

Изследване на алгоритми
за конструиране
на обобщеномрежови модели

Вася Красимилова Атанасова

Дисертационен труд, представен за присъждане
на образователна и научна степен „доктор“ по научна специалност
01.01.12 „Информатика“, професионално направление
4.6 “Информатика и компютърни науки”

Научен ръководител: акад. Иван П. Попчев

Съдържание

Увод.....	2
Цели и задачи	2
Глава 1. Накратко за обобщените мрежи	4
1.1. Обобщени мрежи – поява и развитие	4
1.2. Формални дефиниции.....	8
1.3. Алгоритми за функциониране на преход и обобщена мрежа	11
Глава 2. Развитие на теорията на обобщените мрежи	15
2.1. Представимост на преходите	15
2.2. Обобщени мрежи с ядра с обеми.....	25
2.3. Алгоритми за движение на ядра в интуиционистки размити обобщени мрежи ..	32
2.4. Интуиционистки размити обобщени мрежи с интуиционистки размити капацитети на позициите	43
2.5. Интуиционистки размит подход към прогнозирането чрез обобщени мрежи	49
Глава 3. Приложения на обобщените мрежи	59
3.1. Обобщеномрежов модел с оптимизационен компонент универсалната обобщена мрежа за класа алгоритми по метода на мравките	59
3.2. Обобщена мрежа, синхронизираща хибриден метаевристичен алгоритъм	65
3.3. Обобщеномрежови модели на механизми за разрешаване на редакционни конфликти	69
Глава 4. Уики платформа с приложение за обобщени мрежи	85
4.1. Технологията уики.....	85
4.2. Ifigenia.org – среда за електронно обучение по обобщени мрежи	87
Заключение – основни резултати.....	113
Декларация за оригиналност на резултатите.....	116
Библиография	117
Списък на публикациите по дисертационния труд	125
Списък на забелязани цитирания на публикации по дисертационния труд.....	126
Списък на проекти с участието на докторанта	127

Увод

Понятието „обобщена мрежа“ (ОМ) се появява точно преди 30 години. През това време то е изследвано и като математически обект и като средство за моделиране на реални процеси. Както се вижда и от двете библиографии [127] от 2002 г., и [38] от 2007 г., през годините интересът към понятието нараства и се проявява в повече от 700 публикации с автори от над 20 страни. Въпреки това, теоретичните проблеми, които стоят пред изследователите на ОМ, все още са твърде много, а и непрекъснато се появяват нови идеи и проблеми.

Настоящият дисертационен труд е посветен на съществуващи и новопоявили се теоретични проблеми, на някои възможни приложения на обобщените мрежи, както и на средствата за разпространяване на информацията за обобщени мрежи. Направен е преглед на развитието на алгоритмите за движение на ядра в преход и на функционирането на цяла мрежа, като са обосновавани пунктовете, в които могат тези два алгоритъма да се оптимизират по отношение на броя стъпки за изпълнение. За някои видове интуиционистки размити обобщени мрежи за първи път са описани как трябва да се модифицират стандартните алгоритми в съответствие със спецификите на тези разширения.

Предложени са две нови разширения на понятието „обобщена мрежа“, за които също са представени съответно модифицираните спрямо тях алгоритми за функциониране. Доказани са теореми за консервативност на тези разширения

Използваната в труда терминология от областта на обобщени мрежи е заимствана от [2, 41, 48], а терминологията от областта на интуиционистки размитите множества следва възприетата в [50]. За край на доказателство на теорема ще бъде използван знакът \square .

Цели и задачи

Основните цели на дисертационния труд могат накратко да се обобщят по следния начин:

1. Да се дефинират разширения на понятието „обобщена мрежа“ и да се изследват относно консервативността им спрямо стандартната дефиниция. Да се построят алгоритмите за движение на ядрата в тях на ниво отделен преход и цяла мрежа.
2. Да се разработят обобщеномрежови модели на метаевристични алгоритми и да се изследва приложимостта им за оптимизиране на отделни аспекти на работата на самите обобщени мрежи. Да се разработят модели на алгоритмите за разрешаване на конфликти в системи с контрол на версиите.

За постигане на тези цели са формулирани следните задачи:

1. Да се проучат съществуващите модификации на обобщените мрежи, в това число и алгоритмите за движение на ядрата в тях. Да се построят алгоритми за движение на ядрата в съществуващите досега четири вида интуиционистки размити обобщени мрежи.
2. Да се дефинира и изследва разширението интуиционистки размити обобщени мрежи от пети тип, да се докаже неговата консервативност и да се построят алгоритмите за движение на ядрата на ниво отделен преход и цяла мрежа.
3. Да се дефинира и изследва разширението „обобщени мрежи с обеми на ядрата“, да се докаже неговата консервативност и да се построят алгоритмите за движение на ядрата на ниво отделен преход и цяла мрежа.
4. Да се предложи решение на проблем за представимостта на преходите, в обобщените мрежи, формулиран през 1991 г.
5. Да се опише обобщеномрежовия модел, отразяващ паралелната работа на алгоритъма на мравките и на генетичен алгоритъм, оптимизиращи работата на реален процес.

Глава 1. Накратко за обобщените мрежи

В тази глава ще бъдат дадени кратки исторически бележки за появата и развитието на обобщените мрежи, основните дефиниции, свързани с тях, и алгоритмите за движение на ядрата в тях.

1.1. Обобщени мрежи – поява и развитие

Понятието „обобщена мрежа” (ОМ) е дефинирано в доклад на К. Атанасов от 1982 г., предназначен за конференция на Бермудските острови и публикуван през 1984 г., [52]. Междувременно, К. Атанасов публикува първите си статии по ОМ в България [39, 53–55], а след отпечатването на доклада от Бермудските острови в списанието *Advances in Modelling and Simulation*, AMSE Press, се появява цяла серия негови статии, посветени на дефиницията, основните свойства и първите модификации на ОМ. През 1986 г. той защитава дисертация за образователната и научна степен „доктор“ [3], а през 1997 г. – и дисертация за научната степен „доктор на техническите науки“ [4].

До 1990 г. в списание AMSE са публикувани редица статии на К. Атанасов, някои от които, съвместно с Г. Гаргов, Е. Димитров, С. Стоева, Л. Атанасова и други. През 1991 г. се появява първата монография по ОМ, [41]. В тази книга са събрани всички по-значими резултати, свързани с теорията на ОМ към момента. Година по-късно се появява единствената книга на български език по ОМ [2], а през 1993 година – сборникът статии под редакцията на К. Атанасов, в който са събрани първите приложения на ОМ в различни области: изкуствен интелект, химия, транспорт, компютърна техника, медицина и други, [56]. Показано е, че всяка абстрактна система също може да се опише чрез ОМ, [42].

Работата по прилагане на ОМ в тези области продължава и през следващите години с публикуването на 20 монографии, в които са описани резултатите получени във всяка една от горепосочените, както и в други области. Една от монографиите [48] представя новите резултати в теорията на ОМ.

Изследванията в теоретичен план са свързани с изучаване на свойствата на ОМ като математически обекти. Например, показано е, че функционирането и резултатите от работата на всеки вид краен автомат и машина на Тюринг могат да се опишат чрез ОМ. К. Атанасов и С. Данчев доказват, че функционирането и резултатите от работата на алгоритъма на Колмогоров също могат да се опишат чрез ОМ [60]. Направени са стъпки по създаване на алгебричен, топологически, логически, операторен, методологически и програмен аспекти на теорията на ОМ.

Най-общо казано, ОМ са разширение както на мрежите на К.-А. Петри [126, 141], така и на всички техни разширения и модификации. През 1980^{те} г. е показано, че функционирането и резултатът от работата на класическите мрежи на Петри, както и на много от техните модификации и разширения могат да се опишат чрез ОМ и по-точно: доказано е, че може да се построи по една универсална ОМ за всеки от различните видове мрежи. Тези резултати се дискутират в [41], Глава 4, стр. 123–158.

В рамките на алгебричния аспект са дефинирани редица операции и релации както над преходите на една ОМ (по аналогия с резултатите на В. Котов за регулярни мрежи на Петри [113]), така и операции и релации над отделни ОМ. В теорията на мрежите на Петри това е направено за първи път в [41], Глава 5, стр. 174–210, и по-късно в [48], Глава 4, стр. 105–116.

Топологичният аспект на теорията на ОМ съдържа резултати в две направления. От една страна са описани оператори за сложност, дефинирани над ОМ, които са аналогични на операторите на Трахтенброт, дефинирани над машините на Тюринг [36], а от друга – дефинирани са оператори, които на всяка ОМ съпоставят граф. Чрез изучаване на свойствата му, може да се получи информация за някои от свойствата на ОМ ([41], Глава 6, стр. 211–239).

След публикуването на [41], където са събрани резултатите по логическия аспект на теорията на ОМ (в Глава 7, стр. 240–250), повече публикации в тази посока не се откриват.

През последните около 15 години изследванията по операторния аспект са много интензивни. Над ОМ са дефинирани шест групи оператори: глобални, локални, йерархични, редуциращи, разширяващи и динамични, които променят отделни компоненти на ОМ (ядра, приоритети, капацитети и т.н.) и по този начин влияят на функционирането и резултатите от работата на модифицирания ОМ-модел. Като цяло, тези оператори са описани в [41], Глава 8, стр. 251–318, и в [48], Глава 5, стр. 121–152, но през годините след публикуването на втората монография се появяват и нови резултати.

За първи път се споменава за методологичен аспект на теорията в статии [46, 47] и в монография [2], но там идеите за методология на изграждане на ОМ са дадени в много общ и фрагментарен вид. Впоследствие, в дисертационните трудове на В. Радева [27] и В. Тасева [33] също се правят стъпки по посока на развитие на методологичния аспект, но те са с конкретна насоченост към отделни области на приложения. Резултатите на автора в тази част от теорията на ОМ ще бъдат описани в Глава 4.

Програмният аспект има дълга история – още от 1983 г., но все още не може да се каже, че видът му е достатъчно завършен и приложим. Подробен преглед на реализациите на софтуерен симулатор и други програмни средства за работа с ОМ е направен през 2010 г. от Д. Димитров в [14].

Ново направление на прилагане на ОМ, а именно като инструмент за формална верификация на императивни, процедурни и обектно-ориентирани програми, е предложено от М. Тодорова в [35, 146–149].

Важна част от изследванията по ОМ са свързани с изучаване на *редуцираните ОМ*, т.е. ОМ, от които са отстранени част от компонентите, и на *разширенията на ОМ*, т.е. ОМ, към които са добавени нови компоненти.

Още в средата на 1980^{-те} г. става ясно, че всеки от видовете мрежи на Петри може да се опише чрез някой клас¹ редуцирани ОМ. Изяснява се, че някои редуцирани ОМ имат същите моделиращи възможности както стандартните ОМ, т.е. че чрез тях може да се опишат функционирането и резултатите от работата на стандартна ОМ и е конструиран минималният клас редуцирани ОМ с това свойство [48], Глава 2, стр. 33–35.

През 1983 г. К. Атанасов дефинира понятието интуиционистки размито множество (ИРМ), започва да изучава свойствата му и да търси приложенията му. Така възниква идеята за съчетаване на ИРМ и ОМ, която ще бъде прилагана и тук многократно (Глави 2 и 3). През 1985 г. се появява първото разширение на ОМ – *интуиционистки размити ОМ* (ИРОМ), публикувано в [43]. Тази публикация предшества с три години статията на Looney [117], в която се дефинират първите размити мрежи на Петри. В дисертационния труд за придобиване на образователната и научна степен „доктор“ на В. Гочев [12] е показано, че функционирането и резултатите от работата на всеки вид размита мрежа на Петри може да се опише с ОМ, но има примери на процеси, описуеми чрез ОМ и неописуеми чрез размити мрежи на Петри, например процеси, чието описание изисква наличието на времеви параметри или логически условия, представени в явен вид чрез предикати.

През годините са дефинирани други над 20 различни разширения на ОМ: цветни ОМ, ОМ с глобална памет, ОМ с интервално време за активиране, ОМ със сложна структура, ОМ с оптимизационен компонент, двупосочни ОМ, ОМ с ядра, имащи време на живот и други. Голяма част от тях са описани в [48], Глава 3, стр. 53–103 през 2007 г., но и след публикуването на тази монография се появиха нови разширения на ОМ, при дефинирането на две от които участва и авторът. Те са описани тук в Глава 2, т. 2.2 и 2.4.

*
**

През последните 20 години основното внимание на специалистите, интересувани се от ОМ, е насочено към възможностите за тяхното приложение. Например, в четири монографии [61, 138–140] са описани над 500 ОМ-модела на процеси на поставяне на медицински диагнози в областта на нефрологията, неврологията, общата и вътрешна медицина. Тези изследвания са реализирани в активното сътрудничество между К. Атанасов, д-р Й. Шоршич, проф. А. Shannon от Австралия, както и на лекари специалисти в различни области на медицината от България, Австралия и Южна Корея. В монография [59], започната в края на 1990^{-те} г. от К. Атанасов и Й. Шоршич и публикувана през 2008 г., се описват ОМ модели на функциониране на отделните системи в човешкото тяло. През

¹ За отбелязване е, че съгласно теорията на множествата на Нойман-Бернайс-Гьодел, съвкупността на всички предикати образува *клас*, а не *множество* и по тази причина, съвкупността на всички ОМ е клас, а не множество, докато например съвкупността на мрежите на Петри е множество (само) с \aleph_1 на брой елементи. Същото може да се каже за мрежите на Петри с време, цветните мрежи на Петри, стохастичните мрежи на Петри и т.н.

2002 г. в Австралия е защитена докторска дисертация, посветена на описанието с ОМ на рисковите фактори, свързани с бъбречни усложнения при диабет, [84].

Едни от най-интересните приложения на ОМ са свързани с областта на изкуствения интелект. През 1991 г. пред конференция на Института по микро-процесорна техника К. Атанасов излиза с доклад, включен в монография [2], в който предлага езикът на ОМ да се превърне в език за универсално описание на отделните области на изкуствения интелект. През следващите 20 г. голяма част от резултатите на специалистите, работещи в областта на ОМ, са свързани с реализиране на тази идея. Показано е, че функционирането и резултатите от работата на всяка (стандартна) експертна система могат да се опишат чрез ОМ. На тази основа се появяват идеи за нови видове експертни системи, фактите и правилата на които имат специални приоритети, експертни системи, оценяващи степента на вярност на хипотезите чрез интуиционистки размити оценки, експертни системи, отговарящи на въпроси от вида „винаги ли е бил в сила факт А?“, „някога бил ли е в сила факт А?“, „бил ли е понякога в сила факт А?“, „често ли е бил в сила факт А?“ и други.

Механизмите за работа с различни видове бази данни също са обект на описание чрез ОМ. На тези теми са посветени дисертационните трудове за придобиване на образователната и научна степен „доктор“ на П. Георгиев [11], Б. Колев [19], както и дисертационният труд за придобиване на научна степен „доктор на техническите науки“ на Р. Chountas, защитен в България през 2009 г. [82]. На тази тема са и монографиите [44, 83, 112].

Процесите на машинно обучение са описани чрез ОМ в монография [57] и в дисертационния труд за придобиване на образователната и научна степен „доктор“ на Х. Аладжов [1].

По идея на А. Shannon в началото на века започва интензивна работа по описание с ОМ на основните процеси, протичащи в учебно заведение, като акцент пада на процесите, свързани с електронно обучение. Първите резултати се съдържат в дисертацията на Е. Сотирова [29], а впоследствие, с участието на специалисти от Австралия, Белгия, Великобритания, Индия, Италия, Мароко, Полша, Португалия, САЩ, Словакия, Южна Корея и други, е подготвена серия от статии, основните резултати от които са събрани в [64, 65]. През 2011 г. е публикувана и монографията на Д. Орозова [24], участничка в този международен колектив.

Възможностите за моделиране с ОМ на гъвкави автоматизирани производствени системи и на работи се дискутират в дисертациите на М. Стефанова-Павлова [31], Г. Стамболов [30] и З. Минчев [21].

Е. Койчева показва в дисертационния си труд, защитен в Дрезденския университет [18], че всяка UML-диаграма може да се опише чрез ОМ и да се реализира програмно чрез симулатор за ОМ. Койчева, Т. Трифонов, К. Георгиев, Д. Димитров и В. Гочев са сред основните разработчици на програмни продукти реализиращи ОМ-модели през последните години. През 2012 г. излиза монографията на В. Гочев и П. Гочева върху програмната реализация на ОМ с платформата *Microsoft® .NET Framework*, [13].

Процеси на разпознаване и синтез на образи и говор и на вземане на решения с ОМ са описани в монографиите [66, 136]. Част от резултатите, включени в

тези две монографии, се базират на изследванията на Е. Szmidt, включени в дисертационния ѝ труд за придобиване на научна степен „доктор на техническите науки“, защитен в България през 2000 г. [143].

ОМ-моделите на компютърни архитектури, на процеси в мрежова сигурност и на методи за описание на софтуерна архитектура и оценка на комплексността при разработване на софтуерни системи, са обект на дисертациите на П. Стоянов [32], И. Вардева [10] и А. Димов [17], а процесите на управление и синхронизация на паралелни логистични процеси с ОМ са описани в дисертацията на П. Чешмеджиев [37].

В дисертацията на Д. Будакова [9] се прави опит чрез ОМ да се моделира човешко поведение.

Досега за голяма част от основните видове невронни мрежи е показано, че тяхното функциониране и резултатите от работата им може да се опише с ОМ – по една универсална ОМ за всеки от отделните видове невронни мрежи. Тези резултати са включени в дисертационния труд на С. Сотиров [28], в дисертационния труд за придобиване на научна степен „доктор на техническите науки“ на М. Krawczak, защитен в България през 2003 г., [114] и в монографиите [115, 116].

Дисертационните трудове на Д. Пенева [25], В. Тасева [33], Л. Тодорова [34] и докторанта на А. Shannon в Австралия С. Wong [153] са свързани с ОМ-моделиране на процеси от областта на администриране на здравеопазването.

В дисертациите на Л. Димитрова, защитила в Русия, [16], З. С. Мохамед ал Робиех [22], С. Недев [23] и Т. Пенчева [26] се описват ОМ-моделите на процеси, свързани, съответно, с функционирането на автоматизирана система за резервация и продажба на билети в транспорта, на информационна система за издаване на задгранични паспорти, процеси на вземане на решения в химическо производство и за моделиране на клас биотехнологични процеси като обекти с разпределени параметри [137].

Съществуват и други дисертационни трудове, посветени на модели чрез ОМ, защитени преди 1990 г., но точни библиографски данни за тях липсват.

1.2. Формални дефиниции

През годините дефиницията на ОМ е променяна няколко пъти с цел подобрене. Първоначално в [52] тя е имала и повече компоненти. В следващите публикации [51, 53] видът ѝ придобива сегашната си форма, дадена в [41] и [48], която ще приведем по-долу.

Формално, всеки *преход* (*transition*) се задава чрез наредена седморка от вида:

$$Z = \langle L', L'', t_1, t_2, R, S, \square \rangle,$$

където:

- (а) $L' = \{l'_1, \dots, l'_i, \dots, l'_m\}$ е крайно непразно множество от входните позиции на прехода;

- (b) $L'' = \{l''_1, \dots, l''_j, \dots, l''_n\}$ е крайно непразно множество от изходните позиции на прехода;
- (c) t_1 е текущ момент на активиране на прехода;
- (d) t_2 е текуща стойност на продължителността на активното състояние на прехода;
- (e) R е условие на прехода, определящо кои ядра могат да преминат от входните към изходните му позиции. То се задава чрез *индексирана матрица на предикатите* [40, 49] от вида:

$$R = \begin{array}{c|ccccc} & l''_1 & \dots & l''_j & \dots & l''_n \\ \hline l'_1 & r_{1,1} & \dots & r_{1,j} & \dots & r_{1,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_i & r_{i,1} & \dots & r_{i,j} & \dots & r_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_m & r_{m,1} & \dots & r_{m,j} & \dots & r_{m,n} \end{array}$$

за $i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\}$, където $r_{i,j}$ е предикат, съответстващ на i -та входна позиция на прехода и j -та изходна позиция на прехода. Ако предикатът е верен (има вярностна стойност „истина“), е възможно преминаване на ядро от i -та входна към j -та изходна позиция. В противен случай такова преминаване е невъзможно. Предикатите не могат да зависят от бъдещи събития.

- (f) S е *индексирана матрица на капацитетите на дъгите*, имаща вида:

$$S = \begin{array}{c|ccccc} & l''_1 & \dots & l''_j & \dots & l''_n \\ \hline l'_1 & s_{1,1} & \dots & s_{1,j} & \dots & s_{1,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_i & s_{i,1} & \dots & s_{i,j} & \dots & s_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_m & s_{m,1} & \dots & s_{m,j} & \dots & s_{m,n} \end{array}$$

за $i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\}$, където $s_{i,j}$ са естествени числа, задаващи капацитетите на дъгите от всяка i -та входна позиция на прехода към всяка j -та изходна позиция на прехода;

- (g) \square е булев израз, в който като променливи участват идентификаторите на всички входни позиции на прехода, свързани с логическите операции „и“ (\wedge) и „или“ (\vee). Когато вярностната стойност на типа е „истина“, съответният преход може да се активира, в противен случай – не. Ако булевият израз е от вида $\wedge(l'_{i_1}, l'_{i_2}, \dots, l'_{i_u})$, то това означава, че *във всяка* входна позиция $l'_{i_1}, l'_{i_2}, \dots, l'_{i_u}$ трябва да има най-малко по едно ядро. Обратно, ако булевият израз е от вида $\vee(l'_{i_1}, l'_{i_2}, \dots, l'_{i_u})$, то това означава, че *поне в една* от входните позиции $l'_{i_1}, l'_{i_2}, \dots, l'_{i_u}$ трябва да има най-малко едно ядро.

След като е дефинирано понятието преход като основен градивен елемент на обобщената мрежа, може да се дефинира и понятието обобщена мрежа. Наредената четворка

$$E = \langle \langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2 \rangle, \langle K, \pi_K, \theta_K \rangle, \langle T, t^\circ, t^* \rangle, \langle X, \Phi, b \rangle \rangle,$$

се нарича **обобщена мрежа** (ОМ, *generalized net*, GN), ако:

- (a) A е множество от всички преходи в мрежата.
- (b) π_A е функция, задаваща приоритетите на преходите, т.е., $\pi_A : A \rightarrow \mathcal{N}$, където:
$$\mathcal{N} = \{0, 1, 2, \dots\} \cup \{\infty\}.$$
- (c) π_L – функция, задаваща приоритетите на позициите, като L е множеството от всички позиции на обобщената мрежа; $\pi_L : L \rightarrow \mathcal{N}$, където $L = \text{pr}_1 A \cup \text{pr}_2 A$ и с $\text{pr}_i X$ е означена i -тата проекция на n -мерното множество X , $n \in \mathcal{N}$, $n \geq 1$, $1 \leq k \leq n$. По-общо:

$$\text{pr}_{i_1, i_2, \dots, i_s} X = \text{pr}_{i_1} X \times \text{pr}_{i_2} X \times \dots \times \text{pr}_{i_s} X = \prod_{j=1}^s \text{pr}_{i_j} X,$$

където: $1 \leq i_1 < \dots < i_s \leq n$, n е размерността на X и s е естествено число.

- (d) c е функция, задаваща капацитетите на позициите, т.е., $c : L \rightarrow \mathcal{N}$.
- (e) f е функция, определяща вярностната стойност на предикатите (за използваната тук ОМ, функцията f приема стойности „лъжа“ или „истина“, или елементите на множество $\{0, 1\}$).
- (f) θ_1 е функция, задаваща следващия момент от време, в който може да се активира преходът. Стойността на тази функция се преизчислява в момента, в който завършва активното състояние на прехода:

$$\theta_1(t) = t', \text{ където } t = \text{pr}_3 Z, t' \in [t, T + t^*].$$

- (g) θ_2 е функция, която задава продължителността на активното състояние на даден преход Z и стойността ѝ се изчислява в момента, в който се активира преходът:

$$\theta_2(t) = t', \text{ където } t = \text{pr}_4 Z \text{ и } t' \in [0, t^*].$$

- (h) K е множество на ядрата в ОМ.
- (i) π_K е функция, която задава приоритетите на ядрата; $\pi_K : K \rightarrow \mathcal{N}$.
- (j) θ_K е функция, която задава момента от време, в който определено ядро може да влезе в обобщената мрежа, т.е. $\theta_K(\alpha) = t$, където е в сила $\alpha \in K$ и $t \in [T, T + t^*]$.
- (k) T е момент от време, в който обобщената мрежа започва да функционира. Моментът T се определя по фиксирана времева скала.
- (l) t° е елементарната времева стъпка на фиксираната времева скала.
- (m) t^* е продължителност на функционирането на обобщената мрежа.

- (п) X е множеството на всички начални характеристики, които ядрата могат да получат при влизането си в мрежата.
- (о) Φ е характеристична функция. Тя определя новата характеристика на ядрото при придвижването му от входната позиция на даден преход в изходната.
- (р) b е функция, задаваща максималния брой характеристики, които едно ядро може да получи по време на движението си в обобщената мрежа; $b : K \rightarrow \mathcal{N}$. Ако за едно ядро σ е изпълнено $b(\sigma) = 1$, то това ядро ще влезе в мрежата с *начална (нулева) характеристика*. След това то ще „помни“ само *последната си (текуща) характеристика*. Ако $b(\sigma) = k$ ($k < \infty$), то ядрото σ „помни“ последните си k характеристики. Началната (нулева) характеристика винаги се пази. Ако $b(\sigma) = \infty$, то ядрото σ помни всичките си възможни характеристики.

В описанието на дадена ОМ може да не се съдържат всичките ѝ компоненти. В този случай, на местата на липсващите компоненти се пише „*“ и мрежата се нарича „редуцирана“.

Статичната част на дадена обобщена мрежа се определя от елементите на множество $rg_{1,2,6,7}A$, т.е. от входните и изходните позиции на мрежата, от индексиранията матрица на дъгите и от типа на преходите. Динамичният характер на мрежата се определя от ядрата на ОМ и от условията на преходите. Времевият характер се обуславя от времевите компоненти T , t^o , t^* и от елементите на множество $rg_{3,4}A$. Накрая, компонентите Φ , X и b играят ролята на памет на ОМ. Отделните функции също са свързани с тези четири компоненти на ОМ: функции π_A , π_L , c – със статичната структура; f , π_K – с динамичните елементи, а θ_1 , θ_2 и θ_K – с времевите параметри.

1.3. Алгоритми за функциониране на преход и обобщена мрежа

Алгоритмите за движение на ядрата в ОМ, също както и дефиницията, са се променяли във времето неколкократно. В [41] са изложени алгоритмите разработени през 1980^{-те} г., но в началото на века на няколко пъти те са подобрявани и това е отразено в статията [67] и монографията [48]. През последните години, с участие на автора се появиха още няколко модификации, на които ще се спрем в Глава 2. Тук, следвайки и подобрявайки [48], ще бъдат описани общоприетите алгоритми, които са и програмно реализирани, [15, 151]. Тези алгоритми са два: за движение на ядрата в рамките на един преход и за функциониране на ОМ като цяло.

При реализацията на мрежата на Петри, паралелизмът се редуцира до последователно активиране на преходите на мрежата и като цяло редът на активирането на преходите е вероятно или зависи от приоритети на преходите, ако такива приоритети са въведени. Алгоритмите при ОМ дават възможност за по-детайлно моделиране на описвания процес. Алгоритмите за движение на яд-

рата взимат предвид приоритетите на позициите, преходите и ядрата, т.е. са по-прецизни.

Ще отбележим, че тук, в стъпка (A04) на първия от двата алгоритъма, е предложена за първи път модификация, която цели допълнително увеличаване на бързодействието на алгоритъма: тя се състои в проверка дали ядрата могат или не могат да се разцепват и в случай, че не могат, процесът на проверка на предикатите се преустановява при намиране на първия предикат с вярностна стойност „истина“.

1.3.1. Алгоритъм за функциониране на преход

Алгоритъм А

- (A01) Входните позиции на прехода се подреждат по приоритет.
- (A02) Ядра във входните позиции се подреждат в списък по приоритет. Във всяка позиция има по още един аналогичен списък, който на този етап е все още празен.
- (A03) Генерира се индексирана матрица R^* , която съответства на индексираната матрица R . Елементите на R^* имат стойност „0“, ако те съответстват на:
- празна входна позиция, т.е. не съдържаща ядра, които потенциално могат да преминат до изходните позиции на прехода, следователно всички клетки от R^* по съответния ред имат стойност „0“;
 - пълна изходна позиция, т.е. нямаща капацитет да поеме ядра, които потенциално могат да пристигнат от входните позиции на прехода, следователно всички клетки от R^* по съответния стълб имат стойност „0“;
 - входна и изходна позиции с текущ капацитет на дъгата между тях равен на нула, т.е. клетката от R^* , намираща се на съответния ред и стълб има стойност „0“, т.е. от $s_{i,j} = 0$ следва $R^*_{i,j} = 0$.
- (A04) Започваме с позицията с най-висок приоритет, съдържаща поне едно ядро. Вземаме ядрото с най-висок приоритет. **Ако ядрото може да се разцепва**, проверяваме всички предикати от реда на матрица R , които съответстват на разглежданата позиция. Последователно в реда записваме 1 на местата, които съответстват на предикат, имащ вярностна стойност „истина“ и 0 на тези, чийто съответстващ им предикат има вярностна стойност „лъжа“. Ядрото от входната позиция се разцепва на толкова ядра, колкото са единиците с реда и новите ядра се преместват в съответните изходните позиции на прехода, като веднага получават своите нови характеристики. **Ако ядрото във входната позиция не може да се разцепва**, проверяваме вярностната стойност на предикатите в реда от ИМ R , съответстващ на входната позиция докато намерим първия предикат, имащ вярностна стойност „истина“. Ядрото отива в съответната му изходна позиция, където то получава нова характеристика. Ще отбележим,

че изходната позиция, в която влиза ядрото е с най-високия приоритет сред позициите, в които то би могло да влезе.

- (A05) Ако за дадено ядро всички предикати имат вярностна стойност „лъжа“, това ядро остава във входната си позиция, но се премества във втория ѝ списък и то повече не участва в процесите по време на текущото активиране на прехода.
 - (A06) Текущият брой на ядрата за всяка позиция, в която е постъпило ново ядро на предишната стъпка, нараства с 1.
 - (A07) Текущият брой на ядрата преминали през дадена дъга нараства с 1, ако по нея е преминало ядро.
 - (A08) Ако има входна позиция, с приоритет, по-нисък от този на горната, от която все още не сме придвижвали ядра, преминаваме към стъпка (A04). В противен случай отиваме към стъпка (A09).
 - (A09) Текущата стойност t' на моделното време се увеличава с t° .
 - (A10) Проверява се достигнат ли е момент $t_1 + t_2$.
 - (A11) Ако отговорът на стъпка (A10) е „не“, се отива към стъпка (A04), ако е „да“ – край на функционирането на прехода.
-

1.3.2. Общ алгоритъм на функциониране на обобщена мрежа

Алгоритъм В

- (B01) $t_T = T$.
- (B02) В съответните входни позиции на мрежата се въвеждат всички ядра α , за които $\theta_k(\alpha) \leq t_T$, т.е. всички ядра, които трябва да са постъпили в мрежата към (преди) момента на активирането ѝ.
- (B03) Съставя се абстрактен преход на обобщената мрежа. Това е преход с динамична структура и представлява обединение на активните преходи в определен момент от време.
- (B04) Определят се всички преходи, за които моментът за активиране t_1 е равен на текущия момент от време (ще го означаваме с t_T).
- (B05) Изчислява се булевият израз \square за всички преходи, определени на стъпка (B04).
- (B06) Към абстрактния преход се добавят всички преходи, за които булевите изрази, изчислени на стъпка (B05) имат стойност „истина“;
- (B07) За абстрактния преход се прилага общия алгоритъм за функциониране на преход за една времева стъпка t° .
- (B08) Всички преходи, които трябва да се деактивират се отстраняват от абстрактния преход.

- (B09) Текущото време се увеличава с t° , т.е. $t_T = t_T + t^\circ$.
- (B10) Проверява се дали стойността на текущото време е по-малка от $T + t^*$.
- (B11) Ако отговорът на стъпка (B10) е „да“, се отива към стъпка (B02), ако е „не“, то край на функционирането на обобщената мрежа.
-

1.3.3. Изводи

От прегледа на литературата става ясно, че досега не са дискутирани от методологична гледна точка отделните аспекти на дефиницията и алгоритмите за движение на ядра в преход и мрежа. Затова авторът реши да проведе проучване сред студенти по дисциплината „Обобщени мрежи“ в Техническия университет – София, Университета „Проф. д-р Асен Златаров“ – Бургас, и Бургаския свободен университет доколко се усвояват тези знания. В Глава 3 се дискутират резултатите от проучванията, а в Глава 2 се съдържат някои коментари, изясняващи дефинициите и отделните стъпки на двата алгоритъма, които в явен вид никога не са правени в досегашните публикации, а биха били полезни за по-доброто разбиране на този апарат за математическо моделиране.

Глава 2. Развитие на теорията на обобщените мрежи

Глава 2 съдържа пет резултата на автора, свързани с теорията на ОМ.

В точка 2.1. е решен проблем, формулиран през 1991 г., отнасящ се до представимостта на преходите на ОМ. Този резултат е докладван пред международната конференция на IEEE “Intelligent Systems” 2012 г., [74].

В точка 2.2. е предложено разширение на понятието ОМ, наречено „ОМ с обеми на ядрата“. Даден е алгоритъм за движение на ядрата в новия вид ОМ и е формулирана и доказана теорема, че това разширение на ОМ е консервативно. Резултатите са публикувани в [71].

Текстът на точка 2.3. следва и подобрява публикация [62] с участието на автора, където за първи път е описан алгоритъм за движение на ядрата в интуиционистки размити ОМ от I, II, III и IV вид на ниво преход и цяла мрежа.

Нов, пети, вид интуиционистки размити ОМ е описан в точка 2.4. на базата на [58]. Даден е алгоритъм за движение на ядрата в новия вид ОМ и е формулирана и доказана теорема, че това разширение на ОМ е консервативно.

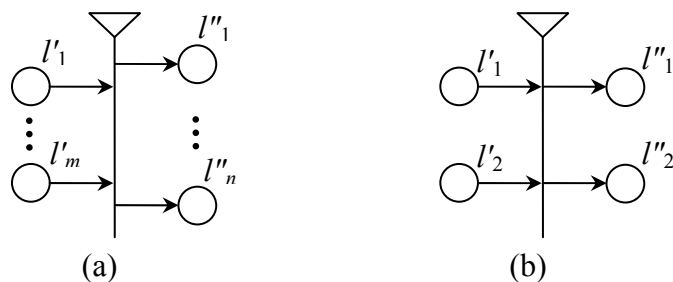
В точка 2.5 е описана публикуваната в [73] процедура за събиране на статистически данни, извлечени от симулация на ОМ модел. Детайлно е описан методът за използване на тази информация за прогностични цели. Използвана е интуиционистка размитост, за да се съставят оценки на преходите на ядрата по време на симулациите, които след това се интерпретират като вероятности за преход на ядрата по време на активиранията на мрежата за прогностична цел.

2.1. Представимост на преходите

Тук се дискутира един проблем за представимостта на преходите, предложен от К. Атанасов през 1991 г. в [41]. През 2005 г. е предложено едно решение на проблема от В. Тасева в [144], но тук предлагаме нов подход, който води до ново – доказано минимално – решение.

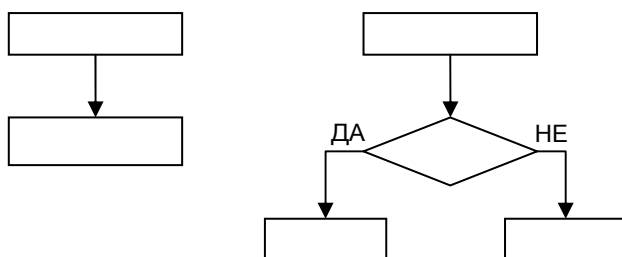
В [144] е въведено означението $H_{m,n}$, за да означаи ОМ преход с m входни и n изходни позиции. Специален случай на $H_{m,n}$ е преходът $H_{2,2}$ (Фиг. 2.1).

В [41] (стр. 264) е формулиран нерешеният проблем да се конструират алгоритми за представяне на произволен преход от вида $H_{m,n}$ посредством преходи от вида $H_{2,2}$.



Фигура 2.1. Преход от вида $H_{m,n}$ (a) и специалният му случай, преход от вида $H_{2,2}$ (b)

В исторически план, за отбелязване е, че идеята за този открит проблем произлиза от статията на С. Вöhm и G. Jakorini [81], която демонстрира, че всяка блок-схема на програма може да бъде представена като композиция само на двата компонента „правоъгълник“ (безусловен преход) и „ромб“, който е еквивалентен на условен преход с възможности ДА/НЕ, свързани със съответните им дъги, както е илюстрирано на Фиг. 2.2.



Фигура 2.2. Единствените два позволени компонента от блоксхеми в подхода на Бьом и Якопини

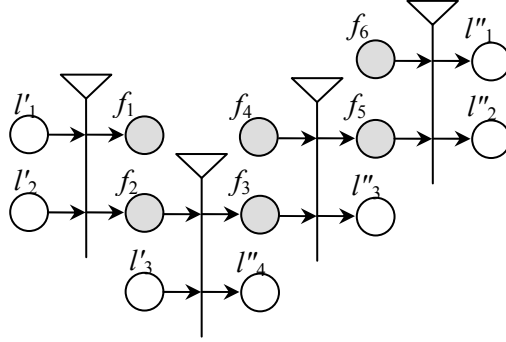
Тази идея, преведена в термините на едно от разширенията на мрежите на Петри, наречено Е-мрежи, изисква да се докаже, че всяка Е-мрежа може да се представи като композиция само от преходи, принадлежащи на един от следните два типа: „един вход–един изход“, „един вход–два изхода“ или „два входа–един изход“. Това доказателство е направено в [89, 109] от R. Grützner и Е. Димитров.

К. Атанасов и Е. Димитров показват в [63], че всяка Е-мрежа може да бъде представена като композиция от преходи, принадлежащи само към типа $H_{2,2}$. В [41] е показано, че този резултат е в сила и в случая на ОМ.

В [41, 63] и на други места, Атанасов разглежда три конфигурации от композициите на преходи от типа $H_{2,2}$. В [144], Тасева разглежда тези и конструира нови подобни конфигурации и им дава имената, които ще се използват и тук, а именно: **X**, **V**, **L**, **I** и **∧**-конфигурации.

2.1.1. Дискусия върху **X**, **V**, **L**, **I** и **∧**-конфигурациите

Да започнем с един пример. Да разгледаме преход от типа $H_{3,4}$, т.е. такъв като показания на Фиг. 2.1. (a) при $m = 3$ и $n = 4$. Представянето му посредством композиция само от $H_{2,2}$ преходи е показано на Фиг. 2.3.



Фигура 2.3. Представяне на един $H_{3,4}$ преход с четири $H_{2,2}$ прехода

На Фиг. 2.3. виждаме че при представянето с $H_{2,2}$ преходи на прехода $H_{3,4}$ се появяват три нови прехода и шест нови позиции, отбелязани в сиво, които до този момент не са съществували. Позициите са наречени „фиктивни“ и отбелязани с f_1, \dots, f_6 . Тези „фиктивни“ позиции са въведени само и единствено с цел да се запазят изискванията на нотацията на $H_{2,2}$ преходите. Можем да отбележим още, че тази конфигурация е от типа \mathbf{V} , по силата на визуалната прилика на новата конструкция с $H_{2,2}$ преходи.

В [144] Тасева въвежда следните две функции: $T(\mathbf{S}, m, n)$ и $P(\mathbf{S}, m, n)$, които изчисляват броя на преходите и „фиктивните“ позиции на дадена конфигурация, представена с $H_{2,2}$ преходи. \mathbf{S} тук отговаря на произволен символ от множеството $\{\mathbf{X}, \mathbf{V}, \mathbf{\Lambda}, l, \lambda\}$. В [144] е доказано, че за $m, n > 2$:

$$\begin{aligned} T(\mathbf{X}, m, n) &= T(\mathbf{V}, m, n) = T(\mathbf{\Lambda}, m, n) \\ &= T(l, m, n) = T(\lambda, m, n) = m + n - 3 \end{aligned} \quad (1)$$

и

$$\begin{aligned} P(\mathbf{X}, m, n) &= P(\mathbf{V}, m, n) = P(\mathbf{\Lambda}, m, n) \\ &= P(l, m, n) = P(\lambda, m, n) = 2m + 2n - 8. \end{aligned}$$

Там е показано, че броят на „фиктивните“ позиции може да бъде редуциран, ако се приложи обединение на позиции. Това обединение може да се приложи само към двойки, съставени от една „фиктивна“ изходна позиция и една „фиктивна“ входна такава. В примера по-горе, такива обединения са възможни само за позиции f_1 и f_4 или между f_1 и f_6 , съответно. Ако се приложи такава оптимизация, броят на преходите остава същият, докато броят на „фиктивните“ позиции се свежда до:

$$\begin{aligned} P(\mathbf{X}, m, n) &= P(\mathbf{V}, m, n) = P(\mathbf{\Lambda}, m, n) \\ &= P(l, m, n) = P(\lambda, m, n) = 2m + n - 6, \end{aligned} \quad (2)$$

където $m \geq n$. Следователно, за всяко $n > 2$ новото представяне съдържа по-малко „фиктивни“ позиции.

Тук правим наблюдението, че m и n участват във функциите T и P симетрично. Следователно, можем да подобрим резултата от [144], като препишем равенство (2) за „фиктивните“ позиции в по-общ вид, а именно:

$$\begin{aligned} P(\mathbf{X}, m, n) &= P(\mathbf{V}, m, n) = P(\mathbf{\Lambda}, m, n) = P(l, m, n) \\ &= P(\lambda, m, n) = 2 \cdot \max(m, n) + \min(m, n) - 6. \end{aligned} \quad (3)$$

2.1.2. Теорема за представимост на $H_{M,N}$ с $H_{2,2}$

Тук ще предложим нова конфигурация за представянето с $H_{2,2}$ преходи. Новостта на представения подход е в наличието на дъги за обратна връзка, които допринасят и за по-стегнатия вид на модела. Ще наричаме това ново представяне **F**-представяне (F от “feedback”, „обратна връзка“).

Да започнем със случая когато $m = n \geq 3$. Следователно, един $H_{m,n}$ преход като показания на Фиг. 2.1. (а) може да бъде **F**-представен като композиция от $H_{2,2}$ преходи, както е показано на Фиг. 2.4.

Веднага се вижда, че за това **F**-представяне броят на преходите и броят на „фиктивните“ позиции са равни съответно на:

$$T(\mathbf{F}, m, m) = m, \quad (4)$$

$$P(\mathbf{F}, m, m) = m. \quad (5)$$

Като цяло, можем да формулираме и докажем следната теорема.

