

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

МЕТОДИ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ КОНСПЕКТ ЛЕКЦІЙ

Навчальний посібник

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра
за освітньою програмою «Електронні компоненти і системи»
спеціальності 171 Електроніка

Укладач: К.С. Клен

Електронне мережне навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2023

Рецензент

Найда С.А., д.т.н., проф.
завідувач кафедри акустичних та мультимедійних електронних систем

Відповідальний
редактор

Вербицький Є.В., д.т.н., доц.

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № 5 від 23.02.2023 р.)
за поданням Вченої ради факультету електроніки
(протокол № 1/2023 від 30.01.2023 р.)*

У навчальному посібнику викладено інформацію щодо основ теорії предикатів; основ теорії графів та дерев; основ теорії скінченних автоматів; принципів роботи машини Тюрінга та основ теорії нечіткої логіки.

Конспект лекцій містить теоретичні відомості до 11 лекцій та список рекомендованої літератури.

Навчальний посібник призначений для здобувачів ступеня бакалавра за освітньою програмою «Електронні компоненти і системи» спеціальності 171 Електроніка.

Реєстр. № НП 22/23-499. Обсяг 5,37 авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Перемоги, 37, м. Київ, 03056
<https://kpi.ua>

Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів
і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

ЗМІСТ

Вступ.....	4
Лекція №1. Загальні положення теорії предикатів. Приклади використання теорії предикатів для доведення теорем теорії чисел.....	5
Лекція №2. Визначення графів та їхні властивості. Дерева та їхні властивості. Використання теорії графів для моделювання електронних схем.....	25
Лекція №3. Скінченні автомати Мілі і Мура	56
Лекція №4. Основи аналізу цифрових автоматів.....	70
Лекція №5. Синтез скінченних автоматів.....	86
Лекція №6. Синтез асинхронних імпульсних автоматів.....	101
Лекція №7. Машина Тюрінга	111
Лекція №8. Методологія нечіткого моделювання	129
Лекція №9. Операції над нечіткими множинами та нечіткі відношення	143
Лекція №10. Загальні положення нечіткої логіки та нечіткі висновки	155
Лекція №11. Приклади використання алгоритмів нечіткої логіки в електронних системах керування складними об'єктами	184

Вступ

Навчальна дисципліна «Методи моделювання інформаційних систем» належить до циклу вибірових освітніх компонент з кафедрального Ф-каталогу за освітньою програмою «Електронні компоненти і системи», читається протягом одного семестру (8).

В процесі вивчення курсу студенти знайомляться з теорією представлення та мінімізації цифрових пристроїв, синтезом та аналізом цифрових автоматів за методикою Мілі та Мура синхронного та асинхронного типів із застосуванням сучасної інтегральної елементної бази.

У конспекті розглянуто основні питання теорії предикатів; теорії графів та дерев; теорії скінченних автоматів; принципу роботи машини Тюрінга та теорії нечіткої логіки.

Методи теорії систем штучного інтелекту, що розглядаються, широко використовуються для формування алгоритмів обробки сигналів в цифрових інформаційних системах. Також ці методи складають теоретичне підґрунтя для проектування вузлів та пристроїв сучасної електронно-обчислювальної техніки. На їхній основі будуються сучасні способи кодування електронних сигналів.

Лекція №1. Загальні положення теорії предикатів. Приклади використання теорії предикатів для доведення теорем теорії чисел

Теорія предикатів – це важливий розділ сучасної дискретної математики, який базується на законах алгебри теорії множин та Булевої алгебри. Теорія предикатів була започаткована у фундаментальних трудах видатних математиків Е.В. Дайкстра та В.М. Глушкова та становить теоретичне підґрунтя всіх сучасних методів програмування. Зокрема, саме на основі теорії предикатів у шістдесятих роках ХХ століття були обґрунтовані методи структурного програмування та створені відповідні мови програмування. Тому сучасна теорія предикатів, як розділ математичної логіки, також є теоретичним підґрунтям для теорії логічних сигналів та для описання програмованих систем їхньої обробки.

Слід відзначити, що часто у класичній арифметиці та у математичній логіці істинність або хибність логічних тверджень не може бути встановленою, оскільки ці твердження містять змінні, значення яких початково не визначаються, а встановлюються в ході обчислювального процесу, і саме на цьому базується теорія предикатів.

Розглянемо базові поняття теорії предикатів та надамо відповідні визначення [1, 2].

Предикатами називаються логічні твердження із змінними, істинність або хибність виконання яких є апіорі невідомою, оскільки вона залежить від значень змінних.

Наведемо конкретні приклади предикатів:

$$P(x): 3x + 2 = 5,$$

$$R(x, y) = x^2 + y^2 \geq 0,$$

$$Q(x, y, z) = x^2 + y^2 > z^2,$$

$$S(x): \sin^2(x) + \cos^2(x) = 1.$$

Місткістю предикату називається кількість змінних у ньому. У наведених вище прикладах $P(x)$ – одномісний предикат, а $Q(x, y, z)$ – тримісний предикат.

Предикати перетворюються на логічні твердження, які можуть бути істинними або хибними, коли змінним присвоюються конкретні значення. Наприклад, предикат $P(x): 3x + 2 = 5$ за умови $x = 1$ стає вірним твердженням, а за умови $x \neq 1$ – хибним.

Якщо у разі підставлення значення змінної до предикату твердження стає істинним, вважається, що це значення змінної *задовольняє* предикату.

Існують предикати, які є істинними для будь-якого значення змінної із множини дійсних чисел, наприклад, такими предикатами є $S(x): \sin^2(x) + \cos^2(x) = 1$ або $R(x, y) = x^2 + y^2 \geq 0$. Такі предикати позначаються через символи математичної логіки як $\forall x S(x)$, де символ $\forall x$ читається як «для всіх значень змінної x ». Іншою формою запису таких предикатів є $\forall x (\sin^2(x) + \cos^2(x))$.

Символ $\forall x$ у теорії предикатів називається *універсальним квантором* або *квантором загальності*.

Можлива множина значень змінної x називається *універсом*, або *предметною галуззю предиката*.

Наприклад, предметною галуззю може бути множина натуральних, цілих, дійсних або комплексних чисел. Зрозуміло, що істинність тверджень із квантором загальності суттєво залежить від предметної галузі. Наприклад, твердження $\forall x (x^2 > 5)$ не є істинним, якщо універсом є значення цілих чисел, проте воно є істинним за умови $x \in \{1, 2\}$. Аналогічно, твердження $\forall x (x^2 \geq 0)$ є істинним за умови раціональних значень змінної x , але хибним, якщо x є комплексним числом. Тобто, необхідною та достатньою умовою істинності твердження із квантором загальності та предикатом є істинність цього твердження для всіх значень змінної x із визначеного універса.

Предикат із квантором називається *квантифікованим предикатом*, або *твердженням*.

Аналогічно, двомісний предикат із квантором $\forall x \forall y S(x, y)$ є твердженням, якщо він є істинним для будь-яких значень із універсів змінних x та y . Іноді, для спрощення розуміння та аналізу логічних виразів із предикатами, квантори загальності пишуться у дужках, наприклад: $(\forall x)(\forall y)S(x, y)$.

Хоча твердження $\forall x S(x)$ для всіх значень x із універса може бути хибним, предикат $S(x)$ може бути істинним твердженням для деяких окремих значень x , хоча б для одного із них. Наприклад, як було показано вище, предикат $P(x): 3x + 2 = 5$ є істинним твердженням за умови $x = 1$.

Якщо існують значення x із універса, для яких предикат $P(x)$ є істинним, такий предикат називається *квантором існування* та позначається $\exists x S(x)$.

Слід відзначити, що істинність тверджень із квантором існування $\exists x$, як і з квантором загальності $\forall x$, залежить від визначеного універса.

Наприклад, вираз $\exists x (x^2 = 4)$ є істинним твердженням на множині натуральних чисел, вираз $\exists x (x^2 = 5)$ – на множині дійсних чисел, а вираз $\exists x (x^2 = -5)$ – на множині комплексних чисел.

Властивість 1.1. Необхідною та достатньою умовою перетворення предиката на твердження є визначеність всіх його змінних, які мають бути пов'язані із квантором універсальності або із квантором існування.

Наприклад, вираз $\exists x \forall y S(x, y, z)$ не є твердженням, а вираз $\exists x \forall y \forall z S(x, y, z)$ – це твердження.

Твердження $\forall x S(x)$ може бути запереченим, якщо існують такі значення x із універса, для яких предикат $S(x)$ є хибним. Заперечення твердження $\forall x S(x)$ у символах математичної логіки позначається як $\sim \forall x S(x)$.

Слід відзначити, що для заперечення твердження $\forall x S(x)$ достатньо знайти хоча б одне значення x із універса, для якого виконується твердження $\sim \forall x S(x)$. Наприклад, твердження $\forall x (x^2 > 0)$ є хибним на множині натуральних чисел, оскільки воно не виконується за умови $x = 0$.

Сформулюємо відповідну теорему теорії предикатів.

Теорема 1.1. Твердження $\forall x S(x)$ є хибним тоді і лише тоді, коли виконується твердження $\exists x (\sim S(x))$.

Наприклад, для розглянутого вище прикладу вірним є твердження $\exists x (\sim (x^2 > 0))$, тому вираз $\forall x (x^2 > 0)$ є хибним.

Зрозуміло, що якщо для предикату $S(x)$ у межах визначеного універса не існує таких значень x , для яких твердження $\forall x S(x)$ є хибним, тоді вірним буде логічний вираз $\sim (\exists x S(x))$. З іншого боку, якщо твердження $\forall x S(x)$ є хибним для всіх значень x у межах визначеного універса, тоді вірним є вираз $\forall x (\sim S(x))$. Тобто, у будь якому разі виконуються наступні тотожності:

$$\sim \forall x (S(x)) \equiv \exists x (\sim S(x)), \quad \sim \exists x (S(x)) \equiv \forall x (\sim S(x)). \quad (1.1)$$

Тобто, згідно із співвідношеннями (1.1), заперечення тверджень із предикатами здійснюється наступним чином. Необхідно замінити квантор із запереченням на зворотний квантор, а знак заперечення поставити перед предикатом. Зрозуміло, що тут квантором, зворотним до квантора універсальності є квантор існування, і навпаки.

Для заперечення тверджень із кількістю кванторів, більшою за 2, співвідношення (1.1) виконуються послідовно. Сформулюємо це правило перетворення у вигляді наступної властивості.

Властивість 1.2. Під час перетворень логічних виразів із предикатами квантори послідовно замінюються на зворотні, а знак заперечення переходить до іншого, внутрішнього квантора. Ці логічні операції проводяться до тих пір, доки знак заперечення не переходить до предиката.

Розглянемо конкретні приклади таких перетворень.

Приклад 1.1. Перетворити наступні логічні вирази і предикатами:
 $\sim (\forall x)(\forall y)R(x, y)$ та $\sim (\exists x)(\forall y)(\exists z)Q(x, y, z)$.

Згідно із співвідношеннями (1.1) та із властивістю 1.2 перетворення записаних виразів із предикатами здійснюються наступним чином:

$$\sim (\forall x)(\forall y)R(x, y) \equiv \exists x(\sim (\forall y)R(x, y)) \equiv (\exists x)(\exists y)(\sim R(x, y));$$

$$\begin{aligned} \sim (\exists x)(\forall y)(\exists z)Q(x, y, z) &\equiv (\forall x)(\sim \forall y)(\exists z)Q(x, y, z) \equiv (\forall x)(\exists y)(\sim \exists z)Q(x, y, z) \equiv \\ &\equiv (\forall x)(\exists y)(\forall z)(\sim Q(x, y, z)). \end{aligned}$$

Розглянемо тепер дві важливі теореми, які описують взаємозв'язки між твердженнями із кванторами та значеннями предикатів у разі підстановки змінних.

Теорема 1.2. Твердження $\forall xS(x)$ є істинним, коли у разі підстановки до предиката $S(x)$ довільного значення x із універса істинним є логічний вираз, заданий предикатом $S(x)$.

Теорема 1.3. Зворотна до теореми 1.2. Якщо твердження $\forall xS(x)$ є істинним, тоді логічний вираз, заданий предикатом $S(x)$, для будь-яких значень x із універса також є істинним.

Розглянемо тепер аналогічні вирази із предикатом та квантором існування. Почнемо із простого прикладу.

Приклад 1.2. Для предикату $S(x): x^2 + 2x - 3 = 0$ перевірити, чи є вірним твердження $\exists x(S(x))$ для універса натуральних чисел.

Квадратне рівняння $x^2 + 2x - 3 = 0$ має корінь $x = 1$. Тому можна сказати, що для предикату $S(x): x^2 + 2x - 3 = 0$ твердження $\exists x(S(x))$ є вірним.

На основі прикладу 1.2 можна сформулювати наступні дві важливі теореми для тверджень із квантором існування.

Теорема 1.4. Якщо твердження $\exists x(S(x))$ для визначеного універса є істинним, тоді для деяких значень a із визначеного універса логічний вираз $S(x = a)$ також є істинним.

Теорема 1.5. Зворотна до теореми 1.4. Якщо логічний вираз $S(x=a)$ є істинним для деяких значень a із визначеного універса, тоді істинним також є твердження $\exists x(S(x))$.

На основі розглянутих визначень та теорем 1.1 – 1.5 можна сформулювати головні властивості предикатів.

Властивість 1.3. Властивість універсальної конкретизації. Із істинності твердження $\forall x(S(x))$ випливає істинність логічного виразу $S(x=a)$ для довільного значення a із визначеного універса.

Властивість 1.4. Властивість універсального узагальнення. Якщо логічний вираз $S(x=a)$ виконується для довільного значення a із визначеного універса, тоді твердження $\forall x(S(x))$ для цього універса також є істинним.

Властивість 1.5. Властивість екзистенціальної конкретизації. Із істинності твердження $\exists x(S(x))$ випливає, що для деяких значень a із визначеного універса істинним є логічний вираз $S(x=a)$.

Властивість 1.6. Властивість екзистенціального узагальнення. Якщо для деяких значень a із визначеного універса істинним є логічний вираз $S(x=a)$, тоді твердження $\exists x(S(x))$ також є істинним.

Розглянемо ще дві важливі теореми теорії предикатів про зв'язки між предикатами, які об'єднані єдиним універсом.

Теорема 1.6. Квантор загальності від кон'юнкції двох предикатів еквівалентний кон'юнкції двох кванторів загальності від першого та другого предикатів, або у математичній формі:

$$\forall x(S(x) \wedge Q(x)) \equiv \forall x(S(x)) \wedge \forall x(Q(x)). \quad (1.2)$$

Теорема 1.7. Квантор існування від диз'юнкції двох предикатів еквівалентний диз'юнкції двох кванторів існування від першого та другого предикатів, або у математичній формі:

$$\exists x(S(x) \vee Q(x)) \equiv \exists x(S(x)) \vee \exists x(Q(x)). \quad (1.3)$$

Сутність теорем 1.5 та 1.6 полягає у наступному. Якщо $p(x)$ – предикат, якому відповідає твердження $T1$, а предикату $q(x)$ відповідає твердження $T2$, тоді вираз $\forall x(p(x) \rightarrow q(x))$ з точки зору алгебри логіки означає наступне: «Для всіх об’єктів, які відповідають твердженню $T1$, виконується твердження $T2$ ». За тих же умов вираз $\exists x(p(x) \wedge q(x))$ з точки зору алгебри логіки означає: «Для всіх об’єктів, які відповідають твердженню $T1$, виконується твердження $T2$ ».

Розглянемо приклади використання теорем 1.5 та 1.6 для формування ускладнених логічних тверджень.

Приклад 1.3. З використанням кванторів загальності та існування сформулювати складні твердження для наступних простих предикатів:

1. $p(x)$ – « x – студент»; $q(x)$ – « x отримує стипендію»;
2. $p(x)$ – « x – мавпа»; $q(x)$ – « x їсть банан»;
3. $p(x)$ – « x – сертифікований електронний прилад»; $q(x)$ – « x – працює нормально»;
4. $p(x)$ – « x – логічний сигнал»; $q(x)$ – « x має значення 1»;
5. $p(x)$ – « x – канал зв’язку»; $q(x)$ – « x є вільним».

На основі наведених предикатів та законів алгебри логіки можна сформулювати наступні складні твердження у разі використання квантору загальності:

1. «Всі студенти отримують стипендію».
2. «Всі мавпи їдять банани».
3. «Всі сертифіковані електронні прилади працюють нормально».
4. «Всі логічні сигнали мають значення 1».
5. «Всі канали зв’язку є вільними».

Відповідно, у разі використання квантору існування можна записати:

1. «Деякі студенти отримують стипендію».
2. «Деякі мавпи їдять банани».
3. «Деякі сертифіковані електронні прилади працюють нормально».
4. «Деякі логічні сигнали мають має значення 1».

5. «Деякі канали зв'язку є вільними».

Легко зрозуміти, що у наведених прикладах твердження 1, 4 та 5 є вірними у разі використання квантору існування, а твердження 2 та 3 – у разі використання квантору загальності.

Тут важливим також є фактор логічного заперечення, який був описаний у визначенні 1.8. Запереченням для твердження «Для всіх» є твердження «Для деяких», і навпаки. Наприклад, якщо не всі студенти у групі отримують стипендію, тоді твердження 1 із квантором загальності: «Всі студенти отримують стипендію» заперечується, що приводить до твердження 1 із квантором існування: «Деякі студенти отримують стипендію».

Аналогічно можна заперечувати твердження 2, 3, 4 та 5.

Для перевірки істинності тверджень із кванторами загальності та існування зручно використовувати діаграми Вейля. Твердженню «Всі $p \in q$ » відповідає діаграма, яка наведена на рис. 1.1, а, а твердженню «Деякі $p \in q$ » – діаграма, яка наведена на рис. 1.1, б.



