B_{12}, \dots), \dots$$

де  $B_1, B_2, \dots, B_3, B_4, \dots$  поведінки,  $a_1, a_2, \dots, a_n \in$  дії.. Необхідно знайти таку поведінку  $Y$ , що  $B_0 = Y;S$  та  $S$  – шаблон вразливості.

Результатом може бути послідовність дій, яка призводить до вразливості та складає деяку трасу або множину трас. Визначивши трасу, маємо ще довести, що такий сценарій поведінки досяжний, тобто існують відповідні значення атрибутів середовища. Доведення досяжності є другим етапом задачі й реалізується методом символного моделювання. Маючи трасу з відповідними формулами станів середовища, можна згенерувати конкретні значення та отримати експлоїт (*англ. exploit*), тобто дані, за яких можливо здійснити атаку.

### Висновки до четвертого розділу

За допомогою інсерційного моделювання розроблено символну й конкретну модель Вальраса, запропоновано підходи до доказу наявності економічного балансу для цієї моделі. Сформульовано умову рівноваги для моделі Вальраса.

Акцентуємо на тому, що класичне трактування моделі Вальраса–Маршала є неповним: не розглядається випадок, коли у виробника немає грошей на купівлю запланованого ресурсу. У цьому варіанті визначимо правило, що виробник у разі браку коштів буде купувати ресурс на стільки, на скільки йому вистачить грошей:

*If* (виробникові грошей бракує на купівлю ресурсу для запланованого плану) *then* (купувати ресурс за доступні гроші).

*If* (споживачеві грошей вистачає на купівлю товару відповідно до функції корисності) *then* (купувати ресурс відповідно до функції корисності).

Унаслідок роботи над токеномічною моделлю *SKILLONOMY* запропоновано підходи до аналізу такого типу моделей, розроблено конкретну й символну моделі. Застосування алгебраїчного підходу вможливило кваліфікування низки структурно-логічних помилок у моделі. Запропоновано принципи до аналізу токеномічних законів із використанням символного моделювання:

1. Неформальний прибуток студентів, які мають середню оцінку «чотири».
2. Початкові коефіцієнти видобутку для кожної групи студентів дуже великі. Використовуючи початкові коефіцієнти моделі, ми отримуємо кількість

жетонів (на період з 2-го по 6-й місяць), які покривають основну купівельну потребу до кінця проєкту. Таким чином, студентам не потрібно купувати жетони, що суперечить вимогам проєкту.

3. Під час роботи над символічною моделлю виявлено й представлено підходи до аналізу надійності та стабільності моделі.

Описані підходи дали змогу більш детально дослідити поведінку спекулянтів, виокремити найбільш негативні сценарії поведінки для цієї моделі, а також дослідити критично важливі змінні моделі в процесі цього сценарію.

Інсерційне моделювання й методи алгебраїчного програмування загалом виявили себе як потужний інструмент для аналізу не тільки класичних моделей економічного балансу, а й для моделей токеномічного напрямку. Під час аналізування токеномічної моделі знайдено низку важливих структурно-логічних помилок, що допоможе покращити ефективність комерційного проєкту.

#### Список джерел до Розділу 4

- [1] S. A. White, "Introduction to BPMN", in BPTrends (website). Ibm Corporation, July 2004. [Online]. Available: <https://www.bptrends.com/introduction-to-bpmn/>. Accessed on: Sept. 14, 2019.
- [2] A. Presley, and D. H. Liles, "The use of IDEF0 for the design and specification of methodologies", in *Proc. of the 4th Industrial Engineering Research Conference (IERC'94)*, May 1995, pp. 442-448.
- [3] *The token economy: A review and evaluation*, A. Kazdin, Ed. NY, USA: Springer Science & Business Media, 2012.
- [4] A. Sunyaev, "Distributed ledger technology", in *Internet Computing. Principles of Distributed Systems and Emerging Internet-Based Technologies*. Cham, Switzerland: Springer, 2020. pp. 265-299.
- [5] V. Kobets, V. Yatsenko, and M. Poltoratskiy, "Dynamic Model of Double Electronic Vickrey Auction", in *Proc. 11-th Int. Conf. ICTERI 2015, CEUR-WS*, May 2015, vol. 1356, pp. 236-251. DOI: 10.13140/RG.2.1.4066.7040.
- [6] М. О. Вінник, В. В. Одінець, та М. Ю. Полторацький, "Використання інформаційних технологій при вивченні популяційної біології", *Проблеми інформаційних технологій*, № 20, с. 168-175, 2016.
- [7] V. Kobets, and M. Poltoratskiy, "Using an evolutionary algorithm to improve investment strategies for industries in an economic system", in *Proc. 5-th Int. Workshop ITER 2016, CEUR-WS*, June 2016, vol. 1614, pp. 485-501.
- [8] S. Mancarella, *Business Process Modelling. Notation – A tutorial*. USA, OMG SOA Healthcare, SPARX Systems, 2011.
- [9] T. Rademakers, *Activiti in Action: Executable business processes in BPMN 2.0*. Manning Publications, 2012. [Online]. Available: <https://www.manning.com/books/activiti-in-action>. Accessed on: Oct. 15, 2019.
- [10] V. Kobets, and V. Yatsenko, "Adjusting Business Processes by the Means of an Autoregressive Model Using BPMN 2.0", in *Proc. 9-th Int. Conf. ICTERI 2013, CEUR-WS*, June 2016, vol. 1614, pp. 518-533.

- [11] Bizagi Studio/Modeler, in *Bizagi (website)*. [Online]. Available: <https://www.bizagi.com/en>. Accessed on: Oct. 15, 2019.
- [12] И. И. Гниломедов, "Моделирование экономических агентов при помощи конечных автоматов", в *Научные доклады конф. «Интегрированные модели, мягкие вычисления, вероятностные системы и комплексы программ в искусственном интеллекте»*, М.: Физматлит, 2009, Т. 2, с. 72-89.
- [13] G. Zimmermann, R. Neuneier, and R. Grothmann, "Multi-agent FX-Market Modeling Based on Cognitive Systems", in *Proc. Artificial Neural Networks – ICANN 2001*, (LNCS, vol. 2130), Aug. 2001, pp. 767-774. Doi: 10.1007/3-540-44668-0\_107.
- [14] R. Draeseke, and D. EA Giles, "A fuzzy logic approach to modelling the New Zealand underground economy", *Mathematics and Computers in Simulation*, vol. 59, iss.1-3, pp. 115-123, 2002. Doi: 10.1016/S0378-4754(01)00399-8.
- [15] A. Hassan, "Relevance of fuzzy logic in the economy", *International scientific journal*, vol. 4 (2), pp. 104-105, 2016.
- [16] D. Nemchenko, V. Kobets, and L. Potravka, "Neuro-Fuzzy Model of Development Forecasting and Effective Agrarian Sector Transformations of Ukraine", in *Proc. 14-th Int. Conf. ICTERI 2018, vol. II: Workshops, CEUR-WS*, May 2018, vol. 2104, pp. 84-99.
- [17] H. Storm, K. Baylis, and T. Heckelei, "Machine learning in agricultural and applied economics", *European Review of Agricultural Economics*, vol. 47, iss. 3, pp. 849-892, July 2020. Doi: 10.1093/erae/jbz033.
- [18] A. Storkey, "Machine learning markets", in *Proc. 14th Int. Conf. Artificial Intelligence and Statistics (AISTATS 2011)*, vol. 15, June 2011, pp. 716-724.
- [19] B. Char, K. Geddes, G. Gonnet, B. Leong, M. Monagan, and S. Watt, *Maple V library reference manual*. NY, USA: Springer Science & Business Media, 2013.
- [20] Z. Chvátalová, and J. Hřebíček, "Computational finance and finance economics with maple", *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 7, iss. 5, pp. 541-550, 2013.



[21] J. Hřebíček, “Mathematical Modeling of Economic Phenomena with Maple”, in *Proc. 30th Int. Conf. Mathematical Methods in Economics (MME’12)*, 2012, pp. 326-331.

[22] J. Luedicke, *POWERSIM: Stata module for simulation-based power analysis for linear and generalized linear models, (Statistical Software Components)*. Boston, USA: Boston College Department of Economics, 2013.

[23] iTHINK, in *isee systems (website)*. [Online]. Available: <https://www.iseesystems.com/store/products/ithink.aspx>. Accessed on: Jan. 09, 2020.

[24] W. D. Kelton, *Simulation with ARENA*. McGraw-hill, 2002.

[25] C. Pîrnău, “Modeling gearing of regional eco-bio-logistic components in bioeconomic management, using Arena simulation software”, *Journal of Knowledge Management, Economics and Information Technology*, vol. 4(1.1), pp. 1-14, Feb.2014.

[26] K. Jilcha, E. Berhan, and H. Sherif, “Workers and Machine Performance Modeling in Manufacturing System Using Arena Simulation”, *J Comput Sci Syst Biol*, vol. 8(4), pp. 185-190, 2015. Doi: 10.4172/jcsb.1000187.

[27] *GPSS: имитационное моделирование систем (сайт)*. [Электронный ресурс]. Доступ: <http://www.gpss.ru/>. Дата обращения: Янв. 09, 2020.

[28] S. A. Vlasov, T. V. Devyatkov, and V. V. Devyatkov, “Cloud technology in simulation studies: GPSS cloud project”, in *7th IFAC Manufacturing Modelling, Management, and Control Conf., IFAC Proc. Volumes*, 2013, vol. 46, iss. 9, pp. 637-641. Doi: [10.3182/20130619-3-RU-3018.00582](https://doi.org/10.3182/20130619-3-RU-3018.00582).

[29] B. Srivastava, J. P. Bigus, and D. A. Schlosnagle, "Using ABLE to Bring Planning to Business Applications", *American Association for Artificial Intelligence (www.aaai.org)*, 2004.

[30] J. P. Bigus, D. A. Schlosnagle, J. R. Pilgrim, W. N. Mills III, and Y. Diao, “ABLE: A toolkit for building multiagent autonomic systems”, *IBM Systems Journal*, vol. 41, iss. 3, pp. 350-371, 2002. Doi: [doi.org/10.1147/sj.413.0350](https://doi.org/10.1147/sj.413.0350).

[31] B. LeBaron, "Agent-based Financial Markets: Matching Stylized Facts with Style", in *Post Walrasian Macroeconomics: Beyond the DSGE Model*, D. Colander, Ed. Cambridge, United Kingdom: Cambridge University Press, 2006, pp. 221-236.