Теорема: За всеки две естествени числа m, n е в сила, че:

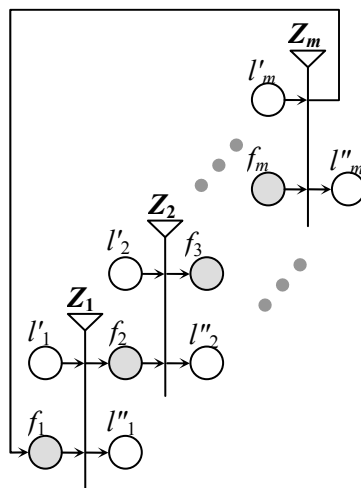
$$T(\mathbf{F}, m, n) = \max(m, n), \quad (6)$$

$$P(\mathbf{F}, m, n) = 2 \cdot \max(m, n) - \min(m, n), \quad (7)$$

и това са минималните брой преходи и брой „фиктивни“ позиции, необходими за **F**-представянето.

Доказателство: Както вече отбелязахме, параметрите m и n участват „симетрично“ във функциите T и P . По тази причина, нека допуснем, че $m \geq n$.

Нека n е фиксирано естествено число и нека $m = n + k$. Случаят $k = 0$ вече беше дискутиран в началото на раздела и илюстриран на Фиг. 2.4. Веднага се вижда, че в този случай равенства (4) и (5) съвпадат с равенства (6) и (7), съответно.



Фигура 2.4. **F**-представяне на един $H_{m,m}$ преход с $H_{2,2}$ преходи

Ще докажем теоремата по индукция.

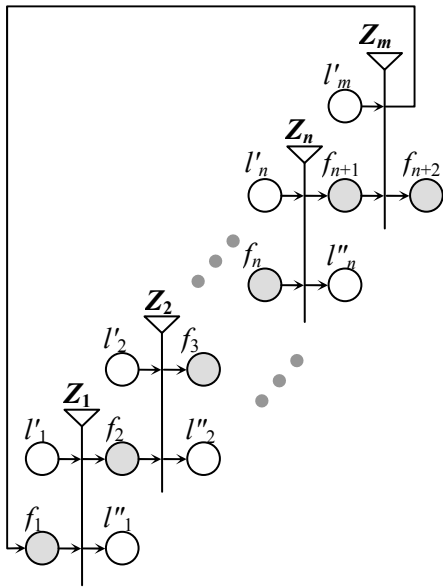
Нека $k = 1$, т.е. $m = n + 1$. Тогава, \mathbf{F} -представянето с $H_{2,2}$ преходи има вида от Фиг. 2.5., показана по-долу. Директно се вижда, че равенства (6) и (7) са верни, тъй като:

$$T(\mathbf{F}, n + 1, n) = \max(n + 1, n) = n + 1,$$

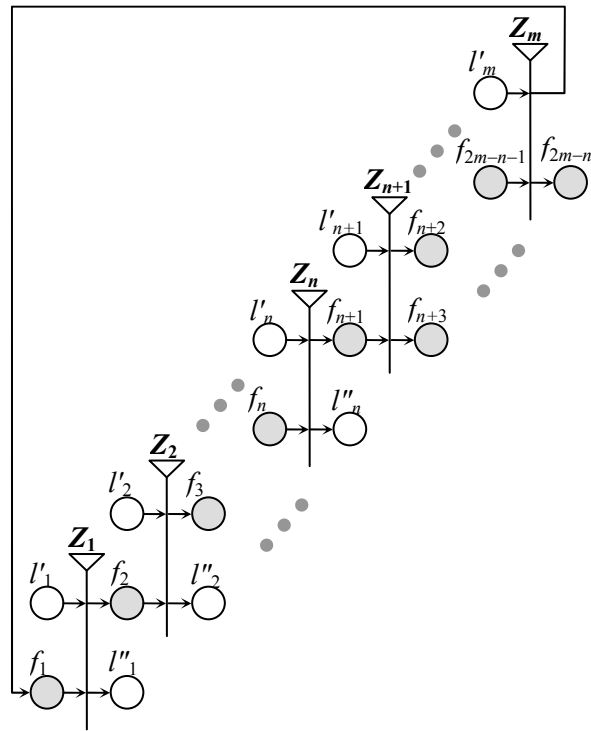
$$\begin{aligned} P(\mathbf{F}, n + 1, n) &= 2 \cdot \max(n + 1, n) - \min(n + 1, n) \\ &= 2 \cdot (n + 1) - n = n + 2. \end{aligned}$$

На Фиг. 2.5. виждаме че всеки от преходите съдържа точно една от входните позиции на оригиналния преход $H_{n+1,n}$, а втората входна позиция на всеки от $H_{2,2}$ преходите се съединява с изходна позиция от предходен $H_{2,2}$ преход. По този начин, за разлика от конфигурацията на Фиг. 2.4., където броят на „фиктивните“ позиции беше равен на броя на преходите, тук се появява още една „фиктивна“ позиция f_{n+2} с единствената цел да запази валидността на нотацията на $H_{2,2}$ преходите. Следователно, не е възможна конфигурация с по-малък брой на $H_{2,2}$ преходи или по-малък брой на „фиктивни“ позиции.

Нека сега допуснем, че формулираното твърдение е в сила за някое $k \geq 1$. Да предположим, че преходът $H_{m,n}$ удовлетворява равенства (6) и (7) и че е изграден от минималния възможен брой $H_{2,2}$ преходи с минималния възможен брой „фиктивни“ позиции. Този случай е илюстриран на Фиг. 2.6.



Фигура 2.5. \mathbf{F} -представяне на един $H_{n+1,n}$ преход с $H_{2,2}$ преходи (база на индукцията)



Фигура 2.6. \mathbf{F} -представяне на един $H_{m,n}$ преход с $H_{2,2}$ преходи (индукционна стъпка)

Ще докажем твърдението за случая $k + 1$, т.е. за прехода $H_{m+1,n}$. Повтаряйки разсъжденията от базата на индукцията, получаваме \mathbf{F} -представянето от Фиг. 2.7. На нея един нов $H_{2,2}$ преход, означен като Z_{m+1} , се добавя към \mathbf{F} -представянето на $H_{m,n}$ прехода, и този нов преход съдържа една входна пози-

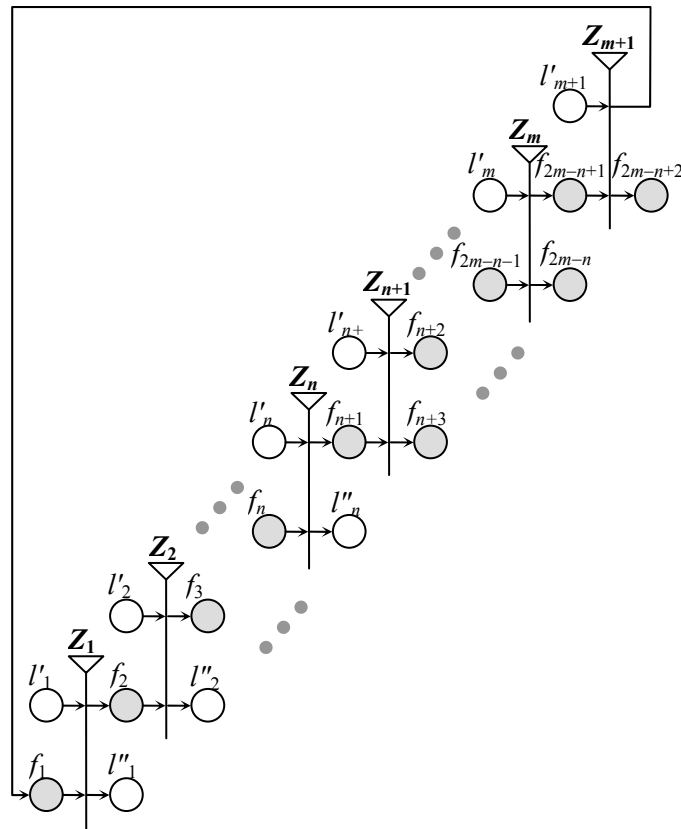
ция отговаряща на $(m + 1)^{\text{вия}}$ вход на прехода $H_{m+1,n}$, а именно l'_{m+1} . Тъй като не можем да пропуснем да добавим този преход и тази входна позиция, следователно новата конфигурация отново притежава минималния възможен брой преходи. За новата конфигурация е в сила, че:

$$\begin{aligned} T(\mathbf{F}, m + 1, n) &= T(\mathbf{F}, m, n) + 1 \\ &= \max(m, n) + 1 = \max(m + 1, n). \end{aligned}$$

По аналогия, втората входна позиция на новия $H_{2,2}$ преход е „фиктивната“ позиция f_{2m-n+1} , която играе роля на свързка с предишния преход Z_m . Двете изходни позиции f_1 и f_{2m-n+2} са „фиктивни“ и служат да допълват необходимия брой изходни позиции за формално коректното $H_{2,2}$ представяне. Следователно, не можем да пропуснем и тях при построяването на коректно \mathbf{F} -представяне на прехода $H_{m+1,n}$. Това означава, че броят на „фиктивните“ позиции е:

$$\begin{aligned} P(\mathbf{F}, m + 1, n) &= P(\mathbf{F}, m, n) + 2 \\ &= 2 \cdot \max(m, n) - \min(m, n) + 2 = 2m - n + 2 = 2(m + 1) - n. \end{aligned}$$

Това на свой ред означава, че броят на „фиктивните“ позиции продължава да е минимално възможният.



Фигура 2.7. \mathbf{F} -представяне на един $H_{m+1,n}$ преход с $H_{2,2}$ преходи (последна стъпка от индукцията)

По този начин се доказва твърдението в Теоремата, но за пълнота ще дискутираме и условията на предикатите на $H_{2,2}$ преходите и начините, по които ядрата се движат при \mathbf{F} -представянето.

Първо ще дадем описание на условията на предикатите в $H_{2,2}$ преходите. За тази цел първо ще споменем, че условието на прехода се задава чрез индексирана матрица, чиито елементи са предикати, определящи възможните пътища за преминаване на ядрата от произволна входна до произволна изходна позиция на прехода. Теорията на индексирани матрици е детайлно представена в [49].

Нека отново разгледаме началния преход от типа $H_{m,n}$ и нека индексиранията му матрица има следния вид:

$$R = \begin{array}{c|cccc} & l_1'' & l_2'' & \dots & l_n'' \\ \hline l_1' & r_{1,1} & r_{1,2} & \dots & r_{1,n} \\ l_2' & r_{2,1} & r_{2,2} & \dots & r_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_m' & r_{m,1} & r_{m,2} & \dots & r_{m,n} \end{array}$$

За да се запази логиката на функционирането на прехода, т.е. възможните пътища на движение на ядрата, трябва да опишем m -те нови индексирани матрици на преходите Z_1, \dots, Z_m формиращи конфигурацията в новопредложеното \mathbf{F} -представяне. Индексиранията матрица R_1 на първия преход Z_1 има вида:

$$R_1 = \begin{array}{c|cc} & l_1'' & f_2 \\ \hline l_1' & r_{1,1} & r_{1,2} \vee r_{1,3} \vee \dots \vee r_{1,n} \\ f_1 & r_{2,1} \vee r_{3,1} \vee \dots \vee r_{m,1} & \bigvee_{i=3}^n (r_{i,2} \vee r_{i,3} \vee \dots \vee r_{i,i-1}) \end{array}$$

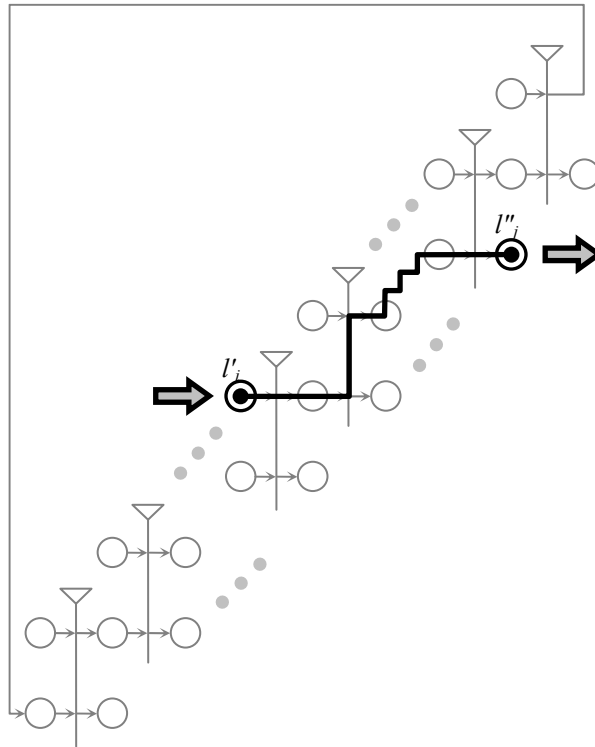
Това доста сложно представяне обаче добре може да се онагледява по следния начин:

$$R = \begin{array}{c|ccccc} & l_1'' & l_2'' & l_3'' & \dots & l_n'' \\ \hline l_1' & r_{1,1} & r_{1,2} & r_{1,3} & \dots & r_{1,n} \\ l_2' & r_{2,1} & r_{2,2} & r_{2,3} & \dots & r_{2,n} \\ l_3' & r_{3,1} & r_{3,2} & r_{3,3} & \dots & r_{3,n} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ l_m' & r_{m,1} & r_{m,2} & r_{m,3} & \dots & r_{m,n} \end{array}$$

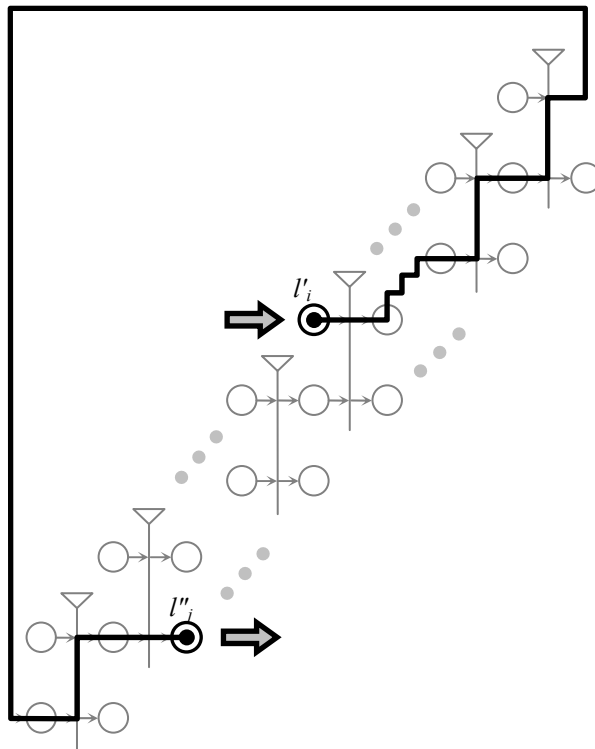
Останалите преходи имат аналогични индексирани матрици със съответните промени в индексите им.

Така построените преходи в \mathbf{F} -представянето осигуряват, че движението на ядрата при него напълно съответства на движението на ядрата в оригиналния $H_{m,n}$ преход. Този факт добре се илюстрира от Фиг. 2.8. и 2.9. Двете демонстрират маршрутите на ядро, което влиза през i -тата входна позиция на $H_{m,n}$, т.е., l_i' , и напуска през неговата j -та изходна позиция l_j'' .

Фиг. 2.8. разглежда случая, когато $i \leq j$. В този случай процесът на движение на ядрото се изпълнява по начин аналогичен на описания в [41] и [144], където ядрата се придвижват само и единствено напред.



Фигура 2.8. Движение на ядро от l'_i към l''_j когато $i \leq j$



Фигура 2.9. Движение на ядро от l'_i към l''_j когато $i > j$

Случаят, при който $i > j$, е илюстриран на Фиг. 2.9. При него наличието на дъги за обратна връзка ефективно се използва, за да се насочи ядро от i -тата входна позиция към крайната изходна позиция на m -тия преход, оттам обратно към първия преход на композицията, и най-сетне към j -тия изход. Това е и прин-

ципната разлика между нашия подход и досега съществуващите подходи. Благодарение на нея, настоящата конструкция съдържа по-малък брой преходи и по-малък брой „фиктивни“ позиции. В допълнение, по индукция се показва, че този брой е минималният възможен.

Трябва да споменем, че ако оригиналният преход $H_{m,n}$ съдържа цикли, т.е. позиции, които едновременно играят роля на вход и изход за прехода, такъв преход също получава своето **F**-представяне по подобен на горния начин.

На последно място, трябва да сравним равенствата (3) и (7). Веднага се вижда, че за $m, n \geq 3$ и за произволно \mathbf{S} от множеството символи, представляващи конфигурациите $\{\mathbf{X}, \mathbf{V}, \mathbf{A}, l, \lambda\}$, е в сила, че:

$$\begin{aligned} & P(\mathbf{S}, m, n) - P(\mathbf{F}, m, n) \\ &= 2 \cdot \max(m, n) + \min(m, n) - 6 - (2 \cdot \max(m, n) - \min(m, n)) \\ &= 2 \cdot \min(m, n) - 6 \geq 0 \end{aligned}$$

Следователно, отново наблюдаваме, че формулата $P(\mathbf{S}, m, n)$ извежда по-голям или равен брой „фиктивни“ позиции в сравнение с броя им получен по формулата $P(\mathbf{F}, m, n)$ при **F**-представянето, т.е. **F**-представянето е минимално при всички положения.

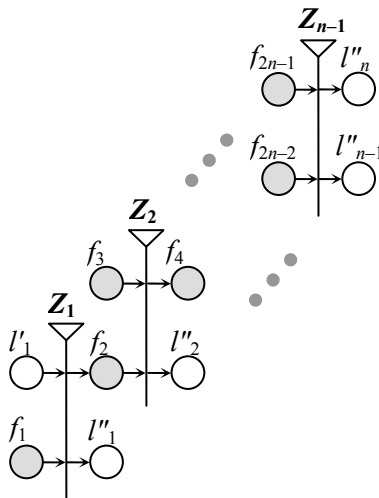
Сега ще дискутираме още четири случая, неотбелязвани досега в [41], [63] или [144]:

- $m = 1, n \geq 3;$
- $m \geq 3; n = 1;$
- $m = 2; n \geq 3;$
- $m \geq 3; n = 2.$

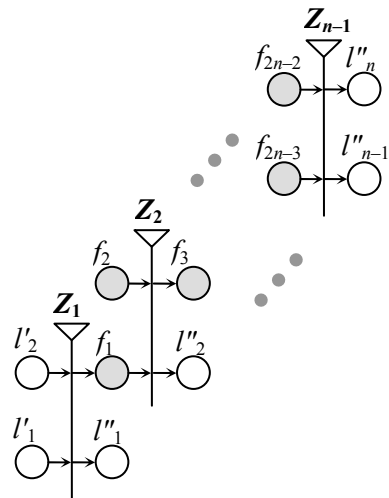
Представимостта в тези случаи е показана на Фиг. 2.10, 2.11, 2.12 и 2.13.

Веднага се вижда, че за тях:

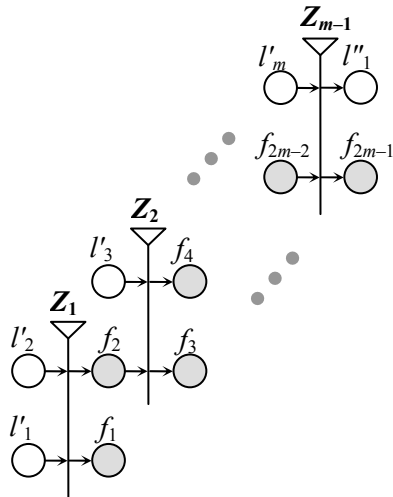
$$\begin{aligned} T(\mathbf{F}, 1, n) &= n - 1, & T(\mathbf{F}, 2, n) &= n - 1, \\ P(\mathbf{F}, 1, n) &= 2n - 1, & P(\mathbf{F}, 2, n) &= 2n - 2, \\ T(\mathbf{F}, m, 1) &= m - 1, & T(\mathbf{F}, m, 2) &= m - 1, \\ P(\mathbf{F}, m, 1) &= 2m - 1, & P(\mathbf{F}, m, 2) &= 2m - 2. \end{aligned}$$



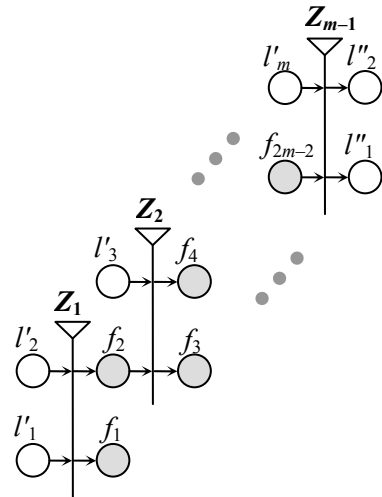
Фигура 2.10. **F**-представяне на един $H_{1,n}$ преход с $H_{2,2}$ преходи



Фигура 2.11. **F**-представяне на един $H_{2,n}$ преход с $H_{2,2}$ преходи

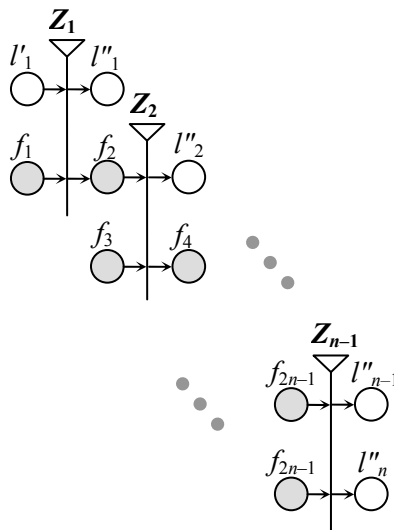


Фигура 2.12. **F**-представяне на един $H_{m,1}$ преход с $H_{2,2}$ преходи



Фигура 2.13. **F**-представяне на един $H_{m,2}$ преход с $H_{2,2}$ преходи

Всяко едно от **F**-представянията, които имат вид I , могат да придобият и вида Λ . При конфигурацията Λ горните равенства остават в сила. Например на Фиг. 2.14. е показана конфигурация Λ , която съответства на Фиг. 2.10.



Фигура 2.14. Λ -представяне на един $H_{1,n}$ преход с $H_{2,2}$ преходи

Накрая за пълнота ще отбележим, че:

$$T(\mathbf{F}, 1, 1) = T(\mathbf{F}, 1, 2) = T(\mathbf{F}, 2, 1) = T(\mathbf{F}, 2, 2) = 1,$$

$$P(\mathbf{F}, 1, 1) = 3,$$

$$P(\mathbf{F}, 1, 2) = P(\mathbf{F}, 2, 1) = 1,$$

$$P(\mathbf{F}, 2, 2) = 0.$$

С това доказателството е завършено. □

*
**

Тук е представено окончателното решение на нерешен проблем, формулиран в [41], свързан с представимостта на произволни ОМ преходи с m входа и n изхода посредством множество от преходи, съдържащи само по два входа и два изхода. В [74] е публикувана основната част от този текст, разширена тук с граничните случаи, когато $m \in \{1, 2\}$ или $n \in \{1, 2\}$.

Основният резултат е формалното доказателство по индукция на откритото минимално решение, с което се подобрява решението, представено в [144]. Като придружаващ резултат, формулите от [144] са леко обобщени, поради наблюдаваната симетрия в двата им параметъра.

За отбелязване е и че един от все още нерешените проблеми в теорията на ОМ дискутира възможността да се покаже как произволна програма, написана на произволен програмен език, може да бъде трансформирана до ОМ. При положение, че този проблем намери своето решение, ще бъде възможно да се строят обобщени мрежи, които извършват верификация на програми. Първите стъпки в това направление са направени от М. Тодорова, [35, 146–149]. Авторът смята, че би било добре ако тези ОМ също бъдат представяни с преходи от вида $H_{2,2}$, което може да направи настоящите теоретични резултати успешно приложими и на практика.

2.2. Обобщени мрежи с ядра с обеми

2.2.1. Въведение и пример

Съществуват реални ситуации, при които процеси, моделирани с апарата на ОМ, изискват ОМ-ядрата, които са динамичните носители на информация за модела, да проявяват някои допълнителни специфични характеристики като „обем“, „размер“ или „тегло“. Когато се конструира един ОМ-модел, ядрата получават разнообразни характеристики в числово-буквен вид, отразяващи специфични аспекти на моделирания процес, и те често представляват записи в база данни. Конкретните стойности на тези характеристики на ядрата обикновено нито повлияват, нито са повлияни от статичните компоненти на мрежата (позиции, преходи или дъги) или съветните им свойства като приоритет или капацитет.

Тук обаче ще разгледаме ситуации, при които ядрата на ОМ проявяват свойства, които имат отношение към свойства на ОМ-инфраструктурата, което води до дефинирането на ново разширение на понятието ОМ, което поема подобни случаи.

В началото да разгледаме за пример прост модел на функционирането на електронна поща. Писмата в електронната кореспонденция съдържат тяло на съобщението (произволен текст, съставен от изпращача) и заглавна част (*header*), структуриран в предефинирани полета като 'From', 'To', 'CC', 'Subject', 'Date', и друга информация за писмото. Дори и размерът на тялото да е нула байта, размерът на заглавната част е ненулев и така размерът на цялото

електронно писмо е ненулев. Електронните писма, в допълнение, често съдържат и прикачени файлове, и те обикновено чувствително увеличават общия размер на писмото. Стандарти за електронна поща като *MIME (Multipurpose Internet Mail Extensions)* не задават никакви ограничения над размера на прикачените файлове, но на практика често неудобство за потребителите се явява невъзможността да изпращат твърде големи файлове. В интернет едно електронно писмо често минава през няколко сървъра и *mail transfer agents* по пътя си от подателя до получателя. Всеки от тези агенти трябва да съхрани писмото преди да го препрати нататък и следователно администраторите на тези сървъри може да са наложили известни ограничения върху размерите на писмата, които потребителите могат да изпращат. Резултатът е, че дори когато големи прикачени файлове могат успешно да преминават вътре в рамките на една компания или организация, това не е сигурно, когато се изпращат през обществената мрежа. От друга страна, и пощенската кутия на получателя може да има ограничен капацитет и при достигането му да не може да приема писма, които надхвърлят определената квота.

Ако преведем така описаните ситуации в термините на ОМ, можем да разглеждаме писмата като динамични *ядра*, които имат някакви характеристики като тяло и полета в заглавната част, както и брой и заглавия на евентуално прикачените файлове. Прикачените файлове обаче имат и размер и тази характеристика (която се явява и характеристика на писмото, т.е. ядрото) не е обикновена характеристика като останалите изброени, а такава, която може да изиграе съществена роля при успешното доставяне на писмото (т.е. движение на ядрото в мрежата), поради ограниченията на капацитетите на дъгите на мрежата (*mail transfer agents*) или на капацитета на изходната позиция (квотата на пощенската кутия на получателя).

2.2.2. Дефиниция на обобщена мрежа с ядра с обеми

Тъй като новопредложеното разширение на ОМ, наречено *обобщена мрежа с ядра с обеми* (ОМЯО, *generalized net with volumetric tokens, GNVТ*) изисква модификация на дефиницията на ОМ преход, ще дадем общата дефиниция, както е дадена в [41, 48], съпоставяйки я със статията на автора, [71], и поставяйки специален акцент върху разликите (отбелязано в сиво).

Формално, всеки преход, както на ОМ, така и на ОМЯО, се задава чрез наредена седморка от вида:

$$Z = \langle L', L'', t_1, t_2, R, S, \square \rangle,$$

където:

- (a) $L' = \{l'_1, \dots, l'_i, \dots, l'_m\}$ е крайно непразно множество от входните позиции на прехода;
- (b) $L'' = \{l''_1, \dots, l''_j, \dots, l''_n\}$ е крайно непразно множество от изходните позиции на прехода;
- (c) t_1 е текущ момент на активиране на прехода;

(d) t_2 е текуща стойност на продължителността на активното състояние на прехода;

(e) R е условие на прехода, задаващо се чрез индексирана матрица от вида:

$$R = \begin{array}{c|ccccc} & l''_1 & \dots & l''_j & \dots & l''_n \\ \hline l'_1 & r_{1,1} & \dots & r_{1,j} & \dots & r_{1,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_i & r_{i,1} & \dots & r_{i,j} & \dots & r_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_m & r_{m,1} & \dots & r_{m,j} & \dots & r_{m,n} \end{array}$$

за $i \in \{1, 2, \dots, m\}$, $j \in \{1, 2, \dots, n\}$, където $r_{i,j}$ е предикат, съответстващ на i -та входна позиция на прехода и j -та изходна позиция на прехода.

(f) S е индексирана матрица на капацитети на дъгите във вида

$$S = \begin{array}{c|ccccc} & l''_1 & \dots & l''_j & \dots & l''_n \\ \hline l'_1 & s_{1,1} & \dots & s_{1,j} & \dots & s_{1,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_i & s_{i,1} & \dots & s_{i,j} & \dots & s_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_m & s_{m,1} & \dots & s_{m,j} & \dots & s_{m,n} \end{array}$$

за $i \in \{1, 2, \dots, m\}$, $j \in \{1, 2, \dots, n\}$, където $s_{i,j}$ ($1 \leq i \leq m$, $1 \leq j \leq n$) е естествено число, по-голямо или равно на нула или знака за безкрайност, ∞ , и означава броя ядра, които могат да преминат през дъгата между i -тата входна и j -тата изходна позиция в рамките на едно активиране на прехода. Този компонент от дефиницията съдържа разликите между стандартната ОМ и новата ОМЯО. В стандартните ОМ, винаги когато едно ядро премине през дадена дъга, чийто капацитет $s_{i,j} \neq \{0; \infty\}$, този капацитет намалява с 1. В случая с ОМЯО капацитетът на дъгата намалява с числовата стойност, изразяваща характеристиката „обем“ на ядрото. Можем, следователно, да смятаме, че стандартните ОМ ядра в ОМЯО са ядра с обем, равен на 1.

(g) \square е булев израз, задаващ типа на прехода.

Продължаваме с формалната дефиниция на ОМЯО. Тя, макар и с малко, се различава от стандартната дефиниция на ОМ. Наредената четворка

$$E_{VT} = \langle \langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2 \rangle, \langle K, K_{VT}, \pi_K, \theta_K, V_{VT} \rangle, \langle T, t^\circ, t^* \rangle, \langle X, \Phi, b \rangle \rangle,$$

се нарича ОМЯО, ако:

- (a) A е множество от всички преходи в мрежата.
- (b) π_A е функция, задаваща приоритетите на преходите.
- (c) π_L е функция, задаваща приоритетите на позициите.
- (d) c е функция, задаваща капацитетите на позициите.

- (e) f е функция, която изчислява върноста на предикатите в условието на прехода.
- (f) θ_1 е функция, задаваща следващия момент от време, когато даден преход Z може да бъде активиран.
- (g) θ_2 е функция, която задава продължителността на активното състояние на даден преход Z .
- (h) K е множеството от стандартните ядра в ОМ.
- (i) K_{VT} в новопредложената модификация е множеството на ядрата на ОМЯО, които проявяват характеристиката “обем”.
- (j) π_K е функция, която задава приоритетите на ядрата.
- (k) θ_K е функция, която задава момента от време, в който определено ядро може да влезе в обобщената мрежа.
- (l) V_{VT} в новопредложената модификация е функцията, която присвоява на ядрата от множеството K_{VT} обеми, които са естествени числа, и стойност 1 на ядрата от множеството K (като това важи и за E , и за E_{VT}).
- (m) T е момент от време, в който обобщената мрежа започва да функционира. Моментът T се определя по фиксирана времева скала.
- (n) t° е елементарната времева стъпка на фиксираната времева скала.
- (o) t^* е продължителност на функционирането на обобщената мрежа.
- (p) X е множеството на всички начални характеристики, които ядрата могат да получат при влизането си в мрежата.
- (q) Φ е характеристична функция, която задава нова характеристика на всяко ядро при придвижването му от входна към изходна позиция на даден преход.
- (r) b е функция, задаваща максималния брой характеристики, които едно ядро може да получи по време на движението си в обобщената мрежа.

2.2.3. Алгоритъм за функционирането на ОМЯО преход

Новопредложената модификация на понятието ОМ като ОМЯО изисква от нас да представим и надлежно модифицирания алгоритъм за функциониране на отделен ОМ преход в даден момент от време.

Отново, по-удобно е да представим и сравним двата алгоритъма за случая на стандартна ОМ [48, 88] и модифицирания случай, предложен от автора в [71] (даден в сиво). Като цяло, разликата се състои във факта, че в случая на стандартна ОМ преминаването на ядрото през прехода води до намаляване на капацитетите на изминатата дъга и на изходната позиция с 1, докато в случая с ОМЯО тези капацитети намаляват с числото, представляващо обема на придвиженото ядро. За тази цел ще използваме дефиницията на функцията $\bar{c}(l, time)$, която дава броя на текущо наличните ядра в позиция l в момент от време $time$.

Алгоритъм А за ОМЯО

- (A01) Входните и изходните позиции на прехода се подреждат по приоритет.
- (A02) Ядрата във входните позиции се подреждат в списък по приоритет. Във всяка позиция има по още един аналогичен списък, който на този етап е все още празен.
- (A03) Генерира се индексирана матрица R^* , която съответства на индексираната матрица R . На елементите на R^* се присвоява стойност „0“, ако те съответстват на:
- празна входна позиция, т.е. не съдържаща ядра, които потенциално могат да преминат до изходните позиции на прехода, следователно всички клетки от R^* по съответния ред имат стойност „0“;
 - пълна изходна позиция, т.е. нямаща капацитет да поеме ядра, които потенциално могат да пристигнат от входните позиции на прехода, следователно всички клетки от R^* по съответния стълб имат стойност „0“;
 - входна и изходна позиции с текущ капацитет на дъгата между тях равен на нула, т.е. клетката от R^* , намираща се на съответния ред и стълб има стойност „0“, т.е. от $s_{i,j} = 0$ следва $R^*_{i,j} = 0$.
- (A04) Започваме с входната позиция с най-висок приоритет, съдържаща най-малко едно ядро, което не е било придвижено към изходна позиция на текущата времева стъпка. Взимаме от нея ядрото с най-висок приоритет, да го означим с α . Последователно проверяваме за всяка от изходните позиции $l'' \in L''$ дали е в сила $V_{VT}(\alpha) > c(l'') - \bar{c}(l'', time)$, т.е. дали обемът на ядрото α е строго по-голям от текущо наличния капацитет на изходната позиция, изразено като разликата между максималния и текущо заетия капацитет на позицията. **Ако да**, тогава изходната позиция не може да приеме толкова обемно ядро (не и по време на текущо активното състояние на прехода), следователно, се присвоява стойност „0“ на съответната клетка от матрицата R^* . **В противен случай**, проверяваме дали текущият капацитет на дъгата между съответната входна и съответната изходна позиция е по-малък от $V_{VT}(\alpha)$. **Ако да**, присвояваме стойност „0“ на съответната клетка от матрицата R^* . **В противен случай**, ако ядрото може да се разцепва, проверяваме всички предикати от реда на матрица R , които съответстват на разглежданата позиция. Последователно в реда записваме 1 на местата, които съответстват на предикат, имащ вярностна стойност „истина“ и 0 на тези, чийто съответстващ им предикат има вярностна стойност „лъжа“. Ядрото от входната позиция се разцепва на толкова ядра, колкото са единиците с реда и новите ядра се преместват в съответните изходни позиции на прехода, като веднага получават своите нови характеристики. Ако ядрото във входната позиция не може да се разцепва, проверяваме вярностната стойност на предикатите в реда от ИМ R , съответстващ на входната позиция докато намерим първия предикат, имащ вярностна стойност „истина“. Ядрото отива в

съответната му изходна позиция, където то получава нова характеристика. Ще отбележим, че изходната позиция, в която влиза ядрото е с най-високия приоритет сред позициите, в които то би могло да влезе.

- (A05) Ако за дадено ядро всички предикати имат вярностна стойност „лъжа“, това ядро остава във входната си позиция, но се премества във втория ѝ списък и то повече не участва в процесите по време на текущото активиране на прехода.
- (A06) Капацитетът на всяка изходна позиция, където са влезли ядра по силата на стъпка (A04), намалява със стойностите на обемите на ядрата. Когато максималният брой ядра, които могат да постъпят в разглежданата изходна позиция, бъде достигнат, тогава на всички клетки в съответния стълб на матрицата R^* се присвоява стойност „0“.
- (A07) Капацитетът на всяка дъга, през която преминава ядро, намалява със стойността на неговия обем. Когато капацитетът на дъгата стане 0, на клетката в матрицата R^* , която съответства на тази дъга, се присвоява стойност „0“.
- (A08) Ако има входна позиция, с приоритет, по-нисък от този на горната, от която все още не са придвижени ядра, преминаваме към стъпка (A04). В противен случай отиваме към стъпка (A09).
- (A09) Текущата стойност t' на моделното време се увеличава с t^o .
- (A10) Проверява се достигнат ли е момент $t_1 + t_2$.
- (A11) Ако отговорът на стъпка (A10) е „не“, се отива към стъпка (A04), ако е „да“ – край на функционирането на прехода.

Алгоритъмът за функциониране на цяла ОМЯО (наричан *Алгоритъм В*) е идентичен с алгоритъма за функциониране на стандартна ОМ, даден в [48, 88].

2.2.4. Теорема за консервативност на ОМЯО

За начало, ще започнем с дефиниция на понятието „консервативно разширение“ на ОМ, което освен тук ще бъде използвано и в т. 2.4.3. Този термин е въведен в теорията на ОМ по аналогия с термина „консервативно разширение на теория“ от математическата логика (вж например [8], стр. 120).

В математическата логика, *теорията* е съвкупност от аксиоми и правила за извод. Тогава, за две теории T_1 и T_2 казваме, че теорията T_1 е *разширение* на теорията T_2 тогава и само тогава, когато всяка аксиома на T_2 е аксиома и на T_1 . Следователно всяка формула, която е изводима в T_2 , е изводима и в T_1 , т.е. двете теории извеждат едни и същи формули.

Казваме още, че теорията T_1 е *консервативно разширение* на T_2 , ако T_1 е разширение на T_2 и всяка формула, която е изводима в T_1 , е изводима и в T_2 .

В областта на ОМ, съгласно [41], обобщена мрежа G се нарича *консервативно разширение на обобщена мрежа H* , ако по някои от своите компоненти (вид на ядрата, сложност на характеристиките, и т.н.) G представлява

разширение на H , но резултатът от работата на G е идентичен с резултата от работата на H .

Ползата от въвеждането на консервативни разширения на ОМ се изразява във възможността с тях да се работи по-лесно и удобно в сравнение с ОМ, дефинирани по стандартната дефиниция. Доказателства за консервативност са направени за всички досега дефинирани разширения на ОМ, в [41].

По-долу ще формулираме и докажем следната:

Теорема: ОМЯО е консервативно разширение на ОМ, т.е. за всяка ОМЯО съществува една ОМ, която представя функционирането и резултатите на ОМЯО.

Доказателство: Нека е дадена ОМЯО G . Конструираме стандартна ОМ H със същите статични компоненти, графична структура, темпорални компоненти и компоненти от паметта като тези на G , но с различно множество ядра. Нека се условим, че на всяко отделно ядро от G с обем 1 съпоставяме отделно ядро от H (където всички ядра по дефиниция са с обем 1), а на всяко отделно ядро α^G от G , чийто обем е $V_{VT}(\alpha^G) = s > 1$, съпоставяме s на брой отделни ядра в H , означени с $\alpha_1^H, \dots, \alpha_s^H$ с обеми 1. Нека приоритетите на $\alpha_1^H, \dots, \alpha_s^H$ да бъдат равни на приоритета на ядрото α^G . Ще коментираме предикатите на преходите, които определят как се придвижват ядрата в мрежата H . Тези предикати трябва да гарантират, че ядрото α^G и ядрото $\alpha_k^H, 1 \leq k \leq s$, ще имат идентично поведение по време на функционирането на двете мрежи.

За тази цел на предиката $r_{i,j}^G$ от ОМЯО G съпоставяме предиката $r_{i,j}^H$ от стандартната ОМ H , който има вида на конюнкция от няколко клаузи:

$r_{i,j}^H = r_{i,j}^G \wedge$ „Ако в позиция l'_i има произволно α^H ядро, тогава всички α^H ядра трябва да се съберат в позиция l'_i до момента на активиране на прехода.“

\wedge „Ако в позиция l'_i има произволно α^H ядро, тогава изходната позиция l''_j трябва да може да поеме поне s ядра (т.е. $c(l'') - \bar{c}(l'', time) \geq s$) и капацитетът на дъгата между l'_i и l''_j трябва да бъде поне s .“

\wedge „Ако в позиция l'_i няма от α^H -ядра, но в позиция l''_j вече има поне едно α^H -ядро, то тогава изходната позиция l''_j трябва в настоящия момент да може да поеме поне s ядра (т.е. $c(l'') - \bar{c}(l'', time) \geq s$).“

Този нов предикат гарантира, че придвижването на едно α^H -ядро ще започне едва след като всички s ядра от пакета бъдат събрани в i -тата входна позиция.

По време на активното състояние на прехода всички или част от α^H -ядрата ще преминат до една или повече изходни позиции, но тяхното движение от новата позиция / новите позиции ще продължи (по същите причини) едва когато всички α^H -ядра от пакета са събрани накуп. Дефинираните горе предикати осигуряват в рамките на текущото активиране на прехода всички α^H -ядра да се

придвижат от съответната входна до съответната изходна позиция / съответните изходни позиции.

Всички s ядра трябва да получат еднакви текущи характеристики в j -тата изходна позиция, и те трябва да съвпадат с характеристиката на съответното ядро от ОМЯО G .

Така описаната процедура показва, че поведението на едно ядро с обем s в ОМ G ще бъде идентично с поведението на съответния пакет от α^H -ядра в ОМЯО H , т.е. се доказва, че новото разширение на понятието ОМ е консервативно разширение. \square

*
**

Новото разширение на понятието ОМ, по аналогия с някои други разширения като интуиционистки размитите ОМ (ИРОМ), предлага широки възможности за по-нататъшни интересни изследвания и приложения. Процесите от горния пример могат да бъдат по-добре моделирани с ОМЯО, отколкото със стандартна ОМ. Новото понятие може да се използва и за проектиране на нови варианти на други добре познати алгоритми, като метаевристичния оптимизационен алгоритъм на мравките. Връзките между ОМЯО и други ОМ разширения също ще са обект на бъдещи изследвания и дискусии.

2.3. Алгоритми за движение на ядра в интуиционистки размити обобщени мрежи

Интуиционистки размитите множества (ИРМ, *intuitionistic fuzzy sets, IFSs*), дефинирани през 1983 г., са разширения на размитите множества ([45, 50]). Те имат две степени: на принадлежност (μ) и на непринадлежност (ν), такива, че сумата им може да е по-малка от 1, т.е. може да бъде дефинирана и трета степен на неопределеност $\pi = 1 - \mu - \nu$. Над ИРМ са дефинирани разнообразие от операции, релации и оператори от модален, топологичен и др. тип.

Първите две разширения на ОМ, предложени през 1985 г., [41, 43, 48], са наречени *интуиционистки размити обобщени мрежи (ИРОМ, intuitionistic fuzzy generalized nets, IFGNs) от първи и втори тип*. През 2001 г. в [68] са описани и още две разширения. Тук се дават подробно алгоритмите за движение на ядра в преход и в цяла мрежа за всеки от четирите типа мрежи. Тези алгоритми са по-ефективни от досегашните, описани в [41].