Рис. 1.1. Діаграми Вейля для кванторів загальності (а) та існування (б)

Розглянемо деякі приклади використання діаграм Вейля для аналізу істинності тверджень, які побудовані з використанням теорії предикатів.

Приклад 7.4. З використанням діаграм Вейля проаналізувати істинність та несуперечність наступних тверджень.

1. Всі студенти НТУУ «КПІ імені Ігоря Сікорського» є неординарними особистостями.
2. Всі вчені є неординарними особистостями.
3. Всі студенти НТУУ «КПІ імені Ігоря Сікорського» є вченими.

Зрозуміло, що третє твердження отримане як результат кон'юнкції предикатів 1 та 2, які вважаються істинними. Тобто, у даному разі універсом вважається група неординарних особистостей, підгрупою якою є вчені, а вже підгрупою вчених – всі студенти НТУУ «КПІ імені Ігоря Сікорського». Для формування цих підгруп використані квантор загальності та принцип кон'юнкції, тобто розглянута вище теорема 1.5. Відповідна діаграма Вейля наведена на рис. 1.2. Наочний аналіз тверджень із цього прикладу з використанням діаграм Вейля показує, що вони є вірними.



Рис. 1.2. Діаграма Вейля для прикладу 1.4

Приклад 1.5. З використанням діаграм Вейля проаналізувати істинність та несуперечність наступних тверджень.

1. Всі поети є щасливими людьми.
2. Деякі поети є ледачими.
3. Деякі щасливі люди ледачі.

У цьому випадку у першому та другому базових твердженнях-предикатах використовується як квантор загальності «Всі», так і квантор існування «Деякі», тому третє твердження формується на основі теорем 1.5 та 1.6. Перевіримо твердження 3 з використанням діаграми Вейля, яка наведена на рис. 1.3.

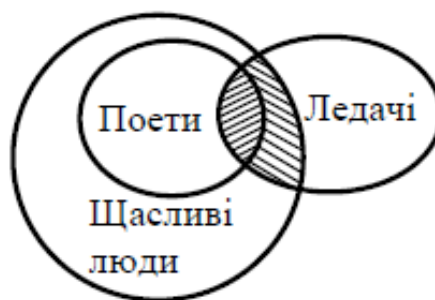


Рис. 1.3. Діаграма Вейля для прикладу 1.5

Із діаграми, наведеної на рис. 1.3, зрозуміло, що, згідно із першим твердженням, коло поетів є ширшим, ніж коло щасливих людей. Із цього випливає, що, якщо коло ледаців, згідно із другим твердженням, перетинає коло поетів, то воно перетинає і коло щасливих людей. Тобто, третє твердження, за умови істинності першого і другого, також є істинним. Також із наведеної діаграми Вейля зрозуміло, що коло ледащих щасливих людей є дещо ширшим, ніж коло ледащих поетів.

Приклад 1.6. З використанням діаграм Вейля проаналізувати істинність та несуперечність наступних тверджень.

1. Деякі поети є невдахами.
2. Деякі вчені є невдахами.
3. Деякі поети є вченими.

Із діаграми Вейля, наведеної на рис. 1.4, зрозуміло, що, незважаючи на те, що коло невдах перетинається як із колом поетів, так і з колом вчених, кола поетів та вчених не перетинаються. Тобто, третє твердження у даному випадку є хибним.



Рис. 1.4. Діаграма Вейля для прикладу 1.6

Приклад 1.7. З використанням діаграм Вейля проаналізувати істинність наступних тверджень.

1. Всі цифрові сигнали є дискретними.
2. Всі сигнали із кодоімппульсною модуляцією є цифровими.
3. Всі сигнали із кодоімппульсною модуляцією є дискретними.

Із діаграми Вейля, наведеної на рис. 1.5, зрозуміло, що у даному випадку використання квантору загальності приводить до вірного результату.

Приклад 1.8. З використанням діаграм Вейля проаналізувати істинність та несуперечність наступних тверджень.

1. Деякі сигнали є сигналами керування об'єктом.
2. Деякі сигнали мають амплітуду, більшу за 0,5 В.
3. Всі сигналами керування об'єктом мають амплітуду, більшу за 0,5 В.

Із діаграми Вейля, яка наведена на рис. 1.6, зрозуміло, що твердження 3 є невірним. Це пов'язано з тим, що у разі використання квантору існування для базових тверджень, у твердженні, яке із них впливає, також має бути використаний квантор існування, а не квантор загальності. Для прикладу, який розглядається, згідно із теоремою 1.6 вірним буде наступне твердження: «Деякі сигналами керування об'єктом мають амплітуду, більшу за 0,5 В».



Рис. 1.5. Діаграма Вейля для прикладу 1.7



Рис. 1.6. Діаграма Вейля для прикладу 1.8

Приклад 1.9. З використанням діаграм Вейля проаналізувати істинність та несуперечність наступних тверджень.

1. Деякі канали зв'язку є вільними.
2. Деякі канали зв'язку призначені для передавання службових повідомлень.
3. Деякі канали зв'язку, призначені для передавання службових повідомлень, є вільними.

Діаграма Вейля для цього прикладу наведена на рис. 1.7.

Із наведеної діаграми зрозуміло, що хоча вільні канали зв'язку та канали зв'язку, призначені для передавання службових повідомлень, належать до одного універса, групи цих каналів можуть не перетинатися, тому третє твердження є невірним. Тобто, можливою є така ситуація, що вільні канали зв'язку існують, проте всі канали, якими можна передавати службові повідомлення, є зайнятими.

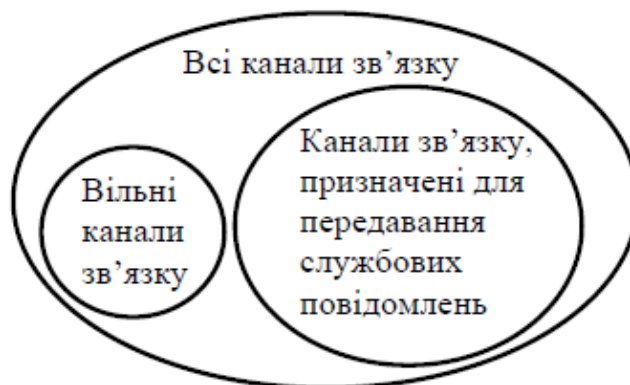


Рис. 1.7. Діаграма Вейля для прикладу 1.9

Наведені приклади дозволяють зрозуміти важливість теорії предикатів для розуміння логіки роботи електронних програмних засобів, для моделювання ситуацій в програмованих електронних пристроях, для кодування сигналів та для формування сигнальних конструкцій.

Приклади використання теорії предикатів для доведення теорем теорії чисел

У цьому підрозділі показано ефективність використання теорії предикатів для доведення важливих теорем та положень теорії чисел. Необхідні логічні міркування здійснюються з використанням властивостей раціональних чисел. Головними із цих властивостей є властивості зімкненості, комутативності, асоціативності, дистрибутивності, а також наявність нейтральних та обернених елементів для операцій додавання та множення.

Приклад 1.10. З використанням теорії предикатів показати, що квадрат парного числа завжди також є парним числом.

З точки зору формалізму теорії предикатів умову поставленої задачі можна переписати наступним чином:

$$\forall n \left((n - \text{парне}) \rightarrow (n^2 - \text{парне}) \right). \quad (1.4)$$

У таблиці 1.1 наведена послідовність логічних міркувань, яка дозволяє довести тотожність (1.4) з використанням теорем теорії предикатів 1.5, 1.6, а також властивостей 1.3 – 1.6.

Приклад 1.11. З використанням теорії предикатів довести тотожність $(a > b) \wedge (c > d) \rightarrow ((a + c) > (b + d))$.

Як і у попередньому прикладі, запишемо послідовність логічних операцій у зручній табличній формі.

Таблиця 1.1. Послідовність логічних операцій теорії предикатів, яка дозволяє довести співвідношення (1.4)

№	Сутність операції	Пояснення
1.	a – ціле число	Визначення універса
2.	a – парне число	Обмеження універса, задане умовою задачі
3.	$\forall n((n - \text{парне}) \rightarrow (\exists k)(k < n, n = 2 \cdot k))$	Визначення парного числа з використанням формалізму теорії предикатів
4.	$a - \text{парне число} \rightarrow (\exists k)(k < a, a = 2 \cdot k)$	Висновок із положень 1 та 3 з урахуванням властивості 1.3
5.	$(\exists k)(k < a, a = 2 \cdot k)$	Висновок із положень 2 та 4
6.	$a = 2 \cdot L$	Висновок із положення 5 з урахуванням властивості 1.5
7.	$\forall r \forall s((r = s) \rightarrow (r^2 = s^2))$	Наслідок аксіом порівняння. Відомий факт із теорії чисел
8.	$(a = 2 \cdot L) \rightarrow (a^2 = (2 \cdot L)^2)$	Висновок із положення 7 та властивості 1.3
9.	$a^2 = (2 \cdot L)^2$	Висновок із положення 8
10.	$(2 \cdot L)^2 = 2 \cdot (2 \cdot L^2)$	Висновок із положення 9 та властивості асоціативності
11.	$\forall x \forall y \forall z((x = y) \wedge (y = z) \rightarrow (x = z))$	Властивість транзитивності рівностей
12.	$((a^2 = (2 \cdot L)^2) \wedge ((2 \cdot L)^2 = 2 \cdot (2 \cdot L^2))) \rightarrow (a^2 = 2 \cdot (2 \cdot L^2))$	Висновок із положень 9, 11 та властивості 1.3
13.	$((a^2 = (2 \cdot L)^2) \wedge ((2 \cdot L)^2 = 2 \cdot (2 \cdot L^2)))$	Висновок із положень 9, 10 з урахуванням властивості кон'юнкції
14.	$a^2 = 2 \cdot (2 \cdot L^2)$	Висновок із положень 12, 13
15.	$(\exists k)(a^2 = 2 \cdot k)$	Висновок із положення 12 з урахуванням властивості 1.5
16.	$\forall x((\exists k)(x = 2 \cdot k) \rightarrow x - \text{парне})$	Визначення парного числа
17.	$(\exists k)(a^2 = 2 \cdot k) \rightarrow a - \text{парне}$	Висновок із положення 12 з урахуванням властивості 1.5 для фіксованого значення a
18.	$a^2 - \text{парне число}$	Висновок із положень 15, 17
19.	$a - \text{парне число} \rightarrow a^2 - \text{парне число}$	Висновок із положень 2, 18
20.	$\forall n((n - \text{парне}) \rightarrow (n^2 - \text{парне}))$	Висновок із положень 1, 19

Таблиця 1.2. Послідовність логічних операцій теорії предикатів, яка дозволяє довести властивість раціональних чисел

№	Сутність операції	Пояснення
1.	$\forall a \forall b ((a > b) \wedge (c > d)) \rightarrow ((a > b > 0) \wedge (c > d > 0))$	Впливає із закону порівняння раціональних чисел
2.	$\forall a \forall b ((a > b > 0) \wedge (c > d > 0)) \rightarrow ((a - b) + (c - d) > 0)$	Впливає із закону порівняння раціональних чисел
3.	$\forall a \forall b ((a - b) + (c - d) > 0) \rightarrow ((a + c) - (b + d) > 0)$	Впливає із закону комутативності для додавання
4.	$\forall a \forall b ((a + c) - (b + d) > 0) \rightarrow ((a + c) > (b + d))$	Впливає із закону порівняння протилежних чисел
5.	$\forall a \forall b ((a + c) > (b + d)) \rightarrow ((a + c) > (b + d))$	Висновок із положень 1, 4

Як видно із таблиць 1.1 та 1.2, останнім кроком доведення алгебраїчних тотожностей з використанням теорії предикатів завжди є встановлення зв'язку між початковим та кінцевим твердженням, а коректність цього зв'язку обґрунтована коректністю ланцюга перетворень логічних виразів. Також слід відмітити, що для здійснення цих логічних перетворень зазвичай використовуються не лише теореми теорії предикатів 1.5 та 1.6, але і важливі теореми та аксіоми предметної області, які у даному випадку зазвичай вважаються відомими. Наприклад, під час розгляду прикладів 1.10 та 1.11 використовувались аксіоми теорії чисел.

Іншим цікавим прикладом використання теорії предикатів для доведення математичних теорем є принцип математичної індукції. З теоретичної точки зору цей принцип є досить простим та зрозумілим. Зазвичай з використанням цього принципу у теорії чисел доводяться функціональні залежності $P(n)$, де n – натуральне число. Прикладами таких залежностей можуть бути степеневі ряди та твірні функції. Алгоритмічна послідовність кроків, які необхідно виконати під час доведення тотожностей за принципом математичної індукції, наведена у таблиці 1.3.

Таблиця 1.3. Послідовність логічних операцій для доведення теорем теорії чисел з використанням принципу математичної індукції

№	Сутність операції
1.	Постановка задачі, визначення функціональної залежності $P(n)$
2.	Показати, що вірною є тотожність $P(1)$
3.	Припустити, що вірною є тотожність $P(k)$, $k \in N$
4.	Показати, що за умови істинності тотожності $P(k)$ істиною буде також тотожність $P(k+1)$

З використанням символіки теорії предикатів послідовність алгоритмічних операцій, описану у таблиці 1.3, можна записати наступним чином:

$$P(1) \wedge (\forall k (P(k) \rightarrow P(k+1))) \rightarrow \forall n P(n). \quad (1.5)$$

Слід відзначити, що принцип математичної індукції, заданий співвідношенням (1.5), використовуються не лише для доведення алгебраїчних співвідношень, але й для розв'язування геометричних задач.

Наведемо декілька прикладів розв'язування алгебраїчних та геометричних задач з використанням принципу математичної індукції.

Приклад 1.12. З використанням принципу математичної індукції довести тотожність:

$$S(n) = \sum_{m=1}^n m^2 = \frac{n \cdot (n+1) \cdot (2 \cdot n + 1)}{6}. \quad (1.6)$$

Легко перевірити, що за умови $n = 1$ співвідношення (1.6) є вірним:

$$1^2 = \frac{1 \cdot (1+1) \cdot (2 \cdot 1 + 1)}{6} = \frac{3 \cdot 2}{6} = 1.$$

Згідно із третім кроком алгоритму, який описаний у таблиці 1.3, припустимо, що співвідношення (1.6) є вірним за умови $n = k$, тобто:

$$\sum_{m=1}^k m^2 = 1^2 + 2^2 + \dots + k^2 = \frac{k \cdot (k+1) \cdot (2 \cdot k + 1)}{6}. \quad (1.7)$$

Тепер, відповідно до четвертого кроку алгоритму, необхідно показати, що, у разі виконання співвідношення (1.7), вираз $S(n)$, визначений

співвідношенням (1.6), також є вірним за умови $n = k + 1$. Дійсно, враховуючи співвідношення (1.5) та (1.7), можна записати:

$$\begin{aligned}
 1^2 + 2^2 + \dots + k^2 + (k+1)^2 &= \frac{k \cdot (k+1) \cdot (2 \cdot k + 1)}{6} + (k+1)^2 = \\
 &= \frac{k \cdot (k+1) \cdot (2 \cdot k + 1)}{6} + \frac{6 \cdot (k+1)^2}{6} = \frac{k \cdot (k+1) \cdot (2 \cdot k + 1) + 6 \cdot (k+1)^2}{6} = \\
 &= \frac{(k+1) \cdot (k \cdot (2 \cdot k + 1) + 6 \cdot (k+1))}{6} = \frac{(k+1) \cdot (2 \cdot k^2 + 7 \cdot k + 6)}{6} = \\
 &= \frac{(k+1) \cdot (2 \cdot k + 3)(k+2)}{6} = \frac{(k+1) \cdot ((k+1)+1) \cdot (2 \cdot (k+1)+1)}{6}
 \end{aligned}$$

тобто

$$S(k+1) = \frac{(k+1) \cdot ((k+1)+1) \cdot (2 \cdot (k+1)+1)}{6}. \quad (1.8)$$

Із співвідношень (1.7) та (1.8) зрозуміло, що згідно із принципом математичної індукції, заданого з використанням теорії предикатів співвідношенням (1.5), початкове співвідношення (1.6) є вірним.

Приклад 1.13. З використанням принципу математичної індукції показати, що площа, яка розділена на області будь-якою кількістю прямих ліній, проведених під різними кутами, може бути зафарбована двома різними кольорами таким чином, що будь-які дві сусідні області, які мають спільну границю, будуть зафарбовані різними кольорами.

Оскільки за умовою задачі кількість ліній m не обмежена, тобто, теоретично можливим є випадок $m = \infty$, комбінаторний перебір всіх варіантів у даному випадку неможливий. Тому скористаємося для розв'язування поставленої задачі принципом математичної індукції (1.5). Переформулюємо поставлену геометричну задачу на основі символіки теорії предикатів. Вираз $P(m)$ можна записати наступним чином:

$$M = \{m : P(m)\} = \{m : \text{креслення, яке виконане з використанням } m \text{ ліній, ... може бути зафарбованим}\} \quad (1.9)$$

Тобто, формулу (1.5) для сформульованої геометричної задачі необхідно застосовувати до твердження (1.9).

Із основ геометрії відомо, що проведена пряма лінія ділить площину на дві частини. Якщо одну із цих частин зафарбувати, наприклад, білим, а другу, відповідно, сірим кольором – це і буде тривіальним розв’язком поставленої задачі за умови $m = 1$. Відповідне креслення показане на рис. 1.8, а. Нескладно також отримати розв’язок задачі для випадку $m = 2$. Для цього необхідно змінити колір кожної області з одного боку введеної лінії L на протилежний. Відповідні перетворення креслення показані на рис. 1.8, б, тут колір був змінений для областей 1 та 2, тобто, для двох областей, які розташовані нижче введеної лінії L . Трошки складніше, але аналогічно здійснюються геометричні перетворення у разі введення третьої лінії. Тут необхідно змінити колір трьох областей, які розташовані з одного боку від введеної лінії, як показано на рис. 1.8, в.