- [32] *AnyLogic* (website). [Online]. Available: <https://www.anylogic.com/>. Accessed on: Jan. 09, 2020.
- [33] J. Ren, Q. Zhao, B. Liu, and C. Chen, "Selection of pallet management strategies from the perspective of supply chain cost with Anylogic software", *PLOS ONE*, 14.6: e0217995, June 2019. Doi: 10.1371/journal.pone.0217995.
- [34] B. Du, and H. Liu, "Modeling and Analysis of Linyi Regional Logistics System Based on Anylogic", *Logistics Technology*, vol. 19, p. 43, 2012.
- [35] Ascape 5.0, in *Ascape Guide* (website). [Online]. Available: <http://ascape.sourceforge.net/>. Accessed on: Jan. 09, 2020.
- [36] MATLAB, in *MathWorks* (website). [Online]. Available: <https://www.mathworks.com/products/matlab.html>. Accessed on: Jan. 09, 2020.
- [37] Y. Li, and S. Malin, "Research on BP Neural Network for Nonlinear Economic Modeling and its Realization Based on Matlab", in *2009 Third Int. Symposium on Intell. Inf. Technol. Application (IITA'09)*, Nov. 2009, vol. 1, pp. 505-508. Doi: [10.1109/IITA.2009.352](https://doi.org/10.1109/IITA.2009.352).
- [38] NetLogo, in *CCL: Center for Connected Learning and Computer-Based Modeling* (website). [Online]. Available: <https://ccl.northwestern.edu/netlogo/>. Accessed on: Jan. 09, 2020.
- [39] B. Head, A. Hjorth, C. Brady, and U. Wilensky, "Evolving agent cognition with Netlogo LevelSpace", in *2015 Winter Simulation Conf. (WSC'15)*, Dec. 2015, pp. 3122-3123.
- [40] J. C. Thiele, "R marries NetLogo: introduction to the RNetLogo package", *Journal of Statistical Software*, vol. 58, iss. 2, pp. 1-41, 2014. Doi: 10.18637/jss.v058.i02.
- [41] L. R. Izquierdo, D. Olaru, S. S. Izquierdo, S. Purchase, and G. N. Soutar, "Fuzzy logic for social simulation using NetLogo", *Journal of Artificial Societies and Social Simulation (JASSS)*, vol. 18, iss. 4, Article 1, 2015. Doi: 10.18564/jasss.2885.
- [42] A. Kopp, and D. Orlovskyi, "A Method for Business Process Model Analysis and Improvement", in *Proc. 15th Int. Conf. ICTERI 2019: PhD Symposium, CEUR-WS*, June 2019, vol. 2403, pp. 1-10.

[43] V. Peschanenko, M. Poltoratskyi, and K. Pryimak, "Formalization and algebraic modeling of university economics", in *Proc. 15th Int. Conf. ICTERI 2019: Workshops (RMSE'19), CEUR-WS*, June 2019, vol. 2393, pp. 585-594.

[44] Кривогуб, А. М., and К. В. Новодережкіна. "Модель загальної рівноваги та методи її дослідження." *Математические машины и системы* 4 (2015).

[45] C. Dannen, *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Berkeley, USA: Apress, 2017. Doi: 10.1007/978-1-4842-2535-6.

[46] C. Dannen, "Solidity programming", in *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Berkeley, USA: Apress, 2017, pp. 69-88.

[47] Z. Yang, and H. Lei. "Lolisa: Formal syntax and semantics for a subset of the solidity programming language in Mathematical Tool Coq", in *arXiv: a free distribution service and an open-access archive*, Mar. 2018. Doi: 10.1155/2020/6191537. [Online]. Available: <https://arxiv.org/abs/1803.09885>. Accessed on: Feb. 12, 2019.

[48] R. Wandmacher, "Tokenomics", in *Cryptofinance and Mechanisms of Exchange. Contributions to Management Science*, S. Goutte, K. Guesmi, S. Saadi, Eds. Cham, Switzerland: Springer, 2019, pp. 113-123. Doi: 10.1007/978-3-030-30738-7\_7.

[49] L. W. Cong, Y. Li, and N. Wang, "Tokenomics: Dynamic adoption and valuation", in *NBER Working Papers*, National Bureau of Economic Research, Inc., 2020, no. 27222.

[50] K. Samani, "The Blockchain Token Velocity Problem", in *CoinDesk (media website)*, Dec. 8, 2017, Updated Jan. 6, 2018. [Online]. Available: <https://www.coindesk.com/blockchain-token-velocity-problem>. Accessed on: Jan. 09, 2020.

[51] O. Letychevsky, V. Peschanenko, V. Radchenko, M. Poltoratzkyi, P. Kovalenko, and S. Mogylko, "Formal Verification of Token Economy Models", in *Proc. 2019 IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, Seoul, Korea,

May 2019. Doi: 10.1109/BLOC.2019.8751318. [Online]. Available: <https://ieeexplore.ieee.org/document/8751318>. Accessed on: Jan. 09, 2020.

[52] O. Letychevskiy, V. Peschanenko, V. Radchenko, M. Poltoratskiy, and Y. Tarasich, "Formalization and algebraic modeling of tokenomics projects", in *Proc. 15th Int. Conf. ICTERI 2019: Workshops (RMSE'19), CEUR-WS*, June 2019, vol. 2393, pp. 577-584.

[53] O. Letychevskiy, V. Peschanenko, M. Poltoratskiy, and Y. Tarasich, "Our Approach to Formal Verification of Token Economy Models", in *Proc. Int. Conf. ICTERI 2019, Communications in Computer and Information Science*, vol. 1175, pp. 348-363. Doi: 10.1007/978-3-030-39459-2\_16.

[54] O. Letychevskiy, V. Peschanenko, M. Poltoratskiy, and Y. Tarasich, "Platform for Modeling of Algebraic Behavior: Experience and Conclusions", in *Proc. 16th Int. Conf. ICTERI 2020: Workshops (RMSEBT'20), CEUR-WS*, Oct. 2020, vol. 2732, pp. 42-57.

[55] H. Davarikia, and M. Barati, "A tri-level programming model for attack-resilient control of power grids", *Journal of Modern Power Systems and Clean Energy*, vol. 6, iss. 5, pp. 918-929, Sept. 2018. Doi: 10.1007/s40565-018-0436-y.

[56] M. Mehar, et al., "Understanding a revolutionary and Flawed Grand Experiment in Blockchain: the DAO Attack", *Journal of Cases on InformationTechnology (JCIT)*, vol. 21, iss.1, pp. 19-32, 2019. Doi: 10.4018/JCIT.2019010102.

[57] G. O. Karame, E. Androulaki, and S. Capkun, "Double-Spending Fast Payments in Bitcoin", in *Proc. 2012 ACM conf. on Computer and commun. security (CCS'12)*, Oct. 2012, pp. 906-917. Doi: 10.1145/2382196.2382292.

[58] C. Liu, H. Liu, Z. Cao, Z. Chen, B. Chen, B. Roscoe, "ReGuard: Finding Reentrancy Bugs in Smart Contracts", in *Proc. 2018 IEEE/ACM 40th Int. Conf. Software Engineering: Companion (ICSE'18 Companion)*, May-June 2018, pp. 65-68. Doi: 10.1145/3183440.3183495.

## ВИСНОВКИ

У першому розділі розглянуто й проаналізовано сучасні підходи до аналізу та моделювання правових вимог. Описано сучасні методи й технології, які використовуються для кодифікації та гармонізації нормативно-правових вимог Європи, визначено переваги й недоліки цих підходів. Проаналізовано наявні системи правового моделювання.

У другому розділі наведено особливості інсерційної семантики, а також особливості розробки технологічної лінії формалізації нормативно-правових документів. Цей підхід уможливив розроблення за допомогою системи інсерційного моделювання й алгебраїчного програмування формальної моделі реєстрації платника ПДВ України. Під час формалізації та аналізу моделі знайдено низку структурно-логічних помилок:

- так, у статті 183.3 визначено, що особа, яка добровільно реєструється й відповідає вимогам 180.1 п.6, має проходити реєстрацію за процедурою 183.7, але, з іншого боку, та сама особа може зареєструватися також через пункти 183.2 та 183.1, проте в тексті законів не вказано, що це – лише для платників податків, які реєструються добровільно й відповідають вимогам 180.1 п.6.

Формалізація згаданих статей та їхнє відображення у вигляді кон'юнкції інформаційно залежних аксіом

$TaxPayerRegistration \ \& \ Condition_{180\_1\_6} \Leftrightarrow Registration_{183} \ \& \& TaxPayerRegistration \ \& Voluntary \ \& Condition_{180\_1\_6} \Leftrightarrow Registration_{187\_1}$ ,

що визначають виконуваність, дозволяє знайти наступну структурно-логічну помилку:

- за умови добровільної реєстрації, згідно зі статтею 183.9 п.2-3 визначається день, із якого нараховується ПДВ. Цей день може бути першим числом наступного місяця після спливання 20 днів із часу подання заяви на добровільну реєстрацію.

У дисертаційному дослідженні представлено підходи для аналізу юридичних прецедентів і, узагалі, підхід до зіставлення юридичних документів

(юридичних постанов, прецедентів) із законами. Проаналізовано податковий прецедент [1] із множинним трактуванням. Під час вивчення цього прецеденту змодельовано наявний у ньому недетермінізм, відображено неповноту закону для випадку  $Activity == TAXI\_SERVICE$ , яка полягає в неявному визначенні, як у цьому разі співвідносяться витрати, а також як визначається податковий кредит.

У *третьому розділі* проаналізовано сучасні підходи до виокремлення й оброблення слабоструктурованих текстів систем автоматизованого анотування. Запропоновано принципи оброблення юридичного тексту та розроблено програмний засіб для трансляції текстів в алгебраїчний вид, а також для побудови моделі траси з тексту судової постанови.

Такий підхід уможливить перевірку правильності вироку судової постанови й може розглядатися як програмний засіб для підтримки прийняття рішення в судовій практиці та юриспруденції загалом.

*Четвертий розділ* присвячено вивченню сучасних підходів і технологій для аналізу економічних моделей. Формалізовано модель Вальраса й запропоновано підхід до доказу наявності економічного балансу в моделі. У процесі дослідження розроблено конкретну та символну моделі Вальраса. Сформульовано й доведено умову рівноваги для моделі Вальраса. Виокремлено низку важливих недетермінізмів у постановці моделі, які було знайдено під час формалізації вимог до цієї моделі:

1. Виробникові бракує коштів на купівлю ресурсу для запланованого плану виробництва.
2. Споживачеві бракує коштів на купівлю товару відповідно до функції корисності.

Запропоновано підхід до розроблення, аналізування токеномічних моделей і токеномічних законів. Під час роботи над моделлю *SKILLONOMY* знайдено низку важливих структурно-логічних помилок:

- неформальний прибуток студентів 2 групи (студентів, які навчаються на «четвірки»): ці студенти накопичують певну кількість токенів (у період від 2 до 6 місяців);

- початкові коефіцієнти видобутку для кожної групи студентів дуже високі. Застосовуючи ці коефіцієнти, ми отримуємо кількість токенів (на період із 2-го до 6-го місяця), які покривають основну потребу до кінця проєкту. Таким чином, студентам не потрібно купувати токени, що суперечить вимогам проєкту.