2.3.1. Основни елементи на интуиционистки размитата логика (ИРЛ)

На всяко съждение (в класическия смисъл) можем да присвоим вярностна стойност: „истина“ – означавана с 1, или „лъжа“ – с 0. В случая с размитата логика вярностната стойност може да е реално число от интервала $[0, 1]$ и може да се

нарече „степен на вярност“ на дадено съждение. В случая на ИРЛ може да бъде добавена още една стойност, „степен на невярност“, която също е реално число в интервала $[0, 1]$. Така две реални числа, $\mu(p)$ и $\nu(p)$, се присвояват на съждението p , като е в сила и следното условие (вж [45, 50]):

$$\mu(p) + \nu(p) \leq 1.$$

Степента на неопределеност се дефинира като

$$\pi(p) = 1 - \mu(p) - \nu(p).$$

Нека над множество от съждения S е дефинирана оценъчна функция V , така че:

$$V(p) = \langle \mu(p), \nu(p) \rangle.$$

2.3.2. Интуиционистки размити обобщени мрежи от първи тип (ИРОМ-1)

Както е посочено в Глава 1, т. 1.1, интуиционистки размитата обобщена мрежа от първи тип (ИРОМ-1) е първото по време разширение на ОМ. То запазва всички компоненти на ОМ, но някои от тях се явяват в модифициран вид. Въпреки това, от 1985 г. досега не са били конструирани алгоритмите за движение на ядра в преход и мрежа, което е обект на настоящата точка от дисертационния труд.

Преход в ИРОМ-1 се нарича наредената седморка:

$$Z = \langle L', L'', t_1, t_2, R, S, \square \rangle,$$

където:

- (a) $L' = \{l'_1, \dots, l'_i, \dots, l'_m\}$ е крайно непразно множество от входните позиции на прехода;
- (b) $L'' = \{l''_1, \dots, l''_j, \dots, l''_n\}$ е крайно непразно множество от изходните позиции на прехода;
- (c) t_1 е текущ момент на активиране на прехода;
- (d) t_2 е текуща стойност на продължителността на активното състояние на прехода;
- (e) R е условие на прехода, задаващо се чрез индексирана матрица от вида:

$$R = \begin{array}{c|ccccc} & l''_1 & \dots & l''_j & \dots & l''_n \\ \hline l'_1 & r_{1,1} & \dots & r_{1,j} & \dots & r_{1,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_i & r_{i,1} & \dots & r_{i,j} & \dots & r_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l'_m & r_{m,1} & \dots & r_{m,j} & \dots & r_{m,n} \end{array}$$

за $i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\}$, където $r_{i,j}$ е предикат, съответстващ на i -та входна позиция на прехода и j -та изходна позиция на прехода. Когато неговата интуиционистки размита оценка удовлетворява поне едно седемте условия, дадени в стъпка (A05) в т. 2.3.3., ядро от i -тата входна позиция преминава до j -тата изходна позиция; в противен случай това не е възможно;

(f) S е индексирана матрица на капацитети на дъгите, имаща вида:

$$S = \begin{array}{c|ccccc} & l_1'' & \dots & l_j'' & \dots & l_n'' \\ \hline l_1' & s_{1,1} & \dots & s_{1,j} & \dots & s_{1,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l_i' & s_{i,1} & \dots & s_{i,j} & \dots & s_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ l_m' & s_{m,1} & \dots & s_{m,j} & \dots & s_{m,n} \end{array}$$

за $i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\}$, където $s_{i,j}$ ($1 \leq i \leq m, 1 \leq j \leq n$) е естествено число, по-голямо или равно на нула или знака за безкрайност, ∞ , и означава броя ядра, които могат да преминат през дъгата между i -тата входна и j -тата изходна позиция в рамките на едно активиране на прехода.

(g) \square е булев израз, задаващ типа на прехода.

Наредената четворка

$$E_{VT} = \langle \langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2 \rangle, \langle K, \pi_K, \theta_K \rangle, \langle T, t^\circ, t^* \rangle, \langle X, \Phi, b \rangle \rangle,$$

се нарича ИРОМ-1, ако:

- (a) A е множество от всички преходи в мрежата.
- (b) π_A е функция, задаваща приоритетите на преходите.
- (c) π_L е функция, задаваща приоритетите на позициите.
- (d) c е функция, задаваща капацитетите на позициите.
- (e) f е функция, която изчислява стойностите на предикатите в условията на прехода. За стандартна ОМ, функцията f има стойности от множеството $\{0, 1\}$, докато за ИРОМ-1 нейните стойности имат вида на наредената двойка $\langle \mu_{i,j}, \nu_{i,j} \rangle$, където $\mu_{i,j}$ и $\nu_{i,j}$ са стойностите на вярност и невярност на предиката, стоящ на (i, j) -то място в индексираната матрица и е в сила $\mu_{i,j} + \nu_{i,j} \leq 1$.
- (f) θ_1 е функция, задаваща следващия момент от време, когато даден преход Z може да бъде активиран.
- (g) θ_2 е функция, която задава продължителността на активното състояние на даден преход Z .
- (h) K е множеството от ядрата в ОМ.
- (i) π_K е функция, която задава приоритетите на ядрата.

- (j) θ_k е функция, която задава момента от време, в който определено ядро може да влезе в обобщената мрежа.
- (k) T е момент от време, в който обобщената мрежа започва да функционира. Моментът T се определя по фиксирана времева скала.
- (l) t° е елементарната времева стъпка на фиксираната времева скала.
- (m) t^* е продължителност на функционирането на обобщената мрежа.
- (n) X е множеството на всички начални характеристики, които ядрата могат да получат при влизането си в мрежата.
- (o) Φ е характеристична функция, която в случая на стандартна ОМ присвоява нови характеристики на всяко ядро, когато то премине от входна към изходна позиция на даден преход, докато в случая с ИРОМ-1 добавя на същите характеристики и степени на вярност и невярност на предикатите, определящи движението на ядрата. Така когато едно ядро приключи движението си в ИРОМ-1, можем да определим степените на вярност и невярност, с които то е извършило този преход през мрежата.
- (p) b е функция, задаваща максималния брой характеристики, които едно ядро може да получи по време на движението си в обобщената мрежа.

Някои от компонентите не са задължителни и е възможно да липсват в дадени случаи. В такива случаи говорим за специален клас от ИРОМ-1, наречени „редуцирани ИРОМ-1“.

2.3.3. Алгоритми за функциониране на преход и цяла ИРОМ-1 мрежа

В [48] е даден по-детайлен алгоритъм за движение на ядрата в стандартна ОМ. Тук, следвайки статия [62], ще въведем алгоритмите за движението на ядрата в ИРОМ-1 на ниво преход и цяла мрежа.

Алгоритъм А за ИРОМ-1

- (A01) Входните и изходните позиции на прехода се подреждат по приоритет.
- (A02) Ядрата във входните позиции се подреждат в списък по приоритет. Във всяка позиция има по още един аналогичен списък, който на този етап е все още празен.
- (A03) Генерира се индексирана матрица R^* , която съответства на индексираната матрица R . Присвоява се стойност „⟨0,1⟩“ (отговаряща на стойността „лъжа“), на всички елементи от индексираната матрица, които се намират:
 - (a) в ред, отговарящ на празна входна позиция; или
 - (b) в стълб, отговарящ на запълнена изходна позиция; или
 - (c) в клетка (i, j) , за която $s_{i,j} = 0$, т.е. текущият капацитет на дъгата между i -тата входна и j -тата изходна позиция е нула.

(A04) Сортираните позиции се обхождат последователно, по реда на приоритета им, започвайки с позицията с най-висок приоритет, която има поне едно ядро и през която не е осъществен преход на текущата времева стъпка. За ядрото с най-висок приоритет (от първия списък) определяме дали то може да се разцепва или не. Този факт се определя от динамичния оператор, дефиниран над конкретната ОМ, и в частност над дадената ИРОМ-1 (вж [48]). Ако не е дефиниран такъв оператор, ще приемем, че ядрото може да се разцепва толкова пъти, колкото е необходимо. След това се проверяват предикатите, отговарящи на съответния ред от индексиранията матрица R . Ако ядрото не може да се разцепва, проверката приключва с откриването на първия предикат с вярностна стойност, различна от „ $\langle 0, 1 \rangle$ “. В противен случай, трябва да изчислим вярностните стойности на всички предикати в реда, за които елементите на R^* са различни от нула.

(A05) В зависимост от изпълнението на оператора за позволяване или забраняване на разцепването на ядрата в мрежата, ядрото от стъпка **(A04)** ще премине или до всички изходни позиции, където му е позволено да премине, или до точно онази от тях, която има най-висок приоритет, следвайки едно от седемте условия по-долу, където за предиката $r_{i,j}$ е в сила, че $f(r_{i,j}) = \langle \mu_{i,j}, \nu_{i,j} \rangle$.

C1 $\mu_{i,j} = 1, \nu_{i,j} = 0$ (случаят на стандартната ОМ);

C2 $\mu_{i,j} > \frac{1}{2} (> \nu_{i,j})$;

C3 $\mu_{i,j} \geq \frac{1}{2} (\geq \nu_{i,j})$;

C4 $\mu_{i,j} > \nu_{i,i}$;

C5 $\mu_{i,j} \geq \nu_{i,i}$;

C6 $\mu_{i,j} > 0$;

C7 $\nu_{i,j} < 1$, т.е. поне $\pi_{i,j} > 0$.

Кое условие ще се използва, зависи от желанието на потребителя. Условието се задава за всеки преход преди активирането на ИРОМ-1. Ако едно ядро не може да премине през даден преход на тази времева стъпка, ядрото се премества във втората група ядра на съответната входна позиция. Ядрата, които се влезли в позицията след активирането на прехода, също се преместват във втората група.

(A06) Стойностите на характеристичната функция на изходните позиции (една или повече), в които са влезли ядра (съгласно стъпка **(A05)**), се изчисляват и следният запис се присвоява като следваща характеристика на ядрото:

„стойност на функцията Φ за съответното ядро, $\mu_{i,j}, \nu_{i,j}$ “

(A07) Присвояваме стойностите „ $\langle 0, 1 \rangle$ “:

- (a) на всички редове на R^* , за които съответната изходна позиция (откъдето ядрото излиза на стъпка **(A05)**) е вече празна; или

- (b) на всички стълбове в R^* , които са запълнени в резултат на придвижването на ядрата на стъпка (A05); или
- (c) в онези клетки от индексиранията матрица, които отговарят на дъги между дискутираните входна и изходни позиции, за които капацитетите на дъгите са станали нула в резултат от придвижването на ядрата.

- (A08) Текущият брой на ядрата във всички входни позиции за текущия преход намалява с 1 за всяко ядро, което ги е напуснало на тази времева стъпка. Ако текущият брой на ядрата за дадената входна позиция е нула, клетките от съответният ред от индексиранията матрица R^* получават стойност „0, 1“.
- (A09) Капацитетите на всички изходни позиции, в които е постъпило ядро по силата на стъпка (A04), намаляват с 1. Ако е достигнат максималният брой ядра за дадена изходна позиция, клетките от съответния стълб на индексиранията матрица R^* стават „0, 1“.
- (A10) Капацитетите на всички дъги, през които е преминало ядро, намаляват с 1. Ако капацитетът на дъга е достигнал нула, на съответната клетка от индексиранията матрица R^* , която отговаря на тази дъга, се присвоява стойност „0, 1“.²
- (A11) Ако продължава да има ядра във входните позиции, които подлежат на трансфериране, и има свободни изходни позиции и има дъги с ненулеви капацитети, то алгоритъмът продължава със стъпка (A12); в противен случай продължава със стъпка (A14).
- (A12) Текущото моделно време t се увеличава с t^o .
- (A13) Текущият момент от време равен или по-голям ли е от $t_1 + t_2$? Ако отговорът е „не“, връщане към стъпка (A04), в противен случай алгоритъмът продължава към стъпка (A14).
- (A14) Край на функционирането на прехода.

2.3.4. Интуционистки развити обобщени мрежи от втори тип (ИРОМ-2)

В *интуционистки развитите обобщени мрежи от втори тип* (ИРОМ-2, [41, 43, 48]), по дефиниция ядрата са заменени от някакви количества вещество, които „протичат“ в мрежата. Това обстоятелство налага някои разлики между определенията на ИРОМ-1 и ИРОМ-2. По-долу ще покажем тези разлики.

² Необходимо е да отбележим, че в статия [62] в стъпки (A08), (A09) и (A10) от алгоритъма е допусната печатна грешка с присвояването на стойност „0“ вместо коректната „0, 1“.

Стойностите на предикатите на условието на прехода могат да бъдат интуиционистки размити, което означава, че могат да имат степени на вярност и невярност както е при ИРОМ-1. ИРОМ-2 има вида:

$$E = \langle \langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2 \rangle, \langle K, \pi_K, \theta_K \rangle, \langle T, t^\circ, t^* \rangle, \langle X, \Phi, b \rangle \rangle$$

където A е множеството от преходите на мрежата (те имат вида на обикновена ОМ само с тази разлика, че тук капацитетите на дъгите на прехода, описани в индексиранията матрица M , са реални числа).

Функциите в първия компонент – $\pi_A, \pi_L, c, f, \theta_1, \theta_2$ – са подобни на тези в ИРОМ-1 и удовлетворяват едни и същи условия, с изключение на функцията c , която ще присвои реално число, отговарящо на „обема“ на позицията, т.е. количеството вещество, което позицията може да поеме.

Съществената разлика между ИРОМ-2 и другите ОМ е множеството K и функциите, с които то е свързано. Тук елементите на K са вид „количества“, които имат като начална характеристика някакъв „тип“ (елементи от множеството X), които не получават други характеристики. Както при стандартните ОМ, функцията θ_K дава момента от време, когато едно ядро ще влезе в мрежата.

Темпоралните компоненти на ИРОМ-2 са същите като тези на другите типове ОМ.

Тук функцията Φ обаче има различен смисъл: тя е свързана с позициите, и чрез тази функция те получават характеристики (количествата ядра от различен тип в съответните позиции). Както при ОМ, тази функция може да се разшири и може също да дава други данни за моделирания процес (например моментите от време на влизане на „количеството вещество“ в позициите). Предвид горните бележки, по-долу ще приведем само точките от дефиницията, които се различават от тези на стандартната ОМ:

- (d) c е функция, даваща капацитетите на позициите, т.е. $c : L \rightarrow \mathcal{R}$, където \mathcal{R} е множеството на реалните (а не естествените) числа.
- (h) K е множеството на ядрата в мрежата, които тук представляват количества вещество.
- (o) Φ е характеристичната функция, която присвоява нови характеристики на всяка позиция, когато ядро извърши преход от изходна позиция към нея, и както е в случая с ИРОМ-1, добавя същата характеристика към степените на вярност и невярност на предиката, който разрешава или не преминаването на ядрото през прехода.
- (p) b е функция, даваща максималния брой характеристики, които дадена позиция (а не ядро) може да получи.

При ИРОМ-2, точно както при ИРОМ-1, е възможно част от гореописаните компоненти да липсват. В тези случаи всички липсващи компоненти ще бъдат изпускани от запис на мрежата. ИРОМ-2 от този вид формират специален клас, наречени „редуцирани ИРОМ-2“.

Всички операции и релации, които се дефинират над ОМ, могат да се дефинират и над ИРОМ-2.

2.3.5. Алгоритми за функциониране на преход и цяла ИРОМ-2 мрежа

Тук ще въведем за първи път формалния алгоритъм за ИРОМ-2. Той, разбира се, е базиран на алгоритмите за трансфер на ядра в стандартни ОМ и ИРОМ-1, но сега има някои съществени разлики и първата от тях е, че преминаването на ядрата се извършва само на една стъпка, но с продължителност, определена от втория темпорален компонент на прехода, ако такъв съществува.

Общият алгоритъм (който ще означаваме с **Алгоритъм А**) за преминаване на ядра през преход в даден момент от време $t_1 = \text{TIME}$ (текущия момент от време на ИРОМ-2) е, както следва

Алгоритъм А за ИРОМ-2

(A01) Непразните входни позиции и незапълнените изходни позиции се сортират по приоритет в намаляващ ред. За разлика от **Алгоритъм А за ИРОМ-1**, всяка позиция може да съдържа само определено количество вещество, което е аналог на единично ядро.

(A02) Генерира се индексирана матрица R^* , която съответства на индексираната матрица R . Присвоява се стойност „ $\langle 0,1 \rangle$ “ (отговаряща на стойността „лъжа“), на всички елементи от индексираната матрица, които се намират:

- (a) в ред, отговарящ на празна входна позиция; или
- (b) в стълб, отговарящ на запълнена изходна позиция; или
- (c) в клетка (i, j) , за която $s_{ij} = 0$, т.е. текущият капацитет на дъгата между i -тата входна и j -тата изходна позиция е нула.

(A03) Сортираните позиции се обхождат последователно по приоритета им, започвайки от позицията с най-висок приоритет, която съдържа някакво количество вещество. Във всяка от обходените позиции, се проверяват предикатите, отговарящи на съответния ред от индексираната матрица R^* . Ако количеството вещество не може да се раздели, проверката приключва с намирането на първия предикат, чиято вярностна стойност е различна от „ $\langle 0, 1 \rangle$ “; в противен случай трябва да изчислим вярностните стойности на всички предикати на реда, за които елементите на R^* са различни от „ $\langle 0, 1 \rangle$ “.

(A04) В зависимост от изпълнението на оператора за позволяване или забраняване на разцепването на ядрата в мрежата, ядрото от стъпка **(A03)** ще премине или до всички позволени за него изходни позиции, или до точно онази от тях, която има най-висок приоритет, следвайки едно от следните седем условия, където за предиката $r_{i,j}$ е в сила, че $f(r_{i,j}) = \langle \mu_{i,j}, \nu_{i,j} \rangle$.

C1 $\mu_{i,j} = 1, \nu_{i,j} = 0$ (случаят на стандартната ОМ);

C2 $\mu_{i,j} > \frac{1}{2} (> \nu_{i,j})$;

C3 $\mu_{i,j} \geq \frac{1}{2} (\geq \nu_{i,j})$;

- C4** $\mu_{i,j} > \nu_{i,j}$;
- C5** $\mu_{i,j} \geq \nu_{i,j}$;
- C6** $\mu_{i,j} > 0$;
- C7** $\nu_{i,j} < 1$, т.е. поне $\pi_{i,j} > 0$.

Условието, което ще бъде използвано, се определя за всеки преход преди началото на активирането на ИРОМ-2.

За разлика от предния алгоритъм, тук е в сила следното специфично изискване:

$$\sum_j \mu_{i,j} \leq 1.$$

Причината да се формулира това условие е, че количеството вещество в i -тата входна позиция ще бъде доставено до j -тата изходна позиция в зависимост от вярностната стойност $\mu_{i,j}$. В i -тата входна позиция ще остане толкова количество вещество, колкото отговаря на степента на невярност $\nu_{i,j}$. По дъгата, свързваща i -тата входна с j -тата изходна позиция, ще се задържи останалото количество вещество, отговарящо на допълнението

$$\pi_{i,j} = 1 - \mu_{i,j} - \nu_{i,j}.$$

- (A05)** Стойностите на характеристичната функция на изходните позиции (една или повече), в които са влезли ядра (съгласно стъпка **(A04)**), се изчисляват и следният запис се присвоява като следваща характеристика на ядрото:

„стойност на функцията Φ за съответното ядро, $\mu_{i,j}, \nu_{i,j}$ “

- (A06)** Присвояваме стойностите „ $\langle 0, 1 \rangle$ “:

- (a) на всички редове на R^* , за които съответната изходна позиция (откъдето ядрото излиза на стъпка **(A05)**) е вече празна; или
- (b) на всички стълбове в R^* , които са запълнени в резултат на придвижването на ядрата на стъпка **(A05)**; или
- (c) в онези клетки от индексиранията матрица, които отговарят на дъги между дискутираните входна и изходни позиции, за които капацитетите на дъгите са станали нула в резултат от придвижването на ядрата.

- (A07)** Текущото количество вещество във всяка входна позиция за текущия преход намалява с количествата вещество, отговарящи на сумата $\sum_j \mu_{i,j}$. Ако капацитетът на позицията стане нула, то в клетките на съответния ред от индексиранията матрица R^* се записва стойност „ $\langle 0, 1 \rangle$ “.

- (A08)** Текущото количество вещество във всяка изходна позиция за текущия преход се увеличава с количествата вещество, отговарящи на сумата $\sum_i \mu_{i,j}$. Ако максималният капацитет на позицията бъде достигнат, то в клетките на съответния стълб от индексиранията матрица R^* се записва стойност „ $\langle 0, 1 \rangle$ “.

- (A09)** Капацитетите на всички дъги, свързващи $i^{-\text{тата}}$ входна с $j^{-\text{тата}}$ изходна позиция, през които е преминало вещество, намаляват пропорционално на сумата $\mu_{i,j} + \pi_{i,j}$. Ако максималният капацитет на дъгата бъде достигнат, в клетката от индексиранията матрица R^* , която отговаря на тази дъга, се записва стойността „ $\langle 0, 1 \rangle$ “.
- (A10)** Ако продължава да има ядра във входните позиции, които подлежат на трансфериране, и има свободни изходни позиции и има дъги с ненулеви капацитети, то алгоритъмът продължава със стъпка **(A11)**, в противен случай продължава със стъпка **(A13)**.
- (A11)** Текущото моделно време t се увеличава с t° .
- (A12)** Текущият момент от време равен или по-голям ли е от $t_1 + t_2$? Ако отговорът е „не“, връщане към стъпка **(A03)**, в противен случай алгоритъмът продължава към стъпка **(A13)**.
- (A13)** Край на функционирането на прехода.

Общият алгоритъм за функциониране на ИРОМ-2 съвпада с алгоритъма за функциониране на ИРОМ-1.

2.3.6. ИРОМ-3 и ИРОМ-4

Тук, следвайки [48, 68], ще въведем ИРОМ от трети (ИРОМ-3) и от четвърти вид (ИРОМ-4). Те са разширения съответно на ИРОМ-1 и ИРОМ-2. Тук, характеристиките на ядрата или, съответно, позициите ще бъдат оценявани в смисъла на интуиционистки размитите множества, т.е. те ще получават за стойности ИР двойки, които представляват степените на вярност и невярност (валидност и невалидност) на характеристиките. Следователно, двата нови типа ОМ позволяват описването на ситуации, при които моделът определя състоянието си със степени на вярност и невярност (валидност и невалидност).

Всяка ИРОМ-3 и всеки от нейните преходи има вида, подобен на вида на ИРОМ-1 от т. 2.3.2.

Например, в ИРОМ-1 стойностите на функцията Φ също се оценяват, т.е. сега има две реални числа $\mu(x_{cu}^p)$ и $\nu(x_{cu}^p)$, такива, че $\mu(x_{cu}^\alpha), \nu(x_{cu}^\alpha) \in [0, 1]$ и $\mu(x_{cu}^\alpha) + \nu(x_{cu}^\alpha) \leq 1$. Едва когато тези оценки удовлетворяват следните условия:

$$\text{C1} \quad \mu(x_{cu}^p) = 1, \nu(x_{cu}^p) = 0 \text{ (случаят на стандартна ОМ);}$$

$$\text{C2} \quad \mu(x_{cu}^p) > \frac{1}{2} \text{ (} > \nu(x_{cu}^p) \text{);}$$

$$\text{C3} \quad \mu(x_{cu}^p) \geq \frac{1}{2} \text{ (} \geq \nu(x_{cu}^p) \text{);}$$

$$\text{C4} \quad \mu(x_{cu}^p) > \nu(x_{cu}^p) \text{;}$$

$$\text{C5} \quad \mu(x_{cu}^p) \geq \nu(x_{cu}^p) \text{;}$$

$$\text{C6} \quad \mu(x_{cu}^p) > 0 \text{;}$$

$$\text{C7} \quad \nu(x_{cu}^\alpha) < 1, \text{ т.е. поне } \pi(x_{cu}^\alpha) > 0.$$

тогава ядрата ще получават характеристики и тези характеристики тук имат вида

$$x_{cu}^{\alpha} = \left\langle \bar{x}_{cu}^{\alpha}, \mu(r_{i,j}), \nu(r_{i,j}), \mu(x_{cu}^{\alpha}), \nu(x_{cu}^{\alpha}) \right\rangle,$$

където \bar{x}_{cu}^{α} е стандартната характеристика на ядрото.

Ще продължим с дефиницията на понятието ИРОМ-4. В тази мрежа ядрата представляват някакви количества вещество, което се движи из мрежата.

Стойностите на предикатите в условието на прехода могат да бъдат ИР двойки, т.е. тези предикати могат да се характеризират със степени на вярност и невярност, както в случаите на ИРОМ-1 и ИРОМ-3. ИРОМ-4 има вида

$$E = \langle \langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2 \rangle, \langle K, \pi_K, \theta_K \rangle, \langle T, t^{\circ}, t^* \rangle, \langle X, \Phi, b \rangle \rangle$$

където A е множеството от преходите в мрежата, които имат вида на ОМ-преходи с една разлика: тук индексиранията матрица S съдържа за елементи реални числа – капацитетите на дъгите на прехода (както в случая с ИРОМ-2).

Функциите в първия компонент – $\pi_A, \pi_L, c, f, \theta_1, \theta_2$ – са сходни с тези в ИРОМ-2 и удовлетворяват същите условия.

Съществената разлика между ИРОМ-2 и ИРОМ-4 от една страна и останалите видове ОМ, от друга, е множеството K на ОМ-ядра и функциите, свързани с него. Тук елементите на K са някакви “количества”, които имат само начални характеристики и не получават текущи по време на функционирането на мрежата. Функцията θ_K определя моментът от време, когато дадено ядро ще постъпи в мрежата, както при стандартните ОМ. Темпоралните компоненти също са както при другите ОМ.

За двата нови вида ОМ, ИРОМ-2 и ИРОМ-4, функцията Φ има нов смисъл, подобен на този в ИРОМ-3: тя присвоява на всяка позиция следната характеристика: количеството ядра в съответната позиция в различните моменти от време и степените на валидност и невалидност на тази характеристика. Както при ОМ, така и тук моделът може да бъде разширен: може да даде и други данни за моделирания процес (например моментите от време на постъпване на „количествата вещество“ в позициите).

Подобно на ИРОМ-2 и ИРОМ-3, функцията Φ в ИРОМ-4 също се оценява. При все това, тук за всяка позиция p , тази функция се оценява с две реални числа $\mu(x_{cu}^p)$ и $\nu(x_{cu}^p)$, такива че $\mu(x_{cu}^p), \nu(x_{cu}^p) \in [0, 1]$ и $\mu(x_{cu}^p) + \nu(x_{cu}^p) \leq 1$. Така горните условия имат вида:

- C1** $\mu(x_{cu}^p) = 1, \nu(x_{cu}^p) = 0$ (случаят на стандартната ОМ);
- C2** $\mu(x_{cu}^p) > \frac{1}{2} \quad (> \nu(x_{cu}^p))$;
- C3** $\mu(x_{cu}^p) \geq \frac{1}{2} \quad (\geq \nu(x_{cu}^p))$;
- C4** $\mu(x_{cu}^p) > \nu(x_{cu}^p)$;
- C5** $\mu(x_{cu}^p) \geq \nu(x_{cu}^p)$;
- C6** $\mu(x_{cu}^p) > 0$;
- C7** $\nu(x_{cu}^p) < 1$, т.е. поне $\pi(x_{cu}^p) > 0$.

и позициите ще получат характеристики, които имат вида

$$x_{cu}^p = \left\langle \bar{x}_{cu}^p, \mu(r_{i,j}), \nu(r_{i,j}), \mu(x_{cu}^p), \nu(x_{cu}^p) \right\rangle,$$

където \bar{x}_{cu}^p е характеристиката на позицията от ИРОМ-2.

2.4. Интуиционистки размити обобщени мрежи с интуиционистки размити капацитети на позициите

По-долу ще конструираме нов тип ОМ. За целите на нашето изследване ще използваме апарата на интуиционистки размитите множества. Както посочихме и в т. 2.3. досега са конструирани четири типа ИРОМ. Тук, следвайки [58], ще бъде предложен нов, пети, тип ИРОМ. За краткост ще го означим с ИРОМ-5. Тези нови ИРОМ ще съдържат позиции, имащи ИР капацитети.

2.4.1. Дефиниция на ИРОМ-5

Като основа на настоящата дефиниция отново ще използваме определението на стандартната ОМ от Глава 1. На съответното място ще откромим (със сив фон) разликите между стандартната и новата дефиниция. Промяната в стандартната дефиниция е малка, но тя повлиява вида на някои други ОМ параметри, а също и на Алгоритъма за движение на ядра в даден преход, и ще ги обсъдим по-долу.

Наредената четворка

$$E = \langle \langle A, \pi_A, \pi_L, c, f, \theta_1, \theta_2 \rangle, \langle K, \pi_K, \theta_K \rangle, \langle T, t^\circ, t^* \rangle, \langle X, \Phi, b \rangle \rangle$$

се нарича ИРОМ-5, ако:

- (a) A е множество от всички преходи в мрежата;
- (b) π_A е функция, задаваща приоритетите на преходите, т.е., $\pi_A: A \rightarrow \mathcal{N}$, където $\mathcal{N} = \{0, 1, 2, \dots\} \cup \{\infty\}$;
- (c) π_L е функция, задаваща приоритетите на позициите, като L е множеството от всички позиции на обобщената мрежа; $\pi_L: L \rightarrow \mathcal{N}$, където $L = \text{pr}_1 A \cup \text{pr}_2 A$ и с $\text{pr}_i X$ е означена i -тата проекция на n -мерното множество X , $n \in \mathcal{N}$, $n \geq 1$, $1 \leq k \leq n$;
- (d) В дефиницията на стандартна ОМ, c е функция, задаваща капацитетите на позициите, т.е. $c: L \rightarrow \mathcal{N}$. Тук, за всяка позиция $p \in L$ се определя интуиционистки размитата тройка $\langle c(p), \mu_c(p), \nu_c(p) \rangle$. Смисълът ѝ е следният. Нека $\bar{c}(p)$ е текущият брой ядра, които стоят в позиция p , т.е. $c(p) \geq \bar{c}(p)$ и нека $[x]$ е цялата част от реалното положително число x . Сигурно е, че $[\mu_c(p) \cdot \bar{c}(p)]$ на брой ядра с най-високи приоритети ще имат възможността да напуснат позиция p , $[\nu_c(p) \cdot \bar{c}(p)]$ е броят ядра, които няма да имат възможност да напуснат позиция p при текущото активиране на прехода, а за останалите

$$\bar{c}(p) - [\mu_c(p) \cdot \bar{c}(p)] - [\nu_c(p) \cdot \bar{c}(p)] \geq 0$$

на брой ядра не е ясно дали ще напуснат или не позиция p . Тази промяна в дефиницията на стандартна ОМ ще се отрази върху дефиницията на θ_2 от

т. (g) по-долу, и следователно върху стойностите на t_2 в т. (c) от дефиницията на стандартен ОМ-преход в Глава 1.

- (e) f е функция, която изчислява вярностната стойност на предикатите в условието на прехода (за описаната тук ОМ нека функцията f има стойност „лъжа“ или „истина“, т.е. стойност от множеството $\{0; 1\}$);
- (f) θ_1 е функция, задаваща следващия момент от време, когато даден переход Z може да бъде активиран, т.е. $\theta_1(Z, t) = t'$, където $\text{pr}_3 Z = t$, $t' \in [T, T + t^*]$ и $t \leq t'$. Стойността на тази функция се изчислява в момента, когато преходът спре да функционира.

- (g) В дефиницията на стандартна ОМ, θ_2 е функция, задаваща продължителността на активното състояние на даден преход Z , т.е. $\theta_2(Z, t) = t'$, където $\text{pr}_4 Z = t \in [T, T + t^*]$ и $t' \geq 0$. Стойността на тази функция се изчислява, когато преходът започне да функционира. Тук за прехода Z , съдържащ позиция $p \in L$ от точка (d), $\theta_2(Z)$ ще бъде числото, за което трябва да важат следните неравенства:

$$\theta_2(Z, t) \leq \max_{p \in \text{pr}_1 Z} [\bar{c}(p) \cdot \mu_c(p)], \quad (8)$$

$$\theta_2(Z, t) \leq \max_{p \in \text{pr}_1 Z} (\bar{c}(p) - [\bar{c}(p) \cdot \nu_c(p)]). \quad (9)$$

Комбинацията на двете неравенства е коректна, защото за всяка позиция p
 $\bar{c}(p) - [\bar{c}(p) \cdot \mu_c(p)] - [\bar{c}(p) \cdot \nu_c(p)] \geq \bar{c}(p) - \bar{c}(p) \cdot \mu_c(p) - \bar{c}(p) \cdot \nu_c(p) \geq 0$.

- (h) K е множеството от ядрата на ОМ;
- (i) π_K е функция, която задава приоритетите на ядрата т.е. $\pi_K: K \rightarrow \mathcal{N}$;
- (j) θ_K е функция, която задава момента от време, в който определено ядро може да влезе в обобщената мрежа, т.е. $\theta_K(\alpha) = t$, където $\alpha \in K$, $t \in [T, T + t^*]$;
- (k) T е момент от време, в който обобщената мрежа започва да функционира. Моментът T се определя по фиксирана времева скала;
- (l) t° е елементарната времева стъпка на фиксираната времева скала;
- (m) t^* е продължителност на функционирането на обобщената мрежа;
- (n) X е множеството на всички начални характеристики, които ядрата могат да получат при влизането си в мрежата;
- (o) Φ е характеристична функция, която задава нова характеристика на всяко ядро при придвижването му от входна към изходна позиция на даден преход;
- (p) b е функция, задаваща максималния брой характеристики, които едно ядро може да получи по време на движението си в обобщената мрежа, $b: K \rightarrow \mathcal{N}$.

Когато ИРОМ-5 съдържа само част от гореописаните компоненти, тя се нарича редуцирана ИРОМ-5. Всички операции, релации и оператори над стандартни ОМ, дефинирани в [41, 48], могат да бъдат трансформирани за случая на ИРОМ-5.

2.4.2. Алгоритми за движение на ядра в преход и за функциониране на ИРОМ-5

Дефиницията на ИРОМ-5 е по-сложна от дефиницията на мрежите на Петри или някоя от техните модификации, както е по-сложна и от дефиницията на стандартна ОМ. В реализацията на мрежа на Петри, паралелизъмът е сведен до последователното активиране на преходи в мрежата и като цяло редът на активирането им е основан на вероятности или зависи от приоритетите на преходите, ако такива съществуват. Алгоритмите на ОМ дават възможност за по-детайлно моделиране на описаните процеси. Алгоритмите за преминаване на ядра през преход отчитат приоритети на позиции и ядра, т.е. са много по-прецизни. В [48, 62, 88] е даден по-детайлен алгоритъм за движението на ядра в обикновена ОМ, отколкото този в [41].

Следвайки [48, 62, 88], ще въведем алгоритми за движение на ядра в преход и цяла мрежа от типа ИРОМ-5.

По аналогия с [41, 48], някои компоненти от дефиницията на ИРОМ-5 не са дадени по-горе, тъй като са свързани с алгоритъма, описан по-долу. Те са специално споменати в текста. Алгоритъмът (който ще означаваме като **Алгоритъм А**) за преминаване на ядра в рамките на един преход след моментът от време $t_1 = TIME$ (текущият момент от време в ИРОМ-5) е същият като за стандартна ОМ. Той е, както следва:

Алгоритъм А за ИРОМ-5

- (A01) Сортират се входните и изходните позиции на прехода според техните приоритети.
- (A02) Сортират се по приоритет ядрата от група P_1 на входните позиции (следвайки последователността от стъпка (A01)). *[Нека както и досега индексиранията матрица R^* отговаря на индексиранията матрица R .]*
- (A03) Стойността 0 се присвоява на всички елементи от R^* , за които е в сила поне едно от трите условия:
 - (a) входната позиция, която отговаря на съответният предикат е празна (списъкът P_1 е празен); **или**
 - (b) изходната позиция, която отговаря на съответният предикат е пълна; **или**
 - (c) текущият капацитет на дъгата между съответни входна и изходна позиции е равен на 0.
- (A04) Сортираните позиции са обходени последователно по низходящ ред на приоритета, започвайки с позицията с най-висок приоритет, която съдържа поне едно ядро и през която не е извършен преход на текущата времева стъпка. За ядрото с най-висок приоритет (от първия списък на позицията) предикатите, отговарящи на съответния ред в R^* се проверяват. Елементите от R , за които елементите на R^* не са нули, се изчисляват.
- (A05) В зависимост от изпълнението на оператора, позволяващ или забраняващ разцепването на ядра в мрежата, ядрото от стъпка (A04) ще премине или до всички разрешени за него изходни позиции, или до онази измежду тях, която има най-висок приоритет. Ако едно ядро не може да премине

през съответната му времева стъпка в рамките на активното състояние на прехода, то то се премества във втория списък от ядра на съответната (входна) позиция. Ядрата, които са навлезли в позицията след активирането на прехода, също биват преместени във втория списък.

- (A06) Капацитетите на всички изходни позиции на дадения преход, които се явяват входни за други активни към момента преходи, се увеличават с 1 за всяко ядро, което ги е напуснало на тази времева стъпка.
- (A07) Капацитетите на всички изходни позиции, в които е постъпило ядро по силата на стъпка (A04), намаляват с 1. Ако бъде достигнат максималният допустим брой ядра за дадената изходна позиция, то на елементите от съответната ѝ колона в матрицата R^* се присвоява стойност 0.
- (A08) Капацитетите на всички дъги, през които е преминало ядро, намаляват с 1. Ако капацитетът на дъга достигне 0, то на съответната ѝ клетка в матрицата R се присвоява стойност 0.
- (A09) Изчисляват се стойностите на характеристичната функция Φ за изходните позиции, в които ядрата са постъпили съгласно стъпка (A05).
- (A10) Ако има още позиции, които могат да бъдат изходни за ядра на тази стъпка, алгоритъмът се връща обратно към стъпка (A04); в противен случай той продължава към стъпка (A11).
- (A11) Текущото моделно време се увеличава с t° .
- (A12) Проверява се дали текущото моделно време $t^\circ \geq t_1 + t_2$. Ако не, се продължава със стъпка (A04), в противен случай функционирането на прехода приключва.

По-долу ще опишем най-общия алгоритъм за функциониране на ОМ (означен като **Алгоритъм В**). За тази цел ще въведем понятието „абстрактен преход“ (АП). Това е преход, който се явява обединение на всички активни преходи на ОМ в даден момент от време. За конструирането му се използва операцията „обединение“ [48].

Алгоритъм В за ИРОМ-5

- (B01) В съответните входни позиции в мрежата се въвеждат всички ядра α , за които $\theta_k(\alpha) \leq t_T$.
- (B02) Конструира се абстрактният преход на ОМ (първоначално преходът е празен).
- (B03) Проверява се дали стойността на текущото моделно време е по-малка от $T + t^*$.
- (B04) Ако отговорът на въпроса на стъпка (B03) е отрицателен, приключва ОМ процесът.
- (B05) Проверяват се всички преходи, за които първият времеви компонент е точно равен на текущия момент от време.
- (B06) Проверяват се типовете на всички преходи, определени на стъпка (B05), като методът на проверка е следният:
 - заменят се имената на всички позиции, които участват в булевия израз на типа на прехода като променливи, съответно със стойности:

- 0, ако съответната позиция не съдържа ядра в текущия момент;
 - 1, в противен случай;
 - изчислява се вярностната стойност на така получения булев израз.
- (B07) Към абстрактния преход се добавят всички преходи от стъпка (B06), за които типовете на преходите са удовлетворени (т.е. имат вярностна стойност „истина“).
- (B08) Над абстрактния преход се прилага **Алгоритъм А**.
- (B09) От абстрактния преход се премахват всички преходи, които са неактивни в текущия момент от време.
- (B10) Текущото време се увеличава с t° .
- (B11) Премахва се към стъпка (B03).

Както се вижда, ИРОМ-5 е разширение на стандартната ОМ, защото ако положим $\mu_c(p) = 1$ за всяка позиция p (и следователно, $\nu_c(p) = \pi_c(p) = 0$) и ако пропуснем ограничения (8) и (9) от т. 2.4.1., то ще получим стандартната ОМ.

2.4.3. Теорема за консервативност на ИРОМ-5

От друга страна, можем да формулираме и докажем следната:

Теорема: За всяка ИРОМ-5 съществува стандартна ОМ, която описва функционирането и резултатите от нейната работа. И така двата типа ОМ са еквивалентни, т.е. ИРОМ-5 са консервативни разширения на ОМ. Накрая ще споменем, че всички съществуващи до момента ОМ-разширения са консервативни.

Доказателство: Доказателството е конструктивно. Нека е дадена ИРОМ-5 G . Строим стандартна ОМ H със същата графична структура, същите динамични и темпорални параметри и същото множество на началните характеристики X . Отново, както в т. 2.2. по-горе, ще използваме Теорема 5.3.1. от [41].

Нека вземем произволен преход Z^G от G с параметри:

$$Z^G = \langle \{l'_1, \dots, l'_m\}, \{l''_1, \dots, l''_n\}, t_1, t_2, R^G, S^G, \square^G \rangle.$$

Нека Z^H е неговият съответен преход в H . Нека към Z^H да добавим една входно-изходна позиция l^H , в която непрекъснато циркулира ядро α . Ядрото α ще има за начална и единствена характеристика интуиционистки размитото множество $\{\langle p, \mu_c(p), \nu_c(p) \rangle \mid p \in \text{rg}_1 Z^H\}$, където функциите μ_c и ν_c са както е описано по-горе.

Тогава преходът Z^H ще придобие вида:

$$Z^H = \langle \{l'_1, \dots, l'_m, l^H\}, \{l''_1, \dots, l''_n, l^H\}, t_1, t_2, R^H, S^H, \wedge(\square^G, l^H) \rangle,$$

където:

- индексиранията матрица на предикатите R^H има вида:

$$R^H = \begin{array}{c|cccc} & l''_1 & \dots & l''_n & l^H \\ \hline l'_1 & & & & false \\ \vdots & & r_{i,j}^H & & \vdots \\ l'_m & & & & false \\ l^H & false & \dots & false & true \end{array}$$

и предикатите $r_{i,j}^H$ имат вида:

$$r_{i,j}^H = r_{i,j}^G \quad \wedge \quad "TIME - t_1 \leq \max_{p \in pr_1 Z} (\bar{c}(p) - [\bar{c}(p) \cdot v_c(p)])"$$

$$\wedge \quad "в момента на активиране на Z^H : $t_2 \geq \max_{p \in pr_1 Z} [\bar{c}(p) \cdot \mu_c(p)]"$.$$

- индексиранията матрица на капацитетите на дъгите S^H има вида:

$$S^H = \begin{array}{c|cccc} & l_1'' & \dots & l_n'' & l^H \\ \hline l_1' & & & & 0 \\ \vdots & & s_{i,j}^H & & \vdots \\ l_m' & & & & 0 \\ l^H & 0 & \dots & 0 & 1 \end{array},$$

т.е. дъгата, която свързва позиция l^H с прехода Z^H , е с капацитет 1, а дъгите, които я свързват с останалите позиции на прехода, са с капацитет 0. Това осигурява ядрото α да не напуска позицията и в тази позиция да не влизат други ядра.