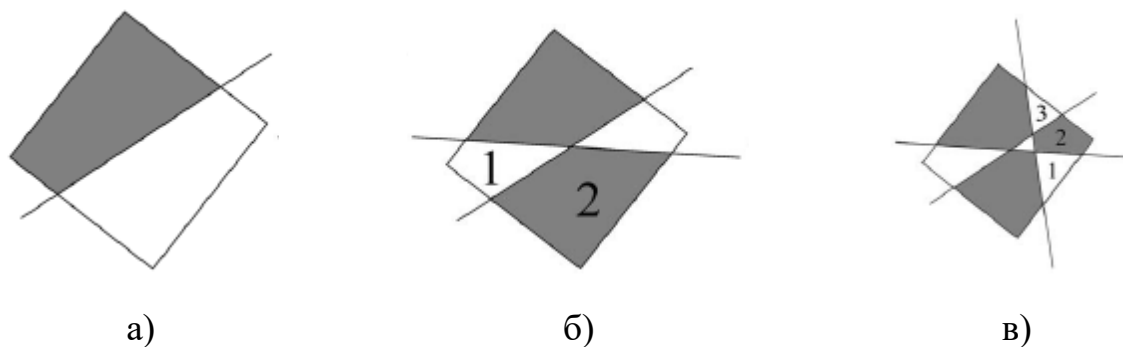


Рис. 1.8. Наочна ілюстрація зафарбовування сусідніх областей площини, яку розбивають на області одна (а), дві (б) та три (в) лінії

Із наведених логічних міркувань та геометричних перетворень легко зрозуміти, що у разі введення додаткової лінії на розфарбовану площину для розв’язування поставленої задачі достатньо змінити колір зафарбованих областей з одного боку введеної лінії, а кількість областей, які перефарбовуються, дорівнює номеру лінії. Отриманий результат означає, що логічний вираз із предикатами (1.9) за принципом математичної індукції (1.5) є вірним.

Контрольні питання

1. Що являє собою теорія предикатів і на яких головних теоретичних положеннях вона ґрунтується? Наведіть власні приклади формування предикатів.
2. Як теорія предикатів пов'язана із математичною логікою та алгеброю Буля? Свою відповідь обґрунтуйте та наведіть доречні приклади.
3. Як теорія предикатів пов'язана із сучасною теорією інформації? Свою відповідь обґрунтуйте та наведіть приклади.
4. Які головні поняття теорії предикатів Вам відомі? Наведіть визначення цих понять та поясніть, які вони мають значення для формування логічних тверджень.
5. У якому випадку вважається, що логічна змінна задовольняє предикату? Наведіть власні приклади таких предикатів та відповідних логічних змінних.
6. Чому істинність або хибність логічних висловлень із предикатами зазвичай апріорі є невідомою? Наведіть власні приклади предикатів та покажіть, у якому випадку логічне твердження, сформоване у вигляді предиката, буде істинним, а у якому – хибним.
7. Що являє собою місткість предикату та як вона визначається? Наведіть власні приклади визначення місткості предикату.
8. Що являє собою квантор загальності, як він позначається та яким чином цей квантор використовується для формування предикатів? Наведіть власні приклади використання квантору загальності.
9. Що являє собою універс і як це поняття використовується у теорії предикатів? Наведіть власні приклади визначення універсу для заданого предикату.
10. Що являє собою квантор існування, як він позначається та яким чином цей квантор використовується для формування предикатів? Наведіть власні приклади використання квантору існування.

11. Чим предикат відрізняється від логічного твердження і як можна перетворити предикат на твердження? Наведіть власні приклади такого перетворення.

12. Що являє собою заперечення логічних тверджень і як це поняття використовується у теорії предикатів? Наведіть свої приклади заперечення твердження, яке сформовано на основі предикату.

13. Яким чином можна заперечити логічне твердження із квантором загальності? Наведіть власні приклади такого використання операції заперечення.

14. Які головні властивості предикатів Вам відомі?

15. У чому полягає несуперечливість логічних тверджень та як вона пов'язана з їх істинністю. Наведіть власні приклади суперечливих та несуперечливих логічних тверджень.

16. Як для аналізу логічних тверджень використовуються діаграми Вейля? Наведіть власні приклади такого використання діаграм Вейля.

17. Яким чином теорія предикатів може бути використана для доведення математичних теорем, зокрема теорем теорії чисел? Наведіть власні приклади такого використання теорії предикатів.

Перелік посилань

1. James A. Anderson. Discrete Mathematics with Combinatorics. – Oxford University Press – 2020, – 198 p.

2. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. Посібник. – Львів: Новий світ – 2000, 2009, – 736 с.

Лекція №2. Визначення графів та їхні властивості. Дерева та їхні властивості. Використання теорії графів для моделювання електронних схем

Гратки Гассе є однією із модифікацій орієнтованих графів, і використовуються для опису відносин між упорядкованими множинами та властивостей груп. Проте поняття графу у математиці є більш широким і загалом воно пов'язано не лише із описанням відносин між множинами. Теорія множин, дійсно, є одним із розділів, де використовуються графи, але теорія графів має власну проблематику і є окремим, важливим розділом математики. Графи також широко використовуються для аналізу особливостей функціонування складних електронних схем та для формування кодів інформаційних повідомлень. У теорії комп'ютерних мереж графи широко використовуються для описання топологічних структур, які визначають структуру провідних або безпроводних з'єднань між вузлами мережі.

Розглянемо визначення графу а також суміжних понять, які, дійсно, тісно пов'язані із теорією множин [1, 2].

Граф являє собою упорядковану множину вершин із заданими відношеннями між ними, які визначаються у вигляді двоелементних підмножин заданої множини вершин.

Задані відношення між вершинами графа називаються його *ребрами*.

Вершини графа, між якими описані відношення у вигляді ребра, називаються *зв'язаними*.

Зазвичай у математичній літературі граф позначається як $G(V, E)$, де G – граф, V – множина його вершин, а E – множина ребер. Часто графи також описують у вигляді креслень, на яких вершини позначаються точками, а ребра – лініями, які проводяться між вершинами згідно із заданими відношеннями. Саме наочність геометричного подання графів робить ефективним їхнє використання для розв'язування практичних інженерних задач, зокрема задач, пов'язаних із аналізом особливостей роботи електронних схем та систем.

Вершини графа a і b , які зв'язані ребром, називаються *кінцями ребра*.

Ребро, яке зв'язує вершини графа a і b , позначається як $\{a,b\}$. Зрозуміло, що за будь-яких умов $\{a,b\} \in E$.

Якщо ребро $\{a,b\}$ з'єднує вершини графа a і b , його називають *інцидентним* до цих вершин, а вершини – *інцидентними*, або *суміжними* до ребра.

Два ребра графа називаються *суміжними*, якщо вони мають спільну вершину.

Наведемо прості приклади побудови графів.

Приклад 2.1. Побудувати граф із множиною вершин $V = \{a,b,c\}$ та ребрами $E = \{\{a,b\}, \{b,c\}\}$.

Визначена структура графа може бути зображена або у вигляді прямої лінії, або у вигляді трикутника, у якому відсутня одна із сторін. Відповідні геометричні побудови наведені на рис. 2.1.



Рис. 2.1. Можливі креслення структури графу для прикладу 2.1

Приклад 2.2. Побудувати граф із множиною вершин $V = \{a,b,c,d,e\}$ та ребрами $E = \{\{a,b\}, \{a,e\}, \{b,e\}, \{b,d\}, \{b,c\}, \{c,d\}\}$.

Одна із можливих структур графу, яка відповідає умові задачі, показана на рис. 2.2.

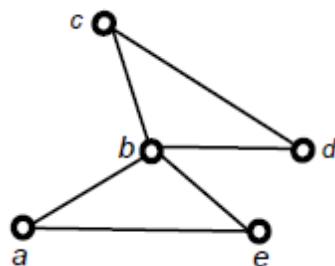


Рис. 2.2. Структура графу для прикладу 7.15

Із наведених прикладів зрозуміло, що для наочного подання зв'язків між вершинами графа важливими є не форма побудованих геометричних фігур, а наявність зв'язків між вершинами, які мають бути відображені. У дискретній математиці такі структури також називаються топологічними. Вони також широко використовуються в інженерній практиці для наочного описання важливих особливостей побудови систем зв'язку, розподілених електронних систем та комп'ютерних мереж.

Може виникнути питання: чому графи настільки ефективно описують електронні системи, системи зв'язку та об'єкти іншої природи, коли вони навіть не відображують чітку геометричну структуру? Тут важливим є саме рівень математичного абстрагування. По-перше, математична абстракція усуває неважливі, несуттєві деталі, що дозволяє інженеру-досліднику концентрувати увагу на дійсно важливих проблемах функціонування складних систем. А по друге, саме абстрактні математичні підходи часто надають можливість шукати ідентичні розв'язки для задач, у яких описуються об'єкти різної фізичної природи. Крім теорії графів, такими абстрактними підходами можна вважати теорію груп, теорію систем масового обслуговування та математичну статистику. Зокрема, теорія графів також широко використовується для аналізу транспортних систем, систем комунальних послуг та для розв'язування інших вкрай важливих практичних задач. Проте задачі проведення комунікацій і практичні задачі сучасної наноелектроніки також тісно пов'язані. Річ у тому, що під час проектування мікросхем важливою задачею є уникнення перетинання електричних з'єднань на кожному із рівнів ієрархії. Така ж сама задача виникає під час розведення складних друкованих плат із великою кількістю мікросхем. Теорія графів завжди дає відповідь на важливе питання: чи можна за умови заданої кількості контактів провести з'єднання між ними без перетинання? Відповідь на це питання дає теорія планарних графів.

Надамо інші важливі визначення теорії графів, які використовуються для розв'язання з використанням неї важливих практичних інженерних задач.

Степінню вершини графа називається кількість його ребер, які є інцидентними до даної вершини.

У математичній літературі степінь вершини графа v зазвичай позначається як $\deg(v)$.

Вершина графа із степінню 0 називається *ізолюваною*.

Розглянемо приклад пошуку у графі суміжних вершин та ребер.

Приклад 2.3. У графі, наведеному на рис. 2.3, знайти суміжні вершини та суміжні ребра.

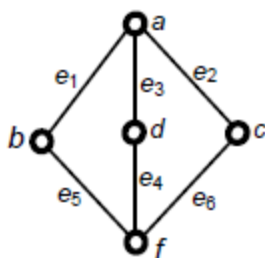


Рис. 2.3. Структура графу для прикладу 2.3

У графі, який наведено на рис. 2.3, суміжними є наступні вершини: $\{a, b\}$, $\{a, c\}$, $\{a, d\}$, $\{b, f\}$, $\{c, f\}$, $\{d, f\}$. Суміжними ребрами є наступні: $\{e_1, e_2, e_3\}$, $\{e_4, e_5, e_6\}$, $\{e_1, e_5\}$, $\{e_5, e_6\}$.

Розглянемо без доведення декілька важливих теорем теорії графів.

Теорема 2.1. Сума степенів всіх вершин графу завжди є парним числом.

Теорема 2.2. У будь-якому графі кількість вершин непарної степені є парним числом.

Граф $G'(V', E')$ називається підграфом графа $G(V, E)$, за умови, що множина V' є підмножиною V , а множина E' є підмножиною E .

Розглянемо приклади підграфів.

Приклад 2.4. Серед графів, наведених на рис. 2.4 а – г, знайти такі, які можна вважати підграфами. Які із цих графів мають ізолювані вершини та ізолювані ребра?

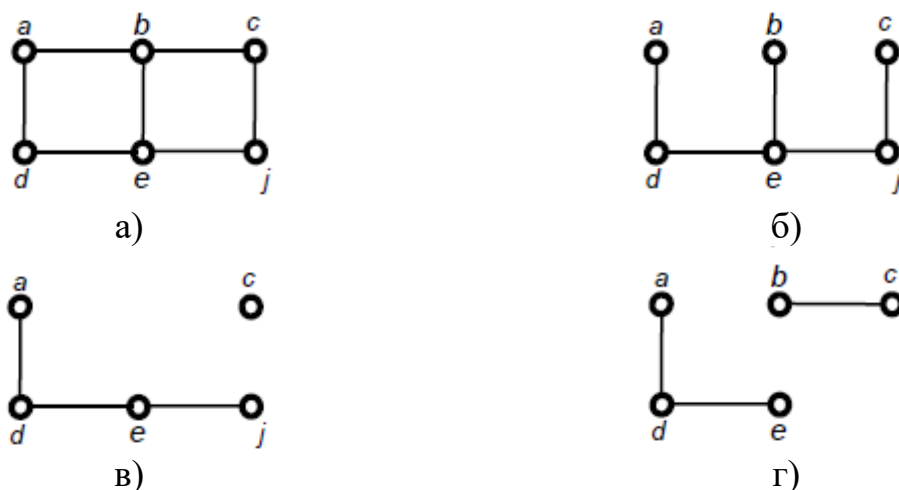


Рис. 2.4. Структури графів для прикладу 2.4

Із аналізу наведених структур графів зрозуміло, що графи, які зображені на рис. 2.4, б, в та г, є підграфами графу, який наведений на рис. 2.4, а. Також зрозуміло, що граф, наведений на рис. 2.4, в, має ізольовану вершину, а граф, наведений на рис. 2.4, г, має ізольоване ребро.

Розглянемо ще декілька важливих визначень, пов'язаних із наявністю та відсутністю шляхів через ребра між вершинами визначеного графа.

У графі, який має вершини $v_0, v_1, v_2, v_3, \dots, v_{k-1}, v_k$ із множини V та ребра $e_1, e_2, e_3, \dots, e_k$ із множини E , шляхом довжини k з вершини v_0 до вершини v_k називається послідовність $P = \{v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_k\}$, для якої ребра e_i лежать між вершинами v_{i-1} та v_i , тобто $e_i = \{v_{i-1}, v_i\}$. Слід відзначити, що послідовність P є надлишковою.

Зазвичай у практичних задачах шлях між двома вершинами вказується лише через вершини, а ребра пропускаються. Тобто, визначена послідовність P записується у спрощеному вигляді наступним чином: $P = \{v_0, v_1, v_2, \dots, v_{k-1}, v_k\}$

Простим шляхом від однієї вершини до іншої називається шлях, у якому немає вершин, що повторюються.

Розглянемо приклад пошуку шляхів у графі.

Приклад 2.5. На графі, структура якого наведена на рис. 2.5, знайти шляхи з вершини v_0 до вершини v_7 , обчислити їхню довжину та виділити серед них найпростіші.

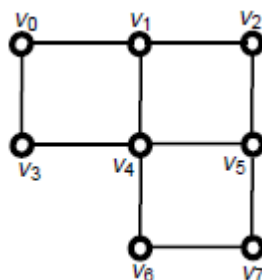


Рис. 2.5. Структура графу для прикладу 2.5

У графі, структура якого наведена на рис. 2.5, із вершини v_0 до вершини v_7 ведуть вісім різних шляхів:

$$\begin{aligned}
 P_1 &= \{v_0, v_1, v_2, v_5, v_7\}; & P_2 &= \{v_0, v_3, v_4, v_6, v_7\}; & P_3 &= \{v_0, v_1, v_4, v_5, v_7\}; \\
 P_4 &= \{v_0, v_1, v_4, v_6, v_7\}; & P_5 &= \{v_0, v_3, v_4, v_1, v_2, v_5, v_7\}; & P_6 &= \{v_0, v_1, v_2, v_5, v_4, v_1, v_2, v_5, v_7\}; \\
 P_7 &= \{v_0, v_1, v_4, v_5, v_2, v_1, v_4, v_6, v_7\}; & P_8 &= \{v_0, v_3, v_4, v_1, v_2, v_5, v_4, v_1, v_2, v_5, v_7\}.
 \end{aligned}$$

Довжина цих шляхів, відповідно, становить:

$$k_1 = k_2 = k_3 = k_4 = 3; \quad k_5 = 5; \quad k_6 = k_7 = 7; \quad k_8 = 9.$$

Зрозуміло також, що серед визначених шляхів $P_1 - P_8$ простими є шляхи P_1, P_2, P_3, P_4 та P_5 , оскільки ці шляхи не містять вершин, які повторюються.

Граф називається *зв'язним*, якщо існує шлях між будь-якими двома його вершинами.

Наприклад, графи, які наведені на рис. 2.1, 2.2, 2.3, 2.4, а, 2.4, б та 2.5 є зв'язними, а графи, наведені на рис. 2.4, в та 2.4, г не є зв'язними.

Слід відзначити, що шляхи $P_6 - P_8$, визначені у прикладі 2.5, можна скоротити, видаливши із них вершини, які повторюються. Наприклад, шлях P_6 скорочується до шляху P_1 . Взагалі будь-який складний шлях у графі можна скоротити видаляючи зімкнені шляхи, які приводять до вершини, з якої вони починалися. Так, із шляху P_6 , щоб перетворити його на простий шлях P_1 ,

необхідно видалити послідовність вершин $\{v_4, v_1, v_2, v_5\}$. У зв'язку з цим у теорії графів існує наступна теорема.

Теорема 2.3. Якщо граф є зв'язаним та у ньому існує шлях із однієї вершини до другої, тоді існує і простий шлях між цими вершинами.

Теорема 2.4. Наслідок теореми 2.3. Граф є зв'язаним тоді і лише тоді, коли в ньому існує простий шлях між будь-якими двома вершинами.

Підграф G' графа G називається його *компонентою*, якщо виконуються наступні умови:

1. G' є непорожнім зв'язним графом.
2. Якщо G'' є зв'язним підграфом графа G , G'' – підграф графа G' і $G'' = G'$, тоді G' є максимальним зв'язним підграфом графа G .

Наприклад, на рис. 2.6 а, б, показані максимальні зв'язні підграфи для графа, наведеного на рис. 2.4, г.