Під час дисертаційного дослідження вирішено такі проблеми:

1. Розроблено технологічну лінію формалізації від тексту до формальної моделі, а також розроблено формальну модель платника податку ПДВ України.
2. Запропоновано підхід до аналізу цілісності й відсутності протиріч, а також до зіставлення юридичних документів із чинними законами.
3. Запропоновано підхід до обробки слабоструктурованого юридичного тексту, розроблено програмний засіб трансляції тексту в алгебраїчний вид.
4. Запропоновано підхід до аналізу класичних економічних моделей, а також моделей токеноміки.
5. Формалізовано модель Вальраса, сформульовано умову рівноваги й математично доведено цю властивість.
6. Формалізовано модель комерційного проєкту *SKILLONOMY*, наведено переваги використання алгебраїчного моделювання для аналізу моделей токеноміки та законів токеноміки. Усунуто низку структурно-логічних помилок у зазначених моделях.

Унікальність нашого підходу полягає у використанні методів алгебраїчного програмування й інсерційного моделювання, які можуть бути ефективно застосовані для аналізу як правових, так і економічних моделей.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] “«Засуджений» ПДВ або Прецедент на користь платника податку”, *Консультант бухгалтера*, № 16 (400), Квіт. 16, 2007. [Електронний ресурс].  
Доступно: <http://cons.parus.ua/d.asp?r=03WMGd7e3c4816c93692e4d5ba8e0b8234dec>. Дата звернення: Січ. 10, 2018.
- [2] *Skillonomy* (сайт). [Электронный ресурс]. Доступно: <https://skillonomy.org/ru/>. Дата обращения: Янв. 10, 2018.
- [3] C. Wass et al., “OpenLaws. eu”, in *Proc. 16th Int. Legal Informatics Symposium (IRIS'13)*, Feb. 2013, pp. 209-211.
- [4] R. Winkels, "The openlaws project: Big open legal data", *Proc. 18th Int. Legal Informatics Symposium (IRIS'15)*, Feb. 2015, pp. 189-196.
- [5] RDF: Resource Description Framework, in *W3C: Semantic Web (website)*. [Online]. Available: <https://www.w3.org/RDF/>. Accessed on: May 19, 2018.
- [6] Standarts, in *W3C: Semantic Web (website)*. [Online]. – Available: <https://www.w3.org/standards/>. Accessed on: May 19, 2018.
- [7] J. Pérez, M. Arenas, and C. Gutierrez, “Semantics and complexity of SPARQL”, *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 16, pp. 1-45, Sept. 2009. Doi: 10.1145/1567274.1567278.
- [8] D. F. Barbieri, D. Braga, S. Ceri, E. D. Valle, and M. Grossniklaus, “C-SPARQL: SPARQL for continuous querying”, in *Proc. 18th Int. World Wide Web Conf. (WWW'09)*, Apr. 2009, pp. 1061-1062. Doi: 10.1145/1526709.1526856.
- [9] R. Fikes, P. Hayes, and I. Horrocks, "DQL-a query language for the semantic web", Knowledge Systems Laboratory, Tech. Rep. KSL-02-05, Nov. 2002.
- [10] N3QL – RDF Data Query Language, in *W3C: Semantic Web (website)*. [Online]. Available: <https://www.w3.org/DesignIssues/N3QL>. Accessed on: May 19, 2018.
- [11] RDQL – A Query Language for RDF, in *W3C: Semantic Web (website)*. [Online]. Available: <https://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>. Accessed on: May 19, 2018.



[12] K. Hailbronner, *EU Immigration and Asylum Law: A commentary*. C. H. Beck/Hart, 2010.

[13] G. Görög, and P. Weisz, “Legal Entity Recognition in an Agglutinating Language and Document Connection Network for EU Legislation and EU/Hungarian Case Law”, *Preprints*, 2019070310, 2019. doi: 10.20944/preprints201907.0310.v1. [Online]. – Available: <https://www.preprints.org/manuscript/201907.0310/v1>. Accessed on: Nov. 12, 2019.

[14] V. Peychev, “XML model for legal documents”, *Problems of Eng. Cybern. and Robot.*, vol. 54, pp. 86–91, 2004.

[15] V. Peychev, “Legal document – a formal model”, *Problems of Eng. Cybern. and Robot.*, vol. 55, pp. 64–70, 2005.

[16] K. E. Petersen, “Experiences with «Lex Dania Live»”, in *From Information to Knowledge: Online Access to Legal Information: Methodologies, Trends and Perspectives*, M. A. Biasiotti, S. Faro, Eds. (Frontiers in Artificial Intelligence and Applications series (FAIA), vol. 236), 2011, p. 69-76.

[17] A. D. Maurer, and U. P. Holenstein, “La publication électronique des actes législatifs suisses. Elaboration d’un modèle structuré pour les textes de loi fédéraux, cantonaux et communaux: le schéma XML – CHLexML”, in *Bundesamt für Justiz, Mitarbeiterinnen und Mitarbeiter*, B. Schindler, Ed. Basel: Helbing & Lichtenhahn, 2006, ss. 391-401.

[18] eLAW, in *LAWi: World Encyclopedia of Law (website)*. [Online]. Available: <https://lawin.org/eLaw/>. Accessed on: May 19, 2018.

[19] E. Francesconi, “A review of Systems and Projects: Management of Legislative Resources”, in *Legislative XML for the semantic Web*, G. Sartor, M. Palmirani, E. Francesconi, M. A. Biasiotti, Eds. (Law, Governance and Technology Series (LGTS), vol. 4), 2011, pp. 173-188. Doi: 10.1007/97894-007\_1887-6\_10.

[20] A. Boer, R. Hoekstra, and R. Winkels, “<sup>META</sup>Lex: Legislation in XML”, in *Legal Knowl. and Inf. Syst.: JURIX 2002*, T. Bench-Capon, A. Daskalopulu, R. Winkels, Eds. (FAIA, vol. 89), Amsterdam: IOS Press, 2002, pp. 1-10.

[21] A. Boer, R. Winkels, and F. Vitali, "MetaLex XML and the Legal Knowledge Interchange Format", in *Computable Models of the Law: Languages, Dialogues, Games, Ontologies*, P. Casanovas, G. Sartor, N. Casellas, R. Rubino, Eds. (Lecture Notes in Computer Science series (LNCS), vol. 4884), 2008, pp. 21-41. DOI: 10.1007/978-3-540-85569-9\_2.

[22] Palmirani, M., Governatori, G., Rotolo, A., Tabet, S., Boley, H., & Paschke, A. (2011, November). LegalRuleML: XML-based rules and norms. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web* (pp. 298-312). Springer, Berlin, Heidelberg.

[23] А. А. Никоненко, "Обзор баз знаний онтологического типа", *Искусств. интеллект*, № 4, с. 208-219, 2009.

[24] J. Breuker, and R. J. Hoekstra, "Epistemology and ontology in core ontologies: FOLaw and LRI-Core, two core ontologies for law", in *CEUR Workshop Proc.*, vol. 118, Oct. 2004. [Online]. Available: <http://ceur-ws.org/Vol-118/paper2.pdf>. Accessed on: May 19, 2018.

[25] A. Gangemi, A. Prisco, M.-T. Sagri, G. Steve, and D. Tiscornia, "Some Ontological Tools to Support Legal Regulatory Compliance, with a Case Study", in *Confeder. Int. Conf. "On the Move to Meaningful Internet Systems" (OTM'03 Workshops)*, (LNCS, vol. 2889), Nov. 2003, pp. 607-620. Doi: 10.1007/978-3-540-39962-9\_64.

[26] D. Tiscornia, "The LOIS project: Lexical Ontologies for Legal Information Sharing", in *Proc. V Legislative XML Workshop*, 2007, pp. 189-204.

[27] R. Hoekstra, J. Breuker, M. Bello, and A. Boer, "The LKIF Core Ontology of Basic Legal Concepts", in *Proc. 2nd Workshop LOAIT'07*, June 2007, pp. 43-63.

[28] R. Hoekstra, J. Breuker, M. Bello, and A. Boer, "LKIF Core: Principled Ontology Development for the Legal Domain", *Law, Ontologies and the Semantic Web: Channelling the Legal Inf.Flood*, vol. 188, pp. 21-52, 2009. Doi: 10.3233/978-1-58603-942-4-21.

[29] C. Griffo, “Ufo-l: A Core Ontology of Legal Concepts Built from a Legal Relations Perspective”, in *Doctoral Consortium on Knowledge Discovery, Knowledge Engineering and Knowledge Management (DC3K), 2015*, pp. 13-20.

[30] T. F. Gordon, "Constructing Legal Arguments with Rules in the Legal Knowledge Interchange Format (LKIF)", in *Computable Models of the Law: Languages, Dialogues, Games, Ontologies*, P. Casanovas, G. Sartor, N. Casellas, R. Rubino, Eds. (LNCS, vol. 4884), 2008, pp. 162-184. Doi: 10.1007 / 978-3-540-85569-9\_11.

[31] I. Horrocks, P. F. Patel-Schneider, S. Bechhofer, and D. Tsarkov, “OWL Rules: A Proposal and Prototype Implementation”, *Journal of Web Semantics*, vol. 3, no. 1, pp. 23-40, 2005. Doi: 10.1016/j.websem.2005.05.003.

[32] OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (Second Edition), B. Motik, and B. Parsia, Eds., in *W3C: Semantic Web (website)*, Dec. 2012. [Online]. Available: <https://www.w3.org/2012/pdf/REC-owl2-syntax-20121211.pdf>. Accessed on: May 19, 2018.

[33] M. Horridge, and S. Bechhofer, “The OWL API: a Java API for Working with OWL 2 Ontologies”, in *Proc. 6th Int. Workshop OWL: Experiences and Directions (OWLED’09)*, Oct. 2009.

[34] M. Horridge, and P. F. Patel-Schneider, “Manchester Syntax for OWL 1.1”, in *OWLED’08, 4th Int. Workshop*, 2008. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.159.1433>. Accessed on: May 23, 2019.

[35] N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Ferguson, and M. A. Musen, “Creating Semantic Web Contents with Protege-2000”, *IEEE intelligent systems*, vol. 16, iss. 2, pp. 60-71, 2001. Doi: 10.1109/5254.920601.

[36] P. Casanovas, M. Palmirani, S. Peroni, T. van Engers, and F. Vitali, “Semantic Web for the Legal Domain: The next step”, *Semantic Web*, vol. 7, no. 3, pp. 213-227, 2016. DOI: 10.3233/SW-160224.

[37] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Groszof, and M. Dean, “SWRL: A Semantic Web Rule Language Combining OWL and RuleML”,

in *W3C: Semantic Web (website): Member submission*, May 2004. [Online]. Available: <https://www.w3.org/Submission/SWRL/>. Accessed on: May 23, 2019.

[38] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A practical OWL-DL reasoner”, *Journal of Web Semantics*, vol. 5, iss. 2, pp. 51-53, 2007. Doi: 10.1016/j.websem.2007.03.004.

[39] T. F. Gordon, “Constructing Legal Arguments with Rules in the Legal Knowledge Interchange Format (LKIF)”, in *Computable Models of the Law*. P. Casanovas, G. Sartor, N. Casellas, R. Rubino, Eds. (LNCS, vol. 4884), 2008, pp. 162-184. DOI: 10.1007/978-3-540-85569-9\_11.