За пълнота ще отбележим, че капацитетът на позиция l^H е 1.

Следва да проверим, че преходите Z^G и Z^H функционират по идентичен начин. Съгласно начина на построяване на прехода Z^H , той ще се активира в същия момент, както преходът Z^G . Вярностната стойност на типовете на двата прехода ще бъде еднаква, защото в позиция l^H по построение винаги има ядро, т.е. $\wedge(\square^G, l^H)$, изчислено като булев израз, ще има същата вярностна стойност като \square^G .

По дефиницията за ИРОМ-5, продължителността на активното състояние на прехода Z^G t_2 (в брой елементарни времеви стъпки) ще удовлетворява неравенствата:

$$\max_{p \in pr_1 Z} (\bar{c}(p) - [\bar{c}(p) \cdot v_c(p)]) \geq t_2 \geq \max_{p \in pr_1 Z} [\bar{c}(p) \cdot \mu_c(p)].$$

Докато това условие е изпълнено, предикатите $r_{i,j}^G$ и $r_{i,j}^H$ ще съвпадат и следователно всички ядра в двата прехода ще се движат по един и същи начин и ще получават едни и същи характеристики. Когато неравенствата в Z^G престанат да са верни, преходът ще престане да функционира, а предикатите в Z^H автоматично ще получат стойност „лъжа“, т.е. ядрата няма да могат да преминават. Двата прехода ще се деактивират в момент $t_1 + t_2$, като очевидно през двата ще преминат еднакъв брой ядра, различни от α . Следователно двата прехода ще имат еднакво поведение, откъдето може да се направи изводът, че мрежата H ще опише функционирането и резултата от работата на мрежата G , с което теоремата е доказана. \square

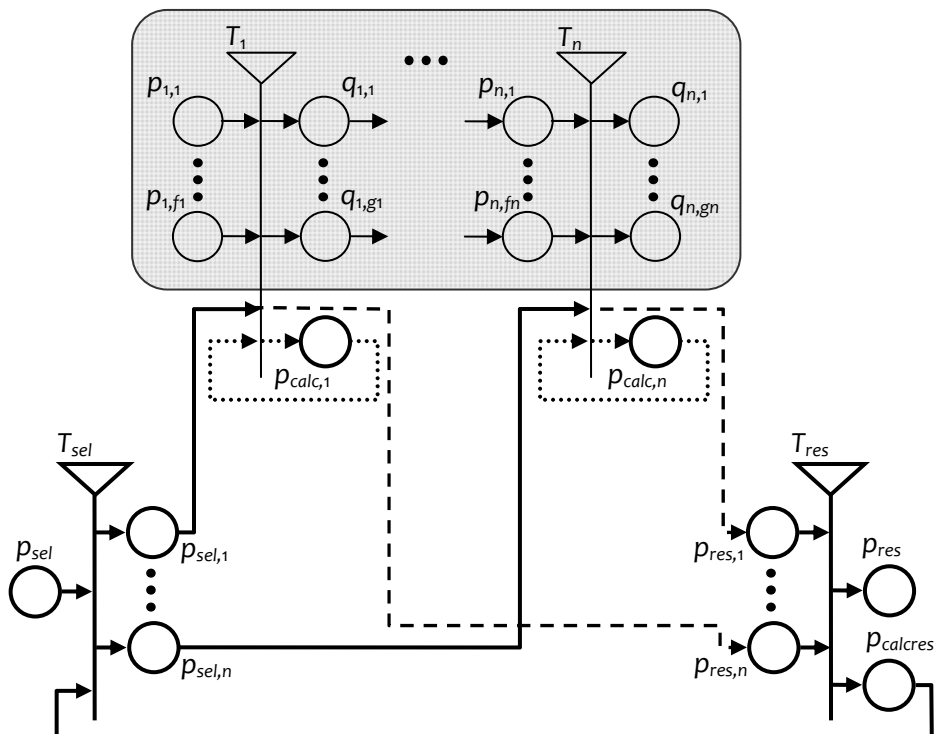
Така дефинираният нов тип ОМ може да бъде модифициран в различни направления. Идеята за тези мрежи може да бъде комбинирана например с идеите за ИРОМ от първи, втори, трети и четвърти тип, цветни ОМ и т.н.

2.5. Интуиционистки размит подход към прогнозирането чрез обобщени мрежи

Нека имаме абстрактен модел на реален процес, конструиран с апарата на ОМ. Моделът се състои от n на брой преходи, T_1, \dots, T_n и множество от позиции, входни и изходни, съвкупно означавани с p_{ij} и $q_{i,k}$ където $i \in \{1, \dots, n\}$, $j \in \{1, \dots, f_i\}$, $k \in \{1, \dots, g_i\}$, където f_i и g_i съответно представляват броя на входните и на изходните позиции на i -тия преход. На Фиг. 2.15. абстрактният ОМ модел е схематично илюстриран в сиво и разширен с допълнителни инфраструктурни елементи, които отразяват новото в текущо представения модел.

Тези допълнителни инфраструктурни елементи са следните:

- два нови прехода T_{sel} и T_{res} ,
- $3n + 3$ нови позиции:
 - p_{sel} (входна позиция за прехода T_{sel} и за цялата ОМ),
 - $p_{sel,1}, \dots, p_{sel,n}$
 - $p_{calc,1}, \dots, p_{calc,n}$ които са n позиции за междинни изчисления, прикачени към всеки от n -те прехода на оригиналната ОМ на моделирания процес,
 - $p_{res,1}, \dots, p_{res,n}$,
 - $p_{calcres}$,
 - p_{res} (изходна позиция за прехода T_{res} и за цялата ОМ).



Фигура 2.15. Предложеното разширение на абстрактния ОМ модел

Преди да продължим с динамиката на новопостроената ОМ, ще коментираме един важен аспект на нейните преходи и позиции: техните приоритети.

Първото съображение, което трябва да се направи, е, че двата нови прехода T_{sel} и T_{res} имат, съответно, най-високия и най-ниския приоритети измежду всички преходи. В рамките на множеството от преходите T_i ($i \in \{1, \dots, n\}$) може да има различни приоритети, определени в зависимост от конкретно моделирания процес, но те не са обект на обсъждане.

Второто съображение е, че в цялата мрежа имаме общо четири нива на приоритет на позициите:

- На първото ниво позициите с най-висок приоритет са $p_{sel}, p_{sel,1}, \dots, p_{sel,n}$ и $p_{calcres}$, т.е. входните и изходните позиции към прехода с най-висок приоритет T_{sel} .
- Второто ниво съдържа съвкупността от позициите на стандартната ОМ, която описва реалния процес, а именно позиции $p_{i,j}$ и $q_{i,k}$, а $i \in \{1, \dots, n\}$ е броят на преходите, и $j \in \{1, \dots, f_i\}$, $k \in \{1, \dots, g_i\}$. На Фиг. 2.15, това са позициите в оцветената в сиво област. Измежду множеството на позициите от тази група може да има позиции с различни приоритети, което се определя според конкретно моделирания процес, но приоритетите на всички тях са съвкупно по-ниски от приоритета на позициите от първото ниво и съвкупно по-високи от приоритета на позициите от третото ниво.
- Третото ниво съдържа позициите $p_{calc,i}$, $i \in \{1, \dots, n\}$, които представляват входове или изходи на прехода T_i .
- Четвъртото ниво съдържа позициите с най-нисък приоритет в новата мрежа, $p_{res,1}, \dots, p_{res,n}$ и p_{res} , т.е. входните и изходните на прехода с най-нисък приоритет T_{res} .

Така описаните допълнителни елементи от статичната природа на ОМ формират нов контур за динамичните елементи (ядрата), които пренасят мета-информация относно оригинално моделирания процес. Да означим тези ядра с α, β и $\gamma_1, \dots, \gamma_n$.

Едно α -ядро влиза в глобалната входна позиция за новоконструирувания ОМ модел p_{sel} с характеристика текущия избор (*selection*, затова и индексът е *sel*) на потребителя за намерението му при пускането на модела: дали целта на пускането е извършване на симулация, или е с цел прогнозиране. Следователно, ядрото α влиза в позицията с точно една от следните две характеристики:

„симулация“ или „прогнозиране“.

Този първи избор е от ключово значение за изпълнението на оригиналния модел, понеже го определя:

- дали информацията за преминаването на ядрата само ще бъде извлечана, статистически обработвана и съхранявана в допълнителните позиции $p_{calc,1}, \dots, p_{calc,n}, p_{calcres}$ ако потребителят е избрал да извърши симулация,
- или събраната информация от предишни симулации ще се използва (без промени) за пускане на модела с прогностична цел.

Така различното поведение на новата ОМ, предопределено от избора, направен в първата входна позиция, диктува, че ще разгледаме тези два сценария поотделно. Преди това, обаче, нека дадем някои нужни работни дефиниции.

Нека отбележим, че съгласно дефиницията на ОМ индексиранията матрица на предикатите на даден преход съдържа информация за възможните маршрути на преминаване на ядрата от входните към изходните позиции на прехода. Клетките от индексиранията матрица могат да съдържат или изричните стойности „истина“ и „лъжа“, или някакви предикати, които подлежат на оценяване и могат да дадат в резултат стойностите „истина“ или „лъжа“ в зависимост от определени условия. Когато дадена клетка съдържа стойността „истина“, това означава, че всяко ядро, намиращо се в съответната входна позиция на прехода, може да премине към изходна позиция; когато стойността е „лъжа“, може просто да се смята, че не съществува дъга, свързваща тази входна и тази изходна позиция.

Ще казваме, че ядро, намиращо се във входна позиция l_i , *подлежи* на трансфер към изходна позиция l_j винаги когато в (i, j) -тата клетка на индексиранията матрица на предикатите на прехода стойността е или “истина” (безусловен преход), или някакъв предикат $W_{i,j}$, чиято вярностна стойност текущо се оценява на „истина“ (условен преход).

Винаги когато дадено ядро подлежи на трансфер, то ще премине през прехода стига да разполага с нужното време на текущата времева стъпка. Тази допълнителна зависимост от времето означава, че някои ядра, които иначе подлежат на трансфериране през прехода, може да не успеят да завършат тази процедура в текущия момент от време, т.е. те ще останат във входната позиция до момента на следващата активация на прехода.

Същото ядро *няма да подлежи* на преход към дадена изходна позиция, когато изчислената вярностна стойност на предиката, свързващ входната му позиция с тази изходна позиция в текущия момент има вярностна стойност „лъжа“, т.е. текущите условия не позволяват преход на ядрото, въпреки че такъв преход би бил позволен при други обстоятелства. Така, отразявайки логиката зад модела, напълно игнорираме случая, когато ядрото безусловно не може да извърши преход, т.е. в съответната клетка в индексиранията матрица на предикатите стои стойност „лъжа“ (а не предикат, който при изчисление евентуално дава стойност „лъжа“).

Както може веднага да се види, и множеството от „успешните“ ядра (да означим кардиналното му число с μ) и множеството на „неуспешните“ ядра (съответно с кардинално число ν) са подмножества на множеството на всички ядра във входните позиции на дадения преход (да означим кардиналното му число с τ). Онова, което не се вижда непосредствено, е че сумата на кардиналните числа на двете множества може и да не се допълва до τ . Възможно е някои ядра във входни позиции да подлежат на преминаване към изходни позиции (поради верността на изчислените предикати в индексиранията матрица на прехода), но да не са могли да преминат поради ограниченото времетраене на активното състояние на прехода. Следователно,

$$0 \leq \frac{\tau - \mu - \nu}{\tau} \leq 1.$$

Тази ситуация прилича на случая на интуиционистки размитите степени на принадлежност (μ/τ), непринадлежност (ν/τ) и неопределеност (допълнението на първите две до 1).

Още едно наблюдение може да бъде направено относно времевата скала и елементарната времева стъпка на новоконструираната ОМ. Удобно е да приемем, че двете са практически идентични с онези на оригинално моделирания процес, но продължителността на функционирането на новата разширена ОМ ще бъде точно една (елементарна) времева стъпка по-дълго от продължителността на функциониране на оригиналната мрежа. Необходимостта от тази конструкция следва от факта, че веднъж след като нужната метаинформация за преходите на симулирания процес се съхрани в ядрата в позиции $p_{res,1}, \dots, p_{res,n}$, е нужно да бъде окончателно натрупана в позиция $p_{calcres}$ (ядра γ_i се сливат обратно в едно ядро γ) и да се съхрани за бъдещи пускания на модела с цел симулация или прогностика.

Ще продължим с описанието на модела. Първо ще го опишем в сценария, при който моделът се пуска с цел *симулация*, а после – в сценария, при който се пуска с цел *прогноза*. Като общо правило са нужни голям брой симулации преди да се проведе проиграване на модела с цел прогноза, което да претендира за правдоподобност.

2.5.1. Сценарий 1: Симулация

Когато потребителят избере да стартира модела със симулационна цел, от него се извлича информация за поведението на оригинално моделирания процес. За целта разглеждаме няколко променливи, свързани с ядрата, циркулиращи в оригиналния ОМ модел.

Първата променлива съответства на общия брой на ядрата (означили сме ги с τ), които стоят във входните позиции на всеки преход. За отбелязване е, че по време на цялостното функциониране на ОМ един преход може да се активира повече от един път и следователно всяко активиране на прехода до края на функционирането на мрежата трябва да бъде отчитано.

Друга променлива съответства на броя на онези ядра, които успешно са извършили преминаването към изходни позиции по време на текущото активно състояние на съответния преход. Тези ядра, да ги наречем μ -ядра, не само са подлежащи на трансфер, но и също са имали достатъчно време, за да го осъществят. Точният маршрут на ядрата не е обект на обсъждане, доколкото е част от вътрешната логика на оригиналния модел.

Поддържа се също така и променлива за броя на ядрата, които не са подлежащи на трансфер, по причина че изчислението на предикатите в съответните клетки от индексирания матрица на предикатите на прехода са дали в резултат стойност „лъжа“. Избираме да ги означаваме с ν .

Тези τ -, μ - и ν -ядра:

- се преброяват постъпково в позиции $p_{calc,i}$ за всеки преход T_i ($i \in \{1, \dots, n\}$) на оригинално моделирания процес, за всяко активиране на преходите в течение на дадената симулация, и
- се събират в позиция $p_{calcres}$ в края на всяка симулация. Дотогава докато моделът се пуска за симулационни цели, информацията от оригиналната ОМ ще се събира в позиция $p_{calcres}$.

По-формално, видът на първия преход T_{sel} е:

$$T_{sel} = \langle \{p_{sel}, p_{res}\}, \{p_{sel_1}, \dots, p_{sel_n}\}, IM_{sel} \rangle,$$

където:

$$IM_{sel} = \begin{array}{c|ccc} & p_{sel,1} & \dots & p_{sel,n} \\ \hline p_{sel} & true & \dots & true \\ p_{calcres} & true & \dots & true \end{array}.$$

Това означава, че в началото на симулацията α -ядрото в позиция p_{sel} се разцепва на n идентични копия, които безусловно преминават към всички позиции $p_{sel,1}, \dots, p_{sel,n}$ на преход T_{sel} , които представляват n допълнителни входни позиции на n -те прехода на моделирания процес (този безусловен преход се кодира в индексиранията матрица на предикатите за T_{sel} като стойността „истина“).

Едновременно с това в позиция $p_{calcres}$, която е втората входна позиция на T_{sel} , стои едно подлежащо на разцепване β -ядро със следната характеристика:

$$x^\beta = "R, VAL_1, VAL_1^*, VAL_2, VAL_2^*, \dots, VAL_n, VAL_n^* ",$$

където:

- R е поредният номер на текущото пускане на модела.
- VAL_1, \dots, VAL_n са n -те масива със съхранени стойности за текущото пускане на модела, които имат вида:

$$VAL_i = " \left\langle R, A_{R,i}, \sum_{s=1}^{A_{R,i}} \tau_{R,i,s}, \sum_{s=1}^{A_{R,i}} \mu_{R,i,s}, \sum_{s=1}^{A_{R,i}} \nu_{R,i,s} \right\rangle ",$$

където $i \in \{1, \dots, n\}$ е поредният номер на прехода, а $A_{R,i}$ е общият брой на активиранията на прехода T_i по време на R^{TOTO} пускане на ОМ модела. Третият параметър, а именно

$$\sum_{s=1}^{A_{R,i}} \tau_{R,i,s},$$

отговаря на общия брой τ на ядрата във всички входни позиции на T_i , натрупани по време на $A_{R,i}$ -те брой активирания на прехода по време на R^{TOTO} пускане на модела. Подобно е значението и на четвъртия и петият параметри, където μ отговаря на онези от ядрата, които подлежат на трансфер и успешно преминават от входна към изходна позиция по време на всичките $A_{R,i}$ активирания на прехода по време на R^{TOTO} пускане на модела. Накрая, ν отговаря на броя на ядрата във входната позиция, които не подлежат на преминаване към изходна позиция.

- VAL_1^*, \dots, VAL_n^* са n масива, които кумулативно натрупват сборовете на съответните стойности на τ -, μ - и ν -ядрата по време на всички предишни пускания на модела, във вида:

$$VAL_i^* = " R, \sum_{s=1}^R A_{i,s}, \sum_{s=1}^R \tau_{i,s}, \sum_{s=1}^R \mu_{i,s}, \sum_{s=1}^R \nu_{i,s} ",$$

където R представляват общия брой на пусканията (симулациите) направени дотук, а останалите стойности са сборовете на съответните стойности за едно пускане. В началото, когато $R = 0$,

$$VAL_i = \langle 0, 0, 0, 0, 0 \rangle$$

и

$$VAL_i^* = \langle 0, 0, 0, 0, 0 \rangle.$$

След това, когато R е естествено число по-голямо от 0,

$$VAL_i = \langle R, 0, 0, 0, 0 \rangle$$

и характеристиката на ядрото β преди поредната симулация е:

$$x^\beta = \langle R, \langle R, 0, 0, 0, 0 \rangle, VAL_1^*, \langle R, 0, 0, 0, 0 \rangle, VAL_2^*, \dots, \langle R, 0, 0, 0, 0 \rangle, VAL_n^* \rangle.$$

Както вече споменахме, двете ядра α и β се разцепват на по n ядра ($\alpha_1, \dots, \alpha_n$ и β_1, \dots, β_n) по време на преминаването им от съответните входни позиции на прехода T_{sel} до изходните позиции $p_{sel,1}, \dots, p_{sel,n}$. Ядрата α_i имат идентични характеристики като ядрото α , а именно „избор“. Ядрата β_i имат характеристики „ VAL_i, VAL_i^* “, които са съответните части от характеристиката на ядрото β . В изходните позиции всяко ядро α_i се слива със съответното β_i -ядро в n нови ядра: γ_i . Началната характеристика на γ_i е „ x^α, x^{β_i} “, където:

- x^α е характеристиката на ядрото α („симулация“), и
- x^{β_i} е характеристиката на ядрото β_i , а именно „ VAL_i, VAL_i^* “. Както беше споменато, в началния момент тези масиви се състоят само от нули.

И така n -те γ -ядра безусловно преминават към n -те входни позиции $p_{sel,1}, p_{sel,2}, \dots, p_{sel,n}$ на n -те преходи, които съставляват оригинално моделирания процес. Да опишем формално само i -тия от тези n прехода след модификациите, въведени от новопредложения подход. За по-лесно разбиране сегментът от индексиранията матрица на предикатите на прехода, който отговаря на оригиналния модел, също е оцветен в сиво, както на Фиг. 2.15.

$$T_i = \langle \{p_{i,1}, \dots, p_{i,fi}, p_{sel,i}, p_{calc,i}\}, \{q_{i,1}, \dots, q_{i,gi}, p_{calc,i}, p_{res,i}\}, IM_i \rangle,$$

където:

$IM_i =$	$q_{i,1}$	\dots	$q_{i,gi}$	$p_{calc,i}$	$p_{res,i}$
$p_{i,1}$	\dots	\dots	\dots	$false$	$false$
\vdots	\vdots	$W_{i,j,k}$	\vdots	\vdots	\vdots
$p_{i,fi}$	\dots	\dots	\dots	$false$	$false$
$p_{sel,i}$	$false$	\dots	$false$	$true$	$false$
$p_{calc,i}$	$false$	\dots	$false$	$W_{calc,i}$	$\neg W_{calc,i}$

където:

- $W_{i,j,k}$ са съответните предикати на оригинално моделирания процес (от чийто конкретен вид се абстрахираме).
- $W_{calc,i}$ = „времметраенето на функционирането на оригиналния ОМ модел още не е приключило“.
- $\neg W_{calc,i}$ е отрицанието на предиката $W_{calc,i}$.

Ядрото γ_i , което стои в позиция $p_{sel,i}$, безусловно постъпва в позиция $p_{calc,i}$ и не подлежи на директен преход към позиция $p_{res,i}$. То може да достигне позиция $p_{res,i}$ единствено ако е направило поне един цикъл през позиция $p_{calc,i}$ (позицията за междинни изчисления на стойностите на масивите „ VAL_i, VAL_i^* “, които отразяват движението на ядрата по време на симулацията). Следователно, както предвижда предикатът $W_{calc,i}$, ядрото ще цикли в позиция $p_{calc,i}$ дотогава, докато функционира оригиналният ОМ модел, като така запазва цялата информация за преминаването на ядрата по време на всички възможни активирания на преход T_i . В момента, в който оригиналният модел спре да функционира, ядрото γ_i , което е циклило в позиция $p_{calc,i}$, записвайки данните за „успешните“ и „неуспешните“ преминавания на ядра през прехода T_i , се придвижва към позиция $p_{res,i}$ получавайки характеристиката „ VAL_i, VAL_i^* “, където съответните стойности на масивите представляват натрупването на тази информация по време на всички активирания на прехода в рамките на цялата симулация на оригиналната ОМ. Това е възможно, тъй като приоритетът на позициите $p_{calc,i}$ ($i \in \{1, \dots, n\}$) е по-нисък от приоритетите на позициите $p_{i,1}, \dots, p_{i,fi}$, $q_{i,1}, \dots, q_{i,gi}$.

Нека сега дискутираме как ще продължат да се придвижват ядрата γ_i във втория новодобавен преход T_{res} . За отбелязване е, че поради най-ниския приоритет на този преход, той ще се активира само когато цялостното функциониране на оригиналната ОМ е напълно завършило, т.е. след края на симулацията. Формалната дефиниция на прехода е следната:

$$T_{res} = \langle \{p_{res,1}, \dots, p_{res,n}\}, \{p_{res}, p_{calcres}\}, IM_{res} \rangle,$$

където:

$$IM_{res} = \begin{array}{c|cc} & P_{res} & P_{calcres} \\ \hline p_{res,1} & W_{sel} & true \\ \dots & \dots & \dots \\ p_{res,n} & W_{sel} & true \end{array}.$$

и предикатът $W_{sel} = „x^\alpha$ е прогнозиране“. Следователно, при първия сценарий този предикат има вярностна стойност „лъжа“.

Ядрата γ_i от входните позиции $p_{res,i}$ безусловно преминават към изходната позиция $p_{calcres}$, където се сливат в едно ядро, β , с характеристика:

$$x^\beta = \langle \overline{R}, \overline{VAL_1}, \overline{VAL_1^*}, \overline{VAL_2}, \overline{VAL_2^*}, \dots, \overline{VAL_n}, \overline{VAL_n^*} \rangle,$$

където:

- $\overline{R} = R + 1$,
- $\overline{VAL_i} = \langle \overline{R}, 0, 0, 0, 0 \rangle$,
- $\overline{VAL_i^*} = VAL_i^* + VAL_i$.

Така окончателните данни за трансферите на ядрата по време на последната симулация на модела не се пазят поотделно, а се обединяват с данните, получени от всички предишни проигравания на модела.

При първия сценарий когато целта на проиграването е „симулация“, както е зададено в характеристиката на началото ядро α , към позиция p_{res} на прехода T_{res} не преминават никакви γ_i ядра. Достигането на тази изходна позиция е възможно само когато целта, с която е пуснат моделът, е „прогнозиране“, което е и вторият сценарий, който ще дискутираме по-долу.

2.5.2. Сценарий 2: Прогнозиране

По време на втория сценарий, функционирането на първия преход T_{sel} е същото както по време на първия сценарий.

Основната разлика между двата сценария е в преходите T_i ($i \in \{1, \dots, n\}$). Преди да ги опишем формално, нека да означим с $r_{i,j,k}$ една случайна променлива от интервала $[0; 1]$, която е свързана с прехода T_i . Тази променлива, отговаряща на j -тата входна позиция и k -тата изходна позиция на i -тия преход, ще се използва за определяне на типа на новите предикати $V_{i,j,k}$, които ще заменят оригиналните предикати, разгледани в първия сценарий.

В зависимост от целта на прогностичното проиграване на модела, има два различни варианта за развитието на процесите в оригиналния ОМ модел. Тези варианти могат условно да се нарекат „размит“ случай и „интуиционистки размит“ случай. И в двата случая формалното представяне на прехода е както следва:

$$T_i = \langle \{p_{i,1}, \dots, p_{i,fi}, p_{sel,i}, p_{calc,i}\}, \{q_{i,1}, \dots, q_{i,gi}, p_{calc,i}, p_{res,i}\}, IM_i \rangle,$$

където:

$IM_i =$	$q_{i,1}$	\dots	$q_{i,gi}$	$p_{calc,i}$	$p_{res,i}$
$p_{i,1}$	\dots	\dots	\dots	<i>false</i>	<i>false</i>
\vdots	\vdots	$V_{i,j,k}$	\vdots	\vdots	\vdots
$p_{i,fi}$	\dots	\dots	\dots	<i>false</i>	<i>false</i>
$p_{sel,i}$	<i>false</i>	\dots	<i>false</i>	<i>true</i>	<i>false</i>
$p_{calc,i}$	<i>false</i>	\dots	<i>false</i>	$W_{calc,i}$	$\neg W_{calc,i}$

където:

- $W_{calc,i}$ = „времето на функциониране на оригиналната ОМ не е приключило“.
- $V_{i,j,k}$ са новите предикати, заменящи тези на оригинално моделирания процес.
 - **Размит случай:**

$$V_{i,j,k} = "r_{i,j,k} \in \left[0; \frac{\sum_{s=1}^R \mu_{i,s}}{R} \right] ",$$

(използваните означения са като гореописаните). В този вариант, продължителността на функциониране на прехода T_i тук съвпада с продълже-

нието на функциониране на T_i в оригиналната мрежа. Това означава, че по време на работата на модела с цел прогнозиране могат да преминат същият брой ядра както в оригиналната мрежа.

- **Интуиционистки размит случай:** В този случай, преходът T_i ще бъде активен толкова на брой времеви стъпки, колкото е общият брой на ядрата в неговите входни позиции. Предикатът $V_{i,j,k}$ ще получи вида си по следното правило:

$$\left\{ \begin{array}{l} \text{ако } r_{i,j,k} \in \left[0; \frac{\sum_{s=1}^R \mu_{i,s}}{\sum_{s=1}^R \tau_{i,s}} \right], \quad \text{то } V_{i,j,k} = \text{„true“ и ядрото} \\ \text{ако } r_{i,j,k} \in \left(\frac{\sum_{s=1}^R \mu_{i,s}}{\sum_{s=1}^R \tau_{i,s}}; \frac{\sum_{s=1}^R \mu_{i,s} + \sum_{s=1}^R \nu_{i,s}}{\sum_{s=1}^R \tau_{i,s}} \right), \quad \text{то } V_{i,j,k} = \text{„false“ и ядрото не} \\ \text{ако } r_{i,j,k} \in \left(\frac{\sum_{s=1}^R \mu_{i,s} + \sum_{s=1}^R \nu_{i,s}}{\sum_{s=1}^R \tau_{i,s}}; 1 \right), \quad \text{то } V_{i,j,k} = \text{„false“ и ядрото не} \end{array} \right. \begin{array}{l} \text{преминава към изходна позиция} \\ \text{преминава (предикатът е „истина“)} \\ \text{преминава (предикатът е „лъжа“)} \\ \text{преминава (поради} \\ \text{недостатъчно време)} \end{array}$$

Посредством това правило ние можем да имитираме реално функциониране на Алгоритъма за движение на ядра в преход (описан в Глава 1).

Накрая третият преход T_{res} запазва вида си, описан в първия сценарий, но сега предикатът $W_{sel} = „x^\alpha \text{ е прогнозиране}“$ се оценява като „истина“. Това променя поведението и характеристиката на ядрата γ_i . Всяко от тях се разцепва на две идентични ядра γ'_i и γ''_i .

Ядрата γ'_i с характеристики „ VAL_i, VAL_i^* “ преминават към позиция p_{res} и се сливат там с ядрото δ , получаващо характеристиката:

$$\text{„} VAL_1^*, VAL_2^*, \dots, VAL_n^* \text{“},$$

която позволява на потребителя да проследява назад и да разбира условията, предопределили развоя на прогностичното проиграване на модела.

Ядрата γ''_i , също с характеристики „ VAL_i, VAL_i^* “, преминават към позиция $p_{caleres}$ и се сливат там в ядро β без да получават нова характеристика. Така при бъдещи пускания на модела, без значение от целта на пускането, това ядро β може отново да бъде използвано като вход в прехода с най-висок приоритет T_{sel} .

*
**

Така конструираният модел описва процеса на събиране и натрупване на статистически данни за преминаването на ядра по време на симулации на произволни ОМ модели. Използвайки тези данни, наред с интуитивни размити оценки, можем да използваме тази конструкция за съставяне на прогнози относно поведението на модела.

Представените тук идеи подсказват конструирането на още по-детайлен подход към събирането на статистически данни, когато информацията се събира не на нивото на преходи, както е тук, а на нивото на входните и изходните позиции на прехода. Подобно детайлизиране ще бъде обект на бъдещи изследвания.

Глава 3. Приложения на обобщените мрежи

В тази глава ще бъдат описани някои нови приложения на ОМ.

В точка 3.1., следваща доклада [77], са описани две обобщени мрежи. Първата от тях представлява универсалната обобщена мрежа G_{ACOA} за класа алгоритми за оптимизация по метода на мравките, описваща начина на функциониране и резултатите от тяхната работа. После дискутираме как да се построи ОМ, която включва G_{ACOA} като оптимизационен компонент, т.е. като нейна подмрежа, чиито резултати от работата определят оптималното движение на ядрата в новата мрежа.

В следващата точка 3.2., която има за основа доклада [78], с ОМ е описан модел на хибриден метаевристичен подход, комбиниращ метода на мравките с генетични алгоритми. Целта е решения, получени с метода на мравките, да се подадат като начална популация за генетичния алгоритъм, с което отчасти да се контролира иначе случайния ѝ характер и да се постигнат по-добри и по-бързи резултати.

В точка 3.3. са представени моделите на двата най-общи и разпространени подхода за разрешаване на редакционни конфликти в системи с контрол на версиите, каквито е вече практика да се използват в процеса на софтуерна разработка, в технологията *wiki*, в системи за управление на съдържанието и на други места. Двата модела са публикувани в [72, 79].

3.1. Обобщеномрежов модел с оптимизационен компонент универсалната обобщена мрежа за класа алгоритми по метода на мравките

В статии [98–103, 105] и монография [104] е показано, че ОМ могат да представят функционирането и резултатите от работата на различни варианти на алгоритми за оптимизация по метода на мравките. От друга страна, в [41] е показано, че можем да конструираме специални видове ОМ, характеризиращи се с оптимизационен компонент. Под „оптимизационен компонент“ ще разбираме подмрежа на дадената ОМ, която описва конкретна оптимизационна задача и резултатите от чието функциониране може да се използва от основната мрежа с цел да бъде оптимизирано поведението ѝ. Досега подобни оптимизационни компоненти са били дефинирани за Транспортната задача и за Задачата за търговския пътник [41].

В доклада [77] с участието на автора за първи път е предложена оптимизационна задача, моделираща посредством ОМ алгоритъма за оптимизация по метода на мравките. Този компонент осигурява, че ОМ функционира оптимално по отношение на резултатите от изпълнението на алгоритъма на мравките. Накратко ще представим самия метод.

3.1.1. Метод на мравките

Оптимизацията по метода на мравките (*Ant Colony Optimization, ACO*) е метаевристичен метод, предложен от М. Дориго [90] през 1992 г. ACO алгоритмите се прилагат към задачи от реалния живот и промишлеността, при които за кратък срок се изискват задоволително добри решения. ACO открива добри решения на оптимизационни задачи с ограничителни условия, каквато задача е например Задачата за раницата.

Метаевристичките стават все по-популярни в различни изследователски области и индустрията. Повечето от тях представляват вдъхновени от биологията и екологията метафори: подходи за решаване на комплексни оптимизационни задачи. За разлика от точните методи, метаевристичките позволяват да се решават мащабни задачи чрез предоставянето на удовлетворителни решения за разумно кратко време.

Методът на мравките представлява многоагентна система от изкуствени „мравки“, в която взаимодействието на ниско ниво между отделните индивиди води до комплексно поведение на цялата колония. Методът имитира поведението на реални мравки, които търсят храна. Мравките са социални насекоми, които живеят на колонии и чието поведение цели в по-голяма степен оцеляването на цялата колония, отколкото оцеляването на отделния индивид. Важен и интересен аспект на метафората е начинът, по който мравките откриват най-краткия път между мравуняка си и източник на храна. Мравките предават помежду си информацията за източниците на храна, използвайки химическото вещество феромон, което те секретират по време на движение.

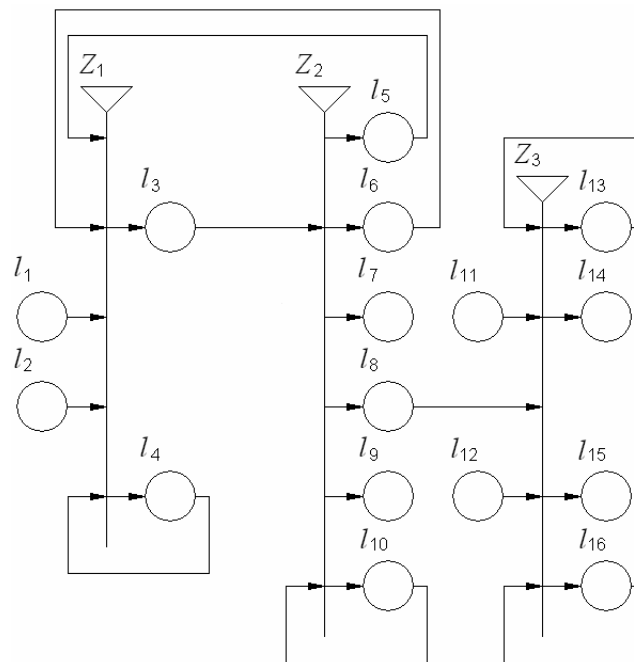
По аналогия ACO се базира на индиректната комуникация на една колония от прости агенти, наречени изкуствени мравки, опосредствана от следите изкуствен феромон. Следите феромон в ACO алгоритмите са представени чрез разпределена числова информация, която мравките използват за да конструират на вероятностен принцип решения на проблема и която информация бива актуализирана от мравките по време на изпълнението на алгоритъма по начин, отразяващ резултатите от тяхното търсене. Изкуствените мравки не само имитират описаното поведение, но те също и прилагат допълнителна евристична информация, специфична за решавания проблем. Идеята за този метод е предложена от Мойсън в [118]. Първият пример на алгоритъм на мравките е Система от мравки (*Ant System*) [92], приложен към решаването на разнообразни тежки задачи за комбинаторна оптимизация.

Към днешна дата са разработени голямо разнообразие от варианти на ACO алгоритми като *Ant colony system* [118], *Ant algorithm with elitist ants* [93], *Max-min ant system* [142], *Ant algorithm with additional reinforcement* [97], *Best-worst ant system* [91], които са прилагани към много оптимизационни задачи.

3.1.2. Универсална ОМ G_{ACO} за класа алгоритми по метода на мравките

Следвайки [101], ще конструираме ОМ, която е универсална за всички стандартни АСО-алгоритми. Тази ОМ ще опише как множество от n изкуствени мравки се придвижват по върховете на граф на АСО-алгоритъм, който се състои от m на брой върха и k на брой ребра. Да допуснем, че предварително е известно дали и къде в графа е останало някакво количество феромон от предишна симулация на процеса.

Нека означим така описаната ОМ с G_{ACO} . Тя има три прехода, 16 позиции (вж Фиг. 3.1) и четири типа ядра (α , β , γ и ε).



Фигура 3.1. ОМ модел на АСО

В ядрото α е заложено да се съдържа информация относно разположението на изкуствените мравки в графа. Ядрото β съдържа информация за броя на върховете (m) и броя на ребрата (k) на графа. Ядрото γ съдържа структурата на графа, т.е. матрицата му на инцидентност, докато ядрото ε съдържа информация на кои върхове/дъги при предишна симулация на процеса е бил положен феромон от мравките, и в какви количествата. Всички тези ядра постъпват във входните позиции на ОМ с определени начални характеристики, които са описани по-долу:

- ядрото α постъпва в позиция l_1 с начална характеристика: „ n -мерен вектор, чиито елементи са разположенията на мравките в графа“;
- ядрото β влиза в позиция l_2 с начална характеристика: „ $\langle t$ -мерен вектор с елементи върховете на графа или l -мерен вектор с елементи ребрата на графа; целева функция“;

- ядрото γ влиза в позиция l_{11} с начална характеристика: „структура на графа с m върха и l ребра“;
- ядрото ε влиза в позиция l_{12} с начална характеристика: „начални данни за местоположението и количеството положен феромон“.

Ще опишем трите прехода последователно. Първият преход Z_1 има вида:

$$Z_1 = \langle \{l_1, l_2, l_4, l_5, l_6\}, \{l_3, l_4\}, r_1 \rangle,$$

където

$r_1 =$		l_3	l_4
l_1		<i>true</i>	<i>false</i>
l_2		<i>false</i>	<i>true</i>
l_4		<i>false</i>	<i>true</i>
l_5		<i>true</i>	<i>false</i>
l_6		<i>true</i>	<i>false</i>

Ядрото α , намиращо се или в позиция l_1 , или в позиция l_5 , или в позиция l_6 , постъпва в позиция l_3 с характеристика:

„вектор на текущите стойности на функцията на прехода
 $\langle \varphi_{1,cu}, \varphi_{2,cu}, \dots, \varphi_{n,cu} \rangle$ “,

докато ядрото ε остава да цикли в позиция l_4 , получавайки характеристиката:

„нов m -мерен вектор с елементи върховете на графа, или
нов l -мерен вектор с елементи ребрата на графа“.

Вторият преход Z_2 има вида:

$$Z_2 = \langle \{l_3, l_{10}\}, \{l_5, l_6, l_7, l_8, l_9, l_{10}\}, r_2 \rangle,$$

където

$r_2 =$		l_5	l_6	l_7	l_8	l_9	l_{10}
l_3		$W_{3,5}$	$W_{3,6}$	$W_{3,7}$	<i>true</i>	$W_{3,9}$	$W_{3,10}$
l_{10}		<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	$W_{10,9}$	$W_{10,10}$

и предикатите в r_2 имат следния смисъл:

- $W_{3,5}$ = „Текущата итерация не е свършила“;
- $W_{3,6} = W_{3,10} = \neg W_{3,5} \vee \neg W_{10,9}$,
- $W_{10,7}$ = „Текущото най-добро решение е по-лошо от глобалното най-добро решение“;
- $W_{10,9}$ = “Вярностната стойност на $C_1 \vee C_2 \vee C_3$ е истина”;
- $W_{10,10} = \neg W_{10,9}$,

а използваните в предиката $W_{10,9}$ изрази C_1 , C_2 и C_3 са следните условия за край:

- C_1 = „Постигнат е максимален брой итерации (изчислително време)“;
- C_2 = „Постигнат е определен брой итерации без да е подобрен резултатът“;
- C_3 = „Ако е известна долна/горна граница, то текущите резултати са близо до тази граница (напр. по-малко от 5%)“.

Ядрото α от позиция l_3 постъпва в позиция l_5 с характеристика:

$$\langle S_{1,ci}, S_{2,ci}, \dots, S_{n,ci} \rangle,$$

където $S_{i,ci}$ е текущото частично решение за текущата итерация, извършена от i -тата мравка ($1 \leq i \leq n$).

Ако предикатът $W_{3,6}$ е истина, то ядрото α се разцепва на три ядра: същото α и ядрата α' и α'' :

- Ядрото α постъпва в позиция l_6 с текуща характеристика:
„нов n -мерен вектор с елементи новите местоположения на мравките“;
- Ядрото α' постъпва в позиция l_8 с последната характеристика на ядрото α (преди разцепването);
- Ядрото α'' постъпва в позиция l_{10} с текуща характеристика наредената двойка:

$$\langle \text{най-доброто решение за текущата итерация}; \\ \text{пореден номер на итерацията} \rangle.$$

Когато предикатът $W_{10,9}$ е истина, ядрото α'' може да постъпи в позиция l_9 , където получава характеристиката „най-добро получено решение“.

В позиция l_7 постъпва онова от двете ядра от позиция l_{10} , което има по-лоши стойности като текуща характеристика, докато в позиция l_{10} остава да цикли ядрото, имащо за текуща характеристика най-добрите стойности.

Третият преход Z_3 има вида:

$$Z_3 = \langle \{l_8, l_{11}, l_{12}, l_{13}, l_{16}\}, \{l_{13}, l_{14}, l_{15}, l_{16}\}, r_3 \rangle,$$

където:

$r_3 =$	l_{13}	l_{14}	l_{15}	l_{16}
l_8	false	false	false	true
l_{11}	true	false	false	false
l_{12}	false	false	false	true
l_{13}	$W_{13,13}$	$W_{13,14}$	false	false
l_{16}	false	false	$W_{16,15}$	$W_{16,16}$

и предикатите в r_3 имат следния смисъл:

- $W_{13,14} = W_{16,15} =$ „Вярностната стойност на израза $C_1 \vee C_2 \vee C_3$ е истина“,
- $W_{13,13} = W_{16,16} = \neg W_{13,14}$.

Ядрата γ от позиция l_{11} и β от позиция l_{12} с гореописаните характеристики постъпват, съответно, в позиции l_{13} и l_{16} без нова характеристика.

Ядрото α от позиция l_8 постъпва в позиция l_{16} и се обединява там с ядрото β (като новото ядро отново е под името β), което получава характеристика:

$$\langle \text{стойност на функцията, актуализираща феромона} \\ \text{съобразно стойностите на целевата функция} \rangle.$$

Ядрата β и γ постъпват, съответно, в позиции l_{14} и l_{15} без да получават нови характеристики.

3.1.3. Обобщени мрежи с оптимизационен компонент G_{ASOA}

На базата на дефиницията на понятието ОМ, ще построим нов тип ОМ – ОМ с оптимизационни компоненти (*GNs with optimization components*); да го означим с *GNOC*. Ядрата в новия тип ОМ се придвижват съобразно класическите алгоритми, дискутирани в [41, 48], базирани на проверката на вярностните стойности на предикатите, определящи условията на преходите.