Рис. 2.6. Максимальні зв'язні підграфи для графа, наведеного на рис. 2.4, г

Циклом називається шлях ненульової довжини, якій з'єднує будь-яку вершину графа із собою і не містить ребер, які повторюються.

Приклад 2.6. Знайти цикли на графі, структура якого наведена на рис. 2.5.

У графі, наведеному на рис. 2.5, циклом є шлях $\{v_4, v_1, v_2, v_5\}$. Зверніть увагу на те, що саме цей шлях був видалений із шляху P_6 для перетворення його до простого шляху P_1 . Іншим циклічним шляхом у цьому графі є шлях $\{v_6, v_4, v_5, v_7\}$.

Граф називається *повним*, або *повнзв'язаним*, якщо будь-які дві його вершини з'єднані ребрами.

Повний граф із кількістю вершин n позначається K_n . На рис. 2.7 а – д показані повнозв'язані графи K_2 , K_3 , K_4 , K_5 та K_6 .

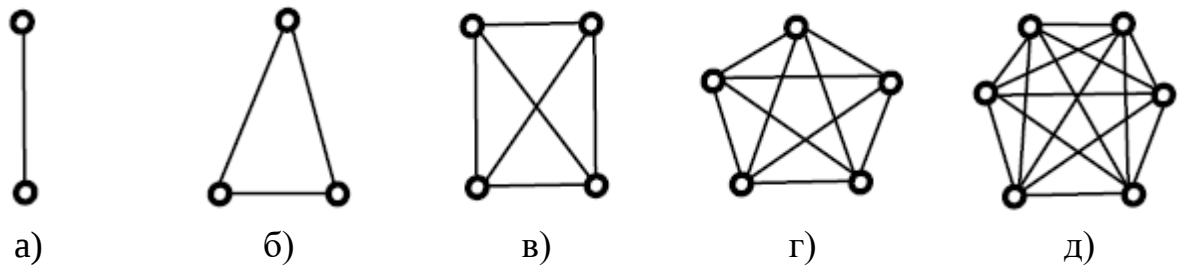


Рис. 2.7. Приклади структур повнозв'язаних графів із двома (а), трьома (б), чотирма (в), п'ятьма (г) та шістьма (д) вершинами

На основі методів комбінаторики можна довести, що кількість ребер у структурі повнозв'язаного графу K_n складає:

$$N_p = \frac{n \cdot (n-1)}{2} = \frac{n^2 - n}{2}. \quad (2.1)$$

У теорії комп'ютерних мереж повнозв'язані графи також називають повнозв'язаними топологіями. Слід відзначити, що такі топології у реальних проектах комп'ютерних мереж рідко використовуються, оскільки вони є вкрай надлишковими та потребують, за умови великої кількості вузлів, наявності великої кількості ліній зв'язу, яка відповідає кількості ребер графу та визначається співвідношенням (2.1). Тому повнозв'язані топології використовують лише у локальних мережах високої надійності із обмеженою кількістю вузлів.

Граф називається *двопільним*, якщо множину вершин V можна розбити на дві підмножини A та B так, що кожне із ребер графу має вигляд $\{a, b\}$ де $a \in A$, $b \in B$.

Інакше кажучи, у двопільному графі вершини із множини A зв'язані із вершинами із множини B , але ніякі дві вершини ні із множини A , ні із множини B не пов'язані між собою.

Граф називається *повним двопільним*, якщо всі вершини із множини A мають зв'язки із усіма вершинами із множини B .

Повний двопільний граф позначається символом K_{mn} , де m – кількість вершин у множині A , n – кількість вершин у множині B .

Приклади повних двопільних графів $K_{1,2}$, $K_{2,3}$, $K_{2,2}$, та $K_{3,3}$ наведені на рис. 2.8.

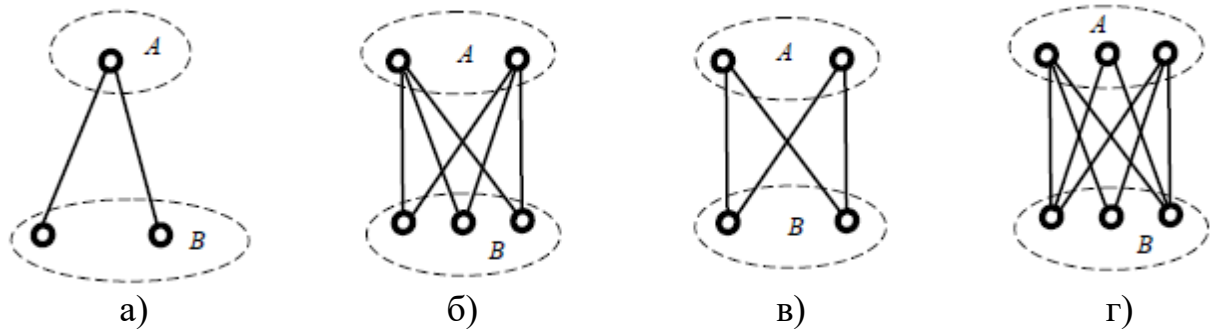


Рис. 2.8. Приклади структур повних двопільних графів $K_{1,2}$ (а), $K_{2,3}$ (б),

$K_{2,2}$ (в), та $K_{3,3}$ (г)

Розглянуті двонаправлені графи широко використовуються у теорії систем та у теорії сигналів для пошуку найкращого розташування вузлів та оптимальних сигнальних конструкцій. Наприклад, як вже було відмічено вище, на основі двонаправлених графів розглядаються принципи побудови комп'ютерних мереж та оптимальні способи з'єднання вузлів мережі.

Дерева та їхні властивості

Дерева є окремим випадком графів, і саме вони часто використовуються у математиці та інформатиці, зокрема у теорії кодування сигналів для формування різноманітних кодів. Наприклад, саме деревоподібні структури є зручними для збереження, сортування та пошуку інформації в ієрархічних даних. Узагальнене визначення дерева є досить простим [1, 2].

Деревом називається зв'язний граф, який не містить циклів. *Лісом* називається незв'язний граф, який складається із кількох дерев.

Приклад дерева показаний на рис. 2.9, а, а лісу – на рис. 2.9, б.

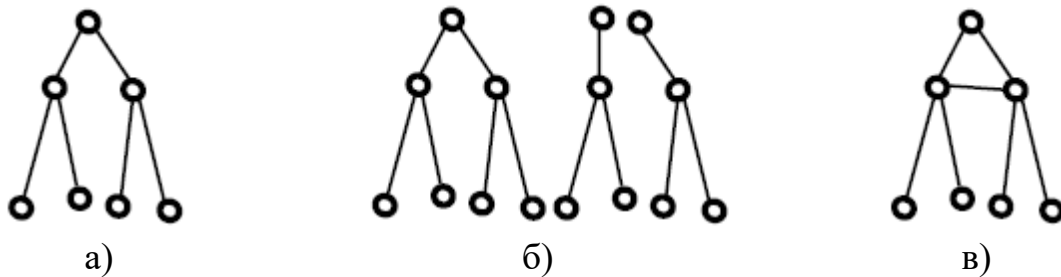


Рис. 2.9. Приклади структур простого дерева (а), лісу (б), та графа із циклом, який не є деревом (в)

Зрозуміло також, що граф, який наведений на рис. 2.9, в, не є деревом, оскільки він містить цикл.

Наприклад, деревом можна вважати родинні генеалогічні структури, тобто зв'язки між предками та нащадками, але тут слід мати на увазі, що у генеалогічному дереві не має бути родинних зв'язків між далекими предками, з метою уникнення циклічних структур. Також деревом можна вважати структуру підприємства або адміністрації університету, якщо вони не містять відділів подвійного підпорядкування. Приклади генеалогічного дерева та дерева, яке відображує структуру підприємства, наведені на рис. 2.10, а, б.

Орієнтованим деревом називається вільний від петель орієнтований граф, співвіднесений граф якого є деревом.

Для неорієнтованих дерев завжди є вірною наступна теорема.

Теорема 2.5. Для будь-яких двох вершин дерева a та b існує єдиний шлях з вершини a до вершини b . Навпаки, якщо для будь-яких вершин графа a та b існує єдиний шлях з вершини a до вершини b , тоді такий граф є деревом.

Підвішуванням дерева відносно вершини a називається розташування всіх його вершин нижче заданої вершини a .

Розглянемо приклад підвішування дерева відносно різних вершин.

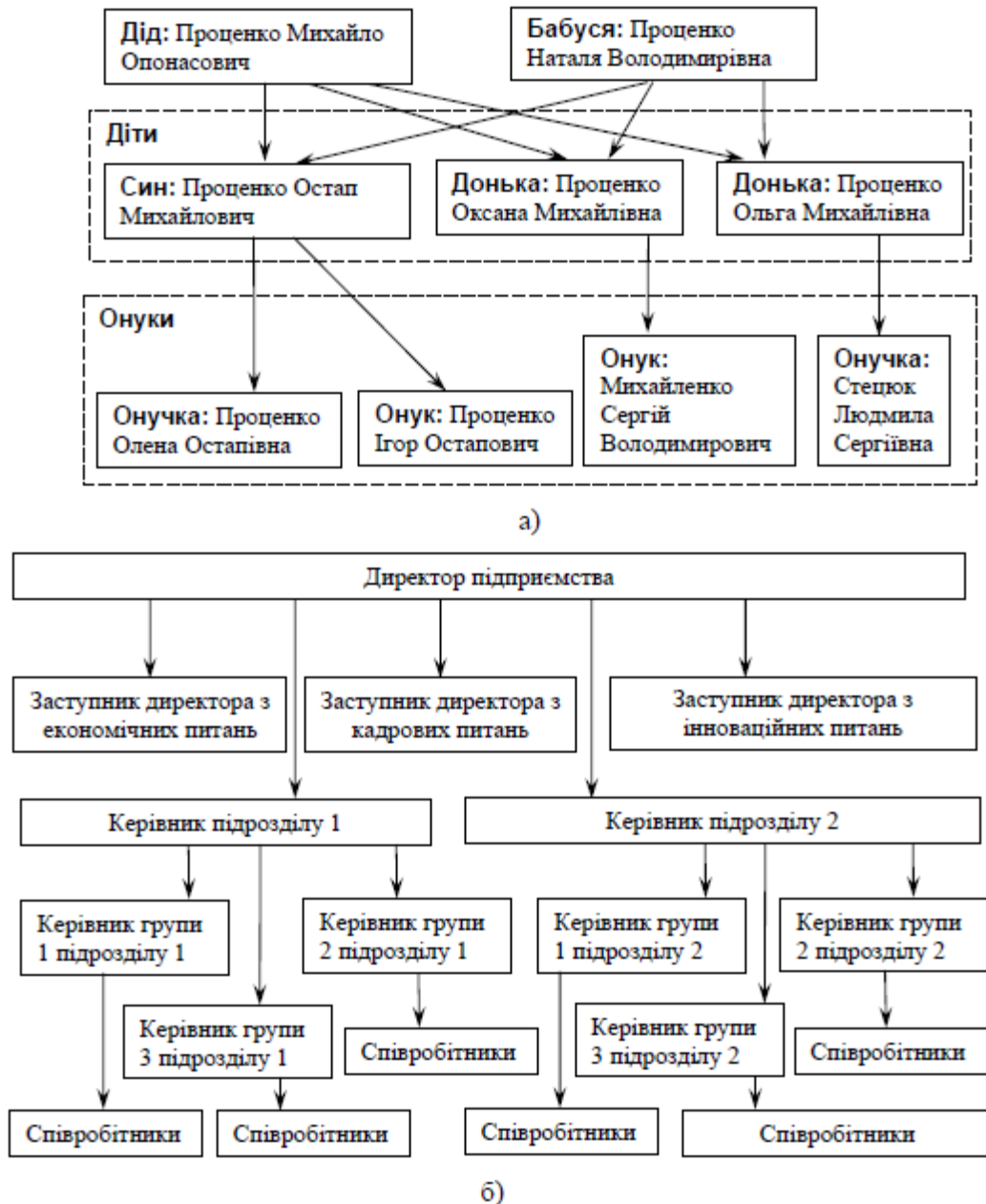


Рис. 2.10. Приклади структур генеалогічного (а) та адміністративного (б) дерев

Приклад 2.7. Підвісити дерево, яке наведене на рис. 2.11, а, відносно вершин v_3 та v_4 .

Зрозуміло, що підвішування дерева приводить до його топологічної реструктуризації, незважаючи на те, що всі шляхи у цих структурах зберігаються, і з точки зору теорії графів такі топологічні структури є еквівалентними. Якщо підвісити граф, показаний на рис. 2.11, а, за вершину v_3

– отримуємо структуру, наведену на рис. 2.11, б, а якщо за вершину v_4 – структуру, наведену на рис. 2.11, в.

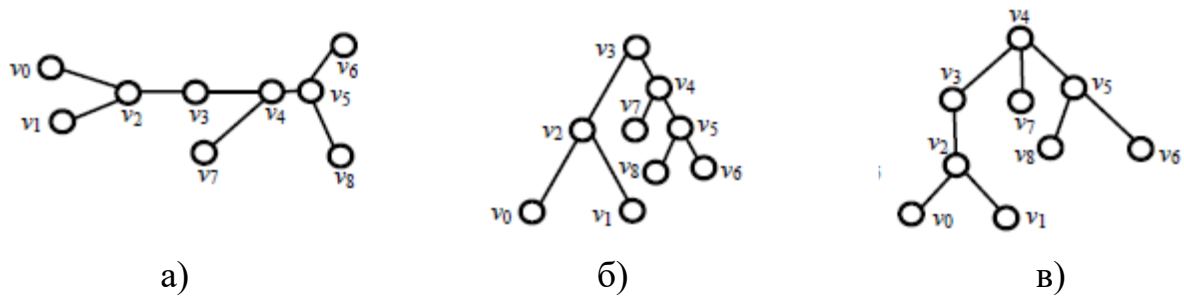


Рис. 2.11. Підвішування дерева, наведеного на рис. 2.11, а, відносно вершин v_3 (б) та v_4 (в)

Надамо інші важливі визначення теорії графів, пов'язані із класифікацією дерев.

Вершина верхньої частини підвішеного дерева називається його *коренем*.

Дерево, для якого визначений корінь, називається *кореневим деревом*.

Якщо граф кореневого дерева є орієнтованим, таке дерево називається *кореневим орієнтованим деревом*.

Рівень вершини v у орієнтованому дереві визначається довжиною шляху від кореня до вершини v , оскільки такий шлях завжди є єдиним.

Висотою орієнтованого дерева називається найдовший шлях від кореня дерева до однієї із його вершин.

Вершина u у орієнтованому дереві називається *батьком* вершини v , а вершина v – *сином* вершини u , якщо існує орієнтоване ребро із вершини u до v .

Якщо у орієнтованому дереві вершини v та v' є синами однієї вершини u , вершини v та v' називаються *братами*.

Якщо у орієнтованому дереві існує шлях із вершини u до вершини v , тоді вершина u називається *предком* вершини v , а вершина v – *нащадком* вершини u .

Степінню виходу орієнтованого дерева називається максимальна кількість ребер m , які виходять із будь-якої із його вершин.

Дерево із степінню виходу m називається m -арним деревом, а дерево із степінню виходу 2 – бінарним деревом.

Нашадок бінарного дерева, який розташований ліворуч від предка, називається *лівим сином*, а нащадок, розташований праворуч – *правим сином*.

Розглянемо приклад аналізу бінарного дерева та визначення його параметрів.

Приклад 2.8. Для бінарного дерева, яке наведено на рис. 2.12, визначити рівні вершин, висоту дерева, синів, братів та нащадків.

Для дерева, яке наведене на рис. 2.12, рівень вершин v_1 та v_2 дорівнює 1, рівень вершин v_3 , v_4 , v_5 та v_6 дорівнює 2, а рівень вершин v_7 та v_8 дорівнює 3.

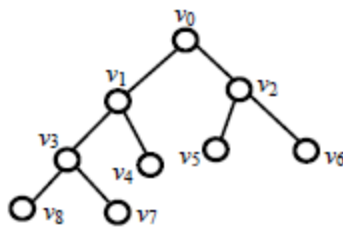


Рис. 2.12. Бінарне дерево для прикладу 2.8

Для вершин v_1 та v_2 предком є вершина v_0 , для вершин v_3 та v_4 – вершина v_1 , для вершин v_5 та v_6 – вершина v_2 , а для вершин v_7 та v_8 – вершина v_3 . Вершини v_1 та v_2 , v_3 та v_4 , v_5 та v_6 , а також вершини v_7 та v_8 є братами.

Вершина v_8 є лівим сином вершини v_3 , а вершина v_7 – правим. Аналогічно, лівим сином вершини v_1 є v_3 , а правим – v_4 , лівим сином вершини v_2 є v_5 , а правим – v_6 , лівим сином вершини v_0 є v_1 , а правим – v_2 .

Розглянемо одну із головних теорем теорії дерев.

Теорема 2.6. Якщо дерево має e ребер та v вершин, тоді $v = e + 1$.

Зворотнє твердження також є вірним, тобто, якщо для графа T є вірним співвідношення $v = e + 1$, тоді T – дерево.

Дерево T називається *остовним деревом графа G* , якщо T є підмножиною G і T – дерево.

Вершини орієнтованого дерева, які не є коренем, називаються *листами*.

Теорема 2.7. Наслідок теореми 2.6. У кожного зв'язаного графа завжди існує підграф, який є деревом.

Розглянемо ще декілька важливих теорем, пов'язаних із теорією дерев.