[40] A. Boer, R. Winkels, and F. Vitali, “Proposed XML standards for law: Metalex and LKIF”, in *JURIX 2007*, A. R. Lodder, L. Mommers, Eds. (FAIA, vol. 165), Vol. 165, 2007, pp. 19-28.

[41] H. Boley, A. Paschke, and O. Shafiq, “RuleML 1.0: The Overarching Specification of Web Rules”, in *Int. Workshop Rules and Rule Markup Languages for the Semantic Web*, (LNCS, vol. 6403), 2010, pp. 162-178. Doi: 10.1007/978-3-642-16289-3\_15.

[42] П. А. Ломов, и А. Г. Олейник, “Разработка технологии проверки и согласования нормативно-правовой базы на основе онтологий”, *Труды Института системного анализа Российской академии наук*, Т. 63, №. 2, С. 62-69, 2013.

[43] O. Corcho, M. Poveda-Villalón, and A. Gómez-Pérez, “Ontology engineering in the era of linked data”, *Bulletin of the Association for Inf.Science and Technology*, vol. 41, nú. 4, pp. 13-17, 2015. Doi: 10.1002/bult.2015.1720410407.

[44] S. Sharifi, A. Parvizimosaed, D. Amyot, L. Logrippo, and J. Mylopoulos, “Symboleo: Towards a Specification Language for Legal Contracts”, in *28th IEEE Int. Requirements Eng. Conf. (RE)*, 2020, pp. 364-369. Doi: [10.1109/RE48521.2020.00049](https://doi.org/10.1109/RE48521.2020.00049).

[45] A. Parvizimosaed, S. Sharifi, D. Amyot, L. Logrippo, and J. Mylopoulos, “Subcontracting, Assignment, and Substitution for Legal Contracts in Symboleo”, in *9th Conceptual Modeling Int. Conf. (ER'20)*, LNCS, vol. 12400, Nov. 2020, pp. 271-285. Doi: 10.1007/978-3-030-62522-1\_20.

[46] M. Montali, “jREC”, in *marcomontali on Free University of Bozen-Bolzano (website)*, 2016. [Online]. Available: <https://www.inf.unibz.it/~montali/tools.html>. Accessed on: May 23, 2019.

[47] T. M. Van Engers, R. Gerrits, M. Boekenoogen, E. Glassée, and P. Kordelaar, “POWER: Using UML/OCL for Modeling Legislation: an application report”, in *Proc. 8th Int. Conf. on Artificial intelligence and law (ICAIL’01)*, 2001, pp. 157-167. Doi: 10.1145/383535.383554.

[48] J. Cabot, R. Clarisó, and D. Riera, “Verification of UML/OCL Class Diagrams using Constraint Programming”, in *2008 IEEE Int. Conf. Software Testing Verification and Validation Workshop (ICSTW’08)*, 2008, pp. 73-80. Doi: 10.1109/ICSTW.2008.54.

[49] G. Soltana, E. Fourneret, M. Adedjouma, M. Sabetzadeh, and L. Briand, “Using UML for modeling procedural legal rules: Approach and a study of Luxembourg’s Tax Law”, in *Int. Conf. Model Driven Engineering Languages and Systems (MODELS’14)*, (LNCS, vol. 8767), 2014, pp. 450-466. Doi: 10.1007/978-3-319-11653-2\_28.

[50] S. Ali, L. C. Briand, A. Arcuri, and S. Walawege, “An Industrial Application of Robustness Testing Using Aspect-Oriented Modeling, UML/MARTE, and Search Algorithms”, in *MODELS’11*, (LNCS, vol. 6981), 2011, pp. 108-122. Doi: 10.1007/978-3-642-24485-8\_9.

[51] G. Soltana, N. Sannier, M. Sabetzadeh, and L. C. Briand, “Model-Based Simulation of Legal Policies: Framework, Tool Support, and Validation”, *Softw Syst Model*, vol. 17, pp. 851–883, 2018. Doi: 10.1007/s10270-016-0542-0.

[52] G. Sileno, A. Boer, and T. van Engers, “Legal Knowledge Conveyed by Narratives: Towards a Representational Model”, in *Proc. Int. Workshop on Computational Models of Narrative (CMN’14)*, 2014, vol. 41, pp. 182-191. Doi: 10.4230/OASICS.CMN.2014.182.

[53] S. Ghanavati, “Legal-URN framework for legal compliance of business processes”, PhD Thesis, University of Ottawa, Ottawa, Canada, 2013.

[54] *Основные юридические понятия Уэсли Н. Хофельда*, М. В. Антонов, Науч. ред., А. А. Гайдамакина, Сост. и пер. с англ. СПб, Россия: Алеф-Пресс, 2016, с. 58-91.

[55] S. Ghanavati, D. Amyot, and A. Rifaut, “Legal Goal-Oriented Requirement Language (Legal GRL) for Modeling Regulations”, in *Proc. 6th Int. Workshop on Modeling in Software Engineering (MiSE'14)*, June 2014, pp. 1-6. Doi: 10.1145/2593770.2593780.

[56] D. Amyot et al., “Towards Advanced Goal Model Analysis with jUCMNav”, in *Int. Conf. Advances in Conceptual Modeling (ER'12)*, (LNCS, vol. 7518), 2012, pp. 201-210. Doi: 10.1007/978-3-642-33999-8\_25.

[57] D. Amyot, and G. Mussbacher, “User Requirements Notation: The First Ten Years, The Next Ten Years”, *JSW*, vol. 6, no. 5, pp. 747-768, 2011. Doi: 10.4304/jsw.6.5.747-768.

[58] L. T. van Binsbergen, L.-Ch. Liu, R. van Doesburg, and T. van Engers, “eFLINT: a domain-specific language for executable norm specifications”, in *Proc. 19th ACM SIGPLAN Int. Conf. Generative Programming: Concepts and Experiences (GPCE'20)*, 2020, pp. 124-136. Doi: 10.1145/3425898.3426958.

[59] L.T. McCarty, “A language for legal Discourse I. basic features”, in *Proc. 2nd Int. conf. ICAIL '89*, May 1989, pp. 180-189. Doi: 10.1145/74014.74037.

[60] D. L. Chen, "Machine learning and the rule of law", in *Law as Data*, M. Livermore and D. Rockmore, Eds. Santa Fe Institute Press, 2019, pp. 433-441. DOI: 10.37911/9781947864085.16.

[61] R. Barros, A. Peres, F. Lorenzi, L. K. Wives, E. Hubert da S. Jaccottet, "Case Law Analysis with Machine Learning in Brazilian Court", in *Recent Trends and Future Technology in Applied Intelligence (IEA/AIE'18)*, (LNCS, vol. 10868), pp. 857-868, 2018. Doi: 10.1007/978-3-319-92058-0\_82.

[62] Možina, M., Žabkar, J., Bench-Capon, T., & Bratko, I. (2005). Argument based machine learning applied to law. *Artificial Intelligence and Law*, 13(1), 53-73.

[63] P. Clark, T. Niblett, The CN2 induction algorithm, *Machine Learning Journal* 4 (3) (1989) 261–283

[64] Blue J Tax Analytics for tax positions [Электронный ресурс]. Доступно <https://www.bluej.com/ca/blue-j-tax>. Дата обращения: 2021.

[65] H. Sutherland, and F. Figari, “EUROMOD: the European Union tax-benefit microsimulation model”, *International journal of microsimulation*, vol. 6 (1), pp. 4-26, Apr. 2013.

[66] C. O'donoghue, M. Baldini, and D. Mantovani, *Modelling the Redistributive Impact of Indirect Taxes in Europe: An Application of EUROMOD*. EUROMOD Working Paper, no. EM7/01, 2004.

[67] R. Taylor, H. Sutherland, and J. Gomulka, “Using POLIMOD to evaluate alternative methods of expenditure imputation”, *Microsimulation Unit Research Note*, MU/RN38, 2001.

[68] J. D. Popple, “SHYSTER: A Pragmatic Legal Expert System”, PhD Thesis, Australian National Univ., Australia, 1993.

[69] T. A. O'callaghan, J. Popple, and E. McCreath, “SHYSTER-MYCIN: a hybrid legal expert system”, in *Proc. 9th int. conf. on Artificial intelligence and law*, 2003, pp. 103-104.

[70] *Legal Knowledge and Inf.Systems: JURIX 2007: The Twentieth Annual Conference*, A. R. Lodder and L. Mommers, Eds. IOS Press, Dec. 2007.

[71] A. Paliwala, “An Intellectual Celebration: A Review of the Jurix Legal Knowledge Based Systems Scholarship”, in *ICAIL 2000*, vol. 8, no. 4, pp. 317-335, Dec. 2000. Doi: 10.1023/A:1011242531937.

[72] R. H. Michaelsen, “An expert system for federal tax planning”, *Expert Systems*, vol. 1, no. 2, pp. 149-167, Oct. 1984. Doi: 10.1111 / j.1468-0394.1984.tb00440.x.

[73] R. M. Wygant, “CLIPS – a powerful development and delivery expert system tool”, *Computers & industrial engineering*, vol. 17, iss. 1-4, pp. 546-549, 1989. Doi: 10.1016/0360-8352(89)90121-6.

[74] L. T. McCarty, “Reflections on TAXMAN: An experiment in artificial intelligence and legal reasoning”, *Harv. L. Rev.*, vol. 90, no. 5, pp. 837-893, Mar. 1977. Doi: 10.2307/1340132.



[75] R. E. Susskind, "Expert systems in law: A jurisprudential approach to artificial intelligence and legal reasoning", *The modern law review*, vol. 49, no. 2, pp. 168-194, 1986. Doi: 10.1111/j.1468-2230.1986.tb01683.x.

[76] A. Greinke, "Legal expert systems: A humanistic critique of mechanical legal inference", *E Law*, n. 16, Nov. 1994. [Online]. Available: <http://www.murdoch.edu.au/elaw/issues/v1n4/greinke14.txt>. Accessed on: Nov. 25, 2019.

[77] А. Ад. Летичевский, "Инсерционное моделирование", *Управляющие системы и машины*, № 6, с. 3-14, 2012.

[78] A. Letichevsky and D. Gilbert, "A general theory of action languages", *Cybern Syst Anal*, vol. 34 (1), pp. 12-30, Jan. 1998. Doi: 10.1007/BF02911258.

[79] A. Letichevsky and D. Gilbert, "A Model for Interaction of Agents and Environments", in *14th Int. Workshop WADT'99, Recent Trends in Algebraic Development Techniques*, (LNCS, vol. 1827), 2000, pp. 311-328.

[80] A. A. Letichevsky and D. R. Gilbert, "Agents and environments", in *Proc. 1st Int. scientific and practical conf. on programming*, Sep. 1998, 10 s.

[81] R. Milner, *A Calculus of Communicating Systems*. Berlin, Heidelberg, Germany: Springer-Verlag, 1980.