Нека E е произволна ОМ и нека искаме да управляваме придвижването на ядрата в нея по оптимален начин, на база решенията, получени от АСО алгоритъма. Можем да обединим (с операцията обединение между ОМ) мрежите E и G_{ASOA} и в новата ОМ можем да организираме оптималния начин за придвижване на ядрата.

Според алгоритъма за придвижване на ядрата от [48], придвижването на ядрата се осъществява на всяка времева стъпка от функционирането на дадената ОМ в рамките на активното състояние на прехода, т.нар. „абстрактен преход“, който е обединение от всички активни ОМ-преходи в текущия момент.

По тази причина можем да опишем функционирането само на един ОМ-преход и в частност, на абстрактния преход.

Нека е даден преходът Z от мрежата E . Всяка от неговите входни позиции (в термините на АСО алгоритъма) ще съответства на един начален връх на графа. Нека определим всички ОМ-ядра във входовете на Z . Броят им нека е началната характеристика на ядрото α от G_{ASOA} (във вида на началните върхове на дадения граф). Информацията за ядрата, които съдържат тези ядра ще бъде зададена като начална характеристика на ядрото β на същата ОМ. Информацията за графа може да бъде въведена в ОМ G_{ASOA} преди началото на процеса и ще бъде съхранявана по време на цялата симулация.

Като се имат предвид характеристичните функции на изходните позиции на прехода Z , можем да определим онези от върховете на графа, които имат достатъчно високи тегла.

Всяко ядро от входните позиции на Z ще съответства на „мравка“, която трябва да търси път до върха от графа, който има максимално тегло.

Компонентите t_1 , t_2 и M от ОМ прехода не са необходими (например, можем да приемем, че стойностите на t_1 са последователни естествени числа, $t_2 = 1$, а елементите от M са равни на 1). Типът на прехода има вида на дизюнкция.

На базата на определените по-горе характеристики на ядрата от G_{ASOA} , тези ядра преминават през мрежата и в резултат определят маршрутите на отделните мравки (а следователно и на ядрата от Z) от началните върхове на графа до върховете му с максимални тегла (които отговарят съответно на входните и изходните позиции на прехода Z). Така, използвайки решенията от мрежата G_{ASOA} , можем да определим начинът, по който да преминават ядрата от входните към изходните позиции на прехода Z .

Тази процедура може да се използва за управление на всеки един от преходите на ОМ E . В този случай в G_{ASOA} ще поставим съответния брой α - и β -ядра, така че техният брой да отговаря на броя преходи в ОМ E .

*
**

Описаната процедура е универсална по своята природа. Приложима е към всяка ОМ. В [41] е описана ОМ с оптимизационен компонент, решаващ Транспортната задача. Настоящият модел е вторият, но в бъдеще подобни модели могат да се конструират и за други оптимизационни процедури, например за Задачата за търговския пътник, Задачата за раницата и т.н.

Предложената тук процедура може да намери разнообразни приложения, например в спешната медицина. Следвайки идеите, предложени в [110, 119] е възможно да се конструират ОМ модели на организацията и управлението на звено за спешна медицинска помощ, а използвайки оптимизационен компонент по метода на мравките може да се търси оптималност при вземането на решения в звеното.

Ще отбележим още, че в [76] в най-общ план е дискутирана и идеята за прилагане на подхода за оптимизация по метода на мравките към движението на ядрата в ОМ.

3.2. Обобщена мрежа, синхронизираща хибриден метаевристичен алгоритъм

Тук ще бъде описан ОМ модел, базиращ се на два от най-добре представящите се метаевристични подхода: алгоритмите за оптимизация по метода на мравките и генетичните алгоритми. В т. 3.1. вече беше описан методът на мравките, затова по-долу ще се спрем само на генетичните алгоритми.

3.2.1. Генетични и хибридни метаевристични алгоритми

Генетичните алгоритми (ГА, *genetic algorithms*, *GA*) са смятани за обещаваща иновативна концепция, съдържаща техника за паралелно глобално търсене, което симулира действието на генетичните оператори в природата [107]. ГА са метод за стохастично търсене, проектиран да намира оптимални решения, използвайки минимална информация, която направлява търсенето на решение на зададен проблем. Генетичните алгоритми се характеризират с използването на популация от множество структури (индивиди), които извършват едновременно търсене в множество различни области от пространството на задачата. В индивидите са кодирани предполагаеми решения, които се обработват на конкурентен принцип с прилагане към тях на стохастични оператори с цел да се намери задоволително решение. Генетичният алгоритъм процедира в итеративен режим, като генерира нова популация от индивиди от тази на предната итерация. Оценъчната функция (още „фитнес функция“, *fitness function*) присвоява стойност на всеки индивид, който има отношение към задачата, която бива решавана. Каноничният алгоритъм прилага стохастични оператори като селекция, кръстосване (кросоувър) и мутация над популацията, с цел да се изчисли цяло ново

поколение от индивиди. Критерият за спиране на итеративния процес обикновено се формулира по някой от следните начини:

- достигане на предварително програмиран брой итерации на алгоритъма, и/или,
- когато в рамките на фиксиран брой итерации не се наблюдава подобряване на резултата, и/или
- когато имаме предварително направена оценка за оптимума и сме достатъчно близо до нея (отклонение в рамките на 5%).

Тъй като алгоритъмът едновременно оценява множество точки в параметричното пространство, е по-вероятно алгоритъмът да схожда към глобално решение. За да функционира, ГА изисква единствено информация, отнасяща се към качеството на полученото решение от всяко параметрично множество. Тъй като ГА не изискват никаква специфична за решавания проблем информация, те са по-гъвкави и адаптируеми в сравнение с повечето методи за търсене.

До този момент, използвайки апарата на ОМ, са били разработени ОМ модела, свързани с изпълнението на ГА, публикувани в [120–125, 129, 132–134]. Например, ОМ моделите от [129–131, 135] описват различни варианти на процедура за търсене с ГА. Последователно се оценяват с модела няколко оценъчни функции, подреждат се индивидите според тях и се избира най-добрата функция съобразно специфичната област на стойностите на задачата.

Изборът на генетични оператори се описва в ОМ модела от [133]. Той може да проверява различни комбинации от дефинираните генетични оператори и да избира най-подходящите измежду тях. Разработеният ОМ модел описва начина на функциониране и резултатите от работата на генетичен алгоритъм и определя реда на прилагане на генетичните оператори в зависимост от разглеждания проблем.

Някои от вариантите на оператора селекция, описани в термини на ОМ, са публикувани в [121, 122], а на оператора кръстосване – в [125].

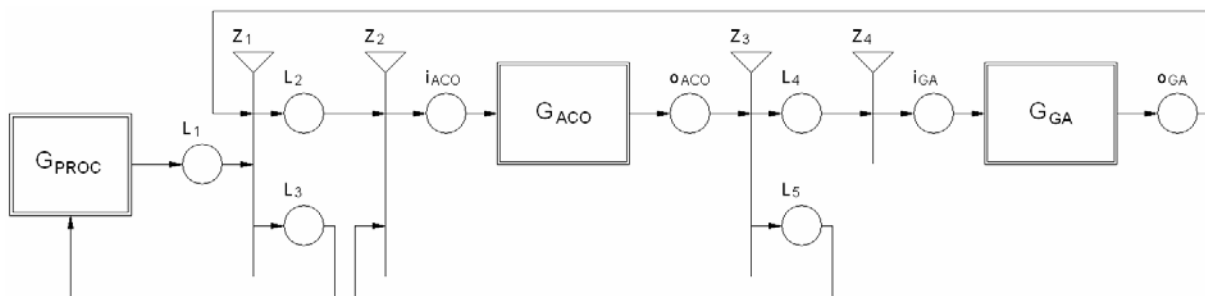
Тенденция през последните години е за подобряване на резултатите често да се използва комбиниране на няколко метода. Умело подобрите комбинации от метаевристики и други оптимизационни техники, наречени хибридни метаевристики (*hybrid metaheuristic*), могат да придадат повече ефективност и гъвкавост на решаването на мащабни и сложни задачи. Като цяло, хибридните метаевристични подходи могат да се класифицират или като колаборативни, или като интегративни комбинации. Колаборативните комбинации (*collaborative combinations*) се основават на обмена на информация между метаевристичния подход и друга техника за оптимизация, като двете се редуват последователно или работят паралелно. Интегративните комбинации (*integrative combinations*) използват друга оптимизационна техника (например, динамично програмиране) като подчинена част от метаевристиката. По посочените по-горе причини, хибридните метаевристики понастоящем се радват на растящ интерес от страна на оптимизационното общество.

Повечето автори комбинират метаевристиките с процедура за локално търсене или някой точен метод [94, 95]. Нашата идея е да комбинираме две метаевристики, АСО и ГА. АСО е конструктивен подход, който не се нуждае от начални решения.

Генетичните алгоритми са базиран на популации метод и в традиционните ГА началната популация се генерира по случаен начин. В тази случайно генерирана популация началните решения могат да бъдат много далеч от оптималните решения и могат да се нуждаят от много на брой итерации, за да се приблизят до тях. Следователно, идеята ни е да генерираме началните решения чрез АСО алгоритъма и после да ги използваме като начална популация за ГА. Така ГА ще започне с популация, която е по-близо до оптималното решение. Понякога след известен брой итерации, ГА стагнира и популацията спира да се подобрява. Така следващата ни идея е да върнем решенията на ГА към АСО алгоритъма и да обновим феромона съобразно, после да стартираме АСО алгоритъма с обновения феромон и така да генерираме нова популация за ГА. По този начин ще извършваме хибридизацията двупосочно: ГА се хибридизира с АСО алгоритъма, а АСО алгоритъмът се хибридизира с ГА. Можем да използваме всяка вариация на АСО алгоритъма и ГА, в зависимост от класа задачи, към които е приложим алгоритъмът.

3.2.2. Обобщена мрежа на хибриден метаевристичен алгоритъм

В [102] и [128] са описани АСО алгоритъмът и ГА в термините на ОМ (тук съответно означени с G_{ACO} и G_{GA}). Тук, следвайки публикацията ни [78], с тяхна помощ конструираме нова мрежа, описвайки хибридният алгоритъм. Процесът в конкретно решаваната оптимизационна задача се описва в рамките на мрежата G_{PROC} , вж. Фиг. 3.2, която се явява подобрене на Фиг. 5.1. от [78].



Фигура 3.2. ОМ с подмрежи G_{PROC} , G_{ACO} и G_{GA}

Всяка от мрежите G_{ACO} и G_{GA} има по една входна и една изходна позиции. Нека ги наричаме, съответно, i_{ACO} ; o_{ACO} ; i_{GA} ; o_{GA} . Нека ядрото α от мрежата G_{PROC} постъпва в позиция l_1 с начална характеристика:

„текущо описание на задачата
(граф на задачата, ограничителни условия, и т.н.)“

Първият преход има вида:

$$Z_1 = \langle \{l_1, o_{GA}\}, \{l_2, l_3\}, r_1 \rangle,$$

където:

$$r_1 = \frac{\quad}{l_1 \mid \begin{array}{cc} l_2 & l_3 \\ \hline true & false \end{array}},$$

$$o_{GA} \mid \begin{array}{cc} W_{GA,2} & W_{GA,3} \end{array}$$

и предикатите в индексиранията матрица имат вида:

- $W_{GA,2} = \text{„Необходима е нова итерация“}$,
- $W_{GA,3} = \neg W_{GA,2}$.

Ядрото α от позиция l_1 постъпва в позиция l_2 с новата характеристика:

*„инициализация на графа на задачата
(начално разположение и количество феромон)“*.

Когато вярностната стойност на предиката $W_{GA,2}$ е истина, α -ядрото от позиция o_{GA} постъпва в позиция l_2 с характеристика:

„обновяване на феромона, съобразно текущите решения“,

докато в позиция l_3 постъпва без да получи нова характеристика.

Вторият преход има вида:

$$Z_2 = \langle \{l_2, l_5\}, \{i_{ACO}\}, r_2 \rangle,$$

където:

$$r_2 = \frac{\quad}{l_2 \mid \begin{array}{c} i_{ACO} \\ true \end{array}}.$$

$$l_5 \mid \begin{array}{c} true \end{array}$$

Всяко едно от α -ядрата от позиции l_2 или l_5 постъпва в позиция i_{ACO} без да получи нова характеристика. То преминава през мрежата G_{ACO} и, напускайки я през позиция o_{ACO} , получава характеристика:

*„текущи решения на АСО алгоритъма
(генерирани популации)“*:

Третият преход има вида:

$$Z_3 = \langle \{o_{ACO}\}, \{l_4, l_5\}, r_3 \rangle,$$

където:

$$r_3 = \frac{\quad}{o_{ACO} \mid \begin{array}{cc} l_4 & l_5 \\ \hline W_{ACO,4} & W_{ACO,5} \end{array}},$$

където предикатите в индексиранията матрица имат вида:

- $W_{ACO,4} = \text{„Условието за край е изпълнено“}$,
- $W_{ACO,5} = \neg W_{ACO,4}$.

Когато вярностната стойност на предиката $W_{ACO,4}$ е истина, ядрото α постъпва в позиция l_4 с характеристика:

*„Представяне на текущите решения (популации) във вид,
подходящ за ГА“*,

докато в позиция l_5 постъпва без да получи нова характеристика.

Четвъртият преход има вида:

$$Z_4 = \langle \{l_4\}, \{i_{GA}\}, r_4 \rangle,$$

където:

$$r_4 = \frac{i_{GA}}{l_4 \mid true}$$

Ядрото α от позиция l_4 постъпва в позиция i_{GA} с характеристика „Текуща популация (решения) на ГА“.

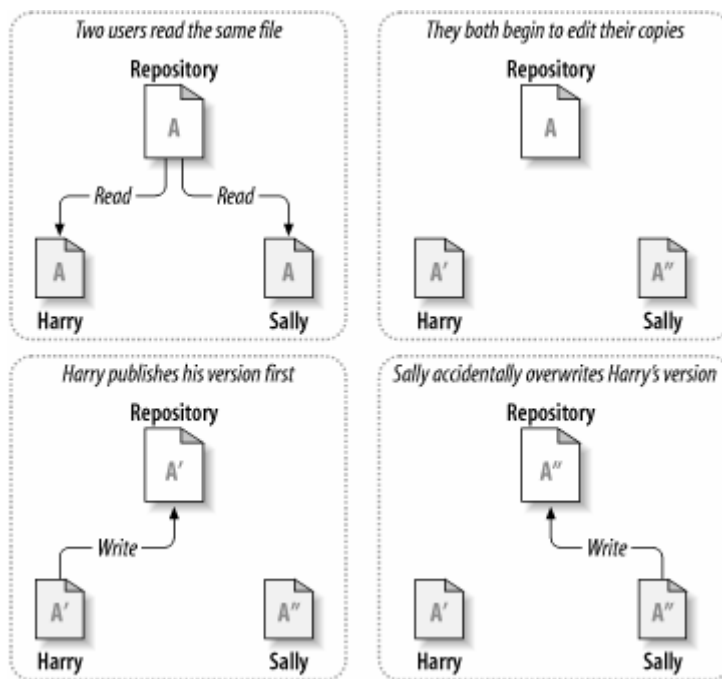
*
**

Предложеният тук нов хибриден метаевристичен подход комбинира две метаевристики, *Ant Colony Optimization* и генетични алгоритми, комбинирайки предимствата на двата съставни подхода. Подходът е приложен към решаване на проблем за параметрична идентификация на процес на култивиране на бактерията *E. coli*, и резултатът, приет за участие в предстоящата 9th International Conference on Large Scale Scientific Computation '2013, показва, че хибридният алгоритъм се нуждае от почти два пъти по-малко изчислително време и приблизително пет пъти по-малки популации от агенти, в сравнение със стандартните алгоритъм на мравките и генетичен алгоритъм.

3.3. Обобщеномрежови модели на механизми за разрешаване на редакционни конфликти

Контролът на версиите (*version control, revision control*) е управлението на промените в документи, програми и друга информация, съхранена под формата на компютърни файлове. Обичайно контрол е необходим и използван в многопотребителски среди като платформи за разработка на софтуер, уикита и системи за управление на съдържанието, където екип от хора могат да си сътрудничат в работата върху споделено множество от файлове. Всяка версия (ревизия, промяна, редакция) на файла от множеството се идентифицира по уникален начин и може да бъде достъпвана и сравнявана с другите версии на същия файл. Също така тази версия се асоциира с времева марка (*timestamp*), потребител (потребителско име или IP-адрес) извършил редакцията, както и други незадължителни данни. Версиите могат да бъдат съхранявани, възвръщани и (при някои файлови типове) сливани.

Системите с контрол на версиите (*version control systems, VCS*) са най-често самостоятелни приложения и по време на процеса на разработката и/или функционирането им един важен въпрос за решаване е конкурентният достъп до общото съдържание от страна на множество потребители, който лесно може да доведе до конфликти на редакции и инцидентна загуба на информация. Редакционните конфликти възникват, когато един потребител започне да редактира документ, но при опита да съхрани версията си се оказва, че друг потребител междуременно вече е променил документа, причинявайки ситуация на конфликт между подготвяната и извършената редакции. Такава ситуация обичайно трябва да бъде разрешена ръчно (Фиг. 3.3).



Фиг. 3.3. Ситуация на конфликт на редакции (Източник: [85]).

3.3.1. Механизми за разрешаване на конфликти на редакции

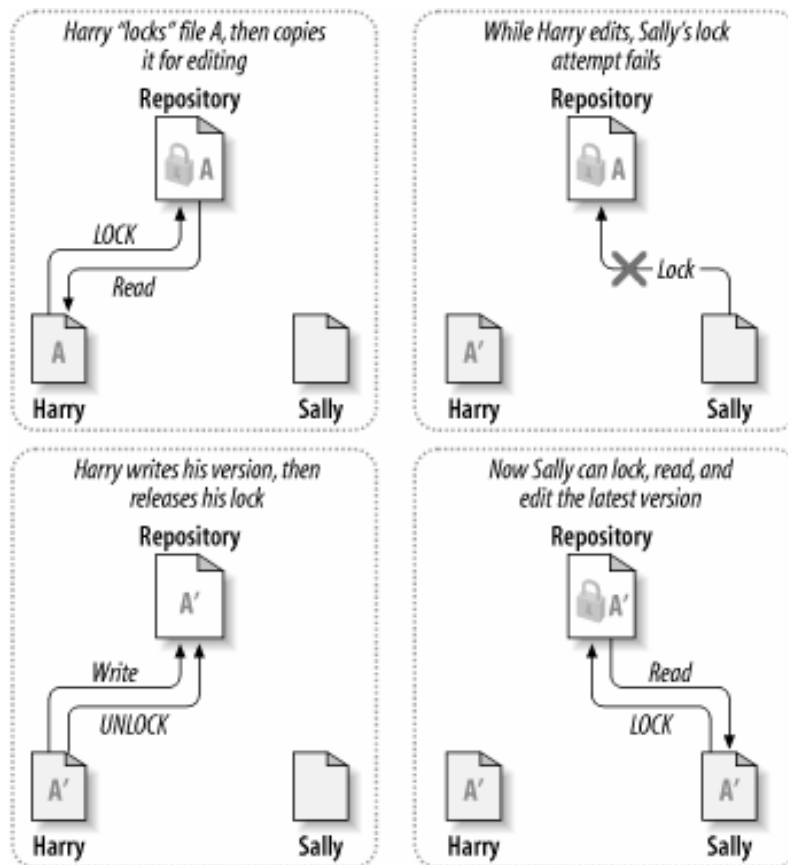
Проблемът с конкурентния достъп за редактиране (*write access*) е фундаментален за всички системи с контрол на версиите и споделени файлови хранилища. С известни вариации, съществуват два генерални подхода за разрешаването на този проблем:

- „Заклучи–Промени–Отключи“ (*Lock–Modify–Unlock, LMU*), и
- „Копирай–Промени–Слей“ (*Copy–Modify–Merge, CMM*).

Механизмът LMU е линейно ориентиран и смятан за песимистичен, докато подходът CMM е граф-ориентиран и смятан за оптимистичен. Всеки от двата подхода има своите предимства и недостатъци и уместността на избора между тях зависи от няколко фактора като типа файлове в хранилището (бинарните файлове не могат да бъдат контекстуално сливани, в сравнение с текстови файлове), броя файлове в хранилището, размера на допринасящата потребителска общност, средното време за извършване на редакция, средният обем извършени промени в байтове, и регистрираната честота на възникващи редакционни конфликти при тези условия.

Заклучи–Промени–Отключи

Подходът LMU (Фиг. 3.4.) е прост и популярен подход за предотвратяване на проблеми с конкурентния достъп. Състои се в заключване на файла, така че в един момент от време само един потребител да има достъп за редактиране до него.



Фигура 3.4. Схема на „Заклучи–Промени–Отключи“ (Източник: [85]).

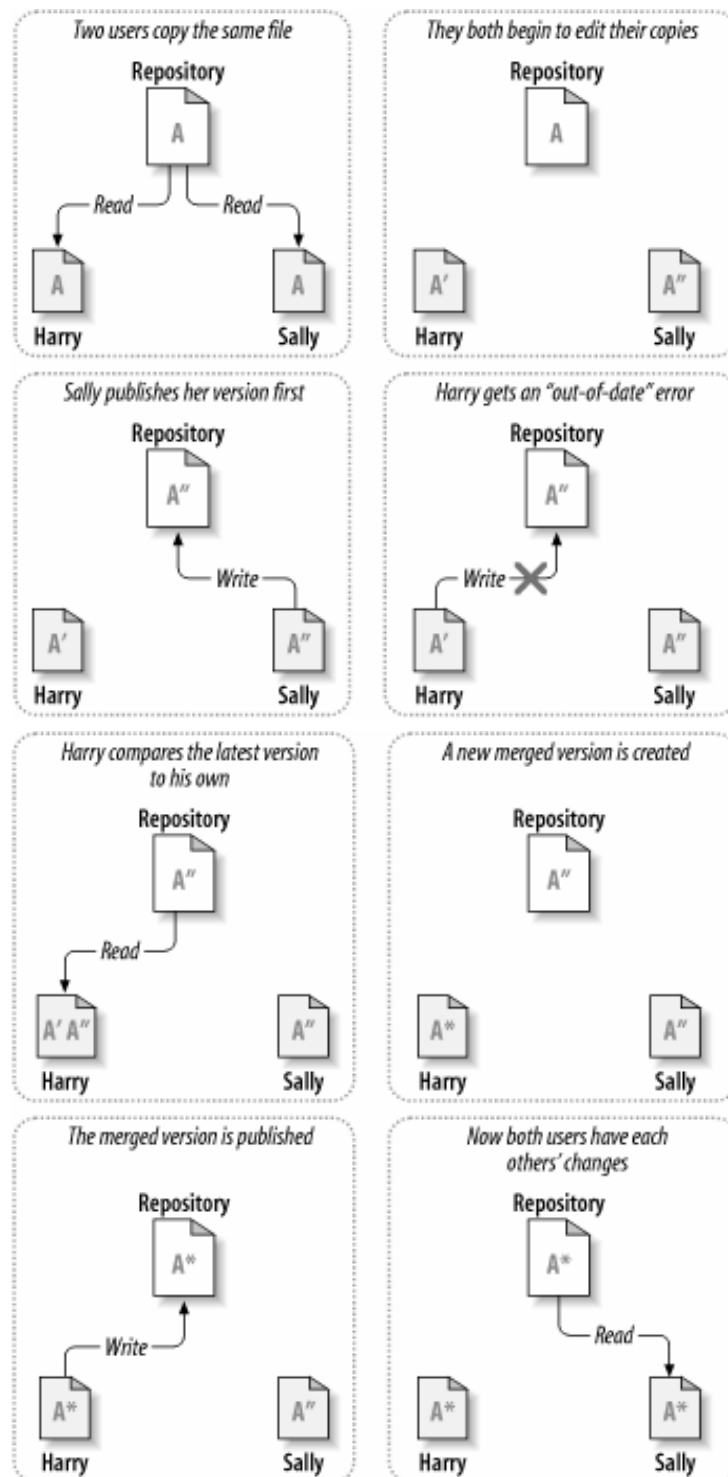
Отварянето на файл за редактиране се нарича „чек-аут“ (*check out*) и веднъж извършено, то предотвратява възможността други потребители също да променят файла, а им дава само достъп за четене (*read access*), докато редакторът не направи „чек-ин“ (*check-in*) на актуализираната версия и отключи файла, или не отмени „чек-аутът“.

Заедно с ползата от лесната реализация на този подход, той има и недостатъци. Той може да се окаже рестриктивен и да доведе до загуба на време и производителност. Понякога първият редактор забравя, че е заключил файл и така може ненужно да възпрепятства други потребители да нанасят своите редакции. Такава нежелана ситуация може да изиска намеса на администратор на системата, който да отключи файла. Освен всичко, заключването на файл може да доведе до т.нар. „фалшиво чувство на сигурност“ в случаи, когато различни потребители заключват и редактират съдържанието на различни документи, зависещи един от друг, и промените в тях са семантично несъвместими.

Копирай–Промени–Слей

В алтернативния подход СММ (Фиг. 3.5) всеки потребител има равнопоставен достъп до един и същ файл от хранилището и при отваряне за редактиране системата създава за този потребител локално работно копие. Така потребителите имат възможност да работят едновременно и независимо един от друг, модифицирайки своите локални копия на документа. Накрая системата се опитва да слее в една окончателна версия отделните версии, като предотврати евентуал-

ни конфликти. Ако системата не е способна автоматично да обедини версиите, тя оставя на потребителите ръчно да се справят с това.



Фигура 3.5. Схема на „Копирай–Промени–Слей“ (Източник: [85]).

Например, в *MediaWiki* (софтуера, с който работи *Wikipedia*), ако на страницата са обособени няколко раздела и различни потребители едновременно редактират различни раздели, системата може да слива редакциите без да причинява конфликт. Ако обаче областите на редактиране се припокриват, например

единият потребител редактира цялата страница, а другият – само негов раздел, или и двамата потребители редактират един и същ раздел, то тогава обичайно настъпва конфликт на редакциите и е необходимо ръчно предотвратяване на конфликта.

При един вариант на този подход, реализиран в някои уики дистрибуции (например *DocuWiki*), се извежда предупредително съобщение, ако даден потребител отвори за редактиране страница, която вече е била отворена за редактиране от друг. Това предупреждение има само превантивна, не ограничителна, цел и в резултат отново е възможно настъпване на редакционен конфликт.

Механизмът СММ може да звучи малко хаотично, но на практика често той преминава съвсем гладко. Потребителите могат да работят паралелно, без да се изчакват един друг и количеството време, което отнема, за да се разреши конфликтът, е много по-кратко отколкото времето, което се губи, докато системата стои заключена, както е при подхода LMU.

3.3.2. Обобщеномрежов модел на подхода LMU

За начало ще бъде направена бележка за означенията, които ще бъдат използвани тук и във втората част от изследването.

Двата ОМ модела съдържа статични преходи, означени с Т и индекс L или С (съответно за моделите LMU или СММ) и номерирани с поредния номер на прехода отляво надясно. Всички позиции, с изключение на DB и ID са маркирани със съответните действия, които се случват в тях, и отново индексирани с L или С. Двата модела неминуемо съдържат общи позиции (Open, Edit, Cancel, Wait, Save, Close), но вторият съдържа и няколко допълнителни позиции (Done, Confl, Approve, Merge). Позициите DB (съответстваща на базата данни на хранилището) и ID (генератор на идентификатори на поредни версии) са оставени без индекс, тъй като в двата модела между тях няма семантични разлики.

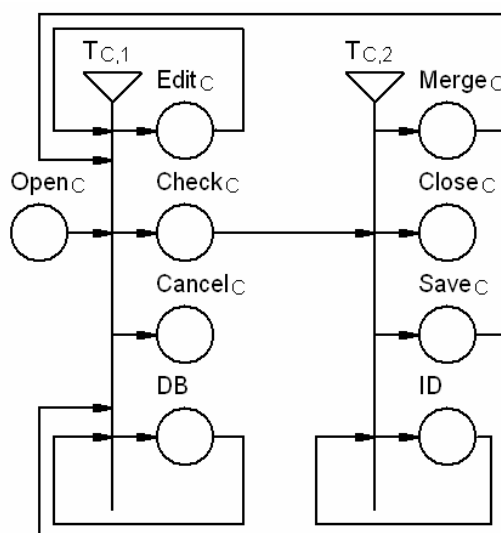
И двата ОМ модела съдържат три типа ядра, които подлежат на разцепване и сливане, и тези ядра са носители на динамичната природа на моделите:

- ε -ядро, което отговаря на редакторите (потребителите), които осъществяват достъп до даден файл. Това ядро отначало влиза в мрежата през позицията с етикет Open.
- δ -ядро, което стои само в позиция DB и представлява базата данни с файлове в хранилището. Отделните инстанции на файлове са ядра, означени с φ , които веднъж щом се отделят от разделящото се ядро δ , могат да се придвижват в останалата част от мрежата.
- ι -ядро, стоящо само в позиция ID и представляващо генератора на идентификатори на версии. Отделните идентификатори на файлови версии са естествени числа, които в ОМ модела са представени чрез ядра, означени с ν . Веднъж щом се отделят от разделящото се ядро ι , те могат да се придвижват из останалата част от мрежата.

Логиката на ОМ модела се съдържа в индексирани матрици на преходите, които са означени с М и индексирани с индексите на съответните им

преходи. Всички предикати в една индексирана матрица, които се различават от „истина“ и „лъжа“, са означени с Р и индексирани с типа на модела, поредния номер на прехода и поредния номер на предиката.

ОМ моделът на подхода LMU (Фиг. 3.6) е по-простият от двата. Той съдържа два прехода, девет позиции и гореописаните три вида ядра.



Фигура 3.6. ОМ модел на „Заклучи–Промени–Отключи“

Всяко ε -ядро влиза в мрежата през позиция Open_C с начална характеристика:

*„Потребителско име (IP-адрес) на потребителя. Поискан файл.
Момент от време на достъп до файла T_{Open} “.*

Едно ядро δ постоянно цикли в позиция DB с характеристика:

*„Списък от файловете в базата данни, асоциирани със състоянието им
Заклучен/Отключен и идентификатора на текущата версия“.*

Първоначално всички файлове в базата данни са в състояние „отключен“, но по време на функционирането на модела всички файлове, които бъдат отворени за редактиране променят състоянието си на „заклучен“. Например:

Таблица 3.1.

Файл	Състояние	ID на текущата версия
file ₁	unlocked	id ₁
...
file _k	locked	id _k
...
file _n	unlocked	id _n

Докато ядро δ представлява цялата база от данни (хранилище), отделните екземпляри на файлове от хранилището ще бъдат представяни с φ -ядра.

Ядро i перманентно цикли в позиция ID с характеристика:

„Брой на редакциите на файла, внесени в системата, считано от даден момент“.

С други думи, това ядро играе ролята на генератор на последователни естествени числа, които идентифицират версиите на всички файлове поддържани в хранилището. По подразбиране, този номер е 0, но ако дадена симулация трябва да започне от по-късен момент от време от функционирането на системата, съответният брой файлови версии, направени до този момент, ще бъдат вложени като начална характеристика на това ядро.

Докато ядрото i представлява генератор на идентификатори, то отделните идентификатори на версии ще бъдат представени с v -ядра.

Първият преход $T_{L,1}$ има четири входни и четири изходни позиции

$T_{L,1} = \langle \{Wait_L, Open_L, Save_L, DB\}, \{Wait_L, Cancel_L, Edit_L, DB\}, M_{L,1} \rangle$
свързани посредством предикатите от следната индексирана матрица:

$M_{L,1} =$	$Wait_L$	$Cancel_L$	$Edit_L$	DB
$Wait_L$	$P_{L,1,3}$	$P_{L,1,4}$	$P_{L,1,1}$	$false$
$Open_L$	$P_{L,1,2}$	$false$	$P_{L,1,1}$	$false$
$Save_L$	$false$	$false$	$false$	$true$
DB	$P_{L,1,2}$	$false$	$P_{L,1,1}$	$true$

където предикатите $P_{L,1,1}, \dots, P_{L,1,4}$ имат следните интерпретации.

- $P_{L,1,1} =$ „Състоянието на поискания файл от текущия потребител е „Отключен“.
- $P_{L,1,2} = \neg P_{L,1,1} =$ „Състоянието на поискания файл от текущия потребител е „Заклучен“.
- $P_{L,1,3} =$ „Потребителят чака файла да бъде отключен.“
- $P_{L,1,4} =$ „Потребителят престава да чака файла да бъде отключен.“

Когато вярностната стойност на предиката $P_{L,1,1}$ е „истина“, ядро ε напуска позиция $Open$ или $Wait$ и влиза в позиция $Edit$. Едновременно с това ядро δ в позиция DB се разцепва на едноименното ядро δ и едно ново ядро φ , което представлява файл от хранилището, поискан за редактиране от текущия потребител. Ядрото φ влиза в позиция $Edit$ с текуща характеристика:

„Име и съдържание на поискания файл.
Други характеристики (незадължителни)“.

Като примери за незадължителни характеристики могат да се посочат някои атрибути на последната версия на файла, например идентификатор, под който тя уникално се идентифицира в базата данни, автор, времева марка (*timestamp*), размер на файла, резюме на последната редакция, идентификатор на предшестващата версия, и т.н., които данни обичайно автоматично се асоциират с идентификатора на текущата версия.

Ядрото δ прави един цикъл и се връща в позиция DB и актуализира характеристиката си, като променя в характеристичната таблица състоянието на файла, съответстващ на φ , от „отключен“ на „заклучен“. За отбелязване е, че

състоянието на файла е важна характеристика, която формира разликата между предикатите $P_{L,1,1}$ и $P_{L,1,2}$.

Когато ядрото ε от позиция $Open_L$ и ядрото φ от позиция DB влязат в позиция $Edit_L$, те се сливат в едно ядро под името ε , което получава характеристиката:

*„Потребителско име. Име на файла. Момент T_{Lock} на заключване на файла.
Състояние: Заключен“*

Предикатът $P_{L,1,2}$ става „истина“, когато едно ε ядро влезе в мрежата през позиция $Open_L$, изисквайки файл от хранилището, но съответният запис от характеристикната таблица на δ -ядрото в позиция DB гласи, че поисканият файл е заключен за редактиране. Следователно, δ -ядрото остава в позиция DB без да се разцепва на нови ядра, докато ядрото ε преминава в позиция $Wait_L$ с характеристика:

„Момент T_{Wait} , когато потребителят е започнал да чака файла да бъде отключен.“

Предикатът $P_{L,1,3}$ е „истина“ дотогава, докогато редакторът чака файла да бъде отключен, т.е. ядрото ε цикли в позиция $Wait_L$ без да получава нови характеристики.

Предикатът $P_{L,1,4}$ става „истина“ когато редакторът реши да се откаже от понататъшно чакане файлът да бъде отключен, като така ядрото ε от позиция $Wait_L$ преминава в позиция $Cancel_L$ с новата характеристика:

„Текущ момент от време $T_{Cancel} \cdot Time_{Wait}$ “

където $Time_{Wait}$ е времето, прекарано напразно в чакане файлът да бъде отключен, и $Time_{Wait} = T_{Cancel} - T_{Wait}$.

Веднъж щом ядрото φ влезе в позиция $Save_L$, на следващия момент от функционирането на ОМ то ще премине безусловно в позиция DB (откъдето и съответната клетка от индексирания матрица $M_{L,1}$ е „истина“, вместо някакъв предикат) с характеристика, която ще бъде коментирана по-късно.

Вторият преход $T_{L,2}$ има две входни и три изходни позиции

$$T_{L,2} = \langle \{ID, Edit_L\}, \{ID, Close_L, Save_L\}, M_{L,2} \rangle$$

свързани посредством предикатите от следната индексирания матрица:

	ID	Close _L	Save _L
$M_{L,2} =$	ID	$P_{L,2,2}$	$P_{L,2,2}$
	Edit _L	$P_{L,2,1}$	$P_{L,2,2}$

където предикатите $P_{L,2,1}$ и $P_{L,2,s}$ имат следните интерпретации:

- $P_{L,2,1} =$ „Потребителят иска да приключи работата си по файла без да запази редакцията си“.
- $P_{L,2,2} =$ „Потребителят иска да приключи работата си по файла и да запази редакцията си“.

Когато едно ядро ε (слято, както беше описано по-горе) влезе в позиция $Edit_L$, то ще остане в тази позиция дотогава, докогато редакторът работи по файла. Когато потребителят приключи работата си по файла, ядрото ще напусне

позиция $Edit_L$ защото или предикатът $P_{L,2,1}$ или предикатът $P_{L,2,2}$ ще са станали „истина“, в зависимост от конкретния сценарий: потребителят може или да реши да затвори файла без да запазва редакцията си, или може да избере да съхрани промените (което и обичайно се случва).

В случай на реална симулация на модела, времето за редактиране може да бъде реализирано, като се избере случайна стойност от област на правдоподобни времетраения и като се присвои тази стойност на ядро ε в позиция $Edit_L$ като втори атрибут на характеристиката. Така когато този времеви интервал изтече, регистриран по времевата скала на модела, точното време, през което файлът е бил в състояние „заклучен“, може да бъде изчислено.

Когато предикат $P_{L,2,1}$ е „истина“, т.е. редакторът избере да напусне без да запазва промените по файла, ядрото ε ще се разцепи на две ядра: същото ядро ε и ново ядро φ . Новополученото ядро ε ще премине към позиция $Close_L$ с характеристиката:

„Момент T_{Close} на затваряне на файла. $Time_{Close}$ “,

където $Time_{Close} = T_{Close} - T_{Lock}$ е продължителността от време, в което файлът е бил заключен, въпреки че не са настъпвали в него редакционни промени.

Новополученото ядро φ ще влезе в позиция $Save_L$ с характеристика:

„Съдържание на файла: Непроменено. Идентификатор на версията:
Непроменен. Състояние на файла: Отключен“.

Когато предикат $P_{L,2,2}$ е „истина“, т.е. потребителят избере да съхрани нанесените промени, се случват три действия. Първо, генераторът на идентификатори на поредни версии ще подаде съответния идентификатор (естествено число), който да бъде присвоен на новосъхранената файлова версия, и после ще увеличи това число с 1 (така че следващата версия на файл от хранилището да получи различен идентификатор). За тази цел ядрото i (с текуща характеристика естественото число $ID_{Current}$), стоящо в позиция ID, ще се разцепи на две ядра: новото ядро v и същото ядро i . Ядрото v ще премине в позиция $Save_L$ с характеристика:

„Текуща стойност на ядрото i : $ID_{Current}$ “.

Ядрото i , получено след разцепването, ще постъпи в позиция ID с характеристика:

„ $ID_{Current} + 1$ “.

От друга страна, ε -ядрото от позиция $Edit_L$ отново ще се разцепи на две ядра: същото ядро ε и ново ядро φ (което отговаря на означеното преди ядро φ дошло от позиция DB). Новополученото ядро ε ще премине в позиция $Close_L$ с характеристиката:

„Момент T_{Save} на съхраняване на файла. $Time_{Save}$ “

където

$$Time_{Save} = T_{Save} - T_{Lock}$$

е продължителността на времето за редактиране на файла, през което той е бил в състояние „заклучен“ за други потребители. (Разликата между $Time_{Close}$ и $Time_{Save}$ е само в тяхната семантика.)

Новополученото ядро φ влиза в позиция $Save_L$, където се слива с ядро ν отново под името φ и получава характеристика:

„Съдържание на файла: Променено. Идентификатор на версията: $ID_{Current}$.
Състояние на файла: Отключен“.

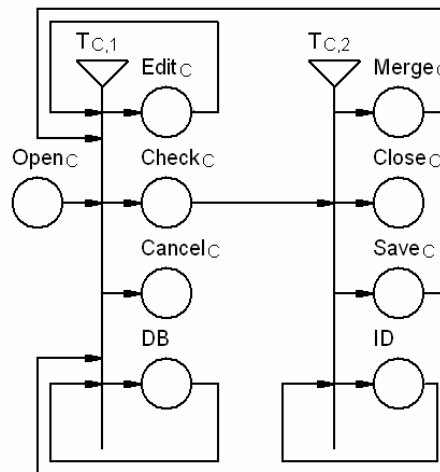
Най-сетне ще се върнем на характеристиката на ядрото φ , получено при безусловния си преход от позиция $Save_L$ обратно в позиция DB. На практика:

- то отразява факта, че файлът бива отключен, без значение бил ли е редактиран или не; и
- в случай на съхранени промени, то добавя нова версия към историята на файла и я прави текущо видима по подразбиране за всички потребителите на системата.

Така ядрото φ от позиция $Save_L$ постъпва в позиция DB, където се слива с наличното там ядро δ с характеристики, зависещи от конкретните стойности на характеристиките, носени от φ ядрото на предходната стъпка от функционирането на модела.

3.3.3. Обобщеномрежов модел на подхода СММ

ОМ моделът на подхода „Копирай–Промени–Слей“ се състои от два прехода и девет позиции (Фиг. 3.7). И той, и първият модел съдържат общи позиции (Open, Edit, Cancel, Save, Close), но вторият модел съдържа и две допълнителни позиции (Conflict и Merge).



Фигура 3.7. ОМ модел на подхода „Копирай–Промени–Слей“

Подобно на ОМ модела на LMU, и моделът тук също оперира с описаните три типа ядра, които могат да се разцепват и сливат.

- ε -ядро, което отговаря на редакторите (потребителите), които достъпват (отварят) даден файл. Това ядро отначало влиза в мрежата през позицията с етикет Open.
- δ -ядро, което стои само в позиция DB и представлява базата данни с файлове в хранилището. Отделните инстанции на файлове са ядра,

означени с φ , които веднъж щом се отделят от разделящото се ядро δ , могат да се придвижват в останалата част от мрежата.

- i -ядро, стоящо само в позиция ID и представляващо генератора на идентификатори на версии. Отделните идентификатори на файлови версии са естествени числа, които в OM модела са представени чрез ядра, означени с v . Веднъж щом се отделят от разделящото се ядро i , те могат да се придвижват из останалата част от мрежата.

Всяко ε -ядро влиза в мрежата през позиция OpenC с начална характеристика:

„Потребителско име (IP-адрес) на потребителя. Поискан файл. Времева марка (timestamp) на момента от време на достъп до файла“.

Едно ядро δ постоянно стои в позиция DB с характеристика:

„Списък от файловете в базата данни, съдържащ:

- (1) Идентификатор на файла. (2) Идентификатор и времева марка на текущата версия на файла. (3) Списък на копията на файла, асоциирани с идентификатор на копието, идентификатор на потребителя и идентификатор и timestamp на версията на файла, която е била текуща към момента на генериране на копието“.*

Таблица 3.2.

Файл (1)	Текуща версия ID и timestamp (2)	Списък на копията на файла, потребители, идентификатори и timestamps (3)
file ₁	id ₁ , ts ₁	[copy ₁ , user _{1,1} , id _{1,1} , ts _{1,1}], [copy ₂ , user _{1,2} , id _{1,2} , ts _{1,2}], ...
...
file _k	id _k , ts _k	[copy ₁ , user _{k,1} , id _{k,1} , ts _{k,1}], [copy ₂ , user _{k,2} , id _{k,2} , ts _{k,2}], ...
...
file _n	id _n , ts _n	[copy ₁ , user _{n,1} , id _{n,1} , ts _{n,1}], [copy ₂ , user _{n,2} , id _{n,2} , ts _{n,2}], ...