Теорема 2.8. Якщо дерево із степінню виходу m має n вершин та i внутрішніх вершин, тоді завжди має місце тотожність:

$$i = \frac{n-1}{m}. \quad (2.2)$$

Теорема 2.9. Наслідок теореми 2.8. Якщо повне дерево із степінню виходу m має n вершин, i внутрішніх вершин та l листів, тоді для такого дерева

$$i = \frac{l-1}{m-1}. \quad (2.3)$$

Теорема 2.10. Повне дерево із висотою h та степінню виходу m має кількість вершин

$$n = \frac{m^{h+1} - 1}{m - 1} \quad (2.4)$$

та кількість листів

$$l = m^h. \quad (2.5)$$

Зрозуміло, що формула (2.4) є формулою геометричної прогресії. Дійсно, якщо у дереві кількість вершин на кожному із рівнів є в m разів більшою, тоді загальна кількість вершин у такому дереві визначається співвідношенням (2.4) за формулою геометричної прогресії. Формула (2.5) відповідає кількості розміщень із повтореннями.

Наступна теорема теорії дерев також пов'язана із основами комбінаторного аналізу та з поняттям ізоморфізму.

Теорема 2.11. Кількість неізоморфних бінарних дерев із коренями із кількістю вершин n дорівнює числу Каталана C_n :

$$C_n = \frac{(2 \cdot n)!}{(n!)^2 \cdot (n+1)}, \quad (2.6)$$

або

$$C_n = \frac{C_n^{2n}}{(n+1)} = C_n^{2n} - C_{n-1}^{2n}. \quad (2.7)$$

Також числа Каталана C_n можна обчислити через рекурентне співвідношення:

$$C_0 = 1; C_n = \sum_{i=0}^{n-1} C_i \cdot C_{n-1-i}, \quad n \geq 1. \quad (2.8)$$

У комбінаториці та топології числа Каталана C_n також визначають кількість розбиттів опуклого багатокутника із кількістю вершин $(n+2)$ на трикутники діагоналями, які не перетинаються. Числа Каталана, поряд із числами Фібоначчі, відіграють величезну роль у теорії числових рядів.

Також у теорії кодування впливові значення мають зважені дерева.

Надамо відповідні визначення.

Дерева, ребрам яких надана відповідна вага, називаються *зваженими*.

Зважені дерева, у яких вага всіх ребер складає 0 або 1, називаються *бінарними*.

Приклади звичайного та бінарного зваженого дерева показані на рис. 2.13.



Рис. 2.13. Приклади звичайного (а) та бінарного (б) зваженого дерева

Теорія дерев широко використовується для побудови натуральних кодів двійкових чисел із імовірнісним передбаченням подій, зокрема натуральних кодів Шеннона – Фано та Хаффмена, а також завадостійких кодів Вітербі.

Використання теорії графів для моделювання електронних схем

Зважені орієнтовані графи також ефективно використовуються для аналізу особливостей роботи електронних схем. Історично одним із перших українських вчених, який у шістдесятих роках минулого сторіччя став ефективно використовувати теорію графів для моделювання аналогових електронних схем, був професор Національного технічного університету України «Київський політехнічний інститут» В.П. Сігорський.

На моделях схем, які подаються у вигляді графа, вершини графа являють собою вузли схеми, а ребра характеризують зв'язок між сусідніми вузлами, якщо такий зв'язок існує. Наприклад, якщо між вузлами схеми x_i та x_j підключений опір R_{ij} , коефіцієнт зв'язку між цими вузлами k_{ij} становить:

$$k_{ij} = G_{ij} = \frac{1}{R_{ij}}. \quad (2.9)$$

У разі, якщо між вузлами схеми x_i та x_j підключений реактивний опір, наприклад ємність C_{ij} , такі схеми також можна аналізувати з використанням апарату комплексних змінних. Так, для ємності C_{ij} коефіцієнт k_{ij} залежить від циклічної частоти ω і складає:

$$k_{ij} = G_{ij} = \frac{1}{j \cdot \omega \cdot C_{ij}}. \quad (2.10)$$

Для індуктивності L_{ij} , підключеної між вузлами x_i та x_j , відповідно можна записати:

$$k_{ij} = G_{ij} = j \cdot \omega \cdot L_{ij}. \quad (2.11)$$

Комплексна змінна j у співвідношеннях (2.10) та (2.11) свідчить про зсув фази сигналу на 90° на частоті ω , тобто, множник $\frac{1}{j}$ відповідає зсуву фази сигналу на -90° .

Співвідношення (2.9) – (2.11) відповідають аналізу лінійних електронних схем. Якщо схеми містять нелінійні компоненти, наприклад, напівпровідникові діоди, відповідні ребра у структурі графу будуть мати коефіцієнти:

$$k_{ij} = f(U_{ij}), \quad (2.12)$$

де U_{ij} – напруга між вузлами. Структури таких зважених орієнтованих графів відповідають системам нелінійних рівнянь.

Як взагалі в теорії графів, вузли у графах, які описують підключення елементів в електронних схемах, позначаються точками, а зв'язки між вузлами – лініями або дугами із направленими стрілками. Напрямок стрілок відповідає напрямку струму в електронній схемі і зазвичай співпадає із напрямком поширення сигналу, тобто, у більшості випадків стрілки направлені із входу схеми до її виходу. У орієнтованих графах, які описують роботу електронних схем, можливою є також наявність петльових структур.

Лінійні та петльові структури, які використовуються на графах для описання підключення елементів в електронних схемах, наведені на рис. 2.14.



Рис. 2.14. Елементарні структури орієнтованих графів, які використовуються для описання підключення елементів в електронних схемах: а – лінійна структура, б – петльова структура

Структурі, наведеній на рис. 2.14, а, відповідає лінійне рівняння:

$$x_j = k_{ij} \cdot x_i. \quad (2.12)$$

а структурі, яка наведена на рис. 2.14, б – рівняння:

$$x_i = k_{ii} \cdot x_i. \quad (2.13)$$

Рівняння (2.12), (2.13) описують лише прості структури зв'язаних орієнтованих графів. У разі більш складних структур необхідно переходити до систем лінійних рівнянь та використовувати способи їхнього розв'язування. Головним правилом, яке використовується для формування таких систем рівнянь, є наступна теорема теорії графів.

Теорема 2.12. Якщо до однієї вершини графа сходиться декілька направлених ребер, значення сигналу у цій вершині розраховується як сума добутків коефіцієнтів відповідних ребер на сигнали у суміжній вершині.

У математичній формі теорема 2.12 записується наступним чином:

$$x_j = \sum_{i=1}^n k_{ij} \cdot x_i = k_{1j} \cdot x_1 + k_{2j} \cdot x_2 + \dots + k_{nj} \cdot x_n. \quad (2.14)$$

Структура графа, який відповідає співвідношенню (2.14), наведена на рис. 2.15.

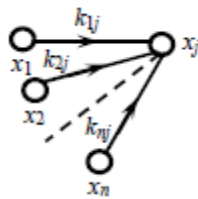


Рис. 2.15. Підграф орієнтованого графу, який відповідає теоремі 2.12 та співвідношенню (2.14)

Розглянемо приклад побудови зв'язаного орієнтованого графу для заданої системи лінійних алгебраїчних рівнянь.

Приклад 2.9. Побудувати зв'язаний орієнтований граф для системи лінійних алгебраїчних рівнянь:

$$\begin{cases} x_2 = a \cdot x_1 + b \cdot x_2 + h \cdot x_3 + c \cdot x_4; \\ x_3 = e \cdot x_2 + k \cdot x_5; \\ x_4 = f \cdot x_3 + d \cdot x_4; \\ x_5 = g \cdot x_4. \end{cases} \quad (2.15)$$

Проаналізуємо систему рівнянь (2.14). Згідно із першим рівнянням цієї системи до вершини x_2 направлені чотири ребра.

1. Ребро $\{x_1, x_2\}$ із вагою a .
2. Ребро $\{x_2, x_2\}$ із вагою b .
3. Ребро $\{x_3, x_2\}$ із вагою h .
4. Ребро $\{x_4, x_2\}$ із вагою c .

Зрозуміло, що ребро $\{x_2, x_2\}$ є петльовим з'єднанням.

Згідно із другим рівнянням системи (2.15) до вершини x_3 направлені два ребра.

1. Ребро $\{x_2, x_3\}$ із вагою e .
2. Ребро $\{x_5, x_3\}$ із вагою k .

Згідно із третім рівнянням системи (2.15) до вершини x_4 направлені два ребра.

1. Ребро $\{x_3, x_4\}$ із вагою f .
2. Ребро $\{x_4, x_4\}$ із вагою d , яке є петльовим з'єднанням.

Згідно із четвертим рівнянням системи (2.15) до вершини x_5 направлено одне ребро $\{x_4, x_5\}$ із вагою g .

Проведений аналіз системи рівнянь (2.15) показує, що можна побудувати орієнтований граф, який описує визначені залежності між змінними $x_1 - x_5$. Одна із можливих структур такого графу показана на рис. 2.16.

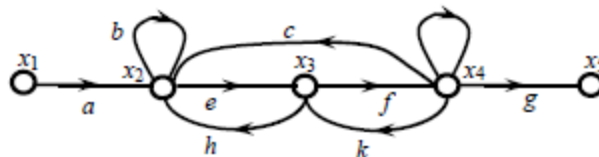


Рис. 2.16. Орієнтований зважений граф для прикладу 2.9

Розглянемо деякі приклади аналізу електронних схем з використанням теорії графів.

Приклад 2.10. Знайти коефіцієнт передачі для схеми із польовими транзисторами, наведеної на рис. 2.17.

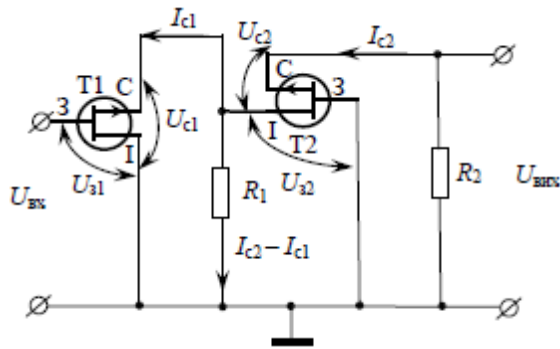


Рис. 2.17. Схема з польовими транзисторами для прикладу 2.10

Перш за все слід відзначити, що польовий транзистор вважається приладом, який керується напругою, і для нього, вважаючи, що струм затвору є незначним та близьким до нуля, можна записати наступну систему рівнянь для балансу струмів:

$$\begin{cases} I_z = 0 \\ I_c = S \cdot U_z + G_i \cdot U_c \end{cases} \quad (2.16)$$

де I_z – струм затвора, I_c – струм стока, U_z – напруга на затворі, U_c – напруга на стоці, S – крутизна характеристики транзистора, G_i – його внутрішній опір.

Щоб описати за допомогою орієнтованого графу схему, наведену на рис. 2.17, з використанням моделі польового транзистора, заданої системою рівнянь (2.16), слід вибрати як вузли графу струми стоків транзисторів I_{c1} та I_{c2} , напруги на їхніх затворах U_{z1} та U_{z2} , а також напруги на стоках транзисторів U_{c1} та U_{c2} . Відповідні електричні параметри схеми вказані на рис. 2.17. Перепишемо систему рівнянь (2.16) у вигляді наступного лінійного алгебраїчного рівняння:

$$U_c = \frac{I_c - S \cdot U_z}{G_i} = R_i \cdot I_c - \mu U_z. \quad (2.17)$$

На основі рівняння (2.17), враховуючи баланс струмів та напруг для схеми, наведеної на рис. 2.16, можна записати наступну систему лінійних рівнянь, які описують роботи схеми, що аналізується:

$$\left\{ \begin{array}{l} U_{вих} = U_{c2} - U_{з2}; \\ U_{c2} = R_{i2} \cdot I_{c2} - \mu_2 \cdot U_{з2}; \\ I_{c2} = -\frac{U_{вих}}{R_2}; \\ U_{з2} = -U_{c1}; \\ U_{c1} = R_{i1} \cdot I_{c1} - \mu_1 \cdot U_{з1}; \\ I_{c1} = I_{c2} + \frac{U_{з2}}{R_1}; \\ U_{вх} = U_{з1}. \end{array} \right. \quad (2.18)$$

Топологічна структура орієнтованого графа, який відповідає системі рівнянь (2.18), наведена на рис. 2.18, а, а результати еквівалентних перетворень цього графа – на рис. 2.18 б – д.

Алгоритм перетворення графа, наведеного на рис. 2.18, а, є наступним.

1. Видалення вершин $U_{з1}$ та I_{c1} . Відповідна структура графа наведена на рис. 2.18, б.

2. Видалення вершини I_{c2} . Відповідна структура графа наведена на рис. 2.18, в.

3. Видалення вершин U_{c1} та U_{c2} . Відповідна структура графа наведена на рис. 2.18, г.

4. Видалення паралельних ребер між вершинами $U_{з2}$ та $U_{вих}$. Відповідна структура графа наведена на рис. 2.18, д.

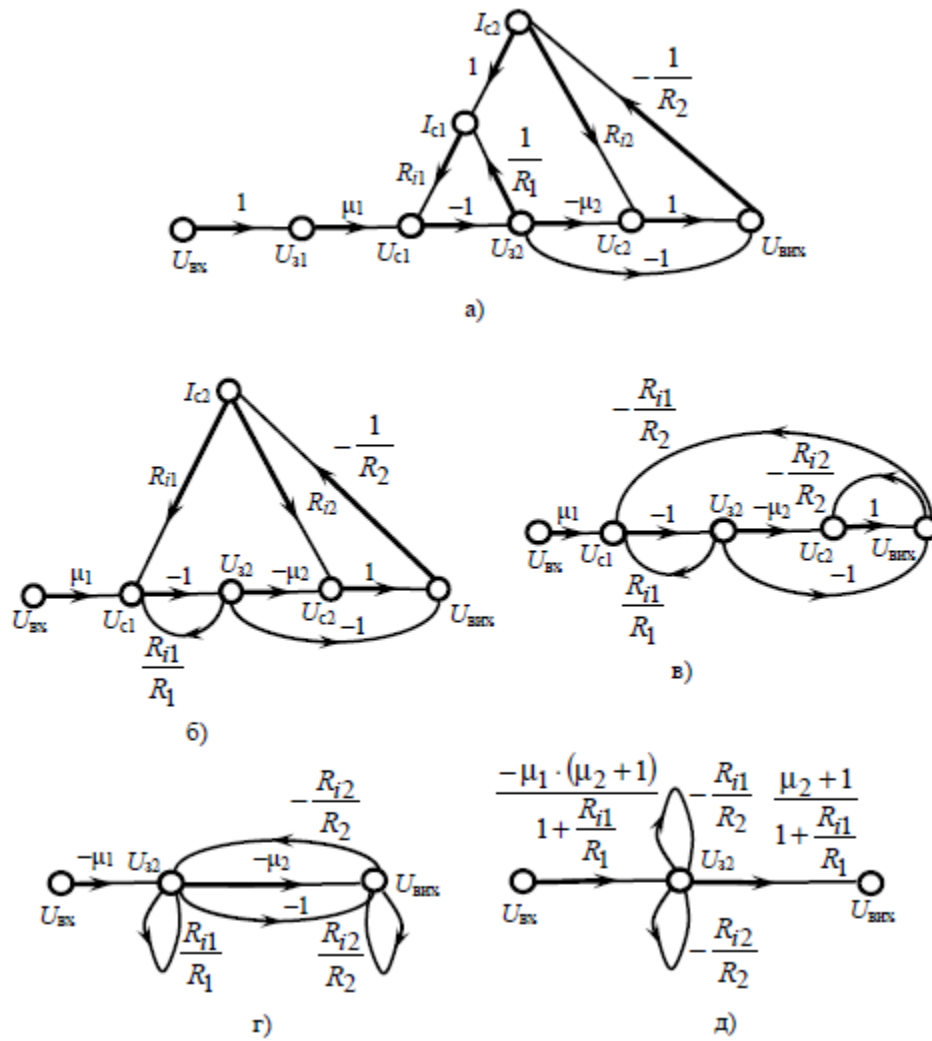


Рис. 2.18. Перетворення топології графа для прикладу 2.10

Відповідно, коефіцієнт передачі сигналу для схеми, яка аналізується, становить:

$$K_U = \frac{U_{вих}}{U_{вх}} = \frac{\frac{-\mu_1(\mu_2 + 1)}{1 + \frac{R_{i1}}{R_1}}}{1 + \left[\frac{(\mu_2 + 1) \cdot \frac{R_{i1}}{R_2}}{1 + \frac{R_{i1}}{R_1}} + \frac{R_{i2}}{R_2} \right]} = \frac{-\mu_1(\mu_2 + 1)}{\left(1 + \frac{R_{i1}}{R_1} \right) \cdot \left(1 + \frac{R_{i2}}{R_2} \right) + (\mu_2 + 1) \frac{R_{i1}}{R_2}}. \quad (2.19)$$

Іншим способом отримання співвідношення (2.19) є аналіз визначника графа, наведеного на рис. 2.18, а. Проаналізуємо наявність контурів у цьому графі. Контур L_1 є найбільшим, тобто охоплює максимальну кількість вершин,

а саме, вершини U_{c1} , U_{32} , U_{c2} , $U_{вих}$, I_{c2} та I_{c1} . Коефіцієнт передачі цього контуру становить:

$$L_1 = -\mu_2 \frac{R_{i1}}{R_2}. \quad (2.20)$$

Другий контур L_2 охоплює вершини U_{c1} , U_{32} та I_{c1} , відповідно, його коефіцієнт передачі визначається через співвідношення:

$$L_2 = -\frac{R_{i1}}{R_1}. \quad (2.21)$$

Для контуру L_3 , який охоплює вершини U_{c1} , U_{32} , $U_{вих}$, I_{c2} та I_{c1} , коефіцієнт передачі обчислюється наступним чином:

$$L_3 = -\frac{R_{i1}}{R_2}. \quad (2.22)$$

Нарешті, для контуру L_4 , який охоплює вершини U_{c2} , $U_{вих}$ та I_{c2} , маємо:

$$L_4 = -\frac{R_{i2}}{R_2}. \quad (2.23)$$

Тоді, враховуючи (2.20) – (2.23), а також те, що всі визначені контури, крім другого та четвертого, перетинаються, для визначника графа можна записати:

$$\begin{aligned} D = 1 - (L_1 + L_2 + L_3 + L_4) &= 1 + \left(-\mu_2 \frac{R_{i1}}{R_2} + \frac{R_{i1}}{R_1} + \frac{R_{i1}}{R_2} + \frac{R_{i2}}{R_2} \right) + \frac{R_{i1}}{R_1} \cdot \frac{R_{i2}}{R_2} = \\ &= \left(1 + \frac{R_{i1}}{R_1} \right) \cdot \left(1 + \frac{R_{i2}}{R_2} \right) + (\mu_2 + 1) \frac{R_{i1}}{R_2} \end{aligned} \quad (2.24)$$

Граф, наведений на рис. 2.18, а, містить два шляхи, коефіцієнт передачі яких, відповідно, складає:

$$P_1 = -\mu_1 \mu_2, \quad P_2 = -\mu_1. \quad (2.25)$$

Тоді, з урахуванням (2.24), (2.25), остаточно отримуємо формулу (2.19).