[82] R. Milner, *Communication and Concurrency*. NJ, USA: Prentice Hall, 1989.

[83] R. Milner, "The polyadic  $\pi$ -calculus: a tutorial", in *Tech. Rep. ECS-LFCS-91-180, Edinburgh University (1991)*, Reprinted in *Logic and Algebra of Specification*, F. Bauer, W. Brauer and H. Schwichtenberg, Eds. Springer-Verlag, 1993, pp. 203-246.

[84] C. A. R. Hoare, *Communicating Sequential Processes*. NJ, USA: Prentice Hall, 1985.

[85] J. A. Bergstra, J. W. Klop, "Process algebra for synchronous communications", *Inf. and Control*, vol. 60 (1/3), pp. 109-137, 1984.

[86] A. A. Letichevsky, O. A. Letychevskiy, and V. S. Peschanenko, "Insertion modeling system", in *Perspectives of System Informatics Int. Andrei Ershov Memorial Conf. (PSI'11)*, (LNCS, vol. 7162), 2012, pp. 262-273.



[87] A. A. Letichevsky et al., “System Specification by Basic Protocols”, *Cybern Syst Anal*, vol. 41, iss. 4, pp. 479-493, July 2005. Doi: 10.1007/s10559-005-0083-y.

[88] A. Letichevsky, “Algebra of behavior transformations and its applications”, *Structural theory of Automata, Semigroups, and Universal Algebra* (NATO Science Series II: Mathematics, Physics and Chemistry (NAII), vol. 207), pp. 241–272, 2005. Doi: 10.1007/1-4020-3817-8\_10.

[89] А. А. Летичевский, А. Ал. Летичевский, В. С. Песчаненко, и А. А. Губа, “Генерація символічних трас в системі інсерційного моделювання”, *Кибернетика и системный анализ*, Т. 51, № 1, с. 7-19, 2015.

[90] А. А. Губа, и К. И. Шушпанов, "Инсерционная семантика плоских многопоточковых моделей языка UCM", *Управляющие системы и машины*, № 6, с. 15-34, 2012.

[91] A. Letichevsky et al., “Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications”, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, iss. 47, pp. 662-675, Dec. 2005.

[92] A. A.Letichevsky et al., “Insertion modeling in distributed system design”, *Проблеми програмування*, № 4, с. 13-38, 2008.

[93] A. A. Letichevsky, O. A. Letychevskiy, , and V. S. Peschanenko, “Insertion modeling system”, in *PSI'11*, (LNCS, vol. 7162), June 2011, pp. 262-273. Doi: 10.1007/978-3-642-29709-0\_23.

[94] O. Letychevskiy, and T. Weigert, "Symbolic verification of requirements in VRS system", in *2014 IEEE 22nd Int. Conf. RE*, Aug. 2014, pp. 331-332. DOI: 10.1109/RE.2014.6912282

[95] А. А. Летичевский, А. Б. Годлевский, и А. А. Летичевский (мл.), С. В. Потенко, В. С. Песчаненко, “Свойства предикатного трансформера системы VRS”, *Кибернетика и системный анализ*, № 4, с. 3-16, 2010.

[96] А. Б. Годлевский, “Предикатные преобразователи в контексте символьного моделирования транзиторных систем”, *Кибернетика и системный анализ*, № 4, с. 91-99, 2010.

[97] A. Letichevsky, A. Godlevsky, A. Guba, A. Kolchin, O. Letychevskiy, and V. Peschanenko, “Invariants in symbolic modeling and verification of requirements”, in *Ninth Int. Conf. on Computer Science and Inf. Technologies Revised Sel. Papers IEEE (SCIT 2013)*, Sept. 2013, pp. 1-6. Doi: 10.1109/CSITechnol.2013.6710332.

[98] V. Peschanenko, A. Guba, and C. Shushpanov, “Specializations and Symbolic Modeling”, in *Proc. 9-th Int. Conf. Information And Communication Technologies In Education, Research And Industry (ICTERI'13), CEUR-WS*. Jun 2013, vol. 1000, pp. 490-505.

[99] Верховна рада України. (1997, Квіт. 03, зі змінами). *Закон України № 168/97-ВР, Про податок на додану вартість*. [Електронний ресурс]. Доступно: <https://zakon.rada.gov.ua/laws/show/168/97-вр>.

[100] A. Letichevsky, O. Letychevskiy, V. Peschanenko, and M. Poltorackij, "An Algebraic Approach for Analyzing of Legal Requirements", in *2017 IEEE 25th Int. Requirements Eng. Conf. Workshops (REW'17)*, Sept. 2017, pp. 209-212. Doi: 10.1109/REW.2017.51.

[101] A. Godlevskiy et al., “Formalization and Algebraic Verification of Legal Requirements”, in *Proc. 13-th Int. Conf. ICTERI 2017, CEUR-WS*. May 2017, vol. 1844, pp. 524-535.

[102] APS APLAN. APS and IMS are best for rewriting and modelling. [Електронний ресурс]. Доступно: <http://apsystems.org.ua>. Дата обращения: Янв. 11, 2010.

[103] A. A. Letichevsky, and J. V. Kapitonova, "Algebraic programming in the APS system", in *Proc. int. symposium on Symbolic and algebraic computation (ISSAC'90)*, July 1990, pp. 68-75. Doi: 10.1145/96877.96896.

[104] A. A. Letichevsky, O. A. Letichevskiy, V. S. Peschanenko, “The Non-Deterministic Strategy of Rewriting”, *Управляющие системы и машины*, № 6, с. 53-58, 2013.

[105] E. Loper, and S. Bird, "NLTK: the Natural Language Toolkit", in *arXiv: a free distribution service and an open-access archive*, May 2002. [Online]. Available: <https://arxiv.org/abs/cs/0205028>. Accessed on: Sept. 14, 2019.

[106] P. Garg, "Sentiment Analysis of Twitter Data using NLTK in Python", PhD Thesis, Thapar University, Patiala – 147004, India, 2016.

[107] *ОРФО (сайт)*. [Электронный ресурс]. Доступ: <https://orfo.ru/>. Дата обращения: Янв. 11, 2020.

[108] А. В. Пруцков, "Генерация и определения форм слов естественных языков на основе их последовательных преобразований", *Вестник Рязанского государственного радиотехнического университета*, № 27, с. 51-58, 2009.

[109] Д. Л. Косов, и В. М. Белов, "Текстовые анализаторы как средство автоматизации правовых экспертиз нормативно-правовых актов и их проектов", *Гражданское общество и правовое государство*, № 1, с. 107-109, 2015.

[110] А. Л. Огарок, "Метод полного лингвистического анализа неструктурированной текстовой информации", *Информатизация и связь*, № 1, с. 91-99, 2018.

[111] M.-C. de Marneffe, and C. D. Manning, "Stanford typed dependencies manual", Stanford University, Stanford, USA, Tech. Rep., Sep. 2008.

[112] B. Carpenter, "Phrasal Queries with LingPipe and Lucene: Ad Hoc Genomics Text Retrieval", in *Proc.13th Text REtrieval Conference (TREC 2004)*, Vol. 1., Oct. 2004. [Online]. Available: <https://trec.nist.gov/pubs/trec13/papers/alias-i.geo.pdf>. Accessed on: Sept. 14, 2019.

[113] А. А. Хорошилов, А. А. Хорошилов, А. А. Хорошилов, Ю. В. Никитин, М. В. Смирнов, и Д. А. Садовников, "Сервер лингвистического ПО МетаФраз R10 (MF Lingware Server R10)", *Свидетельство о гос. регистрации программы для ЭВМ № 2014662743*, Дек. 08, 2014.

[114] А. А. Хорошилов, А. А. Хорошилов, А. А. Хорошилов, Ю. В. Никитин, М. В. Смирнов, и Д. А. Садовников, "Лингвистический комплекс МетаФраз R10 (MF Lingware Complex R10)", *Свидетельство о гос. регистрации программы для ЭВМ № 2014663079*, Дек. 15, 2014.

[115] П. В. Паничева, Е. В. Протопопова, А. Р. Мирзагитова, и О. А. Митрофанова, "Разработка лингвистического комплекса для морфологического анализа русскоязычных корпусов текстов на основе Rymorphy и NLTK", в *Труды междунар. конф. "Корпусная лингвистика – 2015"*, СПб, 2015, с. 361-373.

[116] E. Kuzmenko, "Morphological analysis for Russian: Integration and comparison of taggers", in *Int. Conf. on Analysis of Images, Social Networks and Texts (AIST 2016)*, (Communications in Computer and Information Science (CCIS), vol. 661), Apr. 2016, pp. 162-171. Doi: 10.1007/978-3-319-52920-2\_16.

[117] Brat rapid annotation tool manual, in *Brat rapid annotation tool (website)*. [Online]. Available: <http://brat.nlplab.org/manual.html>. Accessed on: Sept. 14, 2019.

[118] D. S. Zuev, A. A. Marchenko, and A. F. Khasiannov, "Text mining tools in legal documents", in *CEUR Workshop Proc.*, vol. 2022, Oct. 2017, pp. 214-218.

[119] А. В. Рубайло, и М. Ю. Косенко, "Программные средства извлечения информации из текстов на естественном языке", *Альманах современной науки и образования*, № 12, с. 87-92, 2016.

[120] E. V. Politsyna, S. A. Politsyn, and A. S. Porechny, "Development of the Cross-platform Library of Morphological Analysis of the Russian Language Text for Industrial Software", in *Proc. 14th Central and Eastern European Software Eng. Conf.*, 2018, Article no. 11, pp. 1-8. Doi: 10.1145/3290621.3290635.

[121] *Pullenti 4.3 (website)*. [Online]. Available: <http://www.pullenti.ru/>. Accessed on: Sept. 14, 2019.

[122] Р. Стефанчук, Д. Монастырский, В. Рогожа, та В. Ватрас, "Вимоги до змісту юридичного документа", у *Правове письмо, навч. посіб.* Київ, Україна: Алерта; ЦУЛ, 2011. [Електронний ресурс]. Режим доступу: [https://pidru4niki.com/1779070751491/dokumentoznavstvo/vimogi\\_zmistu\\_yuridichno\\_go\\_dokumenta](https://pidru4niki.com/1779070751491/dokumentoznavstvo/vimogi_zmistu_yuridichno_go_dokumenta). Дата звернення: Окт. 20, 2019.

[123] Окружний адміністративний суд Автономної республіки Крим (2013, Черв. 04). *Постанова у справі № 801/3685/13-а*. [Електронний ресурс]. Доступно: <https://reyestr.court.gov.ua/Review/32553284>. Дата звернення: Окт. 20, 2019.

[124] S. A. White, "Introduction to BPMN", in BPTrends (website). Ibm Corporation, July 2004. [Online]. Available: <https://www.bptrends.com/introduction-to-bpmn/>. Accessed on: Sept. 14, 2019.

[125] A. Presley, and D. H. Liles, "The use of IDEF0 for the design and specification of methodologies", in *Proc. of the 4th Industrial Engineering Research Conference (IERC'94)*, May 1995, pp. 442-448.