Например, ако няколко потребители са започнали да редактират копия на един и същ файл, и междувременно един от тези потребители запази първи своята редакция, това ще промени текущата версия, т.е. идентификатора и времевата му марка в (2), но няма да окаже влияние върху идентификатора на версията и времевата марка на оставалите конкурентни редактори в (3). В момента, в който някой от тях се опита да запази своята редакция, разликата между идентификатора на текущата (вече променена) файлова версия и идентификатора на версията, съдържаща тяхното копие, ще отключи механизма за разрешаване на конфликт на системата и той или ще слее двете версии автоматично (ако е способен), или ще изведе на втория потребител съобщение, че е необходимо да разреши конфликта ръчно.

Отново ядрото δ представлява цялата база данни, отделните инстанции на файлове от базата данни ще бъдат представени като φ -ядра. Ядрото i и в този модел цикли постоянно в позиция ID с характеристика:

„Брой редакции на файла, внесени в системата, считано от даден момент“.

Докато ядрото i представлява генератор на идентификатори, то отделните идентификатори на версии ще бъдат и тук представени с v -ядра.

Ще опишем двата прехода в детайли. Първият преход $T_{C,1}$ има следния вид:

$$T_{C,1} = \langle \{Edit_C, Merge_C, Open_C, Save_C, DB\}, \{Edit_C, Check_C, Cancel_C, DB\}, M_{C,1} \rangle$$

където:

$M_{C,1} =$	$Edit_C$	$Check_C$	$Cancel_C$	DB
$Edit_C$	$P_{C,1,2}$	$P_{C,1,4}$	$P_{C,1,3}$	$P_{C,1,3}$
$Merge_C$	$true$	$false$	$false$	$false$
$Open_C$	$P_{C,1,1}$	$false$	$false$	$false$
$Save_C$	$false$	$false$	$false$	$true$
DB	$P_{C,1,1}$	$P_{C,1,5}$	$false$	$true$

и предикатите в индексиранията матрица имат вида:

- $P_{C,1,1} =$ „Потребителят получава копие на файла от хранилището, отворен в режим на редактиране.“
- $P_{C,1,2} =$ „Потребителят продължава да редактира файла.“
- $P_{C,1,3} =$ „Потребителят спира да редактира и не иска да запази редакцията си.“
- $P_{C,1,4} =$ „Потребителят спира да редактира и иска да запази редакцията си.“
- $P_{C,1,5} =$ „Засечен е конфликт на редакции.“

Когато предикатът $P_{C,1,1}$ е истина, ядрото ε напуска позицията $Open_C$ и влиза в позиция $Edit_C$. Същевременно, ядро δ в позиция DB се разцепва на същото ядро δ и едно ново ядро φ , което представлява копие на файла от хранилището, поискан от потребителя за редактиране. Ядрото φ влиза в позиция $Edit_C$ с текуща характеристика:

„{1} Файл. {2} Идентификатор, времева марка и съдържание на текущата версия на файла“.

Ядро δ прави един цикъл и се връща в позиция DB , където обновява характеристичната таблица с добавянето на нов запис в третата колона за новосъздаденото копие на поискания файл φ , който запис се означава с ε и се характеризира с идентификатора на текущата версия на файла, моментът от време (времева марка) на генериране на копието и потребител, поискал копието.

Така когато ε -ядрото от позиция $Open_C$ и φ -ядрото от позиция DB влязат в позиция $Edit_C$, те се сливат като едно ε -ядро, което получава характеристика:

„{1} Потребител. Идентификатор на файл. {2} Идентификатор на версията, времева марка и съдържание на най-скорошната версия на файла. {3} Времева марка и съдържание на копието на файла.“

Разбира се, в първия момент преди потребителят да започне да нанася промени, съдържанието на копието ще бъде идентично със съдържанието на най-скорошната версия на файла. Техните времеви марки, обаче, ще бъдат различни и копието на файла няма да се асоциира с идентификатора на версията.

Когато предикатът $P_{C,1,2}$ получи вярностна стойност „истина“, то това означава, че потребителят продължава да редактира по извлеченото при него персонално копие на файла, т.е. ε -ядрото продължава да цикли в позиция $Edit_C$. Отчитането на момента от време, когато е генерирано копието на файла, е от значение, тъй като това позволява да се сравни локалното копие с онова, което текущо се пази като последно в базата данни. На базата на времевата марка системата може да дава незабавна нотификация на потребителите за промените във файла, по който те работят. На всеки цикъл в позиция $Edit_C$, ядрото ε обновява своята характеристика, особено в последната си част „Съдържание на копието на файла“.

Когато предикатът $P_{C,1,2}$ получи вярностна стойност „лъжа“, то това означава, че може да стане истина или предикатът $P_{C,1,3}$ или $P_{C,1,4}$: потребителят решава да прекрати редактирането на файла. Кой от двата предиката ще стане истина зависи от това дали потребителят иска да съхрани редакцията си или не. Ако потребителят иска да откаже редактирането без да запази промените си, става истина предикатът $P_{C,1,3}$, а ядрото ε от позиция $Edit_C$ се разцепва на две: същото ядро ε и началното ядро φ . Ядрото ε влиза в позиция $Cancel_C$ без никаква характеристика, докато ядрото φ влиза в позиция DB и се слива там с ядрото δ , обновявайки характеристичната таблица с премахването в (3) на съответния запис за копието от съответния файл, поискан от съответния потребител. Веднъж премахнат, записът няма да повлияе на преживяването на останалите редактори с предизвикване на евентуални конфликти на редакции.

Най-сетне потребителят може да реши да съхрани редакцията си, което означава предикат $P_{C,1,4}$ да стане истина, което става причина ε -ядрото от позиция $Edit_C$ да се придвижи до позиция $Conflict_C$, запазвайки последната си характеристика. Там е от значение каква ще бъде стойността на предиката $P_{C,1,5}$. Предикатът $P_{C,1,5}$ съответства на проверка на системата дали е настъпил конфликт на редакции за дискутираното копие на файла, или не. Тази проверка лесно се прави в позиция DB , предвид факта, че характеристичната таблица на ядрото δ поддържа информация не само за идентификаторите на файловете версии, но и за времевите марки на текущата файлове версия и на всяко от произведените копия на файла, когато той е бил отварян за редактиране. Когато нова версия на файла успешно се съхрани в базата данни, времевата марка на текущата (обновена) версия на файла също се обновява, и ако вече са били генерирани копия на файла, техните времеви марки са по-малки като стойност от стойността на времевата марка на обновената версия на файла, следователно предикатът $P_{C,1,5}$ става истина. Това води до разцепването на δ -ядрото в позиция DB на две ядра: оригиналното δ -ядро, носител на пълната информация за системната база данни, и едно ново ядро φ_{cur} , което преминава в позиция $Conflict_C$ с характеристиката:

*„Идентификатор на текущата версия, времева марка
и съдържание на файла“.*

Там ядрото φ_{curr} се слива с ядрото ε , което е пристигнало по силата на стойността „истина“ на предиката $P_{C,1,4}$. Така, позиция $Conflict_C$ съответства на момента, в който системата засича конфликт на редакции и изисква от потребителя ръчно да го разреши.

Някои системи за контрол на версиите са реализирани по начин, че позволяват автоматично разрешаване на настъпили редакционни конфликти. Това изследване няма за цел да описва особеностите на тази процедура и за нашите нужди само изход във вида „Да / Не“ е достатъчен. При все това, изготвянето на прецизен механизъм за автоматично разрешаване на конфликти е интересна бъдеща стъпка на разширяване на ОМ модела. Един възможен подход е сравняването на областите от документа, където потребителите са редактирали конкурентно. Ако тези области не се пресичат, например единият потребител е редактирал в началото на документа, а другият – в края, тогава системата е способна да разреши конфликта автоматически, без дори да има нужда да известява на съхранилия втори потребител, че е възникнал конфликт с първия. Това е чисто технически подход към разрешаването на редакционен конфликт и в общия случай не отчита по никакъв начин *семантиката* на всяка от внесените промени.

И така в позиция $Conflict_C$ новополученото след сливането ядро ε получава комбинираната характеристика:

- „{1} *Потребител. Идентификатор на файл.*
- {2} *Идентификатор на версията, времева марка и съдържание на файловата версия, която е била текуща към момента на генерирането на копие.*
- {3} *Времева марка и съдържание на файловото копие.*
- {4} *Идентификатор на версията, времева марка и съдържание на текущата файлове версия.“*

На практика, това съответства на момента от време, когато потребителят е получил известие за конфликта на редакции, в зависимост и от потребителския интерфейс на конкретната система за контрол на версиите. Например, екранът за известяване на потребителя за настъпил конфликт на редакции в софтуера *MediaWiki* предлага на потребителя три части от информацията: съдържанието на текущата файлова версия {4}, съдържанието на неговото копие на файла със съответните редакции {3} и разликата между двете версии. Както ще видим по-долу, потребителят може или да одобри междуременно съхранената от друг потребител версия, или може да пожелае да продължи да редактира, сливайки своята версия с последно съхранената.

Позиция $Conflict_C$ е единствената позиция в мрежата, която свързва двата прехода в модела, т.е. се явява изходна позиция за преход $T_{C,1}$ и входна за втория преход $T_{C,2}$, който се представя по следния начин:

$$T_{C,2} = \langle \{Conflict_C, ID\}, \{Merge_C, Close_C, Save_C, ID\}, M_{C,2} \rangle$$

където:

$M_{C,2} =$	$Merge_C$	$Close_C$	$Save_C$	ID
$Conflict_C$	$P_{C,2,1}$	$P_{C,2,2}$	$P_{C,2,2}$	<i>false</i>
ID	<i>false</i>	<i>false</i>	$P_{C,2,2}$	<i>true</i>

и предикатите в индексиранията матрица имат вида:

- $P_{C,2,1} = \text{„Потребителят продължава да редактира.“}$
- $P_{C,2,2} = \neg P_{C,2,1}$.

Ако предикатът $P_{C,2,1}$ е истина, това отговаря на сценария, при който потребителят, след като е бил известен за конфликта на редакции, е избрал да продължи да редактира. Конкретните причини за подобно решение могат да бъдат различни, включващи или не качествена оценка на въведените между-временно от друг потребител промени. Например, потребителят не е съгласен с междуременно внесените редакции и иска частично или изцяло да ги отмени; или потребителят е съгласен с направените промени, но планираните от него редакции са в по-голям обем, по-сложни или по някакъв друг начин по-ценни. Каквато и да е причината, резултатът от това предикатът $P_{C,2,1}$ да е истина е, че ε -ядрото от позиция Conflict_C преминава в позиция Merge_C , където, както знаем от първата матрица на предикатите $M_{C,1}$, то безусловно преминава в позиция Edit_C .

В обратния случай, ако предикат $P_{C,2,1}$ е лъжа, т.е. е истина предикатът $P_{C,2,2}$, това отговаря на ситуацията, когато системата позволява на потребителя да съхрани редакцията си, тъй като не е установен конфликт на редакции. Как точно това ще се случи в термините на ОМ, зависи от стойността на предиката $P_{C,1,5}$ от по-горе:

- Ако предикатът $P_{C,1,5}$ е лъжа, т.е. не е засечен конфликт на редакции, то пред потребителя няма технически пречки да съхрани своите промени по файла. Следователно ε -ядрото от позиция Conflict_C се разделя на две ядра: ядро ε , което представлява редактора, и ядро φ_{new} , което представлява новогенерираната версия на файла. По същото време i -ядрото в позиция ID също се разцепва на две: същото ядро i и ново ядро v , което представлява поредния идентификатор на файлова версия. Ядрото i продължава да цикли в позиция ID, като текущата му характеристика се увеличава с 1. Ядрото v преминава от позиция ID в позиция Save_C , където се слива с ядрото φ_{new} , което получава характеристиката:

„Потребител. Идентификатор на файл. Идентификатор на файлова версия. Времева марка. Редактирано съдържание на файла“,

където идентификаторът на версията е получен от характеристиката на v -ядрото, а времевата марка отговаря на текущия момент от време, когато новата версия на файла е съхранена. Нещо повече, както се вижда от индексиранията матрица $M_{C,1}$ на първия преход, от позиция Save_C винаги се преминава към позиция DB (стойност „истина“), с което тези данни се добавят като нов запис в историята на файла в базата данни. Самото ядро ε преминава от позиция Conflict_C към позиция Close_C с малко по-кратка характеристика:

*„Потребител. Идентификатор на файл.
Идентификатор на версията. Времева марка“*,

която на практика съответства на затварянето на сесията за редактиране и добавяне на нов запис към списъка на извършените от потребителя действия в системата (в случай, че системата поддържа такъв дневник).

- Ако предикатът $P_{C,1,5}$ беше истина, т.е. е бил засечен редакционен конфликт в системата, но потребителят въпреки това не иска да продължи да редактира, това отразява ситуацията, когато потребителят или се отказва от редактирането на този (по-късен) етап, или преценява, че одобрява междуременно съхранената версия и може да игнорира собствените си промени, нанесени в локалното копие на файла. В термините на ОМ, това означава, че ε -ядрото от позиция Conflict_C се разцепва на две ядра: ядро ε , което преминава в позиция Close_C без никаква характеристика и ядро φ_{new} , което преминава в позиция Save_C с характеристиката:

*„Премахване от базата данни на съответния запис
за текущото копие на файла, генерирано от този потребител“.*

Веднъж щом се премахне от характеристичната таблица, това файлово копие повече няма да влияе на останалите генерирани от други потребители локални копия на същия файл и няма да причинява при никого от тях конфликти на редакции. Както се вижда в матрицата на преходите на първия преход, преминаването от позиция Save_C към позиция DB е безусловно „истина“.

*
**

В заключение, може да се отбележи, че ясното онагледяване и описване на процесите чрез ОМ позволява прецизно и детайлно планиране на тестването на двата подхода (самото тестване се извършва с друг математически апарат). При наличие на пълнофункционален ОМ симулатор ще е възможна реализацията на двата модела в режим на паралелна симулация и функционалното сравнение между тях. Анализът на резултатите може да помогне на разработчиците на системи с контрол на версиите да направят верния избор, когато решават кой от двата подхода да внедрят в продуктите си. След допълнително доразвиване на тези относително прости ОМ модели посредством йерархични оператори [48], различни варианти и допълнителни опции на всеки от двата подхода могат да бъдат симулирани и тествани, и реализирани на практика, ако бъдат сметнати за уместни.

Глава 4. Уики платформа с приложение за обобщени мрежи

Тук се дискутира реализацията на уебсайт, изграден от автора с мрежовата технология уики, посветен на областите ИРМ и ОМ. Ще посочим как съществуващата технология уики (на английски *wiki*, от хавайското *wiki wiki*, „бързо“) е приложена на практика, за да задоволи един голям диапазон от потребности на тези две научни общности, които имат световен обхват (Австралия, Белгия, България, Великобритания, Германия, Гърция, Индия, Испания, Италия, Китай, Полша, Португалия, Р. Македония, САЩ, Словакия, Южна Корея и др.).

Сайтът <http://www.ifigenia.org> (накратко Ifigenia.org), е първият такъв тематично специализиран уебсайт и още от самото си създаване през октомври 2008 г. започва да предоставя следните възможности:

- изграждане на учебен корпус от контекстуално свързан понятиен апарат (по примера на онлайн енциклопедията Уикипедия),
- база данни от публикации,
- платформа за търсене на партньорства и провеждане на научни дискусии,
- среда за съвместна разработка на научни текстове, проектни предложения, софтуерна документация и други.

През април 2010 г., след направено разширение на ядрото на софтуера, сайтът започва да предлага и опцията за провеждане на онлайн тестове по ОМ, и то в непосредствена близост до учебното съдържание в рамките на уикито.

Пред специализирана аудитория сайтът вече е представян при стартирането си, [70], а по-късно опитът по изграждане на уики сайт с толкова разнообразни възможности е синтезиран и поднесен пред по-широка аудитория, [7], с надеждата да даде тласък за развитие и онлайн присъствие и на други научни общности със сходни потребности.

4.1. Технологията уики

Уики е технология за създаване на колекции от взаимосвързани уебстраници, проектирана така, че всеки да може да ги редактира, като използва само платформено независим браузър, като всички версии на страниците се съхраняват еднозначно в базата данни. Сайтовете, изградени на уики технологията (също наричани *уикита*), често се използват от различни потребителски общности за улесняване на онлайн базираната комуникация и сътрудничество и представляват алтернатива на други системи за управление на съдържание и знание.

Технологията дава на потребителите си множество гъвкави и дружелюбни инструменти за навигация и организиране на съдържанието, за взаимодействие помежду си, за проследяване на промените в реално време и в ретроспекция, и за извличане на информация по различни начини и критерии. Посредством настройването на параметрите на уики софтуера и добавянето на допълнителни разширения (*extensions*) към ядрото му може да се постигне персонализация на уикито по отношение на политиките за регистрация и достъп до съдържанието, потребителските групи и права, именните пространства, допустимите файлови формати, дизайна, и т.н.

Уорд Кънингам, създателят на първото по рода си уики, WikiWikiWeb, оригинално описва технологията като „най-простата онлайн база данни, която изобщо би могла да функционира“ [87]. Свободната онлайн енциклопедия Уикипедия е най-популярният пример за потенциалните възможности на уики софтуера, въпреки съществуващото критично отношение към така наречената „мъдрост на тълпите“. Понастоящем, няколко са дистрибуциите на уики софтуер, които се смятат за най-популярни: *MediaWiki*, *DokuWiki*, *TikiWiki*, *Moinmoin* и други. За Ifigenia.org беше избрана дистрибуцията *MediaWiki*, на която работи и Уикипедия и към която много хора по цял свят активно допринасят с поддръжка, отстраняване на грешки и разширяване с разнообразни софтуерни модули.

Философията зад концепцията за уики може да бъде обобщена в следните три водещи принципа [152].

- Уики насърчава смислените тематични асоциации между различни онлайн обекти, позволявайки интуитивно създаване на връзки както към вече съществуващи уебстраници и документи, така и към все още несъздадени такива. Това на практика представлява онлайн база от данни, даваща отворен достъп за създаване, четене, извличане, търсене и организиране на информация.
- Технологично, уики технологията позволява на всички свои потребители, без значение регистрирани или не, да редактират всяка страница или документ в базата данни в рамките на сайта. При все това, всеки допринасящ редактор бива държан отговорен за своите редакции и действия, тъй като всяка редакция се съхранява в базата данни под уникален идентификатор, асоцииран с датата и часа на редакцията и потребителското име или IP-адреса на потребителя, и всяка редакция може да бъде идентифицирана, проследена, премахната или възвърната при необходимост.
- Уики технологията не служи за създаване на грижливо структурирани и съдържателно завършени уебсайтове, предназначени за статично и еднопосочно доставяне на съдържание от един „производител“ на информация към множество „консуматори“. Уики има за цел да увлече посетителите на уебсайта в един непрестанен интерактивен процес на развиване на съдържанието на сайта и на взаимодействие. Най-вероятните потребители на уики съдържанието са и неговите основни автори.

Общностният дух у допринасящите към уикито се смята от някои автори за най-привлекателния аспект на тази технология. Колкото и необичайно да е за някои автори/издатели на съдържание да предоставят част от контрола в ръцете на читателите си, онези, които съумеят да открият модела, по който да правят

това, биват възнаградени с лоялна и ентузиазизирана потребителска общност, [108]. Този дух се проявява особено силно при по-малки общности от хора, които споделят едни и същи интерес и стремежи.

Приложенията на уики технологията са много на брой и разнообразни. Компании използват уикита за интранет, разработване на продуктова документация, управление на проекти, разпределение и отчитане на задачи, групов електронна поща, корпоративна база от данни или знания, „жълти страници“ и други. Учебни заведения и академични общности използват уикита за съвместна изследователска и развойна дейност, споделяне на резултати, дискусии, генериране и обмен на идеи, провеждане на електронно обучение и т.н., [86]. Български изследвания в областта на приложенията на уики в образованието са например [6, 20].

4.2. Ifigenia.org – среда за електронно обучение по обобщени мрежи

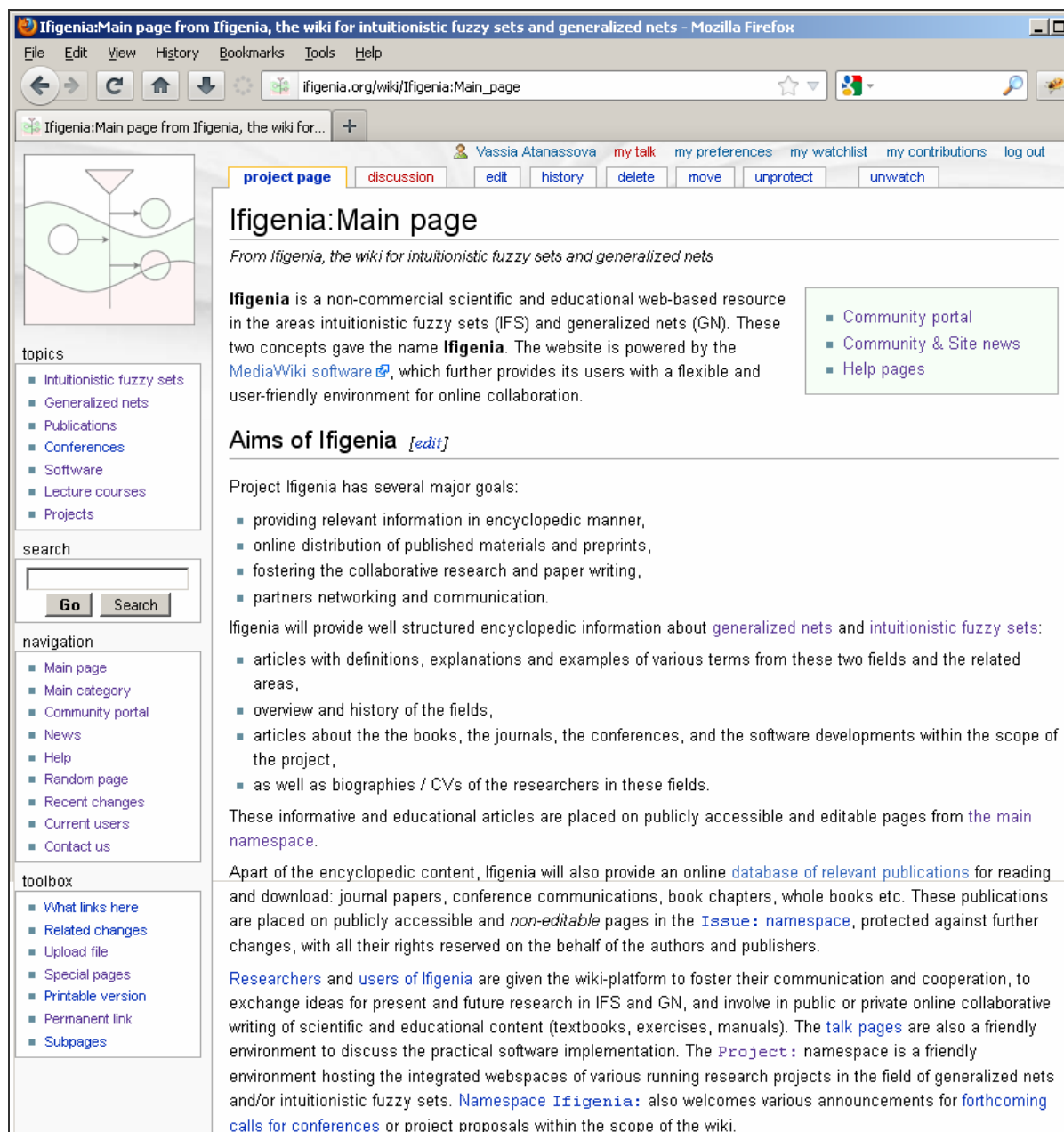
Първоначалната идея за създаването на Ifigenia.org (Фиг. 4.1.) е продиктувана от нуждата да се изгради интегрален онлайн-базиран информационен ресурс за ОМ и ИРМ. И двете научни области са получили разпространение в много страни по света. По теорията и приложенията на ИРМ и ОМ има публикувани множество монографии и статии, провеждат се годишни международни конференции и работни срещи, четат се университетски лекционни курсове, разработва се софтуер. При все това, към момента на създаване на уикито не беше системно работено по задоволяване потребността от онлайн присъствие и достъп до различни ресурси и услуги.

Междувременно натрупаният опит на докторанта с работа на доброволчески начала по българската версия на свободната енциклопедия Уикипедия даде добра основа тази потребност да бъде удовлетворена, използвайки наличните възможности на технологията уики.

Основната инсталация е направена с версия 1.13.2 на MediaWiki, версия 5.0.51a-24+lenny2 на MySQL и версия 5.2.6-1+lenny3 на PHP. По време на периода на тестване и конфигуриране на инсталацията са изпробвани разнообразни разширения на ядрото. Част от идеите са заимствани частично от други уики платформи, а друга част се появяват в процеса на използване на сайта за задоволяване на възникнали потребности. В резултат, ядрото на MediaWiki в Ifigenia.org е надградено със следните разширения, пълния и актуален списък на които е достъпен през специалната страница “Version”³:

- *CategoryTree* – за динамична навигация в категорийната структура.
- *CharInsert* – позволява създаването на JavaScript меню за избор и вмъкване на неклавиатурни символи.
- *Cite* – за въвеждане на цитирания и бележки под линия с XML етикетите `<ref>`, `</ref>` и `<references/>`.

³ Ifigenia.org, Special:Version, <http://www.ifigenia.org/wiki/Special:Version>



Фигура 4.1. Началната страница на Ifigenia.org

- *CurrentUsers* – за проследяване на текущите посетители на сайта.
- *Link Search* – инструмент за откриване по зададен регулярен израз на страници от уикито, съдържащи препратки към външни веб-ресурси.
- *LockDown* – за управление на групови потребителски права според именно пространство.
- *MathStatFunctions* – колекция от математически и статистически функции.
- *Newuserlog* – за поддръжка на дневник на новорегистрирани потребители.
- *ParserFunctions* – колекция от парсерни и логически функции, необходими при създаването на уики шаблони.
- *Renameuser* – възможност за преименуване на потребителски сметки.
- *SpamBlacklist* – за поддръжка на черен списък на спама.

- *SyntaxHighlight Geshi* – за осветяване в различни цветове на отделни синтактични елементи от програмен код.
- *User Stats* – за изготвяне на статистики за приносите на отделни потребители.

Още отначало е взето решение Ifigenia.org да съдържа два типа страници – с публичен и с ограничен достъп. Публичната част на уикито може да се чете от всички потребители, но само регистрирани могат да пишат в нея и да качват файлове. В частта с ограничен достъп могат да четат и редактират само регистрираните, като тези страници не се индексират в търсачките. За целта са добавени три нови именни пространства:

- *Issue*, в което се поставят описателни страници с анотации на качените публикации от областта на ОМ и ИРМ,
- *Project*, в което са обособени уеб-пространствата на различни финансирани проекти от областта на ОМ и ИРМ, и
- *Private*, което е изцяло непублично и на разположение за четене и писане само на регистрираните потребители.

За разлика от други видове колаборативен софтуер и системи за управление на съдържанието, в ядрото на MediaWiki няма реализирана възможност за лични потребителски страници, недостъпни за други потребители. Това обаче е принципно възможно с някои от съществуващите разширения на ядрото (като *AuthorRestriction* или *AuthorProtect*). От съображения за сигурност и за противодействие на различни спам-ботове в Мрежата, регистрацията в уикито не е отворена, а става след заявка по електронна поща, която подлежи на одобрение от администратор. Потребителските сметки са с личните имена на заявителите.

Също от самото начало датира и визията за авторскоправния статут на съдържанието на Ifigenia.org. Тук са в сила три постановления. Първо, публичната част от съдържанието, която се разработва в рамките на самото уики (корпусът от енциклопедични и учебни материали, вж Фиг. 4.2.) е лицензирана под свободния лиценз *Creative Commons Attribution 2.5 (CC-BY 2.5⁴)*, което позволява многократно използване за всякакви цели, модификации и разпространение без необходимост от изрично разрешение, стига да се прави адекватно признание на авторството с обратна връзка към сайта. Изборът на този лиценз кореспондира със свободния лиценз на *MediaWiki (GPL)* и с вижданията на автора за приложението на свободния софтуер в науката, [5]. Второ, качените в Ifigenia.org публикации (Фиг. 4.3, 4.4.) са публично достъпни за четене, но не и за промени, и са защитени с пълните авторски права на своите автори и издатели. По тази причина в колекцията не могат да попаднат всякакви материали, а само такива, за които има разрешение от издателите или които са оригинално публикувани при условия, позволяващи копирането и разпространението за информационни и некомерсиални цели (каквито например са публикациите на EUSFLAT⁵). Трето, материалите с ограничен достъп, поставени в пространство *Private* (обсъждания, чернови на научни статии и на проектни предложения), са с безусловно защитени права на авторите им.

⁴ Creative Commons Attribution, <http://creativecommons.org/licenses/by/2.5/deed.bg>

⁵ EUSFLAT: Copyright, <http://www.eusflat.org/documents/imprint.htm>

the wiki for intuitionistic fuzzy sets and generalized nets - Mozilla Firefox

Bookmarks Tools Help

ifigenia.org/w/index.php?title=Transition&oldid=4327

the wiki for intuitionistic... +

Vassia Atanassova my talk my preferences my watchlist my contributions log out

page discussion edit history delete move protect unwatch

Transition

From ifigenia, the wiki for intuitionistic fuzzy sets and generalized nets (You can help improve this article immediately)

Revision as of 17:21, 22 November 2009 by Vassia Atanassova (Talk | contribs | block)
 (diff) ←Older revision | Current revision (diff) | Newer revision→ (diff)

Transition in the context of *generalized nets* is an object from the static structure of the net, which comprises the conditions of *tokens*' transfer from the transition's input places to its output places.

When tokens enter the input place of a transition, it becomes *potentially fireable* and at the moment of their transfer towards the transition's output places, the transition is being *fired*.

The tokens' transfer through a transition is described by a *special formal algorithm*.

Formal description [edit]

Formally, every transition is described by a 7-tuple:

$$Z = \langle L', L'', t_1, t_2, r, M, \square \rangle$$

where:

- L', L'' are finite, non-empty sets of places: the transition's input and output places, respectively.
- t_1 is the current time-moment of the transition's firing.
- t_2 is the current value of the duration of its active state.
- r is the transition's *condition*, determining which tokens will transfer from the transition's inputs to its outputs. The parameter has the form of an *index matrix*:

	l''_1	...	l''_j	...	l''_n
l'_1					
...					
l'_i			$r_{i,j}$		
...					
l'_m					

where $r_{i,j}$ are predicates, $1 \leq i \leq m, 1 \leq j \leq n$

- M is the index matrix of the capacities of the transition's arcs:

	l''_1	...	l''_j	...	l''_n
l'_1					
...					
l'_i			$M_{i,j}$		
...					
l'_m					

where $M_{i,j} \geq 0$ are natural numbers or ∞ , $1 \leq i \leq m, 1 \leq j \leq n$

- \square is called transition's type, an object having a form similar to a Boolean expression. It may contain as variables the symbols that serve as labels for transition's input places, and it is an expression constructed of variables and the Boolean connectives \wedge and \vee determining the following conditions:
 - $\wedge(l_{i_1}, l_{i_2}, \dots, l_{i_u})$ - each of the places $l_{i_1}, l_{i_2}, \dots, l_{i_u}$ must contain at least one token,
 - $\vee(l_{i_1}, l_{i_2}, \dots, l_{i_u})$ - there must be at least one token in the set of places $l_{i_1}, l_{i_2}, \dots, l_{i_u}$ where $\{l_{i_1}, l_{i_2}, \dots, l_{i_u}\} \subset L'$
 When the value of a type (calculated as a Boolean expression) is "true", the transition can become active, otherwise it cannot.

A GN transition with m inputs and n outputs

References [edit]

- On *Generalized Nets Theory*, Krassimir Atanassov, Prof. Marin Drinov Academic Publishing House, Sofia, 2007

Category: Generalized nets

Фигура 4.2. Страницата “Transition” към версия от 22.11.2009, записана под #4327

ts with places having intuitionistic fuzzy capacities from Ifigenia, the wiki for intuitionistic fuzzy sets and generalized nets - M...

Bookmarks Tools Help

ifigenia.org/wiki/Issue:Generalized_nets_with_places_having_intuitionistic_fuzzy_capacities

with places having in... +

Vassia Atanassova my talk my preferences my watchlist my contributions log out

issue discussion edit history delete move protect unwatch

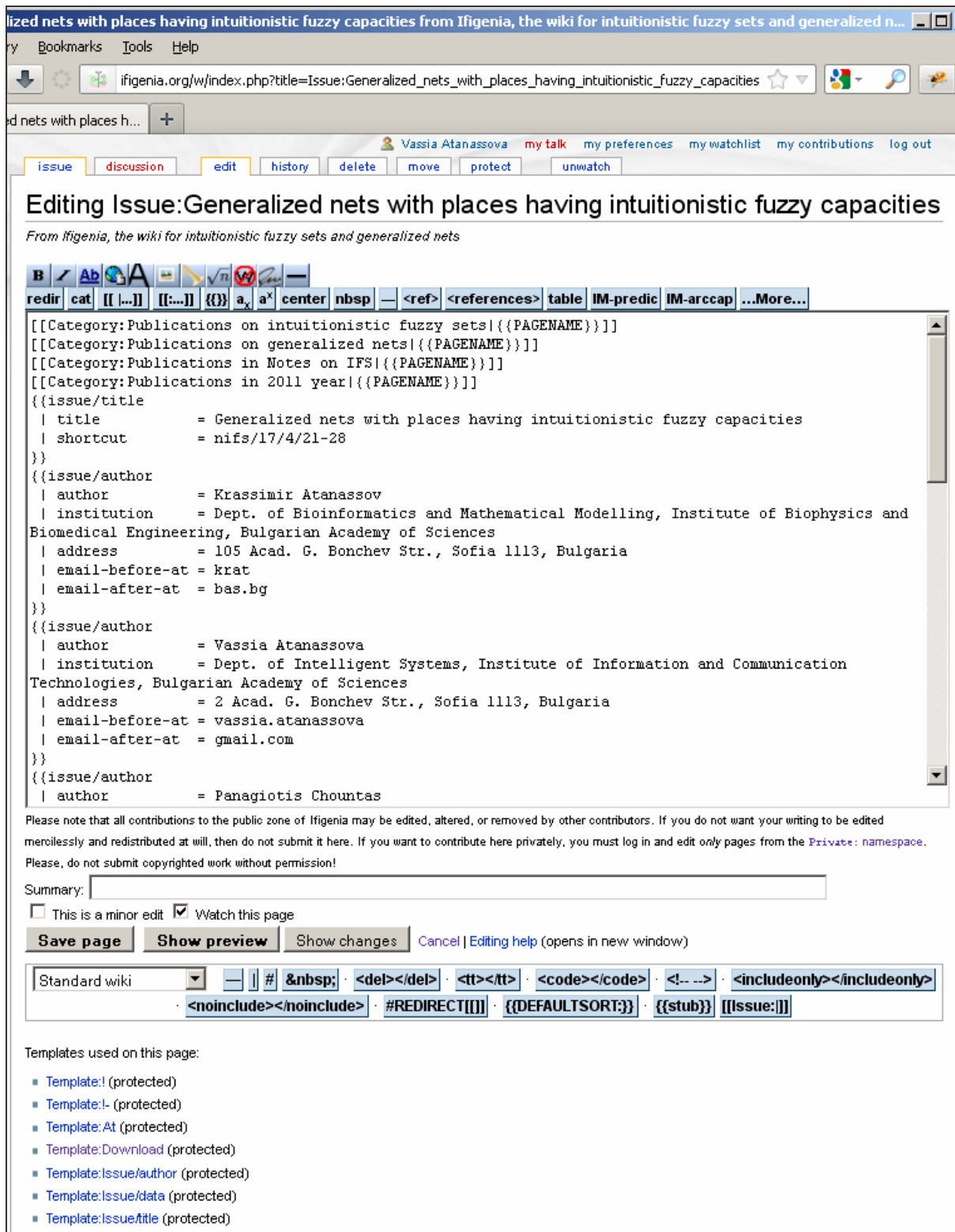
Issue:Generalized nets with places having intuitionistic fuzzy capacities

From Ifigenia, the wiki for intuitionistic fuzzy sets and generalized nets shortcut: ifigenia.org/wiki/issue:nifs/17/4/21-28

Title of paper:	Generalized nets with places having intuitionistic fuzzy capacities
Author(s):	<p>Krassimir Atanassov Dept. of Bioinformatics and Mathematical Modelling, Institute of Biophysics and Biomedical Engineering, Bulgarian Academy of Sciences, 105 Acad. G. Bonchev Str., Sofia 1113, Bulgaria krat@bas.bg</p> <p>Vassia Atanassova Dept. of Intelligent Systems, Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, 2 Acad. G. Bonchev Str., Sofia 1113, Bulgaria vassia.atanassova@gmail.com</p> <p>Panagiotis Chountas HSCS, University of Westminster,, Northwick Park, HA1 3TP, London, UK P. I. Chountas@westminster.ac.uk</p> <p>Anthony Shannon Faculty of Engineering & IT, University of Technology, Sydney, NSW 2007, Australia tshannon38@gmail.com, anthony.shannon@uts.edu.au</p>
Presented at:	7th IMFS, Banska Bystrica, 27 September 2011
Published in:	Conference proceedings, "Notes on IFS", Volume 17 (2011) Number 4, pages 21—28
Download:	PDF (174 Kb, Info)
Abstract:	A new extenmsion of the Generalized Net (GN) called "Generalized nets with places, having intuitionistic fuzzy capacities" is introduced. Because this GN is the fifth type of Intuitionistic Fuzzy GNs, for brevity it is named also IFGN5. Algorithms for tokens transfers of the IFGN5s are given. It is proved a theorem, asserting that for each IFGN5 there exists a standard GN that describes the functioning and the results of its work.
Keywords:	Generalized net , Intuitionistic fuzziness
AMS Classification:	68Q85
References:	<ol style="list-style-type: none"> 1. Atanassov K., Generalized Nets, World Scientific, Singapore, New Jersey, London, 1991. 2. Atanassov, K., On Generalized Nets Theory. Prof. M. Drinov Publishing House, Sofia, 2007. 3. Atanassov, K. Intuitionistic Fuzzy Sets: Theory and Applications. Springer-Verlag, Heidelberg, 1999. 4. Atanassov, K., D. Dimitrov, V. Atanassova, Algorithms for tokens transfer in different types of intuitionistic fuzzy generalized nets Cybernetics and Information Technologies, Volume 10, No 4, 2010, 22–35. 5. Dimitrov, D. Optimized algorithm for token transfer in generalized nets. In: Recent Advances in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics, Vol. 1, Warsaw, SRI PAS, 2010, pp. 63–68.
Citations:	n/a The list of publications, citing this article may be empty or incomplete. If you can provide relevant data, please, write on the talk page .

Categories: [Publications on intuitionistic fuzzy sets](#) | [Publications on generalized nets](#) | [Publications in Notes on IFS](#) | [Publications in 2011 year](#) | [Publications of Krassimir Atanassov](#) | [Publications of Vassia Atanassova](#) | [Publications of Panagiotis Chountas](#) | [Publications of Anthony Shannon](#)

Фигура 4.3. Анотация на качена публикация в именно пространство *Issue* на Ifigenia.org (в режим на преглед; режимът на редактиране за същата страница е на Фиг. 4.4.)



Фигура 4.4. Изгледът на страницата от Фиг. 4.3. в режим на редактиране

От разширенията, по-специално отбелязване заслужава *SyntaxHighlight Geshi*, което дава възможност части от програмен код на поддържаните около

120 програмни езика⁶ да бъдат оцветявани, в зависимост от настройките за езика. В Ifigenia.org са използвани в документация на програмните продукти за ОМ, разработвани от Т. Трифонов, К. Георгиев [150] и Д. Димитров [14, 15], а също и за демонстрация как се кодират ОМ в XML формат, а характеристикните им функции – в специално създадения LISP-подобен език, както е показано на Фиг. 4.5. и в страницата “GN XML format” в Ifigenia.org⁷.

```

Example: test.xml [edit]
<?xml version="1.0"?>
<gn name="A Test GN" time="10">
  <transitions>
    <transition id="Z1">
      <inputs> P1 P2      </inputs>
      <outputs> P2 </outputs>
      <predicates>
        true
        less_than_10
      </predicates>
    </transition>
  </transitions>
  <places>
    <place id="P1"/>
    <place id="P2" char="inc"/>
  </places>
  <tokens>
    <token id="T1" host="P1">
      <char name="Default" type="double"> 0 </char>
    </token>
  </tokens>
  <functions>
    (defun inc "" () ()
      (set (+ token 1)))
    (defun less_than_10 "1;tokens.T1.char" () ()
      (< #0 10))
  </functions>
</gn>

```

Фигура 4.5. Описание на ОМ в XML формат, осветяването на програмния код се получава с командата `<source lang="xml">`

По отношение на дизайна на Ifigenia.org бяха направени някои специфични настройки. Например беше добавено меню *Topics*, извеждащо по-специален акцент върху двете основни тематични направления на сайта – ИРМ и ОМ, а също и подпомагащо навигацията до категориите с публикации, конференции, софтуер, учебни курсове и научни проекти.

По-сериозна персонализация беше направена в режима на редактиране под формата на допълнителни бутони и падащи менюта, програмирани на JavaScript. Новите елементи бяха съобразени с потребността от въвеждане на специфични неклавиатурни Unicode символи като знаци за операции и релации, гръцки букви и т.н. Специални уики шаблони бяха изготвени и за лесната визуализация на индексирани матрици, един характерен по своята графика елемент в теорията на ОМ.

⁶ Mediawiki.org, Extension:SyntaxHighlight GeSHi, http://www.mediawiki.org/wiki/Extension:SyntaxHighlight_GeSHi

⁷ Ifigenia.org, GN XML format, http://ifigenia.org/wiki/GN_XML_format

Бяха направени необходимите изпитания, които да осигурят, че всички скриптове и шаблони в уикито работят коректно под различни браузъри и техни версии.

*
**

Ще отбележим, че ОМ са прилагани за моделирането на разнообразни приложни аспекти на образованието, например изпитване и оценяване на студенти, оценяване на преподаватели, използване на платформи за електронно обучение, управление на информационните и материалните потоци в дигитален университет, и др. Някои от най-съществените приноси по тази тема са събрани в монографиите [24, 136]. Въпреки това доскоро не бяха правени опити за обратното: да се приложат добре утвърдени практики на преподаване и тестване към обучението по теорията на ОМ. Този факт отваря възможности за нови, плодотворни и интересни направления на изследване в областта.

В началото на 2010 г. в [69] авторът предложи новия, *дидактически*, аспект на теорията на ОМ. От една страна той има за цел да адресира и обоснове необходимостта от системен подход към обучението по ОМ като критична предпоставка за формирането на добри специалисти по този апарат за математическо моделиране, а от друга – да очертае посоките за развитие и на методологическия аспект на теорията на ОМ. Като първа стъпка в тази посока бяха предложени стандартизирани обучителни тестове, които взимат предвид особеностите на теорията на ОМ и нивото на настоящата студентска аудитория.