Тобто, два розглянуті підходи до обчислення коефіцієнта передачі графа, оснований на перетвореннях топології графа та на обчисленні його визначника, дійсно є еквівалентними.

Орієнтовані зважені графи можна також використовувати для аналізу схем на біполярних транзисторах, проте з урахуванням фізичних особливостей роботи цих електронних компонент. Річ у тому, що для біполярного транзистора, як для приладу, який керується струмом, а не напругою, струм бази у робочому стані не дорівнює нулю, тому співвідношення (2.16) у цьому випадку не виконуються. Розглянемо математичну модель біполярного транзистора, параметри якої визначені на рис. 2.19. Важливим є те, що ці параметри суттєво залежать від схеми підключення транзистора. Рис. 2.19, а, відповідає схемі підключення із загальним емітером, рис. 2.19, б – схемі із загальним колектором, а рис. 2.19, в – схемі із загальною базою.

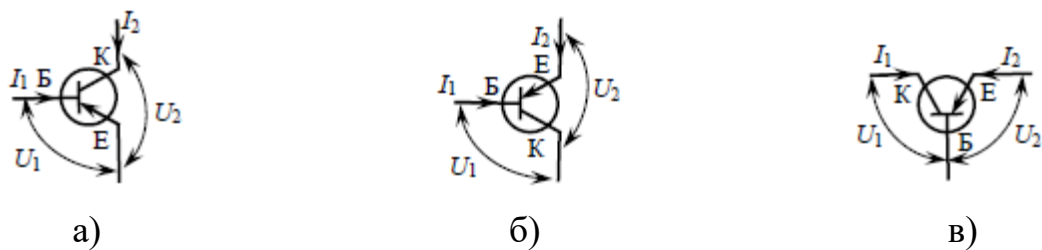


Рис. 2.19. Параметри моделі біполярного транзистора для різних схем підключення

Математичну модель біполярного транзистора можна записати у формі різних лінійних залежностей, залежно від схеми його підключення, і вибір форми подання моделі залежить від того, які параметри задані, а які необхідно обчислити. Взагалі на практиці використовують три форми описання функціональних залежностей між визначеними параметрами U_1 , U_2 , I_1 , I_2 , а саме:

$$1. [I_1, I_2] = F_1(U_1, U_2); 2. [U_1, U_2] = F_2(I_1, I_2); 3. [U_1, I_2] = F_3(U_2, I_1). \quad (2.26)$$

Загалом система рівнянь параметричної моделі транзистора базується на фізико-топологічній моделі Еберса – Мола, яка у загальному вигляді записується наступним чином:

$$\begin{aligned} I_e &= I'_{e0} \left(\exp \left(\frac{U_e}{\phi T} \right) - 1 \right) - \alpha_{0I} I'_{k0} \left(\exp \left(\frac{U_k}{\phi T} \right) - 1 \right); \\ I_k &= \alpha_{0N} I'_{e0} \left(\exp \left(\frac{U_e}{\phi T} \right) - 1 \right) - I'_{k0} \left(\exp \left(\frac{U_k}{\phi T} \right) - 1 \right); \\ I_{\bar{o}} &= (1 - \alpha_{0N}) I'_{e0} \left(\exp \left(\frac{U_e}{\phi T} \right) - 1 \right) - (1 - \alpha_{0I}) I'_{k0} \left(\exp \left(\frac{U_k}{\phi T} \right) - 1 \right), \end{aligned} \quad (2.27)$$

де U_e – напруга на емітері, U_k – напруга на колекторі, $I_{\bar{o}}$ – струм бази, I_e – струм емітера, I_k – струм колектора, I'_{e0} – зворотній струм емітерного переходу транзистора, I'_{k0} – зворотній струм колекторного переходу транзистора, α_{0N} – статичний коефіцієнт передавання струму для нормального включення транзистора, α_{0I} – статичний коефіцієнт передавання струму для інверсного включення транзистора, ϕ – тепловий потенціал, I'_{e0} та I'_{k0} – теплові струми емітерного та колекторного переходів, які обчислюються наступним чином:

$$I'_{e\bar{o}0} = \frac{I_{e0}}{1 - \alpha_{0N} \alpha_{0I}}; \quad I'_{k\bar{o}0} = \frac{I_{k0}}{1 - \alpha_{0N} \alpha_{0I}}. \quad (2.28)$$

З урахуванням аналітичних співвідношень (2.27), (2.28), параметрична модель транзистора, задана функціями (2.26), може бути записана наступним чином:

$$\begin{cases} I_1 = g_{11} \cdot U_1 + g_{12} \cdot U_2; \\ I_2 = g_{21} \cdot U_1 + g_{22} \cdot U_2, \end{cases} \quad (2.29)$$

$$\begin{cases} U_1 = r_{11} \cdot I_1 + r_{12} \cdot I_2; \\ U_2 = r_{21} \cdot I_1 + r_{22} \cdot I_2, \end{cases} \quad (2.30)$$

$$\begin{cases} U_1 = h_{11} \cdot I_1 + h_{12} \cdot U_2; \\ I_2 = h_{21} \cdot I_1 + h_{22} \cdot U_2, \end{cases} \quad (2.31)$$

де $g_{11}, g_{12}, g_{21}, g_{22}, r_{11}, r_{12}, r_{21}, r_{22}, h_{11}, h_{12}, h_{21}, h_{22}$ – внутрішні параметри моделі транзистора, які можна обчислити із системи рівнянь (2.27), (2.28). Параметри моделі для системи рівнянь (2.29) задані на рис. 2.19, а, для системи рівнянь (2.30) – на рис. 2.19, б, а для системи (2.31) – на рис. 2.19, в. Топологічні структури орієнтованих графів, які відповідають системам лінійних рівнянь (2.29) – (2.31), показані на рис. 2.20, а – в.

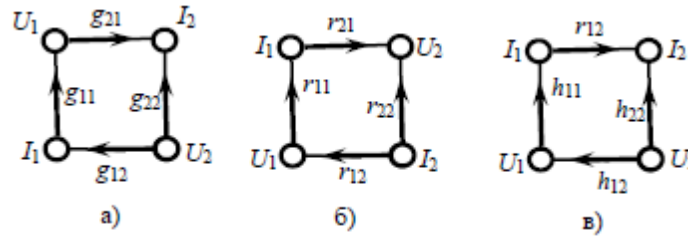


Рис. 2.20. Топологічні структури орієнтованих графів, еквівалентні моделям біполярного транзистора, які описуються системами лінійних рівнянь (2.29) (а), (2.30) (б) та (2.31) (в)

Слід відзначити, що параметри біполярного транзистора g , r та h пов'язані між собою через аналітичні співвідношення. Наприклад, аналітичні вирази для параметрів g через параметри r та h можна записати наступним чином:

$$g_{11} = \frac{r_{22}}{r_{11} \cdot r_{22} - r_{12} \cdot r_{21}} = \frac{1}{h_{11}}; \quad g_{12} = \frac{r_{12}}{r_{11} \cdot r_{22} - r_{12} \cdot r_{21}} = \frac{h_{12}}{h_{11}}; \\ g_{21} = \frac{r_{21}}{r_{11} \cdot r_{22} - r_{12} \cdot r_{21}} = \frac{h_{21}}{h_{11}}; \quad g_{22} = \frac{r_{22}}{r_{11} \cdot r_{22} - r_{12} \cdot r_{21}} = \frac{h_{11} \cdot h_{22} - h_{12} \cdot h_{21}}{h_{11}}. \quad (2.32)$$

Складність використання співвідношень (2.29) – (2.32) для побудови орієнтованих графів для схем із біполярними транзисторами полягає у тому, що, як видно на рис. 2.19, для різних схем підключення транзистора використовуються однакові параметри моделі U_1, U_2, I_1, I_2 . Наприклад, для схеми з загальною базою $I_1 = I_k$, а для схеми із загальним емітером $I_1 = I_e$. Проте, з іншого боку, така систематизація параметрів моделі дозволяє перейти

на більш високий рівень абстракції під час побудови орієнтованих графів. Дійсно, незалежно від способу підключення транзистора його входні та вихідні параметри залишаються незмінними. З іншого боку, у будь-якому разі параметри моделі транзистора відповідають фізико-топологічній моделі Еберса – Мола, заданої співвідношеннями (2.27), (2.28), а зв'язок між внутрішніми параметрами моделі визначається співвідношеннями, аналогічними співвідношенням (2.32). Вибір тієї або іншої моделі транзистора залежить від способу його підключення в електричному колі.

Розглянемо приклад побудови орієнтованого графа для схеми із біполярними транзисторами.

Приклад 2.11. Побудувати орієнтований граф для схеми із біполярними транзисторами, наведеної на рис. 2.21.

Відмінність схем із біполярними транзисторами від схем із польовими транзисторами, аналогічних тій, яка аналізувалась у прикладі 2.10, полягає у тому, що вхідний струм для біполярних транзисторів не дорівнює нулю, тому граф для схеми, наведеної на рис. 2.21, повинен мати дві вхідних вершини, U_{ex} та I_{ex} . Припустимо, що вхідним параметром є U_{ex} , тоді вхідний струм I_{ex} може бути розрахований через параметри графа, але у цьому випадку для визначення струму I_{ex} слід записати додаткове рівняння.

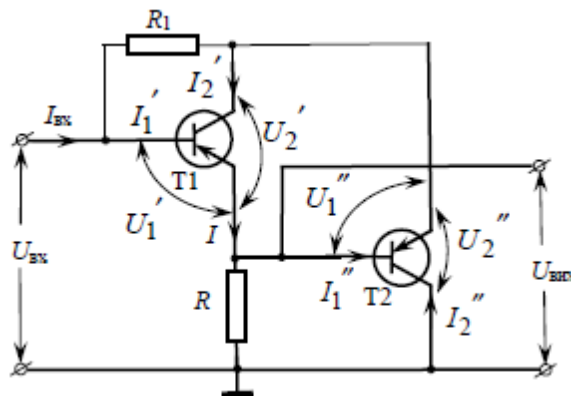


Рис. 2.21. Схема з біполярними транзисторами для прикладу 2.11

Будемо проводити формування системи рівнянь виходячи з того, що транзистори підключені за схемою із загальним емітером. Для вихідного кола

схеми, наведеної на рис. 2.21, згідно із законом балансу напруг, можна записати наступне рівняння:

$$U_{вих} = U_1'' - U_2''. \quad (2.33)$$

З іншого боку, напруги U_2'' та U'' можна явно виразити через струми транзистора Т2 з використанням системи рівнянь (2.30):

$$\begin{cases} U_1'' = r_{11}'' \cdot I_1'' + r_{12}'' \cdot I_2''; \\ U_2'' = r_{21}'' \cdot I_1'' + r_{22}'' \cdot I_2'', \end{cases} \quad (2.34)$$

Рівняння балансу для струмів транзистора Т2 I_1'' та I_2'' , які входять до співвідношення (2.34), згідно із схемою, наведеною на рис. 2.21, можна записати наступним чином:

$$\begin{cases} I_1'' = I_1' + I_2' - \frac{1}{R} \cdot U_{вих}; \\ I_2'' = \frac{1}{R} \cdot U_{вих} - I_{вх}. \end{cases} \quad (2.35)$$

Далі необхідно визначити струми транзистора Т1 I' та I' , які входять до співвідношення (2.35). Використовуючи систему рівнянь (2.29), можна записати:

$$\begin{cases} I_1' = g_{11}' \cdot U_1' + g_{12}' \cdot U_2'; \\ I_2' = g_{21}' \cdot U_1' + g_{22}' \cdot U_2'. \end{cases} \quad (2.36)$$

Але, крім цього, до системи рівнянь (2.35) входить вхідний струм схеми $I_{вх}$, який, згідно із наведеними вище міркуваннями, не є вхідним джерелом сигналу, а в такому разі його необхідно виразити через інші параметри схеми. Рівняння балансу струмів на вході схеми, наведеної на рис. 2.21, можна записати наступним чином:

$$I_{вх} = I_1' + \frac{1}{R_1} \cdot (U_1' - U_2'). \quad (2.37)$$

Тепер необхідно записати систему рівнянь для визначення напруг U' та U_2' на транзисторі Т1. Згідно із рівнянням балансу напруг маємо:

$$\begin{cases} U'_1 = U_{ex} - U_{vix}; \\ U'_2 = -U'. \end{cases} \quad (2.38)$$

Згідно із співвідношеннями (2.33) – (2.38), електронна схема, яка наведена на рис. 2.21, описується наступною системою із десяти лінійних рівнянь з десятьма невідомими, яка є зімкненою, повною та не суперечливою:

$$\begin{cases} U_{vix} = U''_1 - U''_2; \\ U''_1 = r''_{11} \cdot I''_1 + r''_{12} \cdot I''_2; \\ U''_2 = r''_{21} \cdot I''_1 + r''_{22} \cdot I''_2; \\ I''_1 = I'_1 + I'_2 - \frac{1}{R} \cdot U_{vix}; \\ I''_2 = \frac{1}{R} \cdot U_{vix} - I_{ex}; \\ I'_1 = g'_{11} \cdot U'_1 + g'_{12} \cdot U'_2; \\ I'_2 = g'_{21} \cdot U'_1 + g'_{22} \cdot U'_2; \\ I_{ex} = I'_1 + \frac{1}{R_1} \cdot (U'_1 - U'_2); \\ U'_1 = U_{ex} - U_{vix}; \\ U'_2 = -U'. \end{cases} \quad (2.39)$$

Орієнтований граф, побудований за системою лінійних рівнянь (2.39), наведений на рис. 2.22.

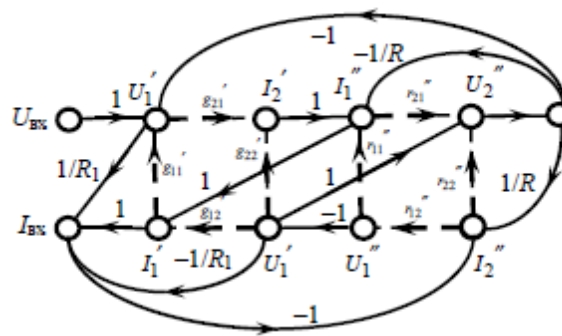


Рис. 2.22. Структура графа для прикладу 2.11

Контрольні запитання

1. Що являє собою теорія графів і як ця теорія використовується для аналізу електронних схем та способів кодування сигналів в електронних системах? Наведіть власні приклади такого використання.
2. Що являє собою граф з точки зору теорії множин? Наведіть власні приклади визначення графу як упорядкованої множини.
3. Що являють собою вершини та ребра графу?
4. Що являє собою підграф та як способи формування підграфів використовуються для аналізу структури графів? Наведіть власні приклади такого використання.
5. Що являє собою шлях у графі та як способи визначення шляхів використовуються для аналізу структури графів? Наведіть власні приклади такого використання.
6. Що являють собою орієнтовані графи та як вони використовуються для аналізу електронних схем та способів кодування сигналів? Наведіть власні приклади такого використання.
7. Як визначається степінь виходу вершини орієнтованого графа, джерело графа та сток графа? Наведіть власні приклади визначення цих параметрів орієнтованих графів.
8. Як визначаються підграфи орієнтованих графів? Наведіть власні приклади виділення підграфу із структури орієнтованого графу.
9. Що являє собою довжина шляху у орієнтованому графі та як визначається довжина шляху орієнтованого графу?
10. Які неорієнтовані графи називаються співвіднесеними до орієнтованих?
11. Що являють собою планарні графи і як вони можуть бути використані у сучасному виробництві електронної техніки? Наведіть власні приклади такого використання планарних графів.
12. Що являють собою грані планарного графа та які грані графа називаються зовнішніми?

13. Що являють собою дерева як різновид графів та як вони використовуються для формування та аналізу двійкових кодів?

14. Які дерева називаються орієнтованими?

15. Які головні параметри визначаються для дерева як топологічної структури?

16. Що являють собою числа Каталана та як вони використовуються для аналізу топологічних структур дерев? Наведіть власні приклади такого використання чисел Каталана.

17. Які дерева називаються бінарними? Наведіть власні приклади бінарних дерев.

18. Які дерева називаються зваженими? Наведіть власні приклади зважених дерев.

19. Як зважені орієнтовані графи можуть бути використані для аналізу роботи електронних схем та розрахунку їхніх параметрів? Наведіть власні приклади такого використання зважених орієнтованих графів.

Перелік посилань

1. James A. Anderson. Discrete Mathematics with Combinatorics. – Oxford University Press – 2020, – 198 p.

2. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. Посібник. – Львів: Новий світ – 2000, 2009, – 736 с.