[126] *The token economy: A review and evaluation*, A. Kazdin, Ed. NY, USA: Springer Science & Business Media, 2012.

[127] A. Sunyaev, "Distributed ledger technology", in *Internet Computing. Principles of Distributed Systems and Emerging Internet-Based Technologies*. Cham, Switzerland: Springer, 2020. pp. 265-299.

[128] V. Kobets, V. Yatsenko, and M. Poltoratskiy, "Dynamic Model of Double Electronic Vickrey Auction", in *Proc. 11-th Int. Conf. ICTERI 2015, CEUR-WS*, May 2015, vol. 1356, pp. 236-251. DOI: 10.13140/RG.2.1.4066.7040.

[129] М. О. Вінник, В. В. Одінецов, та М. Ю. Полторацький, "Використання інформаційних технологій при вивченні популяційної біології", *Проблеми інформаційних технологій*, № 20, с. 168-175, 2016.

[130] V. Kobets, and M. Poltoratskiy, "Using an evolutionary algorithm to improve investment strategies for industries in an economic system", in *Proc. 5-th Int. Workshop ITER 2016, CEUR-WS*, June 2016, vol. 1614, pp. 485-501.

[131] S. Mancarella, *Business Process Modelling. Notation – A tutorial*. USA, OMG SOA Healthcare, SPARX Systems, 2011.

[132] T. Rademakers, *Activiti in Action: Executable business processes in BPMN 2.0*. Manning Publications, 2012. [Online]. Available: <https://www.manning.com/books/activiti-in-action>. Accessed on: Oct. 15, 2019.

[133] V. Kobets, and V. Yatsenko, "Adjusting Business Processes by the Means of an Autoregressive Model Using BPMN 2.0", in *Proc. 9-th Int. Conf. ICTERI 2013, CEUR-WS*, June 2016, vol. 1614, pp. 518-533.

[134] Bizagi Studio/Modeler, in *Bizagi (website)*. [Online]. Available: <https://www.bizagi.com/en>. Accessed on: Oct. 15, 2019.

[135] И. И. Гниломедов, "Моделирование экономических агентов при помощи конечных автоматов", в *Научные доклады конф. «Интегрированные модели, мягкие вычисления, вероятностные системы и комплексы программ в искусственном интеллекте»*, М.: Физматлит, 2009, Т. 2, с. 72-89.

[136] G. Zimmermann, R. Neuneier, and R. Grothmann, "Multi-agent FX-Market Modeling Based on Cognitive Systems", in *Proc. Artificial Neural Networks – ICANN 2001*, (LNCS, vol. 2130), Aug. 2001, pp. 767-774. Doi: 10.1007/3-540-44668-0\_107.

[137] R. Draeseke, and D. EA Giles, "A fuzzy logic approach to modelling the New Zealand underground economy", *Mathematics and Computers in Simulation*, vol. 59, iss.1-3, pp. 115-123, 2002. Doi: 10.1016/S0378-4754(01)00399-8.

[138] A. Hassan, "Relevance of fuzzy logic in the economy", *International scientific journal*, vol. 4 (2), pp. 104-105, 2016.

[139] D. Nemchenko, V. Kobets, and L. Potravka, "Neuro-Fuzzy Model of Development Forecasting and Effective Agrarian Sector Transformations of Ukraine", in *Proc. 14-th Int. Conf. ICTERI 2018, vol. II: Workshops, CEUR-WS*, May 2018, vol. 2104, pp. 84-99.

[140] H. Storm, K. Baylis, and T. Heckelei, "Machine learning in agricultural and applied economics", *European Review of Agricultural Economics*, vol. 47, iss. 3, pp. 849-892, July 2020. Doi: 10.1093/erae/jbz033.

[141] A. Storkey, "Machine learning markets", in *Proc. 14th Int. Conf. Artificial Intelligence and Statistics (AISTATS 2011)*, vol. 15, June 2011, pp. 716-724.

[142] B. Char, K. Geddes, G. Gonnet, B. Leong, M. Monagan, and S. Watt, *Maple V library reference manual*. NY, USA: Springer Science & Business Media, 2013.

[143] Z. Chvátalová, and J. Hřebíček, "Computational finance and finance economics with maple", *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 7, iss. 5, pp. 541-550, 2013.

[144] J. Hřebíček, "Mathematical Modeling of Economic Phenomena with Maple", in *Proc. 30th Int. Conf. Mathematical Methods in Economics (MME'12)*, 2012, pp. 326-331.

[145] J. Luedicke, *POWERSIM: Stata module for simulation-based power analysis for linear and generalized linear models, (Statistical Software Components)*. Boston, USA: Boston College Department of Economics, 2013.

[146] iTHINK, in *isee systems (website)*. [Online]. Available: <https://www.iseesystems.com/store/products/ithink.aspx>. Accessed on: Jan. 09, 2020.

[147] W. D. Kelton, *Simulation with ARENA*. McGraw-hill, 2002.

[148] C. Pîrnău, “Modeling gearing of regional eco-bio-logistic components in bioeconomic management, using Arena simulation software”, *Journal of Knowledge Management, Economics and Information Technology*, vol. 4(1.1), pp. 1-14, Feb.2014.

[149] K. Jilcha, E. Berhan, and H. Sherif, “Workers and Machine Performance Modeling in Manufacturing System Using Arena Simulation”, *J Comput Sci Syst Biol*, vol. 8(4), pp. 185-190, 2015. Doi: 10.4172/jcsb.1000187.

[150] *GPSS: имитационное моделирование систем (сайт)*. [Электронный ресурс]. Доступ: <http://www.gpss.ru/>. Дата обращения: Янв. 09, 2020.

[151] S. A. Vlasov, T. V. Devyatkov, and V. V. Devyatkov, “Cloud technology in simulation studies: GPSS cloud project”, in *7th IFAC Manufacturing Modelling, Management, and Control Conf., IFAC Proc. Volumes*, 2013, vol. 46, iss. 9, pp. 637-641. Doi: [10.3182/20130619-3-RU-3018.00582](https://doi.org/10.3182/20130619-3-RU-3018.00582).

[152] B. Srivastava, J. P. Bigus, and D. A. Schlosnagle, "Using ABLE to Bring Planning to Business Applications", *American Association for Artificial Intelligence (www.aaai.org)*, 2004.

[153] J. P. Bigus, D. A. Schlosnagle, J. R. Pilgrim, W. N. Mills III, and Y. Diao, “ABLE: A toolkit for building multiagent autonomic systems”, *IBM Systems Journal*, vol. 41, iss. 3, pp. 350-371, 2002. Doi: [doi.org/10.1147/sj.413.0350](https://doi.org/10.1147/sj.413.0350).

[154] B. LeBaron, "Agent-based Financial Markets: Matching Stylized Facts with Style", in *Post Walrasian Macroeconomics: Beyond the DSGE Model*, D. Colander, Ed. Cambridge, United Kingdom: Cambridge University Press, 2006, pp. 221-236.

[155] *AnyLogic (website)*. [Online]. Available: <https://www.anylogic.com/>. Accessed on: Jan. 09, 2020.



[156] J. Ren, Q. Zhao, B. Liu, and C. Chen, "Selection of pallet management strategies from the perspective of supply chain cost with Anylogic software", *PLOS ONE*, 14.6: e0217995, June 2019. Doi: 10.1371/journal.pone.0217995.

[157] B. Du, and H. Liu, "Modeling and Analysis of Linyi Regional Logistics System Based on Anylogic", *Logistics Technology*, vol. 19, p. 43, 2012.

[158] Ascape 5.0, in *Ascape Guide (website)*. [Online]. Available: <http://ascape.sourceforge.net/>. Accessed on: Jan. 09, 2020.

[159] MATLAB, in *MathWorks (website)*. [Online]. Available: <https://www.mathworks.com/products/matlab.html>. Accessed on: Jan. 09, 2020.

[160] Y. Li, and S. Malin, "Research on BP Neural Network for Nonlinear Economic Modeling and its Realization Based on Matlab", in *2009 Third Int. Symposium on Intell. Inf. Technol. Application (IITA'09)*, Nov. 2009, vol. 1, pp. 505-508. Doi: [10.1109/IITA.2009.352](https://doi.org/10.1109/IITA.2009.352).

[161] NetLogo, in *CCL: Center for Connected Learning and Computer-Based Modeling (website)*. [Online]. Available: <https://ccl.northwestern.edu/netlogo/>. Accessed on: Jan. 09, 2020.

[162] B. Head, A. Hjorth, C. Brady, and U. Wilensky, "Evolving agent cognition with Netlogo LevelSpace", in *2015 Winter Simulation Conf. (WSC'15)*, Dec. 2015, pp. 3122-3123.

[163] J. C. Thiele, "R marries NetLogo: introduction to the RNetLogo package", *Journal of Statistical Software*, vol. 58, iss. 2, pp. 1-41, 2014. Doi: 10.18637/jss.v058.i02.

[164] L. R. Izquierdo, D. Olaru, S. S. Izquierdo, S. Purchase, and G. N. Soutar, "Fuzzy logic for social simulation using NetLogo", *Journal of Artificial Societies and Social Simulation (JASSS)*, vol. 18, iss. 4, Article 1, 2015. Doi: 10.18564/jasss.2885.

[165] A. Kopp, and D. Orlovskyi, "A Method for Business Process Model Analysis and Improvement", in *Proc. 15th Int. Conf. ICTERI 2019: PhD Symposium, CEUR-WS*, June 2019, vol. 2403, pp. 1-10.



[166] V. Peschanenko, M. Poltoratskyi, and K. Pryimak, "Formalization and algebraic modeling of university economics", in *Proc. 15th Int. Conf. ICTERI 2019: Workshops (RMSE'19)*, CEUR-WS, June 2019, vol. 2393, pp. 585-594.

[167] Кривогуб, А. М., and К. В. Новодережкіна. "Модель загальної рівноваги та методи її дослідження." *Математические машины и системы* 4 (2015).

[168] C. Dannen, *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Berkeley, USA: Apress, 2017. Doi: 10.1007/978-1-4842-2535-6.

[169] C. Dannen, "Solidity programming", in *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Berkeley, USA: Apress, 2017, pp. 69-88.

[170] Z. Yang, and H. Lei. "Lolisa: Formal syntax and semantics for a subset of the solidity programming language in Mathematical Tool Coq", in *arXiv: a free distribution service and an open-access archive*, Mar. 2018. Doi: 10.1155/2020/6191537. [Online]. Available: <https://arxiv.org/abs/1803.09885>. Accessed on: Feb. 12, 2019.

[171] R. Wandmacher, "Tokenomics", in *Cryptofinance and Mechanisms of Exchange. Contributions to Management Science*, S. Goutte, K. Guesmi, S. Saadi, Eds. Cham, Switzerland: Springer, 2019, pp. 113-123. Doi: 10.1007/978-3-030-30738-7\_7.