Тестовите бяха изпробвани в реални условия със студенти от Технически университет – София, Университет „Проф. д-р Асен Златаров“ – Бургас и Бургаски свободен университет, където от години се четат курсове по ОМ. За отделните типове задачи беше направен анализ на обучителните цели, следващ таксономията на В. Bloom [80], както и анализ на най-често допусканите грешки, съдържащ както полезна обратна връзка за провежданото обучение, така и идеи, свързани с методологическия аспект на теорията на обобщените мрежи. Резултатите от това проучване са публикувани в [75] и ще бъдат накратко представени по-долу.

През април 2010 г. към ядрото на Ifigenia.org е инсталирано разширението *Quiz*⁸, което позволява всички тези типове тестови задачи да бъдат интегрирани на практика в уикито и да се използват за провеждане на електронни тестове.

4.2.1. Съображения при проектиране на тестове по обобщени мрежи

Когато се изготвят оценителни тестове за обучителните курсове по ОМ, следва да се спазват основните принципни препоръки за тестовите формати. Различни публикации в научни списания и мейнстрийм сайтове предлагат успешни практики, предимства и недостатъци и съвети за писане на тестови въпроси, върху които няма да се спираме в детайли тук, [96, 106, 111, 145]. Ще бъде

⁸ Mediawiki.org, Extension:Quiz, <http://www.mediawiki.org/wiki/Extension:Quiz>

проведена (или поне започната) обаче дискусиия върху особеностите на ОМ като научна област, тъй щото тези съображения да бъдат взети предвид и оптимално експлоатирани в процеса на изготвяне на тестовете.

Без претенции за изчерпателност, някои от най-популярните типове тестове са въпросите с множество отговори (*Multiple-choice questions*), въпросите от типа „истина или лъжа“ (*True/False*), тест за попълване на пропуснати думи (*Cloze test*), тест за намиране на съответствия (*Matching test*), въпрос със свободен отговор (*Free response*), и други. Авторът смята тези тестови типове много подходящи средства за оценяване на обучаемите по ОМ по много причини, включващи, но не ограничени до посочените по-долу:

- Въпросите от вида „истина или лъжа“ са от полза за оценяване на разбирането от страна на студентите на твърдения или факти, и по-специално такива, които по естествен начин предлагат две логически алтернативи, както и за изпитване на разбирането на популярни погрешни схващания. Съществуват много теореми и твърдения в теорията на ОМ, които могат лесно да бъдат формулирани като въпроси от този вид. При все това, този тестов елемент е свързан с високи нива на налучкване и е трудно да се разграничат студентите, които знаят верния отговор, от онези, които не го знаят и са го налучкали.
- Тестът за намиране на съответствия също е подходящ за проверка на нивото на знанията и понякога – когато е съставен добре – за нивото на разбиране. Този вид тест може да се използва за оценяване на знанията на обучаемите за базовите свойства или компоненти от дефиницията на ОМ, за различните разширения на мрежите на Петри, и разширенията на самите ОМ.
- Тестовите въпроси с множество възможни отговори са ефективно и все-странно средство за изпитване, способно да покрие голям диапазон от учебно съдържание. Често се смята че този тип тестове са трудни за изготвяне, защото добрите въпроси от този тип трябва да предлагат известен брой (обикновено около 3–4) правдоподобно звучащи алтернативи на верния отговор („дистрактори“), а такива алтернативни отговори са по-трудни за формулиране. Теорията на ОМ обаче предлага относително дълъг списък дефиниции и означения, които при правилно комбиниране формират изчерпателно множество от тестови въпроси и дистрактори. Нещо повече, въпросите с много отговори не са ограничени до текстови или числови изрази. Тъй като ОМ включват графични представяния на моделите, тези графики могат успешно да бъдат включвани в тестовете. При този тип тестове има по-ниски нива на налучкване, поради по-големия брой алтернативни отговори. Във всеки случай, ако от студентите се иска и да обосновават отговора си, това е винаги показателно как са достигнали до него.
- Въпросите за попълване на пропуснати думи са отворени въпроси, които изискват от студентите да разбират и интерпретират изречения и логически да достигнат до липсващата дума. Така възможното налучкване е намалено и могат да се проверят умения от по-високо ниво. Една стъпка напред в тази посока е да се задават въпроси, които предполагат свободен отговор в едно или две изречения. Областта на ОМ предлага добри възможности за генериране на подобни отворени въпроси.

Други популярни техники за изпитване като скала на Ликърт (*Likert scale*) и писане на есе, обаче, авторът смята за неподходящи. Теорията на ОМ е силно формализирана и трудно могат в нея да се формулират твърдения, на които да се отговаря със степен на съгласие или несъгласие, както е в скалата на Ликърт. Поради привичната им субективност, есетата също са неприложими към тази научна област, въпреки че те открояват уменията на студентите да подбират, разработват и организират идеите си, да сравняват, противопоставят, аргументират и анализират информация. За отбелязване е все пак, че такива умения са от критично значения по време на интервюта между ОМ моделиери и експерти, които съставляват началната фаза при изграждането на ОМ модел.

4.2.2. Примерни тестови задачи по обобщени мрежи

Следва списък на примерни тестови задачи от различните типове, описани по-горе и в [69]. Всеки тип е представен с по три въпроса, като са дадени и верните отговори.

Таблица 4.1. Примери на тестови въпроси от вида „истина или лъжа“

Въпрос	Истина	Лъжа
1. Всяко разширение на мрежите на Петри е консервативно разширение.		✓
2. Всяка цветна мрежа на Петри може да бъде представена чрез редуцирана обобщена мрежа.	✓	
3. Понятието „абстрактен преход“ представлява сечение на всички текущо активни преходи.		✓

Таблица 4.2. Примери на тест за намиране на съответствия

В таблицата по-долу отбележете с ✓ кой компонент от дефиницията на обобщена мрежа отразява нейната (A) статика, (B) динамика, (C) хронология, (D) памет.	Статика A	Динамика B	Хронология C	Памет D
Функцията θ_k , даваща следващия момент от време, когато дадено ядро ще влезе в мрежата		✓		
Моментът от време T , когато мрежата започва да функционира			✓	
Функцията θ_2 , даваща продължителността на активното състояние на даден преход	✓			
Функция b , даваща максималния брой на характеристиките, пазени в историята на ядрото				✓
Функция c , даваща капацитетите на позициите	✓			
Функция π_A , даваща приоритетите на преходите	✓			

Таблица 4.3. Примери на тестови въпроси с множество отговори

<p>1. Как се определя елементарната времева стъпка на модел, получен като композиция на два модела с различни времеви стъпки?</p>	
A. Средноаритметичното на двете елементарни времеви стъпки.	<input type="checkbox"/>
B. Частното на по-голямата и по-малката елементарна времева стъпка.	<input type="checkbox"/>
C. По-малката от двете елементарни времеви стъпки.	<input type="checkbox"/>
D. Най-големият общ делител на двете елементарни времеви стъпки.	<input checked="" type="checkbox"/>
<p>2. Кое може да бъде променено с използването на локален оператор?</p>	
A. Множеството от началните характеристики на ядрата.	<input type="checkbox"/>
B. Структурата на преход.	<input checked="" type="checkbox"/>
C. Стратегиите за трансфер на ядра през преход.	<input type="checkbox"/>
D. Характеристичната функция на изходна позиция.	<input type="checkbox"/>
<p>3. Коя от следните графични структури на обобщена мрежа съдържа грешка?</p>	
<p>A.</p>	
<p>B.</p>	
<p>C.</p>	
<p>D.</p>	
A.	<input checked="" type="checkbox"/>
B.	<input type="checkbox"/>
C.	<input type="checkbox"/>
D.	<input type="checkbox"/>

Таблица 4.4. Примери на въпроси за попълване на пропуснати думи

1. Всяка обобщена мрежа съдържа поне един *преход* .
2. Типът на прехода \square е обект от *булев* тип.
3. Самомодифициращите се обобщени мрежи представляват *консервативно* разширение на обобщените мрежи.

Таблица 4.5. Примери на въпроси със свободен отговор

1. Какви са разликите между обикновената матрица и индексиранията матрица?

 Индексиранията матрица има редове и стълбове, които са означени с индекси, позволявайки по-гъвкаво манипулиране на тези обекти, в сравнение с обикновените матрици. Алгебричните операции с индексиранията матрици са винаги коректно дефинирани, без ограничения върху размерите на матричните операнди.

2. Какви са компонентите на минималната редуцирана обобщена мрежа?

 Тя съдържа само онези компоненти, които са строго задължителни: $\langle A', K, X, \Phi \rangle$ където A' е множеството на преходите (също с редуцирано множество от компоненти, а именно множество на входни позиции, множество на изходни позиции и индексиранията матрица на предикатите); K е множеството от ядрата; X е множеството от началните характеристики на ядрата; Φ е функцията, която присвоява нови стойности на характеристиките на ядрата. Всички други компоненти на обобщената мрежа могат да бъдат изразени, използвайки посочените.

3. Какви са условията да се присвои 0 на всички елементи от индексиранията матрица, съответстваща на матрицата на предикатите в алгоритъма за функциониране на преход от мрежата?

 Тези условия са три: (1) входната позиция, която отговаря на съответния предикат е празна, или (2) съответната изходна позиция е пълна, или (3) капацитетът на дъгата, свързваща съответен вход и изход, е нула.

4.2.3. Задачи по обобщени мрежи, базирани на графичните им структури

Теорията на ОМ е богата на дефиниции и теореми, които предлагат множество начини да се оценят когнитивните нива на познаване и разбиране на студентите, и предразполагат към достигане на по-високите когнитивни нива на приложение, анализ, синтез и оценяване (по В. Bloom), необходими за практикуването на ефективно математическо моделиране [69]. Един забавен и

предизвикателен, и все пак непроучен начин за това е използването на графичните структури на ОМ, поставени в центъра на някои от тестовите задачи.

На базата на изследването на литературата по ОМ, може да се твърди, че графичните структури на ОМ моделите не играят централната роля в процеса на моделиране чрез ОМ. Те имат само спомагателна роля за илюстриране на модела, тъй като не съдържат и не представят цялата необходима информация за функционирането на модела, която информация се съдържа в съпътстващите индексирани матрици на предикатите и капацитетите на преходите, описанията на началните характеристики на ядрата и характеристичните функции, които определят следващите им стойности и т.н.

Графичната структура на обобщеномрежовия модел обаче е неговото най-непосредствено и потребителски приветливо представяне, тъй като тя предлага с един поглед разбиране на последователността на модела и относителната му сложност, изразени с броя на преходите, позициите, дъгите (прави и обратни), както и възможните повтарящи се елементи, в които се съдържат потенциални възможности за опростяване и оптимизиране на модела.

Дотук авторът е формулирал два вида задачи, целящи да се изпитат нивата на знаене и разбиране на обучаемите при:

- изграждане на графичните структури на ОМ (Таблица 4.3., задача 3), и
- прилагане на граф-операторите, дефинирани над ОМ (в [41], §6.2, стр. 222–239).

Тези тестови задачи са били апробирани в реални изпити и резултатите са публикувани в [75]. За отбелязване е, че трите групи са подредени според времето, когато са обучавани и изпитвани, и че обратната връзка от всяка група обучаеми е вземана предвид при обучението на всяка следваща група.

Този визуален подход с възможността за непосредствено представяне и разбиране на теорията на ОМ е особено полезен при начинаещи обучаеми или такива, които нямат предварителни познания по изкуствен интелект, математическо моделиране или програмиране, но се нуждаят от този апарат за решаването на практически задачи от тяхната област на компетенции.

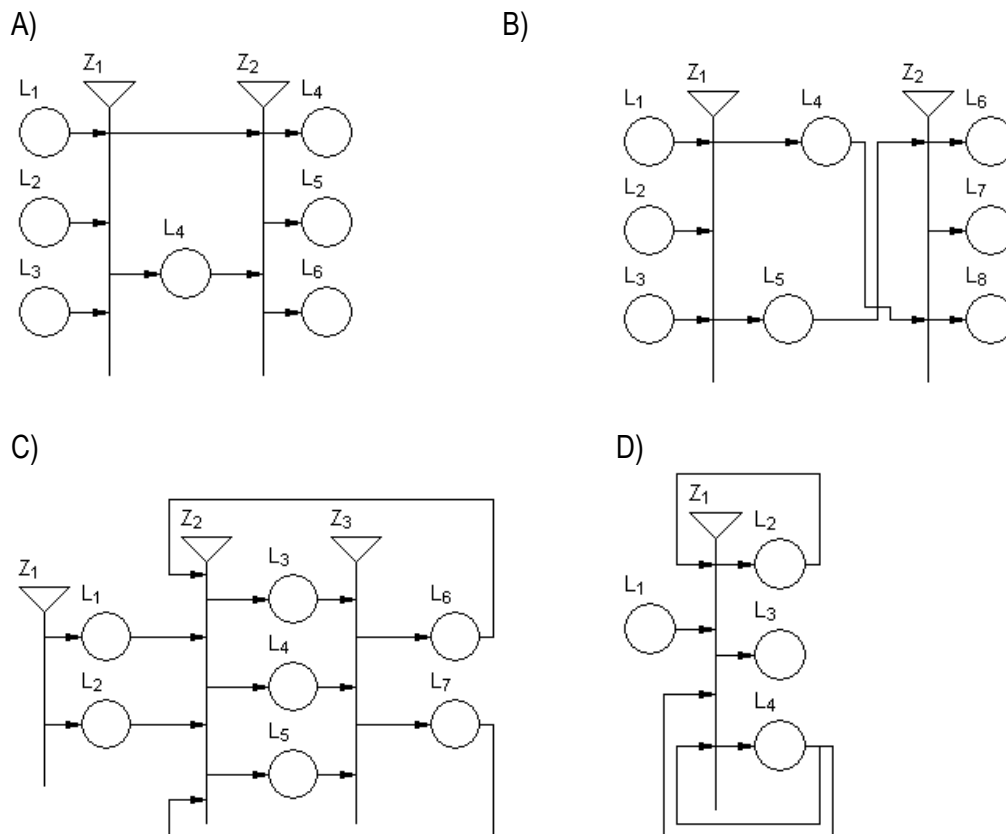
Въпроси за избор измежду множество отговори

Първият вид задачи са заимствани от описаните в т. 4.2.1 типове въпроси с множество отговори. На студентите се предлагат четири относително прости графики на ОМ модели, измежду които или точно една е вярна, или точно една е грешна. От обучаемите се иска да различат която е вярната (респективно, грешната) и да придружат отговора си с обосновка. Верните отговори без такава обосновка не се вземат предвид.

Пример за въпрос от този вид е даден на Фиг. 4.6.

От обучаемите се очаква да могат да определят верния ОМ модел, съдейки по графичната му структура, и да обяснят в какво се състоят грешките на другите три места.

Посочете коя от графичните структури на ОМ модели е вярна и обяснете отговора си:



Фигура 4.6. Примерен въпрос за графични структури на ОМ с четири възможни отговора, от които един правилен и три грешни („дистрактори“).

Верният отговор в случая е В). В А) грешките са две: първо, преходи Z_1 и Z_2 са свързани директно, без да се преминава през позиция, и второ, две позиции има, означени по еднакъв начин с етикет L_4 . Грешката в С) е в липсата на входна позиция за преход Z_1 , а оттам и за целия модел, докато в Д) грешката е в двойната дъга, която излиза от позиция L_4 .

Следната таблица представя разпределението на обучаемите в трите изследвани групи студенти в зависимост от техните отговори.

Таблица 4.6.

	Правилен отговор		Частичен или грешен отговор		Без отговор		Общо
	Брой	Процент	Брой	Процент	Брой	Процент	
Група 1	3	27,27%	1	9,09%	7	63,64%	11
Група 2	9	56,25%	5	31,25%	2	12,50%	16
Група 3	11	100%	0	0%	0	0%	11
Общо	23	60,53%	6	15,79%	9	23,68%	38

Следните типове грешки бяха регистрирани.

- Грешки при означаването на позиции и преходи. Някои студенти не отчитат етикетите, с които позициите и преходите се означават, като *уникални* идентификатори на обекти в рамките на ОМ. Например, може да не обърнат внимание на факта, че две различни позиции в мрежата са означени по еднакъв начин. Други студенти се объркват от факта, че са възможни пропуски и номерацията на индексите, например, L_1, L_2, L_4, L_5 , където липсата на L_3 се посочва като грешка, макар че не е. Тези грешки са индикатор, че обучаемите не са съвсем наясно как да интерпретират ОМ в термините на други по-познати понятия от програмирането.
- Липсващи или излишни компоненти на ОМ. Някои студенти забравят, че според дефиницията на ОМ, два прехода могат да бъдат свързани само посредством позиция, както и две позиции могат да бъдат свързани само посредством преход. Други не разпознават, когато от една позиция излизат повече от една дъга и не разпознават когато дъга за обратна връзка е грешно изчертана да влиза от дясната, вместо от правилната лява страна на прехода. Някои студенти не интерпретират добре дефиницията на ОМ, според която всеки преход трябва да има поне една входна и поне една изходна позиция и не виждат никакъв проблем с преход Z_1 в С). Този клас грешки могат лесно да се преодолеят, като се изясни дефиницията на ОМ и правилата за построяване на мрежата от нейните графични компоненти.
- Неясноти относно графичната структура. Някои обучаеми са склонни да придават неоправдано голямо значение на *конкретното разположение* на преходите, позициите и дъгите, вместо да се фокусират върху изразените графично *връзки* между тях. Някои мислят, че ако една входна и една изходна позиция са разположени на едно и също ниво по дължината на прехода (например както са дъгите от / към позиции L_1 и L_4 в пример В) по-горе), то ядрата, излизащи от тази входна позиция задължително ще продължат движението си само и единствено към намиращата се отсреща изходна позиция на прехода. Тази линия на разсъждение е погрешна, но показателна, тъй като демонстрира липсващата връзка между графичната структура на ОМ модела и индексирания матрица на предикатите, която диктува фактически възможните маршрути на движение на ядрата в прехода, в зависимост от логиката на моделирания процес.
- Погрешни предположения за динамиката на модела, хронологичните му компоненти и паметта. Когато не става въпрос за задачи за изследване на казуси, задачите с графични структури не съдържат никаква допълнителна информация за примерните модели, като индексирани матрици на предикатите на преходите, характеристики на ядрата, времена и прочее, а е дадена само и единствено графична структура. Затова е изненадващо, когато някои обучаеми твърдят, че са открили в моделите грешки, причинени от липса на ядра в позициите или определени стойности на предикатите с вярностна стойност „лъжа“. Някои студенти дори твърдят, че според тях грешката на модела се състои в излишни или липсващи преходи или позиции. Такива грешки са показателни за недостатъчното разбиране на разликата между логиката на моделирания процес и неговата графична

визуализация; между статичната и динамичната природа на ОМ модела, както и недоразбиране на това коя част от информацията за модела в какъв вид се съхранява и как може да се извлече и интерпретира.

Приложение на граф оператори над структурата на ОМ модела

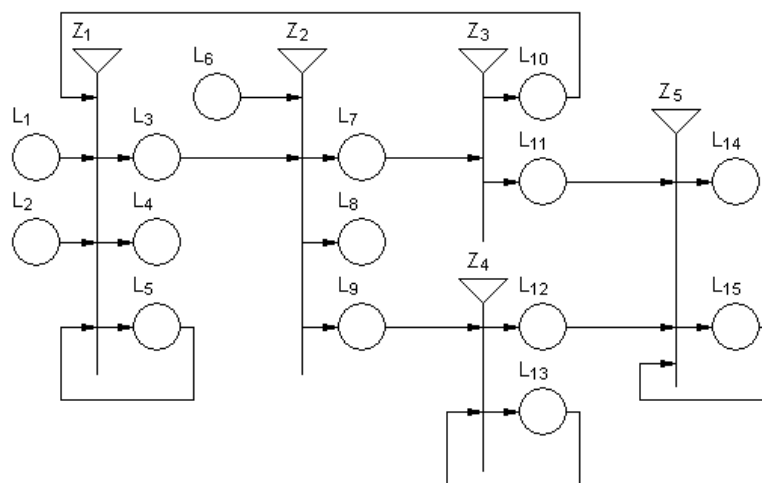
За разлика от задачите за избор измежду няколко отговора, вторият вид задачи, включени в проектираните досега тестове, е *конструктивна задача*, т.е. по-сложна и касаеща по-високото когнитивно равнище на приложение, според таксономията на В. Bloom на когнитивните цели. Поради конструктивния си характер обаче този тип задача не е включена в електронните тестове, реализирани с разширението *Quiz* в Ifigenia.org и засега въпросът дали това може да се случи в бъдеще остава отворен.

Този тип задача има за цел да упражни или провери разбирането на студентите за приложението на трите граф оператора, които съпоставят на дадена мрежа:

- един двуделен ориентиран граф, чиито множества от елементи представят съответно множеството на позициите и множеството на преходите;
- един насочен граф, чието множество от елементи представя само множеството от позициите на мрежата; и
- един насочен граф, чието множество от елементи представя само множеството от преходите на мрежата.

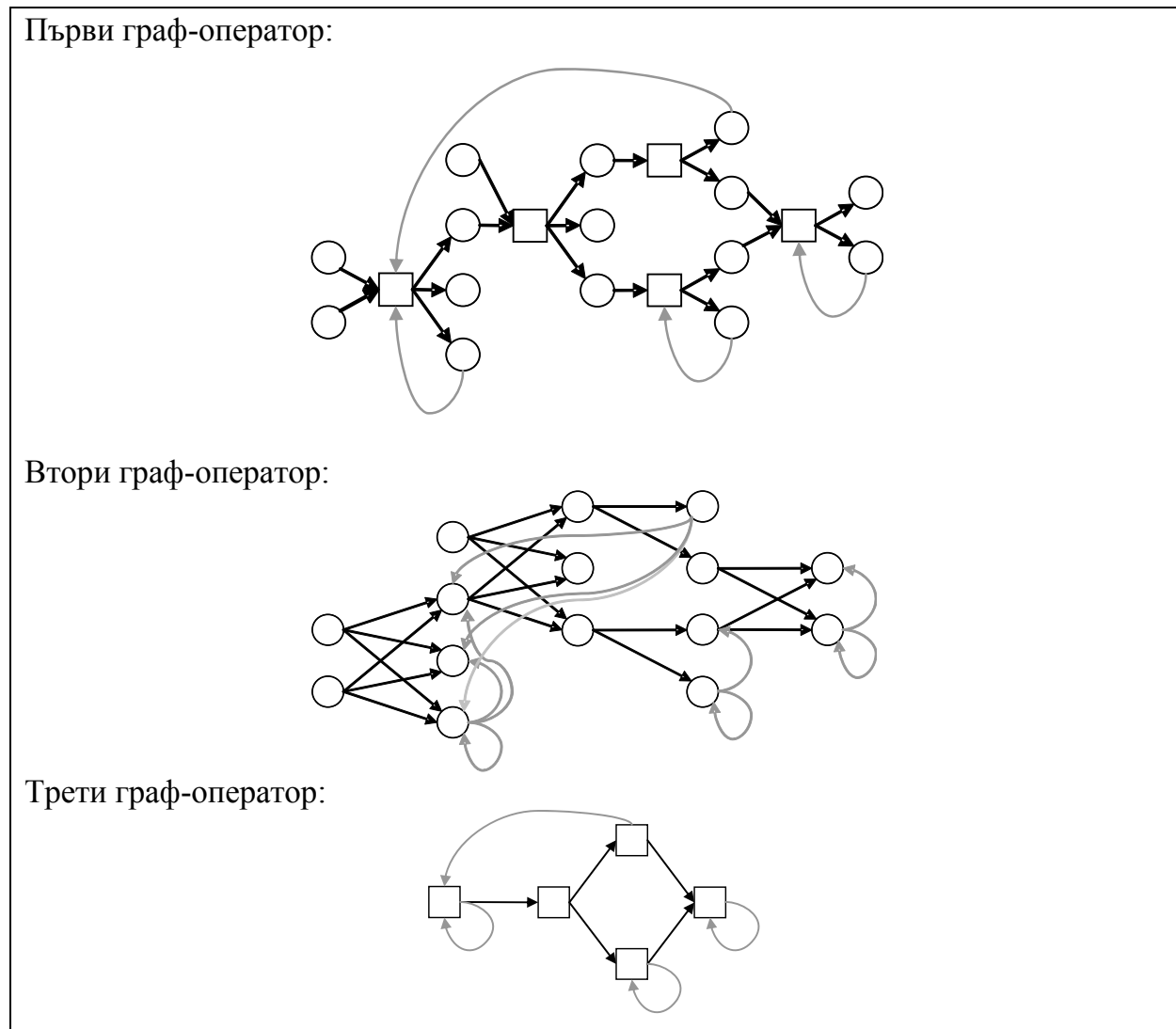
Примерна задача от този вид е дадена на Фиг. 4.7., а отговорите са дадени на Фиг. 4.8. Като цяло всички задачи от този вид съдържат по 5 прехода с 14-15 позиции и 3-4 дъги за обратна връзка. Тези задачи проверяват разбирането на обучаемите на тема графи (графичната структура на ОМ е по своему вид двуделен ориентиран граф) и способността им правилно да проследяват всички съществуващи връзки между позиции и преходи.

Приложете граф-операторите към следната обобщена мрежа:



Фигура 4.7. Примерна конструктивна тестова задача за приложение на процедурите на трансформация на ОМ модел в три различни графа, в зависимост от различните множества от елементи от статичната структура на мрежата.

Правилните отговори на поставената във Фиг. 4.7. задача са следните:



Фигура 4.8. Отговори на конструктивната задача от Фиг. 4.7.

Отново, таблицата по-долу представя разпределението на студентите от трите обучавани групи според представянето им.

Таблица 4.7.

	Правилен отговор		Частичен или грешен отговор		Без отговор		Общо
Група 1	1	9,09%	2	18,18%	8	72,73%	11
Група 2	5	31,25%	8	50,00%	3	18,75%	16
Група 3	5	31,25%	4	36,36%	2	18,18%	11
Общо	11	28,95%	14	36,84%	13	34,21%	38

Очаквано, нивата на успешно справяне с тази задача са по-ниски отколкото на първия тип задача. Обособени са следните класове от типични грешки

- Грешки, свързани с понятието за граф. Обичайна грешка е при прилагането на първия граф-оператор да не се прави разлика между позициите и преходите в мрежата и ги представят като (визуално) различни множества от върхове на двуделния граф, или смесват особеностите на първия и втория граф-оператор. Още по-често срещана грешка е пропусъкът дъгите на насочения граф да бъдат ориентирани.
- Грешки при изграждането на връзката между обобщените мрежи и графите. Някои студенти показват по-ниски възможности да свързват понятия от ОМ и графите, когато се представя един и същ абстрактен модел в термините на тези две различни области. Например, много студентите пропускат дъгите за обратна връзка в трите резултатни графа, особено във втория (което беше най-често допусканата грешка). Други просто игнорират части от модела, които намират за повтарящи се, или обратно, добавят допълнителни възли и дъги в графите, които не присъстват в оригиналния модел.

*
**

Би било интересно да се открият и други показателни типови грешки при отговорите на обучаемите и да се анализира процесът на преподаване на ОМ, както и образователните и практически предпоставки за прослушването на курс по ОМ. Особено предизвикателство би било сравняването на тестови резултати за една конкретна група обучаеми с представянето им при изграждане на реални ОМ модели, тъй като двете дейности оценяват различни когнитивни равнища, според йерархията на В. Bloom [80] на образователните цели: знаене → разбиране → приложение → анализ → синтез → оценка. Използвайки тестове, могат да се адресират и оценят само трите най-ниски от общо шестте когнитивни равнища. За сравнение, процесът на конструиране на ОМ модел изисква в добавка и умения от трите най-високи когнитивни равнища.

4.2.4. Реализация на електронни тестове по ОМ в средата на Ifigenia.org

С помощта на разширението *Quiz* на ядрото на софтуера на Ifigenia.org, *MediaWiki*, става възможна реализацията на всички описани по-горе тестови задачи от Таблицы 4.1.–4.4. в електронен тест за самооценяване. Задачите със свободен отговор като представените в Таблица 4.5., както и конструктивните задачи за прилагане на граф-оператори над ОМ, предложени в т. 4.2.3., на този етап не са реализирани в електронен вариант.

Тук накратко е представен синтаксисът на разширението *Quiz* за различните типове въпроси, следвайки документацията⁹.

Вмъкването на тест в произволна страница на уикито става, като се ограда съдържанието на теста с етикетите `<quiz>` и `</quiz>`. С това се вмъква стан-

⁹ Help:Quiz, Wikiversity, <http://en.wikiversity.org/w/index.php?title=Help:Quiz&oldid=892620>

дартизирано начало на теста, указващо броя точки, които се дават при правилен отговор (по подразбиране 1), броя точки, които се отнемат при грешен отговор (по подразбиране 0), както и бутон “Shuffle questions” за разбъркване на въпросите (активен само до момента преди да започне решаването на теста). За премахване на всяка от тези две опции по подразбиране, в етикета <quiz> могат да се добавят съответно параметрите display=simple и shuffle=none. Тестът по подразбиране съдържа и цетова идентификация при правилен, грешен и неподаден отговор, съответно зелено, червено и синьо.

В края на блока с въпроси стои бутонът “Submit” за приключване на теста, с което отговорите се маркират със съответните цветове. Всичко между етикетите <quiz> и </quiz> подлежи на разбор (*parsing*) от разширението *Quiz* и се превръща във въпроси, когато отговаря на синтаксиса.

Таблица 4.8.

В режим на редактиране	В режим на преглед
<pre><quiz> {Въпросите се поставят тук.} </quiz></pre>	
<pre><quiz display=simple> {Въпросите се поставят тук.} </quiz></pre>	

Между отделните въпроси трябва да има по един празен ред.

Въпросите, които имат *точно един* правилен отговор измежду произволен брой възможни, се представят със стандартния елемент “radio button” („радиобутон“) от графичния потребителски интерфейс. Формата на елемента е окръжност, която е запълнена с точка, в случай, че потребителят е избрал този отговор, или празна – в противен случай. С избирането на една от алтернативите, евентуално избраната преди това се отменя. В синтаксиса на *Quiz* този тип въпроси се означават с type="()", правилните отговори с плюс, грешните – с минус.

Въпросите, които могат да имат *повече от един* правилен отговор измежду произволен брой възможни, се представят със стандартния елемент “check box” („поле за отметка“) от графичния потребителски интерфейс. Формата на елемента е квадратна кутийка, която е запълнена с отметка, в случай, че потребителят е избрал този отговор, или празна – в противен случай. Избирането на втора и следваща алтернатива не отменя първоначално избраната.

В синтаксиса на *Quiz* този тип въпроси се означават с `type="[]"`, правилните отговори с плюс, грешните – с минус.

Таблица 4.9.

В режим на редактиране	<p>{Всяко разширение на мрежите на Петри е консервативно разширение. type="()"} - Истина + Лъжа</p> <p>{Всяка цветна мрежа на Петри може да бъде представена чрез редуцирана обобщена мрежа. type="()"} + Истина - Лъжа</p> <p>{Понятието „абстрактен преход“ представлява сечение на всички текущо активни преходи. type="()"} - Истина + Лъжа</p>
В режим на преглед (преди подаване на отговорите)	<p>1. Всяко разширение на мрежите на Петри е консервативно разширение.</p> <p><input type="radio"/> Истина <input type="radio"/> Лъжа</p> <p>2. Всяка цветна мрежа на Петри може да бъде представена чрез редуцирана обобщена мрежа.</p> <p><input type="radio"/> Истина <input type="radio"/> Лъжа</p> <p>3. Понятието „абстрактен преход“ представлява сечение на всички текущо активни преходи.</p> <p><input type="radio"/> Истина <input type="radio"/> Лъжа</p> <p style="text-align: center;"><input type="button" value="Submit"/></p>

**В режим на преглед
(след подаване на отговорите)**

1. Всяко разширение на мрежите на Петри е консервативно разширение.
 - Истина
 - Лъжа
2. Всяка цветна мрежа на Петри може да бъде представена чрез редуцирана обобщена мрежа.
 - Истина
 - Лъжа
3. Понятието „абстрактен преход“ представлява сечение на всички текущо активни преходи.
 - Истина
 - Лъжа

Submit Reset

Your score is 1 / 3

Тестовите задачи за намиране на съответствия могат да бъдат оформени, следвайки модела:

```
{Въпрос
| type="()" }
| A | B | C
+-- Алтернатива 1 (Отговор: A)
--+ Алтернатива 2 (Отговор: C)
-+- Алтернатива 3 (Отговор: B)
```

Така задачата от Таблица 4.2 в синтаксиса на *Quiz* добива вида:

Таблица 4.10.

В режим на редактиране

```
{В таблицата по-долу отбележете кой компонент от дефиницията на
обобщена мрежа отразява нейната (A) статика, (B) динамика, (C)
хронология, (D) памет
| type="()" }
| Статика<br/>A | Динамика<br/>B | Хронология<br/>C | Памет<br/>D
-+-- Функцията '' $\theta_K$ '', даваща следващия момент от време,
когато дадено ядро ще влезе в мрежата
--+- Моментът от време ''T'', когато мрежата започва да
функционира
+--- Функцията '' $\theta_2$ '', даваща продължителността на
активното състояние на даден преход
---+ Функция ''b'', даваща максималния брой на характеристиките,
пазени
в историята на ядрото
+--- Функция ''c'', даваща капацитетите на позициите
+--- Функция '' $p_A$ '', даваща приоритетите на преходите
```

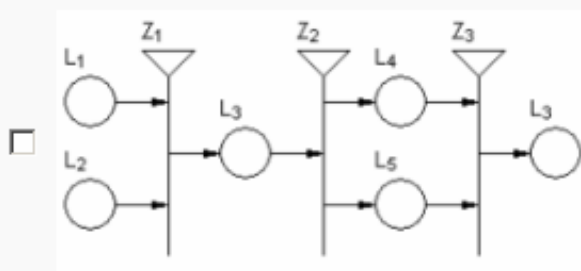
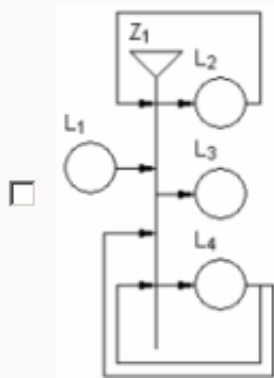
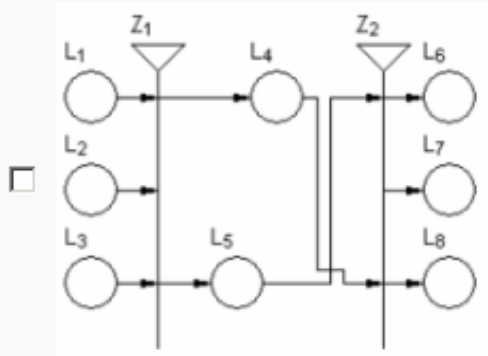
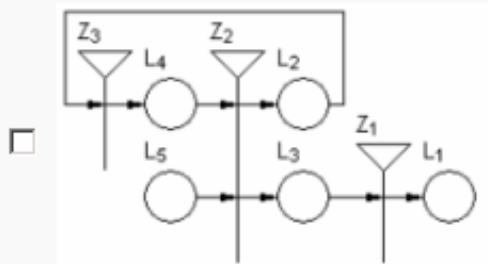
В режим на преглед (преди подаване на отговорите)	1. В таблицата по-долу отбележете кой компонент от дефиницията на обобщена мрежа отразява нейната (A) статика, (B) динамика, (C) хронология, (D) памет				
	Статика	Динамика	Хронология	Памет	
	A	B	C	D	
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Функцията θ_k , даваща следващия момент от време, когато дадено ядро ще влезе в мрежата
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Моментът от време T , когато мрежата започва да функционира
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Функцията θ_2 , даваща продължителността на активното състояние на даден преход
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Функция b , даваща максималния брой на характеристиките, пазени
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Функция c , даваща капацитетите на позициите	
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Функция λ_d , даваща приоритетите на преходите	
<input type="button" value="Submit"/>					

За отбелязване при реализацията на задача 3 от Таблица 4.3. е, че възможните отговори са изображения. За целта първо в уикито трябва да бъдат локално качени файловете и да се извикат с уики-командата `[[Image:...]]`. Добавянето на анимирани изображения, аудио или видео е аналогично. За илюстриране и на случая с повече от един правилен отговор, ще дадем как изглежда кодът и резултатът със задача, различна от тази под номер 3 в Таблица 4.3.

Таблица 4.11.

В режим на редактиране	{Отбележете в кои от следните графични структури на обобщеномрежови модели има допуснати грешки?
	<code> type="[]"</code>
	<code>- [[Image:Gn-test-1a.gif]]</code>
	<code>- [[Image:Gn-test-1b.gif]]</code>
	<code>+ [[Image:Gn-test-1c.gif]]</code> <code>+ [[Image:Gn-test-1d.gif]]</code>

1. Отбележете в кои от следните графични структури на обобщеномрежови модели има допуснати грешки?



Submit

Задачите за попълване на пропуснати думи (*Cloze test*) се получават чрез синтаксиса на *Quiz*, като им се зададе тип `|type="{}"` и очакваният правилен отговор се зададе между двойка фигурни скоби на съответното оставено празно място в изречението. За отбелязване е, че по подразбиране очакваният отговор е чувствителен на главни и малки букви, но разликата може да бъде премахната с добавката на параметър `(i)`, който идва от английското *case-insensitive*. Дължината на празното поле, респективно броят позволени символи за въвеждане в

Таблица 4.13.

<p>В режим на редактиране</p>	<pre>{Съберете следните две индексирани (0,1)-матрици.
 [[Image:Gn-test-im.gif]] type="{ }" &nbsp; { p _1 } { q _1 } { r _1 } { s _1 } { t _1 } ----- { a _1 } { 1 _1 } { 1 _1 } { 1 _1 } { 0 _1 } { 0 _1 } { b _1 } { 1 _1 } { 1 _1 } { 0 _1 } { 1 _1 } { 1 _1 } { c _1 } { 1 _1 } { 0 _1 } { 1 _1 } { 1 _1 } { 0 _1 } { d _1 } { 1 _1 } { 0 _1 } { 1 _1 } { 1 _1 } { 0 _1 } { e _1 } { 1 _1 } { 0 _1 } { 0 _1 } { 1 _1 } { 0 _1 }</pre>																																																																																										
<p>В режим на преглед (преди подаване на отговорите)</p>	<p>1. Съберете следните две индексирани (0,1)-матрици.</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">a</td> <td style="padding: 5px;">p</td> <td style="padding: 5px;">q</td> <td style="padding: 5px;">r</td> <td style="padding: 5px;">s</td> <td style="padding: 5px;">\oplus</td> <td style="border-right: 1px solid black; padding: 5px;">b</td> <td style="padding: 5px;">s</td> <td style="padding: 5px;">t</td> <td style="padding: 5px;">p</td> <td style="padding: 5px;">$=$</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">a</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">0</td> <td></td> <td style="border-right: 1px solid black; padding: 5px;">b</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">1</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">b</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">0</td> <td></td> <td style="border-right: 1px solid black; padding: 5px;">a</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">c</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">1</td> <td></td> <td style="border-right: 1px solid black; padding: 5px;">e</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">d</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">1</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table> <div style="margin-top: 10px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; width: 5%;"></td> <td style="width: 15%;"><input type="text"/></td> <td style="width: 15%;"><input type="text"/></td> <td style="width: 15%;"><input type="text"/></td> <td style="width: 15%;"><input type="text"/></td> <td style="width: 15%;"><input type="text"/></td> <td style="width: 15%;"><input type="text"/></td> </tr> <tr> <td style="border-right: 1px solid black;"><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td style="border-right: 1px solid black;"><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td style="border-right: 1px solid black;"><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td style="border-right: 1px solid black;"><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> </table> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Submit"/> </div> </div>	a	p	q	r	s	\oplus	b	s	t	p	$=$	a	1	1	1	0		b	1	1	1		b	0	1	0	0		a	0	0	1		c	1	0	1	1		e	1	0	1		d	1	0	1	1								<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
a	p	q	r	s	\oplus	b	s	t	p	$=$																																																																																	
a	1	1	1	0		b	1	1	1																																																																																		
b	0	1	0	0		a	0	0	1																																																																																		
c	1	0	1	1		e	1	0	1																																																																																		
d	1	0	1	1																																																																																							
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>																																																																																					
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>																																																																																					
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>																																																																																					
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>																																																																																					
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>																																																																																					

На последно място, възможностите на *Quiz* не се изчерпват само с тези типове въпроси. Разширението дава възможност да се задават и въпроси, чиито отговори са точни числа или числа в определени интервали, или с +/- определено отклонение, зададено в проценти.

Възможно е и фино регулиране на точките, които се дават или отнемат при верни/грешни отговори, като и настройка на тегловите коефициенти на отделните отговори в рамките на въпрос и на отделните въпроси в рамките на теста. Тъй като *Quiz*, както и *MediaWiki*, е софтуер под свободен лиценз, при нужда и желание е възможно да бъде обогатен и с други допълнителни настройки и добавки.

*
**

Понастоящем базата от данни на Ifigenia.org предлага свободен достъп до над 50 теми от понятийния апарат на ИРМ и ОМ и до повече от 300 публикации от тези области в PDF формат (част от които специално цифровизирани за качване в сайта, тъй като до този момент са били трудно достъпни, макар и основополагащи за изследванията в тези две научни области). В рамките на различни проекти с участието на автора тази база понастоящем е, а и в бъдеще ще бъде допълвана.

Откакто съществува сайтът, чрез новинарската му секция са били обявени повече от 10 конференции и докторантски спецкурсове. Също е обявявано и излизането от печат на нови книги в областта на ОМ и ИРМ.

В страници от *Private* частта на сайта досега са разработени четири проектни предложения от различни екипи. В публичната част на сайта се намират уебстраниците на няколко спечелени проекта, финансирани от Фонд „Научни изследвания“ на МОМН, и на един вътрешноуниверситетски проект на Университет „Проф. д-р Асен Златаров“ – Бургас, чиято тематика изцяло попада в обхвата му. По този начин се подпомага видимостта на получените резултати и се създава органична връзка между проектите и научните колективи, които работят по тях, от една страна, и научните общности, които в най-силна степен се ползват от научните достижения по проектите и ресурсите на Ifigenia.org, от друга страна.

Синтезираният опит по изграждането на Ifigenia.org – едно уики в помощ на световните научни общности по ИРМ и ОМ – може да се използва при създаване и на други подобни уебсайтове, целящи да отговорят на потребността от информационни ресурси, разнообразие от услуги и насърчаване на общностен дух. По отношение администрирането на сайта би било от полза наличие на известен предварителен опит с ползването, инсталирането и конфигурирането на уики. По-важно е обаче администраторите да имат експертния опит в проблемната област на уикито, тъй като доста от настройките и допълнителната му персонализация се правят основно с оглед особеностите на тематиката и нуждите на редакторската общност при създаване на свързано с темата уеб съдържание. За крайните потребители и редактори на уикито обаче повечето от тези подробности са спестени и остава удоволствието и ползата от конструктивното сътрудничество, което уики софтуерът успешно стимулира.