Лекція №3. Скінченні автомати Мілі і Мура

Існує велика кількість різноманітних схем, у яких використовується декілька тригерів. У такому випадку кількість внутрішніх станів схеми зростає в пропорції 2^m , де m – кількість тригерів. При цьому, зрозуміло, зростає складність “реакції” такого пристрою на вхідні сигнали, оскільки кожного разу внутрішні стани тригерів можуть бути іншими.

Цифрові пристрої з m тригерами ($m > 1$), стан виходів яких залежить не тільки від значень вхідних сигналів у заданий момент часу, а й від стану використовуваних тригерів у даний та попередній моменти часу, називаються *цифровими автоматами*.

У загальному плані *цифровим автоматом* називається пристрій, який формує ряд вихідних дискретних сигналів у відповідності із вхідною бінарною комбінацією сигналів [1, 2]. Найпростішим з таких автоматів є перетворювач кодів, і він називається *комбінаційним*.

У подальшому розглядатимуться автомати з пам'яттю, завдяки якій значення їх виходів залежать не лише від вхідних сигналів, що надходять у визначений момент часу, а також від внутрішнього стану автомата. Оскільки кількість внутрішніх станів залежить від кількості використовуваних тригерів, то цифрові автомати поділяються на *скінченні*, що мають обмежену кількість станів, і автомати з *нескінченною кількістю станів*, або *послідовнісні машини*.

Будь-який цифровий автомат є сукупністю елементів пам'яті (тригерів) та комбінаційних схем. Тригери містять інформацію про особливості попередньої роботи автомата. Комбінаційні схеми на основі вхідних автоматів та інформації, яка береться з тригерів, формують вихідні сигнали і сигнали для формування нових станів тригерів. Таким чином, однією з особливостей цифрових автоматів є те, що вони мають свої внутрішні стани, від яких залежить реакція на вхідні сигнали. Одним з найпростіших прикладів цифрових автоматів є кодовий замок, реакція на черговий сигнал якого залежить від попередніх сигналів.

Скінченні автомати можуть бути *синхронними*, зміна станів яких відбувається в тактові моменти часу, що задаються зовнішнім генератором. В *асинхронних* автоматах зміна станів відбувається внаслідок дії входних сигналів без затримок. Тому асинхронні автомати вважаються більш швидкодіючими і знаходять використання у швидкодіючих інформаційних пристроях вимірювання і керування різноманітними процесами, де необхідна миттєва реакція на зміну входних сигналів.

Синхронні автомати в силу специфіки своєї роботи вносять у процес вимірювання або керування затримку, що визначається величиною періоду синхросигналу.

Здебільшого асинхронні автомати будуються на основі асинхронних елементів пам'яті – асинхронних тригерів. Синхронні автомати будуються з використанням синхронних тригерів.

Математичною моделлю цифрового автомата є абстрактний автомат, в якому враховуються його входні та вихідні сигнали, а також внутрішні стани. Будемо розглядати детерміновані автомати, які завжди мають початковий стан Q_0 , з якого починають працювати при дії входних сигналів і при повторі перебору входних сигналів повторюють послідовність станів і вихідних сигналів.

Абстрактний автомат задається *множиною внутрішніх станів* $Q = \{ Q_1, Q_2, \dots, Q_m \}$ (*алфавітом станів*), *множиною входних сигналів* $X = \{ X_1, X_2, \dots, X_p \}$ (*вхідним алфавітом*), *множиною вихідних сигналів* $Y = \{ Y_1, Y_2, \dots, Y_k \}$ (*вихідним алфавітом*) і *початковим станом* Q_0 . Перехід з одного стану в інший визначається функцією переходів f_p , що визначає стан автомата Q_s , в який він переходить з попереднього стану Q_m при дії сигналу X_p :

$$Q_s = f_p (Q_m, X_p).$$

Значення виходів автомата задається функцією виходів λ , що залежить від стану автомата Q_m і входного сигналу X_p :

$$Y_k = \lambda (Q_m, X_p).$$

Абстрактний автомат працює в дискретному часі, який задається цілими позитивними числами $t = 0, 1, 2, \dots$

У кожний момент дискретного часу t , який зветься *тактом*, автомат перебуває у деякому стані $Q(t)$ з множини станів автомата Q .

У початковий момент часу ($t = 0$) він завжди знаходиться в стані $Q(0) = Q_0$. Вважається, що реакція автомата на вхідні сигнали не залежить від інтервалів часу між тактовими моментами часу.

У момент t , знаходячись у стані $Q(t)$, автомат сприймає на своєму вході сигнали, які називаються *буквами (літерами) вхідного алфавіту* $X_p \in X$. У відповідності до функції переходів f_p , він перейде у стан $Q_s(t)$, що описується алфавітом станів, тобто $Q_s(t) \in Q$.

Аналогічно, у відповідності з функцією виходів λ , на виході отримається сигнал Y_k , де $Y_k \in Y$.

Приклад 3.1. Розглянути RS-тригер як скінчений автомат.

Розв'язання. RS-тригер має два стани із множини станів Q : $Q_1 = 0$ і $Q_2 = 1$, де $Q_1, Q_2 \in Q$. Автомат має чотири значення входів X_p з алфавіту входів X_1, X_2, X_3, X_4 , де $X_1 = \bar{S} \cdot \bar{R} = 00$; $X_2 = S \cdot \bar{R} = 10$; $X_3 = \bar{S} \cdot R = 01$; $X_4 = S \cdot R = 11$. Вихідної комбінаційної логіки автомат не має, тобто залежність $Y_k = \lambda(Q_m, X_p)$ визначається формулою: $Y = Q$.

Скінчена множина букв вхідного алфавіту, вихідного алфавіту і станів називається *словами*. Для скінченого автомата кількість символів цих алфавітів обмежена і однакова. Повний перебір символів вхідного слова повинен привести автомат до початкового стану Q_0 . Автомат, який завжди починає працювати з початкового стану, називається *ініціальним*.

Поняття *станів* в описі автоматів пов'язане з необхідністю враховувати типи і характер попередніх сигналів, тобто сигналів, які діяли на один або декілька тактів раніше. Стани автомата і є тими відповідними елементами пам'яті, що характеризують попередні сигнали. Введення поняття станів дозволяє усунути час як явну змінну і визначати вихідний сигнал як функцію

внутрішнього стану і входу в тактовий момент часу. Такий спосіб опису автоматів має великі переваги перед іншими для асинхронних пристроїв, в яких інтервали часу між сусідніми тактами можуть мати довільні значення, що значно відрізняються між собою. З такої точки зору комбінаційні пристрої відносяться до автоматів, в яких вихід не залежить від попередніх сигналів і повністю визначається комбінацією вхідних сигналів.

У практиці роботи автоматів часто мають місце випадки, коли деякі комбінації значень вхідних символів подавати неприпустимо (згадаємо таблицю станів *RS*-тригера). Такі комбінації є *забороненими* для автомата. Автомати, що мають заборонені вхідні слова, називаються *частковими*. Наприклад, деякий абстрактний автомат має вхідний алфавіт X , що складається з двох символів X_0 і X_1 , кожен з яких може приймати значення **0** або **1**. Слова $\overline{X_1} \overline{X_0}$, $\overline{X_1} X_0$, $X_1 \overline{X_0}$ є *дозволеними*, а слово $X_1 X_0$ – *забороненим*.

Наявність заборонених станів зменшує кількість слів вхідного алфавіту, зменшує кількість станів і дозволяє будувати часткові автомати з меншими затратами, ніж повні, для яких вказаних обмежень не існує.

Функція $Y_k = \lambda(Q_m, X_p)$ задає значення виходів автомата з алфавіту Y . Кількість символів алфавіту Y не співпадає з кількістю символів алфавіту X . За аналогією з перетворювачами кодів, автомат забезпечує відображення слів вхідного алфавіту в слова вихідного алфавіту. Якщо прийняти можливу кількість слів вхідного алфавіту за P_x , вихідного алфавіту за K_y , то в залежності від співвідношення між P_x та K_y автомати можуть за функціональним призначенням розділятися на три групи.

При $K_y < P_x$ маємо автомати, які призначені для розв'язання задач керування, стиснення інформації, розпізнавання повідомлень. Наприклад, цифровий кодовий замок може мати 3...5 вхідних символів і лише один вихідний. Інший приклад – розпізнавання введеного пароля при вході в комп'ютер.

Якщо $K_y > P_x$, то маємо ситуацію, коли довжина вихідних слів буде більшою, ніж вхідних, оскільки кількість слів повинна бути однаковою. Тобто

вихідні слова будуть нести збиткову інформацію, що використовується, як відмічалось в попередніх розділах, для синтезу перешкодостійких кодів.

При $K_y = P_x$ кількість і довжина вхідних і вихідних слів однакові. Такі автомати використовуються для деяких кодових перетворень (наприклад, перетворення двійкового коду в код Грея, і т. д.), для реалізації деяких методів захисту даних від несанкціонованого доступу (метод підстановок) і т. п.

Автомати Мілі і Мура. Закони функціонування автоматів

У практиці використання цифрових автоматів можна виділити невелику кількість типових алгоритмів їх функціонування. Найбільшого розповсюдження набули два типи автоматів – *автомати Мілі* і *Мура* [1, 2].

Закон функціонування автомата Мілі задається рівняннями:

$$\begin{cases} Q(t+1) = f_p [Q(t), X(t)]; \\ Y_k(t) = \lambda [Q(t), X(t)]; \end{cases} \quad t = 0, 1, 2, \dots \quad (3.1)$$

Закон функціонування автомата Мура описується рівняннями:

$$\begin{cases} Q(t+1) = f_p [Q(t), X(t)]; \\ Y_k(t) = \lambda [Q(t)]; \end{cases} \quad t = 0, 1, 2, \dots \quad (3.2)$$

Різниця між автоматами Мілі і Мура полягає лише в тому, що вихідний сигнал останнього залежить тільки від внутрішнього стану, у той час як в автоматі Мілі значення вихідного сигналу залежить також і від вхідного сигналу.

Структура автомата Мілі зображена на рис. 3.1.

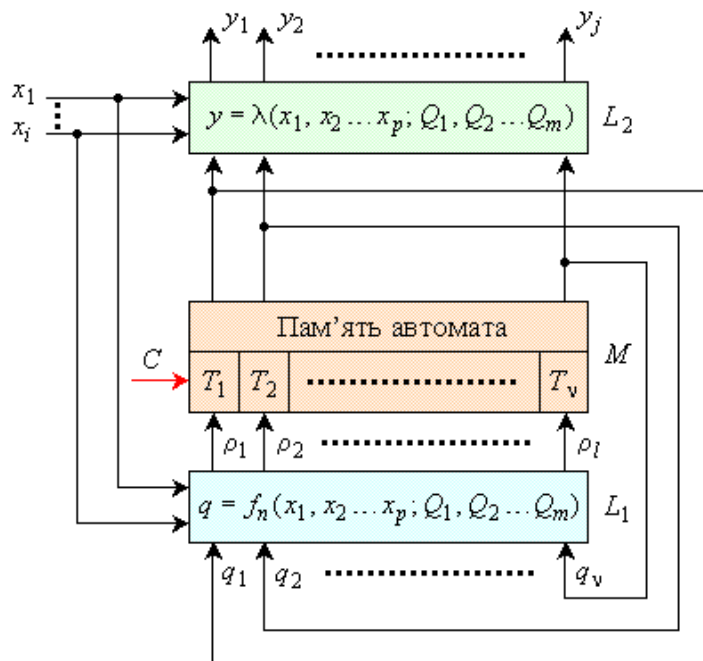


Рис. 3.1. Структура автомата Мілі

Автомат складається з набору елементарних автоматів (тригерів $T_1, T_2, \dots, T_n$), стани яких $q_1, q_2, \dots, q_n$ в кожний момент часу визначають внутрішній стан автомата в цілому. Під дією вхідних сигналів $x_1 \dots x_i$, які подаються в дискретні моменти часу, відбувається формування сигналів $\rho_1 \dots \rho_l$, що забезпечують послідовне перемикання тригерів $T_1 \dots T_n$. Таким шляхом відбувається послідовна зміна станів автомата. Оскільки комбінаційна схема L_1 формує керуючі сигнали тригерів в залежності від значень виходів тригерів у момент подачі тактового сигналу (для синхронних схем) або в моменти подачі вхідних сигналів (для асинхронних), то значення виходів тригерів у послідовний момент часу i , відповідно, стан автомата однозначно визначається вхідними сигналами і станом автомата в попередній момент часу.

Сигнали керування тригерами $\rho_1 \dots \rho_l$ формуються комбінаційним пристроєм L_1 , структура якого визначає функцію переходів автомата. Функція виходів реалізується комбінаційним пристроєм L_2 , який формує сигнали

$y_1 \dots y_j$ як функції виходів пам'яті (для автоматів Мура) або як функції виходів пам'яті і вхідних сигналів (для автоматів Мілі).

Переходячи до відносного часу роботи, формули (3.1), (3.2) можна зобразити у вигляді:

- для автомата Мілі:

$$\begin{cases} Y_{n+1} = L_2(Q_{n+1}, X_n); \\ Q_{n+1} = L_1(Q_n, X_n); \end{cases} \quad (3.3)$$

- для автомата Мура:

$$\begin{cases} Y_{n+1} = L_2(Q_{n+1}); \\ Q_{n+1} = L_1(Q_n, X_n). \end{cases} \quad (3.4)$$

Рівняння (3.3), (3.4) можуть бути заданими аналітично або у вигляді таблиць станів (таблиць відповідностей).

Взаємозв'язок між поточними Q_n і послідовними значеннями Q_{n+1} виходів визначається характеристичними рівняннями використовуваних тригерів.

У табл. 3.1 приведені характеристичні рівняння тригерів.

Таблиця 3.1. Характеристичні рівняння тригерів

Тип використовуваного тригера	Характеристичне рівняння
Асинхронний RS -тригер	$Q_{n+1} = S_n + \overline{R_n} Q_n$
Синхронний статичний D -тригер	$Q_{n+1} = C_n D_n$
Динамічний D -тригер	$Q_{n+1} = D_n$
Динамічний JK -тригер (JK - MS -тригер)	$Q_{n+1} = J_n \overline{Q_n} + \overline{K_n} Q_n$
T -тригер	$Q_{n+1} = C_n \overline{Q_n}$
T -тригер з дозволяючим входом	$Q_{n+1} = C_n (V \overline{Q_n} + \overline{V} Q_n)$

Поєднуючи характеристичні рівняння тригерів і рівняння комбінаційних схем, можна проводити аналіз роботи існуючого автомата або виконувати його синтез.

Способи опису роботи автоматів

В практиці аналізу і синтезу цифрових автоматів використовують різні способи опису їх роботи [3, 4]. Найбільш поширеними є *табличний* і *графічний* способи.

Розглянемо спочатку опис роботи автомата з використанням таблиць переходів та виходів. Стовбці (рядки) цих таблиць позначають символами з множини Q , а рядки (стовбці) – символами з множини X .

Кількість рядків таблиці переходів визначається кількістю комбінацій вхідних сигналів ρ , а кількість стовбців – відповідно, кількість станів M автомата.

У табл. 3.2 зображена таблиця автомата з $\rho = 3$, $M = 2^m = 4$.

В кожній клітинці таблиці переходів записується стан, в який переходить автомат з попереднього стану (стану, що стоїть у заголовку стовпця) при дії відповідного вхідного сигналу. Так, наприклад, якщо автомат знаходиться у стані Q_0 , то при дії сигналу X_1 він перейде в стан Q_3 ; при дії сигналу X_2 залишиться в стані Q_0 , а при дії сигналу X_3 перейде в стан Q_1 .

Таблиця 3.2. Таблиця автомата з $\rho = 3$, $M = 2^m = 4$

$X \backslash Q$	Q_0	Q_1	Q_2	Q_3
X_1	Q_3	Q_2	Q_1	Q_0
X_2	Q_0	Q_3	Q_2	Q_1
X_3	Q_1	Q_0	Q_3	Q_2

Таблиця виходів (табл. 3.3) відрізняється від таблиці переходів лише тим, що у кожній клітці записується відповідне значення вихідного сигналу автомата.

Таблиця 3.3. Таблиця виходів

$X \backslash Q$	Q_0	Q_1	Q_2	Q_3
X_1	Y_0	Y_1	Y_2	Y_3
X_2	Y_3	Y_0	Y_1	Y_2
X_3	Y_2	Y_3	Y_0	Y_0

Таблиці переходів і виходів автомата Мілі можуть бути представлені у вигляді однієї суміщеної таблиці, у клітках якої вказані значення як станів, так і виходів.

Функції переходів і виходів автомата Мура задаються однією таблицею переходів, яка будується так само, як і таблиця переходів автомата Мілі. Різниця полягає лише в тому, що над заголовками кожного стовпця встановлюється окремим рядком значення виходів автомата (табл. 3.4).

Таблиця 3.4. Таблиця переходів

	Y_2	Y_1	Y_0
$Q \backslash X$	Q_0	Q_1	Q_2
X_1	Q_2	Q_1	Q_2
X_2	Q_0	Q_2	Q_1
X_3	Q_1	Q_0	Q_0

Для частково заданих автоматів, у яких функції виходів і функції переходів визначені не для всіх комбінацій Q_m і X_p , відповідні клітки залишаються незаповненими.

Як приклад, розглянемо більш детально автомати, що описуються табл. 3.2 – табл. 3.4.

При двійкових вхідних сигналах входи автоматів X_1 , X_2 , X_3 можуть бути задані вхідними логічними змінними x_1 та x_2 , тобто з множини $J = 2$. В алфавіті X матимемо $2^J = 4$ різних слова, а саме: $X_1 = \overline{x_1} \overline{x_2}$, $X_2 = x_1 \overline{x_2}$, $X_3 = \overline{x_1} x_2$, $X_4 = x_1 x_2$. Звідси витікає, що $X_4 = x_1 x_2$ – це заборонене слово для автоматів, що розглядаються. Але така заборона може існувати не для всіх станів автомата, а тільки для деяких. Приклад такого автомата заданий таблицею переходів табл. 3.5. З таблиці витікає, що при стані Q_0 на вхід автомата не повинен надходити сигнал X_2 , оскільки перехід у такому випадку не визначений.