[172] L. W. Cong, Y. Li, and N. Wang, "Tokenomics: Dynamic adoption and valuation", in *NBER Working Papers*, National Bureau of Economic Research, Inc., 2020, no. 27222.

[173] K. Samani, "The Blockchain Token Velocity Problem", in *CoinDesk (media website)*, Dec. 8, 2017, Updated Jan. 6, 2018. [Online]. Available: <https://www.coindesk.com/blockchain-token-velocity-problem>. Accessed on: Jan. 09, 2020.

[174] O. Letychevsky, V. Peschanenko, V. Radchenko, M. Poltoratzkyi, P. Kovalenko, and S. Mogylko, "Formal Verification of Token Economy Models", in *Proc. 2019 IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, Seoul, Korea,

May 2019. Doi: 10.1109/BLOC.2019.8751318. [Online]. Available: <https://ieeexplore.ieee.org/document/8751318>. Accessed on: Jan. 09, 2020.

[175] O. Letychevskiy, V. Peschanenko, V. Radchenko, M. Poltoratskiy, and Y. Tarasich, "Formalization and algebraic modeling of tokenomics projects", in *Proc. 15th Int. Conf. ICTERI 2019: Workshops (RMSE'19)*, CEUR-WS, June 2019, vol. 2393, pp. 577-584.

[176] O. Letychevskiy, V. Peschanenko, M. Poltoratskiy, and Y. Tarasich, "Our Approach to Formal Verification of Token Economy Models", in *Proc. Int. Conf. ICTERI 2019, Communications in Computer and Information Science*, vol. 1175, pp. 348-363. Doi: 10.1007/978-3-030-39459-2\_16.

[177] O. Letychevskiy, V. Peschanenko, M. Poltoratskiy, and Y. Tarasich, "Platform for Modeling of Algebraic Behavior: Experience and Conclusions", in *Proc. 16th Int. Conf. ICTERI 2020: Workshops (RMSEBT'20)*, CEUR-WS, Oct. 2020, vol. 2732, pp. 42-57.

[178] H. Davarikia, and M. Barati, "A tri-level programming model for attack-resilient control of power grids", *Journal of Modern Power Systems and Clean Energy*, vol. 6, iss. 5, pp. 918-929, Sept. 2018. Doi: 10.1007/s40565-018-0436-y.

[179] M. Mehar, et al., "Understanding a revolutionary and Flawed Grand Experiment in Blockchain: the DAO Attack", *Journal of Cases on InformationTechnology (JCIT)*, vol. 21, iss.1, pp. 19-32, 2019. Doi: 10.4018/JCIT.2019010102.

[180] G. O. Karame, E. Androulaki, and S. Capkun, "Double-Spending Fast Payments in Bitcoin", in *Proc. 2012 ACM conf. on Computer and commun. security (CCS'12)*, Oct. 2012, pp. 906-917. Doi: 10.1145/2382196.2382292.

[181] C. Liu, H. Liu, Z. Cao, Z. Chen, B. Chen, B. Roscoe, "ReGuard: Finding Reentrancy Bugs in Smart Contracts", in *Proc. 2018 IEEE/ACM 40th Int. Conf. Software Engineering: Companion (ICSE'18 Companion)*, May-June 2018, pp. 65-68. Doi: 10.1145/3183440.3183495.

## ДОДАТКИ

## Додаток А

## Список публікацій здобувача

*Основні наукові результати дисертації*

1. V. Kobets, and M. Poltoratskiy "Forming an evolutionarily stable firm strategy under Cournot competition using social preferences", *Communications in Computer and Information Science*, Springer, Cham, 2014, 2014, (CCIS, vol. 469), Juny 2014, pp. 343-361, DOI: 10.1007/978-3-319-13206-8\_17.
2. V. Kobets, V. Yatsenko, and M. Poltoratskiy, "Dynamic Model of Double Electronic Vickrey Auction", in *Proc. 11-th Int. Conf. ICTERI 2015, CEUR-WS*, May 2015, vol. 1356, pp. 236-251. DOI: 10.13140/RG.2.1.4066.7040.
3. V. Kobets, and M. Poltoratskiy, "Using an evolutionary algorithm to improve investment strategies for industries in an economic system", in *Proc. 5-th Int. Workshop ITER 2016, CEUR-WS*, June 2016, vol. 1614, pp. 485-501.
4. Godlevskiy et al., "Formalization and Algebraic Verification of Legal Requirements", in *Proc. 13-th Int. Conf. ICTERI 2017, CEUR-WS*, May 2017, vol. 1844, pp. 524-535.
5. Letichevsky, O. Letychevskiy, V. Peschanenko, and M. Poltorackij, "An Algebraic Approach for Analyzing of Legal Requirements", in *2017 IEEE 25th Int. Requirements Eng. Conf. Workshops (REW'17)*, Sept. 2017, pp. 209-212. DOI: [10.1109/REW.2017.51](https://doi.org/10.1109/REW.2017.51).
6. V. Peschanenko, M. Poltorackiy, "Use of methods of algebraic programming for the formal verification of legal acts", *Проблеми програмування*, № 2–3, с. 109-114, 2018. DOI: 10.15407/pp2018.02.109.
7. O. Letychevsky, V. Peschanenko, V. Radchenko, M. Poltoratzkyi, P. Kovalenko, and S. Mogytko, "Formal Verification of Token Economy Models", in *Proc. 2019 IEEE Int. Conf. Blockchain Cryptocurrency (ICBC'19)*, May 2019, pp. 201-204. DOI: 10.1109/BLOC.2019.8751318.
8. O. Letychevskiy, V. Peschanenko, V. Radchenko, M. Poltoratskiy, and Y. Tarasich, "Formalization and algebraic modeling of tokenomics projects", in

*Proc. 15th Int. Conf. ICTERI 2019: Workshops (RMSE'19), CEUR-WS*, June 2019, vol. 2393, pp. 577-584.

9. V. Peschanenko, M. Poltoratskyi, and K. Pryimak, "Formalization and algebraic modeling of university economics", in *Proc. 15th Int. Conf. ICTERI 2019: Workshops (RMSE'19), CEUR-WS*, June 2019, vol. 2393, pp. 585-594.
10. O. Letychevskyi, V. Peschanenko, M. Poltoratskyi, and Y. Tarasich, "Our Approach to Formal Verification of Token Economy Models", *Communications in Computer and Information Science*, vol. 1175, pp. 348-363. DOI: 10.1007/978-3-030-39459-2\_16.
11. O. Letychevskyi, V. Peschanenko, M. Poltoratskyi, and Y. Tarasich, "Platform for Modeling of Algebraic Behavior: Experience and Conclusions", in *Proc. 16th Int. Conf. ICTERI 2020: Workshops (RMSEBT'20), CEUR-WS*, Oct. 2020, vol. 2732, pp.42-57.

**Апробація матеріалів дослідження  
на конференціях, семінарах, круглих столах тощо**

<b>Рівень заходу та його назва</b>	<b>Місце й дата проведення</b>	<b>Форма участі</b>
Міжнародна наукова конференція ICTERI 2014	м. Львів, травень 2014 р.	Очна, 1 публікація в матеріалах конференції
Міжнародна наукова конференція ICTERI 2015	м. Львів, травень 2015 р.	Очна, 1 публікація в матеріалах конференції
Міжнародна наукова конференція ICTERI 2016	м. Львів, 2016 р.	Очна, 1 публікація в науковому виданні, що індексується Scopus
Міжнародна наукова конференція ICTERI 2017	м. Київ, 2017 р.	Очна, 1 публікація в науковому виданні, що індексується Scopus
2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)	м. Лісабон, Португалія, 2017 р.	Заочна, 1 публікація в науковому виданні, що індексується Scopus
2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)	м. Сінгапур, 2019	Заочна, 1 публікація в науковому виданні, що індексується Scopus
Міжнародна наукова конференція ICTERI 2019	м. Харків, 2019 р.	Очна, 1 публікація в науковому виданні, що індексується Scopus

## Модель реєстрації платника податку ПДВ

### 1. Формалізація середовища:

*environment*(

*types: obj*(

*TENSE:(GONNA\_BE, DOES, DID, DOES\_NOT),*

*RESPONSIBILITY:(TAX\_PAYMENT, KEEP\_TAXFREE\_REQUIREMENTS, OTHER),*

*USAGE\_TYPES:(NOT\_TARGETED, CONDITIONS\_VIOLATION, TARGETED),*

*PRESENT:(PERSON, DIRECTOR, REPRESENTER, OTHER),*

*ACTIVITIES:(JOINT\_ACTIVITY\_CONTRACT, CONTROL\_ESTATE, CONFISCATE, OTHER)*

*);*

*attributes: obj* (

*limitVAT:int,*

*limitVATimport:int,*

*timeActual:obj*(

*day:int,*

*month:int,*

*year:int*

*),*

*A180\_1:bool,*

*A180\_1\_1:bool,*

*A180\_1\_2:bool,*

*A180\_1\_3:bool,*

*A180\_1\_3\_1:bool,*

*A180\_1\_3\_2:bool,*

*A180\_1\_3\_3:bool,*

*A180\_1\_4:bool,*

```

A180_1_5:bool,
A180_1_6:bool,
A181_1:bool,
VIOLATION_CONDITION_1:bool,
VIOLATION_CONDITION_2:bool,
VIOLATION_CONDITION_3:bool,
VAT_CANCELATION_CONDITION_1:bool,
VAT_CANCELATION_CONDITION_2:bool,
VAT_CANCELATION_CONDITION_3:bool
);
agent_types: obj (
  Subject: obj(
    commercialActivity:obj (
      status:TENSE,
      totalSalesSumTaxable: (int, int) -> int,
      sale: (int) -> int
    ),
    registrationVAT:obj(
      status:TENSE,
      decisionVoluntaryVAT:bool,
      necessity:bool,
      presentApplication:PRESENT
    ),
    cancelationVAT:obj(
      elimination bool,
      singleTaxRegistration bool,
      courtDecision bool
    ),
    importActivity:obj(
      status:TENSE,

```

```
responsibility:RESPONSIBILITY,  
taxModeViolation:bool,  
taxExemption:obj(  
    given:bool,  
    usage:USAGE_TYPES  
),  
taxSalesCultureValues:bool,  
totalImportSumTaxable: (int,int) -> int  
),  
otherActivity:ACTIVITIES,  
timeVATstart:obj(  
    day:int,  
    month:int,  
    year:int  
),  
timeVATcancelation:obj(  
    day:int,  
    month:int,  
    year:int  
),  
timeApplication:obj(  
    day:int,  
    month:int,  
    year:int  
),  
timeVATperiod:obj(  
    day:int,  
    month:int,  
    year:int  
),
```



```

timeChangingAttributes:obj(
    day:int,
    month:int,
    year:int
),
taxVATPayer:bool,
singleTaxGroup1:bool,
singleTaxGroup2:bool,
singleTaxGroup3:bool,
totalVendorSumTaxable: (int,int)->int,
VAT: (int)->int,
taxObligation: int,
sales: (int)->int,
saleNumber: int,
expenses: (int)->int,
changingAttributes:bool
)
);
agents: obj (Subject: obj(x));
instances: ( Nil);
axioms: obj(
    A180_1 : (A180_1 <=> (x.taxVATPayer <=> (A180_1_1 // A180_1_2 //
A180_1_3 & A180_1_3_3 // (A180_1_3 & (A180_1_3_1 // A180_1_3_2) &
A180_1_3_3) //A180_1_4 // A180_1_5))),
    A180_1_1 : (A180_1_1 <=> (((x.commercialActivity.status = GONNA_BE) //
(x.commercialActivity.status = DOES)) & (x.registrationVAT.status = DOES) &
x.registrationVAT.decisionVoluntaryVAT)),
    A180_1_2 : (A180_1_2 <=> (x.registrationVAT.necessity //
(x.registrationVAT.status = DID))),