Заклучение – основни резултати

Дисертационният труд включва дефиниции на две нови разширения на понятието „обобщена мрежа“ и доказателствата, че те са консервативни. Дадени са алгоритми за движение на ядрата в петте дефинирани до сега интуиционистки размити обобщени мрежи, както и решение на един открит проблем, съществуващ в теорията на обобщените мрежи повече от 20 години. Представени са възможностите, оптимизационните процедури по алгоритъма на мравките да се използват за оптимизиране на движението на ядрата в обобщени мрежи, както и обратно – чрез обобщени мрежи да се оптимизира работата на алгоритъма на мравките и генетичните алгоритми при решаване на конкретни проблеми.

При изготвянето на дисертационния труд се появиха редица идеи за бъдещи изследвания. Например, проблемът за представимостта на произволни обобщеномрежови преходи чрез композиция на $H_{2,2}$ преходи, от една страна, и проблемът за консервативността на разширенията на обобщените мрежи, от друга, навеждат на въпроса за възможностите да се конструират идентични („синонимни“) по обхват и действие обобщеномрежови модели, които обаче се различават по някои от своите елементи: инфраструктурни елементи (брой преходи и позиции, функции на приоритета и капацитета им), логически елементи (индексирани матрици с предикати за условията на преходите, тип на преходите), динамични елементи (ядра, начални стойности на характеристиките на ядрата и характеристични функции), темпорални елементи (времева скала, елементарна времева стъпка, продължителност на изчисление) и т.н. На практика въпросът се свежда до това, с кои от (незадължителните) компоненти в дефиницията на ОМ може да се направи компромис, в какви ситуации и на каква цена (от гледна точка на изчислително време и памет). Този проблем не е бил обект на разглеждане до момента.

Освен от чисто методологически интерес така поставеният въпрос е важен и от гледна точка на новопредложения дидактически аспект на теорията на обобщените мрежи. Изясняването му би дало възможност както за формиране на задълбочено разбиране у обучаеми и потенциални бъдещи специалисти по моделиране с обобщени мрежи за „относителната тежест“ на всеки от компонентите в дефиницията, така и за повече адекватност, гъвкавост и стандартизираност при изграждането на обобщеномрежови модели на реални процеси.

Приносителите в настоящия дисертационен труд са от научен, научно-приложен и приложен характер.

Научните приноси като цяло са ориентирани към дефиниране на нови разширения на понятието „обобщена мрежа“, доказване на тяхната консервативност и построяване на алгоритмите за движение на ядрата в тях на ниво отделен преход и цяла мрежа.

1. За съществуващите интуиционистки размити обобщени мрежи от всеки един дефинираните досега четири типа са построени липсващите до момента алгоритми за движение на ядрата в техните преходи и в целите мрежи.
2. Дефиниран е нов, пети, вид на интуиционистки размити обобщени мрежи. Доказана е теорема, че този вид обобщена мрежа е консервативно разширение на стандартното понятие за обобщена мрежа, т.е. функционирането и резултатите от работата ѝ може да се опише в термините на стандартна обобщена мрежа. Построени са алгоритмите за движение на ядрата в интуиционистки размитите обобщени мрежи от пети вид на ниво отделен преход и цяла мрежа.
3. Дефинирана е обобщена мрежа с обеми на ядрата. Доказана е теорема, че този вид обобщена мрежа е консервативно разширение на стандартното понятие за обобщена мрежа. Описани са съответните за това разширение два алгоритъма за движение на ядрата – на ниво отделен преход и на ниво цяла мрежа.
4. Предложено е цялостно решение на проблема за представимост на преходите в обобщените мрежи, формулиран през 1991 г., който изисква да се намери процедура, еднозначно представяща произволно сложен преход като композиция от преходи с два входа и два изхода. Доказано е, че предложеното решение е минималното възможно решение.
5. Формулирано е предложение как чрез присвояването на интуиционистки размити оценки на проведени обобщеномрежови симулации да се съставят прогнози за бъдещото изпълнение на обобщеномрежов модел.

Научно-приложните приноси са следните:

1. Предложена е универсалната обобщена мрежа за класа алгоритми за оптимизация по метода на мравките, описваща начина на функциониране и резултатите от тяхната работа. Построена е обобщена мрежа, която включва предложената мрежа като оптимизационен компонент, т.е. подмрежа, чиито резултати от работата определят оптималното движение на ядрата в новата мрежа.
2. Създаден е обобщеномрежов модел, отразяващ паралелната работа на реален процес, на алгоритъм на мравките и на генетичен алгоритъм. Съчетани в един хибриден метаевристичен алгоритъм, тези двата алгоритъма оптимизират работата на реалния процес в рамките на обобщеномрежовия модел.
3. Създадени са и два обобщеномрежови модела на основните алгоритми за предотвратяване на редакционни конфликти в системи с контрол на версиите, а именно „Заклучи–Промени–Отключи“ и „Копирай–Промени–Слей“.

Приложният принос се състои в предлагането на цялостно изложение за нов дидактичен аспект на теорията на обобщените мрежи и в показаните началните стъпки по реализацията му в уеб среда, изградена с технологията уики. За усилията си в тази посока докторантът получава II награда на VIII национална научно-практическа сесия на Федерацията на научно-техническите съюзи (2010 г.) и Наградата на БАН „Иван Евстратиев Гешов“ за най-млади учени до 30 години в областта на техническите науки (2011 г.).

Декларация за оригиналност на резултатите

Декларирам, че настоящата дисертация съдържа оригинални резултати, получени при проведени от мен научни изследвания (с подкрепата и съдействието на научния ми ръководител). Резултатите, които са получени, описани и/или публикувани от други учени, са надлежно и подробно цитирани в библиографията.

Настоящата дисертация не е прилагана за придобиване на научна степен в друго висше училище, университет или научен институт.

Подпис:

Вася К. Атанасова

Библиография

- [1] Аладжов, Х., Обобщено-мрежови модели на обучение и самоорганизация и приложението им за управление на двигателна активност. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ЕКТ, София, 2002.
- [2] Атанасов, К., Въведение в теорията на обобщените мрежи. Бургас, Понтика Принт, 1992.
- [3] Атанасов, К. Обобщени мрежи. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по математика, София, 1986.
- [4] Атанасов, К. Обобщени мрежи: теория и приложения. Дисертационен труд за придобиване на научната степен „Доктор на техническите науки“, СНС по ЕКТ, София, 1997.
- [5] Атанасова, В., Свободен софтуер в науката, Годишник на секция „Информатика“, Съюз на учените в България, Том 2, 2009, 10–13.
- [6] Атанасова, В., Уики технологии в електронното обучение, Годишник на секция „Информатика“, Съюз на учените в България, Том 1, 2008, 116–122
- [7] Атанасова, В. Уики технологии в помощ на науката. Примерът с Ifigenia.org, Сборник от 8^{ма} Национална младежка научно-практическа сесия на ФНТС, 10-11 май 2010, 83–88.
- [8] Барвайс, Д. (ред.), Справочная книга по математической логике. Часть 4: Теория доказательств и конструктивная математика, Москва, „Наука“, 1983.
- [9] Будакова, Д., Моделиране и изследване аспекти на поведение. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ЕКТ, София, 2005.
- [10] Вардева, И., Модели и процеси в мрежова сигурност чрез обобщени мрежи. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, ИБФБМИ-БАН, София, 2012.
- [11] Георгиев, П., Обобщено-мрежови модели на системи базирани на знания. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ЕКТ, София, 1998.
- [12] Гочев, В., Обобщени мрежи и размити мрежи на Петри. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по АСУ, София, 2005.
- [13] Гочев, В., П. Гочева .NET реализация на обобщени мрежи. Изд. на Висше училище „Колеж по телекомуникации и пощи“, 2012, ISBN 978-954-9332-75-9.
- [14] Димитров, Д., Софтуерни продукти, реализиращи обобщените мрежи, Годишник на секция „Информатика“ към Съюза на учените в България, Том 3, 2010, 37–50.
- [15] Димитров, Д., Графична среда за моделиране и симулация с обобщени мрежи, Годишник на секция „Информатика“ към Съюза на учените в България, Том 3, 2010, 51–66.
- [16] Димитрова, Л., Алгоритмическое и программное обеспечение для анализа и принятия решения при создании технологических схем непрерывной ректификации. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, Московской государственной академии тонкой химической технологии им. М. Ломоносова, Москва, 2001.
- [17] Димов, А., Редуциране на комплексността при разработване на софтуерни системи – методи за описание на софтуерната архитектура. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ИММ, 2006.

- [18] Койчева, Е., Entwurfsbergleitende Leistungsanalyse mit der Modellierungssprache UML, dem MARTE-Profil und generalizierten Netzen. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, Технически университет, Дрезден, 2010.
- [19] Колев, Б., Обобщено-мрежови модели на процеси в системи за управление на релационни бази данни и интуиционистки размити релационни бази данни. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ЕКТ, София, 2006.
- [20] Маринова, М., Уикипедия – възможност за внедряване на нов образователен ресурс със световна значимост във формалното обучение по английски език в България, Сборник доклади на конф. „Диалогът между поколенията и обществените структури чрез училищната институция“, София, България, 16–17 април 2010.
- [21] Минчев, З., Обобщено-мрежови модели и алгоритми с интуиционистки размити множества за представяне и управление на мобилни роботи в непозната среда. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по АСУ, София, 2006.
- [22] Мохамед ал Робиех, З. С., Изследване на ефективността на информационна система за издаване на задгранични паспорти чрез обобщени мрежи. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ЕКТ, София, 2002.
- [23] Недев, С., Базова архитектура и специализирани апаратни средства в автоматизирана система за резервация и продажба на билети в транспорта. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ЕКТ, София, 1994.
- [24] Орозова, Д., Обобщеномрежови модели на интелигентни среди за обучение, Академично издателство „Проф. Марин Дринов“, София, 2011.
- [25] Пенева, Д. Обобщено-мрежово моделиране на структури и процеси в системата „е-здраве“. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по АСУ, София, 2010.
- [26] Пенчева, Т., Моделиране на клас биотехнологични процеси като обекти с разпределени параметри. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по АСУ, София, 2003.
- [27] Радева, В., Методи за изграждане на обобщено-мрежови модели. Обобщено-мрежови модели на реални системи от областта на изкуствения интелект. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ЕКТ, София, 2003.
- [28] Сотиров, С., Представяне на невронни мрежи с обобщени мрежи и тяхното приложение в диагностиката на електронни схеми. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ЕКТ, София, 2006.
- [29] Сотирова, Е., Моделиране на информационните процеси в университет чрез обобщени мрежи. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ЕКТ, София, 2004.
- [30] Стамболов, Г., Методология за изграждане на реконфигуриращи се производствени системи. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по МТМ, София, 2005.
- [31] Стефанова-Павлова, М., Моделиране на гъвкави автоматизирани производствени системи с обобщени мрежи. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ЕКТ, София, 2001.
- [32] Стоянов, П., Изследване на вътрешни конфликти в пространствени комутатори за асинхронни компютърни мрежи. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ЕКТ, София, 2005.
- [33] Тасева, В., Моделиране на процеси в здравни структури чрез обобщени мрежи. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по АСУ, София, 2007.

- [34] Тодорова, Л., Система за анализ на състоянието на пациенти при отвикване на продължителна механична вентилация. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по АСУ, София, 2006.
- [35] Тодорова, М. Построяване на коректни обектно-ориентирани програми чрез изграждане на техни обобщени мрежови модели, Годишник на секция „Информатика“, Съюз на учените в България, том 4, 2011, 1–28.
- [36] Трахтенброт, Б., Я. Барздинь, Конечные автоматы – поведение и синтез, Москва, Наука, 1970.
- [37] Чешмеджиев, П., Управление и синхронизация на паралелни логистични процеси с обобщени мрежи. Дисертационен труд за придобиване на образователната и научна степен „Доктор“, СНС по ЕКТ, София, 2010.
- [38] Alexieva, J., Choy, E., Koucheva, E., Chapter 8 Review and bibliography of generalized nets theory and applications in A Survey of Generalized Nets. Choy, E., Krawczak, M., Shannon, A., Szmidt, E. (eds.) Raffles KvB Monograph, No. 10, Sydney, 2007.
- [39] Atanassov, K., Conditions in Generalized nets. Proc. of the XIII Spring Conf. of the Union of Bulg. Math., Sunny Beach, April 1984, 219–226.
- [40] Atanassov K., Generalized index matrices, Comptes rendus de l'Academie Bulgare des Sciences, Vol. 40, 1987, No.11, 15–18.
- [41] Atanassov, K., Generalized Nets. World Scientific, Singapore, 1991.
- [42] Atanassov, K. Generalized Nets and Systems Theory. "Prof. M. Drinov" Academic Publishing House, Sofia, 1997.
- [43] Atanassov, K. Generalized nets and their fuzzings. AMSE Review, Vol. 2, 1985, No 3, 39–49.
- [44] Atanassov, K. Generalized Nets in Artificial Intelligence. Vol. 1: Generalized nets and Expert Systems. "Prof. M. Drinov" Academic Publishing House, Sofia, 1998.
- [45] Atanassov, K. Intuitionistic Fuzzy Sets: Theory and Applications. Springer-Verlag, Heidelberg, 1999.
- [46] Atanassov, K., Methodological aspect of the theory of generalized nets. I. AMSE Review, Vol. 12, No. 3, 1990, 59–64.
- [47] Atanassov K., Methodological aspect of the theory of generalized nets. II, AMSE Review, Vol. 20, No. 4, 1992, 53–64.
- [48] Atanassov, K., On Generalized Nets Theory. "Prof. M. Drinov" Academic Publishing House, Sofia, 2007.
- [49] Atanassov, K. On index matrices, Part 1: Standard cases. Advanced Studies in Contemporary Mathematics, Vol. 20, 2010, No. 2, 291–302.
- [50] Atanassov, K. On Intuitionistic Fuzzy Sets Theory. Springer, Berlin, 2012.
- [51] Atanassov, K., On the concept "Generalized net", AMSE Review, Vol. 1 (1984), No. 3, 39–48.
- [52] Atanassov, K., Theory of Generalized nets (an algebraic aspect), Advances in Modelling & Simulation, AMSE Press Vol. 1 (1984), No. 2, 27–33.
- [53] Atanassov, K., Theory of Generalized nets (an algebraic aspect. II). Proc. of II Int. Symp. "Automation and Scientific Instrumentation", Varna, May 1983, 391–396.
- [54] Atanassov, K., Theory of Generalized nets (a functional aspect). Сб. докл. от IX Нац. младежка школа „Математически методи в информатиката“, Варна, авг. 1983, 112–113.
- [55] Atanassov, K., Theory of Generalized nets (a logical aspect). Proc. of Summer School on Math. Logic and its Applications, Primorsko, Sept. 1983, 26–29.
- [56] Atanassov, K. (Ed.) Applications of Generalized Nets, World Scientific, Singapore, 1993.
- [57] Atanassov, K., H. Aladjov. Generalized Nets in Artificial Intelligence. Vol. 2: Generalized nets and Machine Learning. "Prof. M. Drinov" Academic Publishing House, Sofia, 2001.
- [58] Atanassov, K., V. Atanassova, P. Chountas, A. Shannon. Generalized nets with places, having intuitionistic fuzzy capacities. Notes on Intuitionistic Fuzzy Sets, Vol. 17, 2011, No. 4, 21–28.
- [59] Atanassov, K., V. Chakarov, A. Shannon, J. Sorsich. Generalized Net Models of the Humab Body. "Prof. M. Drinov" Academic Publishing House, Sofia, 2008.

- [60] Atanassov, K., S. Dantchev, Generalized net realization of Kolmogorov's algorithm. *Issues in Intuitionistic Fuzzy Sets and Generalized Nets*, Vol. 4, 2007, 65-74.
- [61] Atanassov, K., M. Daskalov, P. Georgiev, S. Kim, Y. Kim, N. Nikolov, A. Shannon, J. Sorsich. *Generalized Nets in Neurology*. "Prof. M. Drinov" Academic Publishing House, Sofia, 1997.
- [62] Atanassov, K., D. Dimitrov, V. Atanassova, Algorithms for tokens transfer in different types of intuitionistic fuzzy generalized nets. *Cybernetics and Information Technologies*, Volume 10, No 4, 2010, 22–35.
- [63] Atanassov, K., E. Dimitrov, Theorem for the normal form of the transitions in E-nets, *Proc. of International School on Automation of the Scientific Research*, Varna, 1982, 129–134
- [64] Atanassov, K., G. Gluhchev, S. Hadjitodorov, A. Shannon, V. Vassilev. *Generalized Nets and Pattern Recognition*. KvB Visual Concepts Pty Ltd, Monograph No. 6, Sydney, 2003.
- [65] Atanassov, K., G. Gluhchev, S. Hadjitodorov, J. Kacprzyk, A. Shannon, E. Szmidt, V. Vassilev. *Generalized Nets Decision Making and Pattern Recognition*. Warsaw School of Information Technology, Warszawa, 2006.
- [66] Atanassov, K., G. Gluhchev, S. Hadjitodorov, A. Shannon, V. Vassilev. *Generalized Nets and Pattern Recognition*. KvB Visual Concepts Pty Ltd, Monograph No. 6, Sydney, 2003.
- [67] Atanassov, K., V. Tasseva, T. Trifonov. Modification of the Algorithm for Token Transfer in Generalized Nets. *Cybernetics and Information Technologies*, Vol. 7, 2007, No. 1, 62–66.
- [68] Atanassov K., N. Nikolov. Intuitionistic fuzzy generalized nets: definitions, properties, applications. In: *Systematic Organization of Information in Fuzzy Systems* (P. Melo-Pinto, H.-N. Teodorescu and T. Fukuda, Eds.), IOS Press, Amsterdam, 2003, 161–175.
- [69] Atanassova, V., Design of Training Tests on Generalized Nets, *Proc. of 5th International IEEE Symposium 'Intelligent Systems'*, 7–9 July 2010, London, UK, 327–330, 2010.
- [70] Atanassova, V., Ifigenia – Doing IFS and GN the wiki way, *Advanced in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics*. Volume 2: Applications, EXIT Publ. House, Warsaw, Poland, 2008, 13–18.
- [71] Atanassova, V. Generalized Nets with Volumetric Tokens, *Comptes rendus de l'Academie Bulgare des Sciences*, Vol. 65, 2012, No. 11, 1489–1498.
- [72] Atanassova, V., Generalized net models of conflict resolution approaches in version control systems. Part 1: Lock-Modify-Unlock, *Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics*. Vol. II: Applications, 2010, 13–24.
- [73] Atanassova, V., On Intuitionistic Fuzzy Approach to Generalized Net Prognostics. In: *New Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics*. Volume II: Applications. (K. Atanassov, M. Baczynski, J. Drewniak, J. Kacprzyk, M. Krawczak, E. Szmidt, M. Wygralak, S. Zadrozny, Eds.), SRI-PAS, Warsaw, 2012, 1–12, ISBN: 9788389475411
- [74] Atanassova, V. The Minimal Solution of a Problem in Generalized Nets, *Proc. of the 6th IEEE Int. Conference on Intelligent Systems '2012*, 6–8 Sept. 2012, Sofia, Bulgaria, Vol. II, 159–163.
- [75] Atanassova, V. Utilizing the Model Graphic Structures when Teaching Generalized Nets, In: *Recent Advances in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics*, Vol. 2, 2010, 7–17.
- [76] Atanassova, V., K. Atanassov. Ant Colony Optimization Approach to Tokens' Movement within Generalized Nets, *Lecture Notes in Computer Science*, Vol. 6046, Springer, 2011, 240–247.
- [77] Atanassova, V., S. Fidanova, P. Chountas, K. Atanassov. A generalized net with an ACO-algorithm optimization component, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 7116, 2012, 190–197.
- [78] Atanassova, V., S. Fidanova, I. Popchev, P. Chountas, Chapter 5. Generalized Nets, ACO Algorithms, and Genetic Algorithms, In: *Monte Carlo Methods and Applications*, *Proc. of the 8th IMACS Seminar on Monte Carlo Methods*, August 29 – September 2, 2011, Borovets, Bulgaria (Sabelfeld, K., I. Dimov, Eds.), De Gruyter, 2012, 39–46.

- [79] Atanassova, V., Georgiev, P., Generalized net models of conflict resolution approaches in version control systems. Part 2: Copy–Modify–Merge, Proc. of 10th Int. Workshop on Generalized Nets, 5 Dec. 2009, 14–21.
- [80] Bloom, B. S., Taxonomy of Educational Objectives. Allyn and Bacon, Boston, MA, 1984.
- [81] Böhm, C., G. Jacopini. Flow diagrams, Turing machines and languages with only two formation rules, Comm. ACM, Vol. 9, 1966, pp. 366–371.
- [82] Chountas, P. On Uncertain Data & Knowledge Engineering: The Case for Intuitionistic Fuzzy Sets. Дисертационен труд за придобиване на научната степен „Доктор на техническите науки“, СНС по ЕКТ, 2009.
- [83] Chountas, P., B. Kolev, E. Rogova, V. Tasseva, K. Atanassov. Generalized Nets in Artificial Intelligence. Vol. 4: Generalized nets, Uncertain Data and Knowledge Engineering. "Prof. M. Drinov" Academic Publishing House, Sofia, 2007.
- [84] Choy, Y.-H. Risk Factors Associated with Renal Disease in Diabetes. Mellitus, Ph.D. thesis, Univ. of Technology, Sydney (Australia), 2002.
- [85] Collins-Sussman B., Fitzpatrick B. W., Pilato C. M. (2004), Version Control with Subversion, O'Reilly Media, <http://svnbook.red-bean.com/>
- [86] Cooney, L., Wiki as a Knowledge Management Tool (PhD thesis), <http://cooney.wikidot.com/>, 31.08.2006.
- [87] Cunningham, W., What is wiki, Wiki.org, <http://wiki.org/wiki.cgi?WhatIsWiki>, 27.06.2002.
- [88] Dimitrov, D. Optimized algorithm for token transfer in generalized nets. In: Recent Advances in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics, Vol. 1, Warsaw, SRI PAS, 2010, 63–68.
- [89] Dimitrov, E. Grundlegende Strukturen von Netzsimulationssystemen und ihre Implementation für Experimente mit Modellen auf Basis modifizierter Petri-Netze: das System MARS, Dissertation, Humboldt-Uni, Berlin, 1985.
- [90] Dorigo, M. Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Italy, 1992 (in Italian).
- [91] Dorigo, M., Di Caro G. Gambardella M. (1999) Ant Algorithms for Distributed Discrete Optimization. J. of Artificial Life. 5, 137–172.
- [92] Dorigo, M. L. M. Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computing. 1, 1999, 53–66.
- [93] Dorigo, M., V. Maniezzo, A. Colomi. The Ant System: Optimization by a Colony of Cooperative Agents. IEEE Transactions on Systems, Man and Cybernetics part B. 26(1), 1996, 29–41.
- [94] Dorigo, M., T. Stützle. Ant Colony Optimization, The MIT Press, 2004.
- [95] El-Ghazali Talbi, Metaheuristics: From Design to Implementation, John Wiley and Sons, 2009.
- [96] Felder, R. Designing tests to maximize learning. J. Prof. Issues in Engr. Education & Practice, Vol. 128, 2002, No. 1, 1–3. <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/TestingTips.htm>
- [97] Fidanova S. ACO Algorithm with Additional Reinforcement. In Dorigo M., Di Caro G. Sampels M. eds. From Ant Colonies to Artificial Ants, Lecture Notes in Computer Science. 2463, 2002, 292–293.
- [98] Fidanova, S., Atanassov, K. Generalized nets as tools for modelling of the ant colony optimization algorithms. Lecture Notes in Computer Science, Vol. 5910, 2010, 326–333.
- [99] Fidanova S., Atanassov K. Generalized net models and intuitionistic fuzzy estimation of the process of ant colony optimization Issues on Intuitionistic Fuzzy Sets and Generalized Nets, Vol 8, 2010, 109–124.
- [100] Fidanova, S., Atanassov K., Generalized net models for the process of hybrid ant colony optimization, Comptes Rendus de l'Academie Bulgare des Sciences, Vol. 62, 2009, No. 3, 315–322.

- [101] Fidanova, S., Atanassov, K. Generalized net models of the process of ant colony optimization with intuitionistic fuzzy estimations. Proceedings of the Ninth International Workshop on Generalized Nets (K. Atanassov and A. Shannon, Eds.), Sofia, 4 July 2008, Vol. 1, 41–48.
- [102] Fidanova, S., Atanassov K. Generalized Net Models of the Process of Ant Colony Optimization, Issues on Intuitionistic Fuzzy Sets and Generalized Nets, Vol 7, 2008, 108–114.
- [103] Fidanova, S., K. Atanassov, P. Marinov. Intuitionistic fuzzy estimations of the ant colony optimization. Proc. of 3rd Annual Meeting of the Bulgarian Section of SIAM, Dec. 22-23, 2008, Sofia, Demetra Pub. House, 2009, 33–36.
- [104] Fidanova, S., K. Atanassov, P. Marinov. Generalized Nets in Artificial Intelligence. Vol. 5: Generalized nets and Ant Colony Optimization. "Prof. M. Drinov" Academic Publishing House, Sofia, 2011.
- [105] Fidanova, S., Marinov P., Atanassov K., Generalized Net Models of the Process of Ant Colony Optimization with Different Strategies and Intuitionistic Fuzzy Estimations, Proc. Jangjeon Math. Soc., Vol. 13(1), 2010, 1–12.
- [106] Frary, R. B. Hints for designing effective questionnaires. Practical Assessment, Research & Evaluation, Vol. 5, 1996, No. 3, <http://PAREonline.net/getvn.asp?v=5&n=3>
- [107] Goldberg, D. Genetic algorithms in search, optimization and machine learning. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
- [108] Graubart, B. 50 Content companies that matter – Wikipedia, Content Matters, 24.08.2005. http://www.contentmatters.info/content_matters/2005/08/50_content_comp_2.html.
- [109] Grützner, R. Konzeptionelle Grundlagen der Modellierung und Simulation von Software auf der Basis modifizierter Petri-Netze, Dissertation, Berlin: AdW, ZfR-Informationen, ZfR 84.06, 1984.
- [110] Ibri, S., H. Drias, M. Nourelfath. Integrated Emergency Vehicle Dispatching and Covering: A Parallel Ant-Tabu Approach. 8th International Conference of Modeling and Simulation (MOSIM'10) May 10–12, 2010, Hammamet, Tunisia, "Evaluation and optimization of innovative production systems of goods and services"
- [111] Kehoe, J. Writing multiple-choice test items. Practical Assessment, Research & Evaluation, Vol. 4, 1995, No. 9, <http://PAREonline.net/getvn.asp?v=4&n=9>.
- [112] Kolev, B., E. El-Darzi, E. Sotirova, I. Petronias, K. Atanassov, P. Chountas, V. Kodogianis. Generalized Nets in Artificial Intelligence. Vol. 3: Generalized nets, Relational Data Bases and Expert Systems. "Prof. M. Drinov" Academic Publishing House, Sofia, 2006.
- [113] Kotov, V. An algebra for parallelism based on Petri nets, Lect. Notes in Comp. Sci. 64 (1978), 39–55.
- [114] Krawczak, M. Generalized Nets Representation of Multilayer Neural Networks. Дисертационен труд за придобиване на научната степен „Доктор на техническите науки“, СНС по ЕКТ, 2003.
- [115] Krawczak, M., Multilayer Neural Systems and Generalized Net Models. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2003.
- [116] Krawczak, M., S. Sotirov, K. Atanassov. Multilayer Neural Networks and Generalized Nets. Warsaw School of Information Technology, Warsaw, 2010.
- [117] Looney, C. Fuzzy Petri Nets for rule-based decision making, IEEE Trans. on Systems Man and Cybernetics, Vol. 22, 1988, 178–183.
- [118] Moyson, F., B. Manderick. The Collective Behavior of Ants: An Example of Self-Organization in Massive Parallelization. In Proceedings of AAAI Spring Symposium on Parallel Models of Intelligence, Stanford, California, 1988.
- [119] Nan Liu, Bo Huang, Xiaohong Pan. Using the Ant Algorithm to Derive Pareto Fronts for Multiobjective Siting of Emergency Service Facilities. Transportation Research Record: Journal of the Transportation Research Board, No. 1935, Transportation Research Board of the National Academies, Washington, D.C., 2005, 120–129.
- [120] Pencheva, T. Generalized Nets Model of Crossover Technique Choice in Genetic Algorithms, Issues in Intuitionistic Fuzzy Sets and Generalized Nets, Vol. 9, 2011, 92–100.

- [121] Pencheva, T., K. Atanassov, A. Shannon, Modelling of a Stochastic Universal Sampling Selection Operator in Genetic Algorithms Using Generalized Nets, Proceedings of the Tenth International Workshop on Generalized Nets, December 5, 2009, Sofia, Bulgaria, 1–7.
- [122] Pencheva, T., K. Atanassov, A. Shannon, Modelling of a Roulette Wheel Selection Operator in Genetic Algorithms Using Generalized Nets, Bioautomation, 2009, 13(4), 257–264.
- [123] Pencheva, T., K. Atanassov, A. Shannon, Generalized Net Model of Offspring Reinsertion in Genetic Algorithms, Annual of “Informatics” Section of Union of Scientists in Bulgaria, 2011, 4, 29–35.
- [124] Pencheva, T., K. Atanassov, A. Shannon, Generalized Net Model of Selection Function Choice in Genetic Algorithms, in: Recent Advances in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics, Volume II: Applications, Systems Research Institute, Polish Academy of Sciences, Warsaw, 2011, 193–201.
- [125] Pencheva, T., O. Roeva, A. Shannon, Generalized Net Models of Crossover Operators in Genetic Algorithms, Proceedings of the Ninth International Workshop on Generalized Nets, Sofia, Bulgaria, July 4, 2008, 2, 64–70.
- [126] Petri, C.-A. Kommunikation mit Automaten. Ph.D.diss., Univ. of Bonn, 1962; Schriften des Inst. fur Instrument. Math., No 2, Bonn, 1962.
- [127] Radeva, V., M. Krawczak, E. Choy, Review and bibliography on generalized nets theory and Applications. Advanced Studies in Contemporary Mathematics, Vol. 4, 2002, No. 2, 173–199.
- [128] Roeva, O., K. Atanassov, Generalized Net Model of a Modified Genetic Algorithm, Issues in Intuitionistic Fuzzy Sets and Generalized Nets, Vol. 7, 2008, 93–99.
- [129] Roeva, O., K. Atanassov, A. Shannon. Generalized net for evaluation of the genetic algorithm fitness function. Proceedings of the Eighth Int.Workshop on Generalized Nets, Sofia, 26 June 2007, 48–55.
- [130] Roeva, O., K. Atanassov, A. Shannon, Generalized Net for Selection of Genetic Algorithm Operators, Annual of “Informatics” Section of Union of Scientists in Bulgaria, Vol. 1, 2008, 117–126.
- [131] Roeva, O., T. Pencheva, Generalized Net Model of a Multi-population Genetic Algorithm, Issues in Intuitionistic Fuzzy Sets and Generalized Nets, 2010, 8, 91–101.
- [132] Roeva, O., T. Pencheva, K. Atanassov, Generalized Net of a Genetic Algorithm with Intuitionistic Fuzzy Selection Operator, New Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Vol. I: Foundations (Eds. K. T. Atanassov, M. Baczyński, J. Drewniak, J. Kacprzyk, M. Krawczak, E. Szmiedt, M. Wygralak, S. Zadrozny), IBS PAN/SRI PAS, Warsaw, 2012, 167–178.
- [133] Roeva, O., T. Pencheva, K. Atanassov, A. Shannon, Generalized Net Model of Selection Operator of Genetic Algorithms, 2010 IEEE International Conference on Intelligent Systems (IS 2010), July 7–9, 2010, University of Westminster, London, UK, 286–289.
- [134] Roeva, O., A. Shannon, T. Pencheva, Description of Simple Genetic Algorithm Modifications Using Generalized Nets, Proc. of 6th IEEE Intelligent Systems IS’12, 6–8 Sept. 2012, Sofia, Bulgaria, Vol. II, 178–183.
- [135] Roeva, O., A. Shannon, A Generalized Net Model of Mutation Operator of the Breeder Genetic Algorithm, Proceedings of the Ninth International Workshop on Generalized Nets, Sofia, July 4, 2008, 2, 59–63.
- [136] Shannon, A., K. Atanassov, D. Orozova, M. Krawczak, E. Sotirova, P. Melo-Pinto, I. Petrounias, T. Kim. Generalized nets and information flow within a university. Warsaw School of Information Technology, Warsaw, 2007.
- [137] Shannon A., O. Roeva, T. Pencheva, K. Atanassov, Generalized Nets Modelling of Biotechnological Processes, “Prof. Marin Drinov” Publishing House of Bulgarian Academy of Sciences, Sofia, 2004.
- [138] Shannon, A., J. Sorsich, K. Atanassov. Generalized Nets in Medicine. "Prof. M. Drinov" Academic Publishing House, Sofia, 1996.
- [139] Shannon, A., J. Sorsich, K. Atanassov, N. Nikolov, P. Georgiev. Generalized Nets in General and Internal Medicine. "Prof. M. Drinov" Academic Publishing House, Sofia, Vol. 1 1998, Vol. 2, 1999 and Vol. 3, 2000.

- [140] Sorsich J., A. Shannon, K. Atanassov. Generalized Net in Child Neurology. "Prof. M. Drinov" Academic Publishing House, Sofia, 2002.
- [141] Starke, P. Petri-Netze. Berlin, VEB Deutscher Verlag der Wissenschaften, 1980.
- [142] Stütze, T., H. H. Hoos, MAX MIN Ant System, Future Generation Computer Systems, Vol. 16, 2000, 889–914.
- [143] Szmidt, E. Application of Intuitionistic Fuzzy Sets in Decision Making. Дисертационен труд за придобиване на научната степен „Доктор на техническите науки”, СНС по ЕКТ, 2000.
- [144] Tasseva, V. On the generalized nets transitions, Issues in the Representation and Processing of Uncertain and Imprecise Information, EXIT Academic Press, Warsaw, 2005, 373–385.
- [145] The University of Tennessee, “Designing Test Questions”, 04.02.2033 <http://www.utc.edu/Administration/WalkerTeachingResourceCenter/FacultyDevelopment/Assessment/test-questions.html>.
- [146] Todorova, M. Methodological Aspects of an Approach for Verification of Object-Oriented Programs”, Proc. of 6th International Conference IEEE Intelligence Systems’12, 6–8 Sept. 2012, Sofia, Bulgaria, 153–158.
- [147] Todorova, M. Model Checker of Object-Oriented Programs based on Generalized Nets, In: Recent Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Vol. II: Applications, 10–20.
- [148] Todorova, M. Simulation of the Behaviour of a Pure Imperative Synchronous Programming Language by Means of Generalized Nets with Stop-Conditions, Proc. of 6th Int. Conf. Information Systems & Grid Technologies, 1–3 June 2012, Sofia, 312–325.
- [149] Todorova, M. Verification of procedural programs via building their generalized nets models. Proc. of the 41st Spring Conference of the Union of Bulgarian Mathematicians, Mathematics And Education In Mathematics, 2012, 259–265.
- [150] Trifonov, T., K. Georgiev. GN Ticker – A Software Tool for Efficient Interpretation of Generalized Net Models. – Issues in Intuitionistic Fuzzy Sets and Generalized Nets, Vol. 3, Warsaw, 2005, 71–78.
- [151] Trifonov, T., K. Georgiev, K. Atanassov. Software for modelling with generalized nets. Issues in Intuitionistic Fuzzy Sets and Generalized Nets, Vol. 6, 2008, 96–110.
- [152] Wikipedia, The Free Encyclopedia, “Wiki”, 28.04.2010. <http://en.wikipedia.org/w/index.php?oldid=358114642>.
- [153] Wong, C. K., Mathematical modelling of peptide kinetics in diabetes mellitus, Ph.D. thesis, Univ. of Technology, Sydney (Australia), 1995.

Списък на публикациите по дисертационния труд

1. Atanassova, V., Georgiev, P., Generalized net models of conflict resolution approaches in version control systems. Part 2: Copy-Modify-Merge, *Proc. of 10th Int. Workshop on Generalized Nets*, 5 Dec. 2009, 14–21, ISSN: 1313-6860.
2. Atanassova, V., Generalized net models of conflict resolution approaches in version control systems. Part 1: Lock-Modify-Unlock, *Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Vol. II: Applications*, Warsaw, 2010, 13–24, ISBN: 978-8389475305.
3. Atanassova, V., Design of training tests on generalized nets, *Proc. of 5th IEEE Conf. on Intelligent Systems*, 7–9 July 2010, London, UK, 327–330, ISBN: 978-1-4244-5163-0.
4. Atanassova, V. Utilizing the model graphic structures when teaching generalized nets. *Recent Advances in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics, Vol. II: Applications*, 2010, 7–17, ISBN: 978-83-89475367.
5. Атанасова, В. Уики технологии в помощ на науката. Примерът с Ifigenia.org, *Сборник от 8^{ма} Национална младежка научно-практическа сесия на ФНТС*, 10–11 май 2010, 83–88, ISSN: 1314-0698.
6. Atanassov, K., Dimitrov, D., Atanassova, V., Algorithms for Tokens Transfer in Different Types of Intuitionistic Fuzzy Generalized Nets, *Journal of Cybernetics and Information Technologies*, Vol. 10, 2010, No.4, 22–35, ISSN: 1311-9702 (print), 1314-4081 (online).
7. Atanassov, K., V. Atanassova, P. Chountas, A. Shannon. Generalized nets with places having intuitionistic fuzzy capacities. *Notes on Intuitionistic Fuzzy Sets*, Vol. 17, 2011, No.4, 21–28, ISSN: 1310-4926.
8. Atanassova, V., S. Fidanova, P. Chountas, K. Atanassov. A generalized net with an ACO-algorithm optimization component, *Lecture Notes in Computer Science*, Vol. 7116, 2012, 190–197, ISSN: 0302-9743.
9. Atanassova, V., S. Fidanova, I. Popchev, P. Chountas, Chapter 5. Generalized Nets, ACO Algorithms, and Genetic Algorithms, In: Monte Carlo Methods and Applications, *Proc. of the 8th IMACS Seminar on Monte Carlo Methods*, August 29 – September 2, 2011, Borovets, Bulgaria (Sabelfeld, K., I. Dimov, Eds.), De Gruyter, 2012, 39–46, eBook ISBN: 9783110293586.
10. Atanassova, V., On Intuitionistic Fuzzy Approach to Generalized Net Prognostics. In: *New Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets*,

Generalized Nets and Related Topics. Volume II: Applications, SRI-PAS, Warsaw, 2012, 1–12, ISBN: 978-8389475411.

11. Atanassova, V. The minimal solution of a problem in generalized nets, *Proc. of the 6th IEEE Int. Conference on Intelligent Systems*, 6–8 Sept. 2012, Sofia, Bulgaria, Vol. II, 159–163, ISBN: 978-1-4673-2277-5.
12. Atanassova, V. Generalized Nets with Volumetric Tokens, *Comptes rendus de l'Academie Bulgare des Sciences*, Vol. 65, 2012, No. 11, 1489–1498, ISSN: 1310-1331.

Списък на забелязани цитирания на публикации по дисертационния труд

Забелязани са общо 7 цитирания на 4 публикации.

Atanassov, K., D. Dimitrov, V. Atanassova, Algorithms for Tokens Transfer in the Different Types of Intuitionistic Fuzzy Generalized Nets. Cybernetics and Information Technologies, Vol. 10, 2010, No. 4, 22-35, ISSN 1311-9702 (print), 1314-4081 (online).

1. Todorova, M., K. Kanev, Educational Framework for Verification of Object-oriented Programs, The Joint International Conference on Human-Centered Computer Environments HCCE'2012, 8-13 march, Hamamatsu, Japan, 2012, 23-27, ISBN 978-1-4503-1191-5.
2. Тодорова, М. Построяване на коректни обектно-ориентирани програми чрез изграждане на техни обобщеномрежови модели, Годишник на секция “Информатика” към Съюза на учените в България, том 4, 2011, 1-28, ISSN 1313-6852.
3. Todorova, M. Model checker of object-oriented programs based on generalized nets, In: *New Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Volume II: Applications.* (K. Atanassov, M. Baczynski, J. Drewniak, J. Kacprzyk, M. Krawczak, E. Szmidt, M. Wygalak, S. Zadrozny, Eds.), SRI-PAS, Warsaw, 2012, 309-320, ISBN: 9788389475411.
4. Todorova, M. Verification of Procedural Programs via Building there Generalized Nets Models. Proceedings of the 41st Spring Conference of the Union of Bulgarian Mathematicians, Mathematics And Education In Mathematics, 2012, 259-265, http://www.math.bas.bg/smb/2012_PK/tom_2012/pdf/259-265.pdf

Atanassova, V. Design of Training Tests on Generalized Nets. –In: Proc. of the 5th IEEE Int. Conf. on Intelligent Systems, 7-9 July 2010, London, UK, 327-330, ISBN 978-1-4244-5163-0.

5. Орозова, Д. Обобщеномрежови модели на интелигентни системи за обучение. София, Акад. издателство „Проф. Марин Дринов”, 2011, ISBN 978-954-322-481-4.

Atanassova, V., S. Fidanova, P. Chountas, K. Atanassov A generalized net with an ACO-algorithm optimization component, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 7116, 2012, 190-197.

6. Орозова, Д. Обобщеномрежови модели на интелигентни системи за обучение. София, Акад. издателство „Проф. Марин Дринов”, 2011, ISBN 978-954-322-481-4.

Atanassova, V., S. Fidanova, I. Popchev, P. Chountas, Chapter 5: Generalized Nets, ACO Algorithms, and Genetic Algorithms, In: Monte Carlo Methods and Applications, Proceedings of the 8th IMACS Seminar on Monte Carlo Methods, August 29 - September 2, 2011, Borovets, Bulgaria (Sabelfeld, K., I. Dimov, Eds.), De Gruyter, 2012, pp. 39-46, eBook ISBN: 9783110293586.

7. Орозова, Д. Обобщеномрежови модели на интелигентни системи за обучение. София, Акад. издателство „Проф. Марин Дринов”, 2011, ISBN 978-954-322-481-4.

Списък на проекти с участието на докторанта

- Проект с Фонд „Научни изследвания“ № ДИД-02-29 / 2009 „Моделиране на процеси с фиксирани правила на развитие“;
- Проект с Фонд „Научни изследвания“ № ДТК-02-44 / 2009 „Ефективни Монте Карло методи за големи научно-изследователски задачи“;
- Проект с Фонд „Научни изследвания“ № БИИ-02-09 / 2009 „Проектиране и разработка на интуиционистки размити логически средства в информационните технологии“;
- Проект с ЕСФ и МОМН № BG051PO001-3.3.04/40 „Изграждане на висококвалифицирани млади изследователи по съвременни информационни технологии за оптимизация, разпознаване на образи и подпомагане вземането на решения“.
- Проект по Royal Society, International Joint Project № JP100372 „Обобщеномрежови модели и интуиционистки размити множества в интелигентните системи“.