```

**A180\_1\_3** : (*Exist* (*i:int*) ((**A180\_1\_3**  $\Leftrightarrow$  (*x.importActivity.status* = *DOES*) & ((*x.importActivity.totalImportSumTaxable*)(*x.timeVATstart.month* - 11 + *i, timeActual.month*)  $\geq$  *limitVATimport*) & (*i*  $\geq$  0) & (*i*  $\leq$  11) & (*x.importActivity.responsibility* = *TAX\_PAYMENT*))))),

**A180\_1\_3\_1** : (**A180\_1\_3\_1**  $\Leftrightarrow$  (*x.importActivity.responsibility* = *KEEP\_TAXFREE\_REQUIREMENTS*) & *x.importActivity.taxModeViolation*),

**A180\_1\_3\_2** : (**A180\_1\_3\_2**  $\Leftrightarrow$  *x.importActivity.taxExemption.given* & ((*x.importActivity.taxExemption.usage* = *NOT\_TARGETED*) //

(*x.importActivity.taxExemption.usage* = *CONDITIONS\_VIOLATION*))),

**A180\_1\_3\_3** : (**A180\_1\_3\_3**  $\Leftrightarrow$  !(*x.importActivity.taxSalesCultureValues*)),

**A180\_1\_4** : (**A180\_1\_4**  $\Leftrightarrow$  (*x.otherActivity* = *JOINT\_ACTIVITY\_CONTRACT*)),

**A180\_1\_5** : (**A180\_1\_5**  $\Leftrightarrow$  (*x.otherActivity* = *CONTROL\_ESTATE*)),

**A180\_1\_6** : (**A180\_1\_6**  $\Leftrightarrow$  (*x.otherActivity* = *CONFISCATE*)),

**VIOLATION\_CONDITION\_1** : (**VIOLATION\_CONDITION\_1**  $\Leftrightarrow$  !((*timeActual.month* = *x.timeVATstart.month*) // (*timeActual.month* = *x.timeVATstart.month* + 1) & (*timeActual.day*  $\leq$  *C\_NEXT\_WORK\_DAY*(10,*x.timeVATstart.month* + 1,*x.timeVATstart.year* + 1))),

**VIOLATION\_CONDITION\_2** : (**VIOLATION\_CONDITION\_2**  $\Leftrightarrow$  !(((*x.registrationVAT.decisionVoluntaryVAT* // (*x.registrationVAT.necessity* & (*x.otherActivity*=*CONFISCATE*)))) & (*C\_DAYS*(*timeActual.day, timeActual.month, timeActual.year, x.timeApplication.day, x.timeApplication.month, x.timeApplication.year*)  $\geq$  20))))),

**VIOLATION\_CONDITION\_3** : (**VIOLATION\_CONDITION\_3**  $\Leftrightarrow$  !((*x.registrationVAT.presentApplication* = *PERSON*) // (*x.registrationVAT.presentApplication* = *DIRECTOR*) // (*x.registrationVAT.presentApplication* = *REPRESENTER*))),

**VAT\_CANCELATION\_CONDITION\_1** : (**VAT\_CANCELATION\_CONDITION\_1**  $\Leftrightarrow$  (*timeActual.month* >

```

x.timeVATperiod.month + 12) &
(x.totalVendorSumTaxable(timeActual.month,timeActual.month - 12) <
limitVAT)),
VAT_CANCELATION_CONDITION_2 :
(VAT_CANCELATION_CONDITION_2 <=>
(x.cancelationVAT.elimination & (x.taxObligation = 0)) //
(x.cancelationVAT.singleTaxRegistration) //
(x.cancelationVAT.courtDecision)),
VAT_CANCELATION_CONDITION_3 :
(VAT_CANCELATION_CONDITION_3 <=>
Forall (i:int) ((timeActual.month - i>0&timeActual.month - i<=12) ->
(x.VAT(timeActual.month - i)=0)))
);
logic_formula: obj( (limitVATimport = 1000000) & (limitVAT = 1000000) &
!(x.registrationVAT.necessity) & (timeActual.month = 1) & (timeActual.day =1)
& (timeActual.year = 2016) &(x.registrationVAT.status = DOES_NOT) &
!(x.registrationVAT.decisionVoluntaryVAT) & (x.saleNumber = 1)
&(x.timeVATstart.day=0)
);
reductions: (Nil);
initial: env(
  obj(
    attributes: obj(Nil);
    agent_parameters: obj(Nil)
  ),
  state(
    Subject(x, Idle)
  )
);

```

## 2. Мова дій:

```

BP181_1 = (Forall (i:int) (((x.totalVendorSumTaxable)(timeActual.month - 11 + i,
timeActual.month) >= limitVAT) & (i >= 0) & (i <= 11) & !(x.singleTaxGroup1)
& !(x.singleTaxGroup2) & !(x.singleTaxGroup3) & !((x.importActivity.status =
DOES) & ((x.importActivity.totalImportSumTaxable)(timeActual.month - 11 + i,
timeActual.month) >= limitVATimport)))) -> ("181_1")
(C_MOVE_DATE (x.timeVATstart, timeActual);x.registrationVAT.necessity))
),
BP182_1 =(1 -> ("182_1") (x.registrationVAT.decisionVoluntaryVAT)),
BP183_1 = (1 -> ("183.1 Give the registration application")
(C_MOVE_DATE(x.timeApplication,timeActual))
),
BP183_5 =(1 -> ("Set the desired date of VAT start")(
(x.timeVATstart.day > timeActual.day) & (x.timeVATstart.month >
timeActual.month) & (x.timeVATstart.year > timeActual.year))
),
BP183_8 =((VIOLATION_CONDITION_1 // VIOLATION_CONDITION_2 //
VIOLATION_CONDITION_3 // !(x.taxVATPayer)) ->
("183.8 Cancelation of registration application") 1
),
BP183_9 = ((!(VIOLATION_CONDITION_1 // VIOLATION_CONDITION_2 //
VIOLATION_CONDITION_3) // !(x.taxVATPayer)) ->
("Registration of VAT payer") 1
),
BP183_9_DATE1 = ((x.registrationVAT.decisionVoluntaryVAT &
(x.timeVATstart.day > 0))->("183.9 Definition of the start VAT tax period for
voluntary with desired date") (x.timeVATperiod.day :=
x.timeVATstart.day;x.timeVATperiod.month :=
x.timeVATstart.month;x.timeVATperiod.year := x.timeVATstart.year)

```

),

**BP183\_9\_DATE2** = ((*x.registrationVAT.decisionVoluntaryVAT* &  
*(x.timeVATstart.day = 0)*)->("183.9 Definition of the start VAT tax period for  
voluntary without desired date")  
(*C\_SET\_DATE(x.timeVATperiod,C\_GET\_FIRST\_DAY\_NEXT\_MONTH(x.timeA  
pplication.day + 20,x.timeApplication.month,x.timeApplication.year))*))

),

**BP183\_9\_DATE3** = ((*x.registrationVAT.necessity*)->  
("183.9 Definition of the start VAT tax period for necessary")  
(*C\_SET\_DATE(x.timeVATperiod,x.timeApplication.day + 3,  
x.timeVATperiod.month, x.timeApplication.year)*))

),

**BP183\_10** = ((*x.timeApplication.day = 0*)-> ("183\_10 Responsibility") 1),

**BP183\_15** = (*x.changingAttributes* -> ("183.15 Re-registration")->  
(*C\_MOVE\_DATE(timeActual,x.timeChangingAttributes)*))

),

**BP184\_1** = (1 -> ("184.1 a. Cancellation as VAT payer due to total taxes. Submit  
application to Taxation Office")

(*C\_MOVE\_DATE(x.timeVATcancellation, timeActual)*))

),

**BP184\_1\_A** = (*VAT\_CANCELATION\_CONDITION\_1* ->

("Satisfaction of VAT registrarion cancellation")

(*x.registrationVAT.status := DOES\_NOT*)

),

**BP184\_1\_OTHER** = (*VAT\_CANCELATION\_CONDITION\_2*->

("VAT registrarion cancellation") (*x.registrationVAT.status := DOES\_NOT*)

),

**BP184\_1\_G** = (*VAT\_CANCELATION\_CONDITION\_3*->

("VAT registrarion cancellation") (*x.registrationVAT.status := DOES\_NOT*)

),

```

BP184_3 = ( !(VAT_CANCELATION_CONDITION_1 //
VAT_CANCELATION_CONDITION_2 //
VAT_CANCELATION_CONDITION_3) -> ("Refuse of application") 1
),
BP_NEXT_TIME = (1 -> ("Next Period")
(C_SET_DATE(timeActual,timeActual.day + 1,
timeActual.month,timeActual.year))
),
BP_SALES_1 = (1 -> ("Sales") (saleNumber := saleNumber + 1))

```

### 3 Алгебра поведінки:

```

B0 = BP181_1.B1_NEC + BP182_1.B1_VOL + BP_NEXT_TIME.B0,
B1_NEC = BP183_1.B2_NEC + BP_NEXT_TIME.B1_NEC,
B2_NEC = BP183_8.B_TERMINATION + BP183_9.B3_NEC,
B3_NEC = BP183_9_DATE3.B2,
B1_VOL = BP183_1.BP183_5.B2_VOL + BP183_1.B2_VOL +
BP_NEXT_TIME.B1_VOL,
B2_VOL = BP183_8.B_TERMINATION + BP183_9.B3_VOL,
B3_VOL = BP183_9_DATE1.B2 + BP183_9_DATE2.B2,
B2 = BP183_15.BP_NEXT_TIME.B2 + BP_NEXT_TIME.B2 + BP_NEXT_TIME.B3,
B3 = BP184_1.B4,
B4 = BP184_1_A.B0 + BP184_1_G.B0 + BP184_1_OTHER.B0 + BP184_3.B2

```

## Модель SKILLONOMY

Оскільки формалізація моделі *SKILLONOMY* частина має значний обсяг, її опубліковано на публічному акаунті Google Drive. Із формалізацією можна ознайомитися за посиланням:

<https://drive.google.com/drive/folders/13dI4t5MDP0b3aDM6BaxhB901Dgp1EyNU?usp=sharing>