Таблиця 3.5. Таблиця переходів неповністю заданого автомата

$X \backslash Q$	Q_0	Q_1	Q_2
X_1	Q_1	Q_2	Q_0
X_2	—	Q_2	Q_1

Можна прийняти, що $X_1 = x = 1$, а $X_2 = \bar{x} = 0$ і, відповідно, лише значення $X_1 = 1$ може змінювати стан автомата. З табл. 8.5 однозначно визначається і послідовність зміни станів автомата при дії X_1 : $Q_0 \Rightarrow Q_1 \Rightarrow Q_2 \Rightarrow Q_0$. Для цього необхідно задати послідовність X_1, X_1, X_1 . З цього витікає, що автомат закінчує свою роботу переходом до початкового стану. Якщо функція $\lambda = 1$, то виходи автомата Мілі одночасно визначатимуться значеннями його внутрішніх станів. В тому випадку, коли X_1 є тактовим сигналом, що діє лише на тригери, а стани автомата змінюються упорядковано в зростаючому або спадаючому напрямку, автомат називається *лічильним автоматом*, або *лічильником*.

Більш наочним є спосіб опису автоматів з допомогою графів, подібно до того, як описувалася робота тригерів. Різниця полягає в тому, що автомат може мати суттєво більшу кількість станів. На рис. 3.2 показані граф-схеми автоматів Мілі і Мура, які задані табл. 3.2 – 3.3 (рис.3.2.а), табл.3.4 (рис.3.2.б).

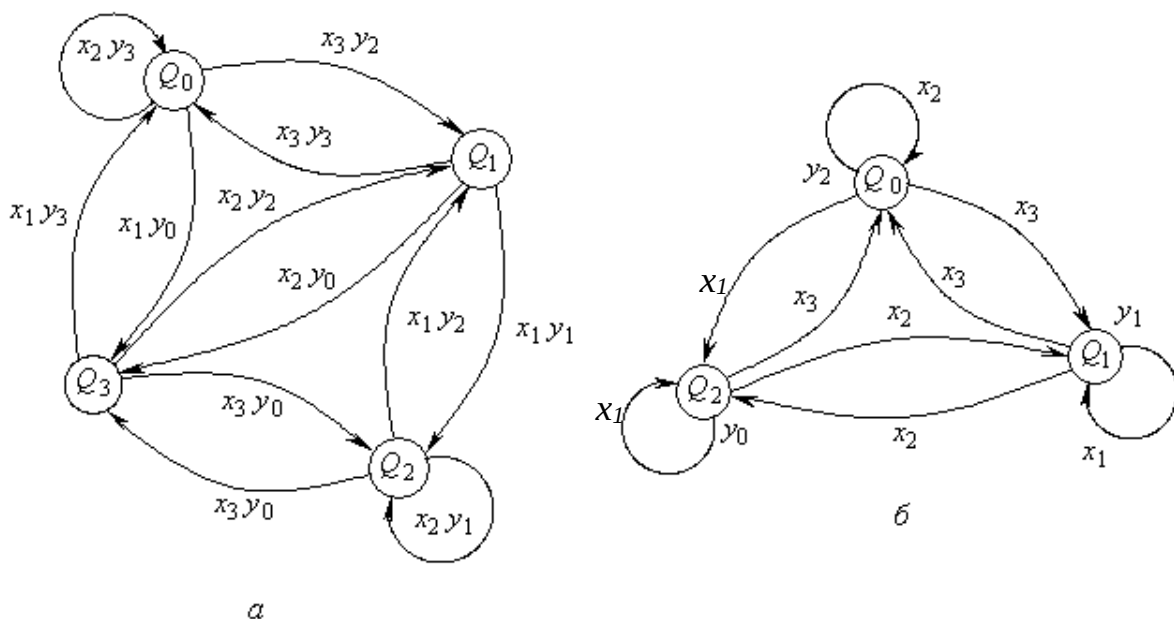


Рис. 3.2. Граф-схеми автоматів Мілі і Мура

Граф-схеми широко використовуються як при аналізі, так і при синтезі автоматів, а також при переході від словесного до формалізованого їх опису.

За допомогою таблиць переходів і виходів, як і за допомогою граф-схеми завжди можна знайти вихідну реакцію автомата на будь-яке вхідне слово, що належить множині X .

Робота автомата може описуватися у часі за допомогою спеціальної таблиці (табл. 3.6), яка називається *стрічкою цифрового автомата*.

Таблиця 3.6. Стрічка цифрового автомата

Такт	0	1	2	3	4	5	6	7	8	9	10
Вхідний сигнал	X_2	X_1	X_1	X_2	X_3	X_1	X_1	X_3	X_2	X_1	X_1
Стан	Q_0	Q_2	Q_3	Q_0	Q_2	Q_0	Q_3	Q_1	Q_2	Q_3	Q_3
Вихідний сигнал	Y_0	Y_1	Y_2	Y_0	Y_0	Y_1	Y_2	Y_2	Y_1	Y_1	Y_2

Особливість такої стрічки полягає в тому, що для будь-якої пари сусідніх тактів i та $(i+1)$ можна виділити четвірку символів (виділена у табл. 3.6 жирною лінією), яка показує, в який стан перейде цифровий автомат в $(i+1)$ -ому такті і який вихідний сигнал буде сформований під дією вхідного сигналу.

Однією з форм зображення автомата є його *дерево переходів і виходів*. Дерево може мати декілька ярусів, у кожному з яких за допомогою гілок показуються можливі переходи, починаючи з нульового стану.

На рис. 3.3 приводиться приклад дерева переходів, що відповідає табл. 3.2.

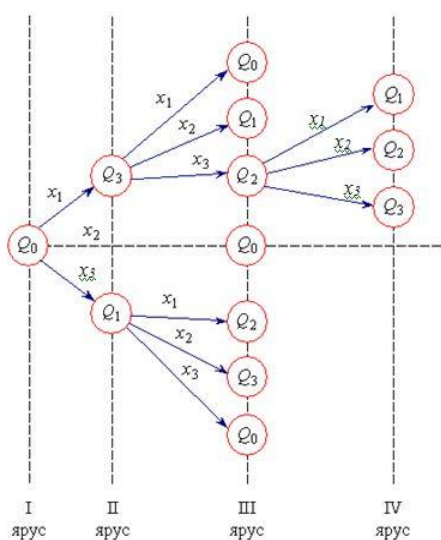


Рис. 3.3. Дерево переходів

Еквівалентні перетворення автоматів

Для будь-якого автомата Мілі можна побудувати відповідний автомат Мура, і навпаки [3, 4].

Розглянемо алгоритми таких переходів на прикладі автомата Мура, що заданий табл. 3.7 і табл. 3.8.

Таблиця 3.7. Таблиця переходів автомата Мура

$X \backslash Q$	Q_0	Q_1	Q_2	Q_3
x_1	Q_1	Q_2	Q_0	Q_2
x_2	Q_3	Q_0	Q_0	Q_0
x_3	Q_2	Q_3	Q_2	Q_3

Таблиця 3.8. Таблиця виходів автомата Мура

$X \backslash Q$	Q_0	Q_1	Q_2	Q_3
x_1	Y_3	Y_2	Y_1	Y_1
x_2	Y_2	Y_1	Y_3	Y_3
x_3	Y_1	Y_2	Y_2	Y_3

Поставимо у відповідність кожній парі Q_m та X_p автомата Мілі стан Q_{mp} автомата Мура. До множини станів автомата Мура включимо початковий стан автомата Мілі Q_0 , але позначимо його Q_{00} . Для прикладу, що розглядається, така відповідність приведена в табл. 3.9.

Таблиця 3.9. Таблиця відповідності автомата Мілі автомату Мура

$X \backslash Q$	Q_0 Q_{00}	Q_1	Q_2	Q_3
x_1	Q_1 Q_{01}	Q_2 Q_{11}	Q_0 Q_{21}	Q_2 Q_{31}
x_2	Q_3 Q_{02}	Q_0 Q_{12}	Q_0 Q_{22}	Q_0 Q_{32}
x_3	Q_2 Q_{03}	Q_3 Q_{13}	Q_2 Q_{23}	Q_3 Q_{33}

Якщо автомат Мілі має m станів і p вхідних сигналів, то відповідний йому автомат Мура матиме $mp + 1$ станів.

З табл. 3.9 витікає той факт, що стан Q_0 автомата Мілі співпадає зі станами Q_{00} , Q_{21} , Q_{12} , Q_{22} , Q_{32} автомата Мура. Тобто має місце наступна тотожність:

$$Q_0 = Q_{00} = Q_{21} = Q_{12} = Q_{22} = Q_{32}.$$

Аналогічно,

$$Q_1 = Q_{01}; \quad Q_2 = Q_{03} = Q_{11} = Q_{23} = Q_{31}; \quad Q_3 = Q_{02} = Q_{13} = Q_{33}.$$

Така відповідність означає, що одному переходу автомата Мілі з Q_0 в Q_1 повинні бути відповідними всі переходи автомата Мура зі станів Q_{00} , Q_{21} , Q_{12} , Q_{22} , Q_{32} в стан Q_{01} ; переходу з Q_1 в Q_2 повинні бути відповідні всі переходи автомата Мура зі стану Q_{01} в стани Q_{03} , Q_{11} , Q_{23} , Q_{31} , і т.д. Зі сказаного витікає наступний висновок: якщо стан Q_{mp} входить до множини станів, що співпадає зі станом, наприклад, Q_1 , то стовбець таблиці переходів для стану Q_{mp} співпадатиме з стовбцем таблиці переходів для стану Q_1 . Значення m функції виходів для еквівалентного автомата Мура:

$$\lambda(Q_{mp}) = \lambda(Q_m, X_p) \text{ при } Q_{mp} \neq Q_{00}.$$

Наприклад, для стану Q_{01} функція виходу буде Y_3 , оскільки вона відповідає набору Q_0 і X_1 ; для стану Q_{02} функція виходу буде Y_2 , що відповідає набору Q_0 і X_2 , і т. д.

Для початкового стану Q_{00} значення вихідного сигналу вибирається довільно.

Внаслідок описаних відповідностей і перетворень можна побудувати таблицю переходів і виходів еквівалентного автомата Мура (див. табл. 3.10).

Таблиця 3.10. Таблиця переходів і виходів еквівалентного автомата Мура

	Y_3	Y_3	Y_2	Y_1	Y_2	Y_1	Y_2	Y_1	Y_3	Y_2	Y_1	Y_3	Y_3
	Q_{00}	Q_{01}	Q_{02}	Q_{03}	Q_{11}	Q_{12}	Q_{13}	Q_{21}	Q_{22}	Q_{23}	Q_{31}	Q_{32}	Q_{33}
X_1	Q_{01}	Q_{11}	Q_{31}	Q_{21}	Q_{21}	Q_{01}	Q_{31}	Q_{01}	Q_{01}	Q_{21}	Q_{21}	Q_{01}	Q_{31}
X_2	Q_{02}	Q_{12}	Q_{02}	Q_{22}	Q_{22}	Q_{02}	Q_{32}	Q_{02}	Q_{02}	Q_{22}	Q_{22}	Q_{12}	Q_{32}
X_3	Q_{03}	Q_{13}	Q_{33}	Q_{23}	Q_{23}	Q_{03}	Q_{33}	Q_{33}	Q_{33}	Q_{23}	Q_{23}	Q_{33}	Q_{33}

Розглянемо особливість переходу від автомата Мура до еквівалентного автомата Мілі.

Якщо f_{p1} і λ_k функції переходів і виходів автомата Мура, то функції переходів $f_{p2}(Q, X)$ і виходів $\lambda_2(Q, X)$ еквівалентного автомата Мілі:

$$f_{p2}(Q, X) = f_{p1}(Q, X); \quad \lambda_2(Q, X) = \lambda_1(f_{p1}(Q, X)).$$

Звідси витікає, що таблиця переходів еквівалентного автомата Мілі співпадає з таблицею переходів автомата Мура, а в кожному клітку таблиці виходів записується символ, яким відмічений стан автомата Мура в заданій клітці.

При такому перетворенні граф автомата Мілі відрізняється від графу автомата Мура лише тим, що вихідні сигнали з вузлів графа перенесені на всі гілки, що входять у даний вузол.

Контрольні запитання

1. У чому полягає принципова різниця між автоматами Мура та Мілі?
2. Після мінімізації внутрішніх станів автомата їх кількість зменшилась до 12. Яка кількість тригерів достатня для їх реалізації?
3. Автомат Мілі має m станів і p вхідних сигналів. Скільки станів матиме еквівалентний йому автомат Мура?

Перелік посилань

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.
2. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.К. Прикладна теорія цифрових автоматів. – К. НАУ, 2007. – 364 с.

Лекція №4. Основи аналізу цифрових автоматів

Аналіз роботи автомата виконується з метою визначення його стану в послідуєчий тактовий момент часу і передбачення послідуєчих станів [1 – 4]. Такі задачі з'являються при вивченні роботи невідомих схем, при налагодженні пристроїв цифрової схемотехніки. Задачі аналізу є досить складними і розв'язуються поетапно.

Рекомендуються наступні кроки:

- визначаються стани на послідуєчому тактовому моменті часу і значення виходів вхідної та вихідної комбінаційних функцій L_1 і L_2 ;
- використовуються функції L_1 і L_2 для побудови таблиці станів, яка повністю визначає послідуєчий стан і значення виходу схеми для кожної комбінації поточного стану і виходів;
- будується діаграма станів, яка містить інформацію з попереднього кроку (граф переходів).

Аналіз автоматів з D-тригерами

Розглянемо особливості аналізу на конкретних прикладах скінченних автоматів.

Виконаємо аналіз схеми, що приведена на рис. 4.1 і містить вхідну комбінаційну функцію L_1 , що реалізована на елементах DD1, DD2, вихідну комбінаційну функцію L_2 , реалізовану на DD5, і модуль пам'яті з використанням динамічних D-тригерів DD3 і DD4. Тобто схема, що розглядається, відноситься до групи автоматів Мілі.

Розглянемо поведінку схеми в роботі.

За фронтом синхросигналу C_n кожен з тригерів приймає сигнал, що попередньо був поданий на D-вхід, і передає його на свій вихід в відповідності до алгоритму. Сигнали q_1 і q_2 є встановлюючими (збуджуючими) для D-тригерів у кожний тактовий момент часу.

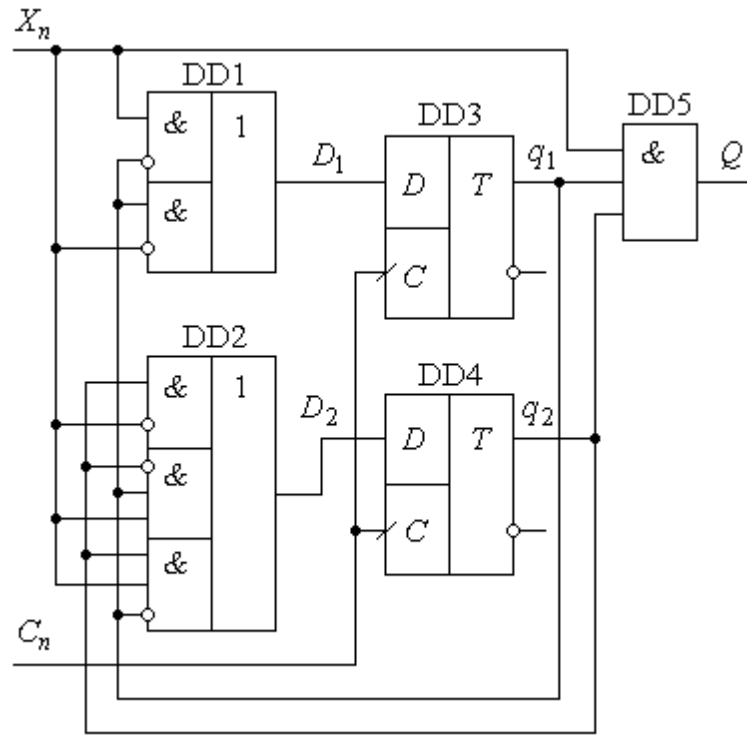


Рис. 4.1. Схема автомата на D -тригерах

Логічні рівняння, що визначають встановлюючі сигнали як функції поточного стану і вхідних сигналів, називаються *збуджуючими рівняннями* (*Excitation equations*) і можуть бути отримані з принципової схеми:

$$D_{1n} = X_n \overline{q_{1n}} + \overline{X_n} q_{1n} = X_n \oplus q_{1n}; \quad (4.1)$$

$$D_{2n} = \overline{X_n} q_{2n} + X_n q_{1n} \overline{q_{2n}} + X_n \overline{q_{1n}} q_{2n} = \overline{X_n} q_{2n} + X_n (q_{1n} \oplus q_{2n}). \quad (4.2)$$

Функція L_1 , що описується збуджуючими рівняннями, називається *функцією збудження*.

Використовуючи характеристичні рівняння D -тригерів, можемо записати:

$$q_{1(n+1)} = D_{1n} \cdot C_n; \quad (4.3)$$

$$q_{2(n+1)} = D_{2n} \cdot C_n. \quad (4.4)$$

Підставляючи значення D_{1n} , D_{2n} з (9.1) – (9.2), отримуємо:

$$q_{1(n+1)} = C_n \cdot (X_n \oplus q_{1n}); \quad (4.5)$$

$$q_{2(n+1)} = C_n \cdot (\overline{X_n} q_{2n} + X_n (q_{1n} \oplus q_{2n})). \quad (4.6)$$

Отримані рівняння (4.5) – (4.6), що виражають значення стану автомата в $(n + 1)$ -й момент часу в залежності від значень його входів і стану в n -й момент часу, називаються *перехідними рівняннями*. Їх суть полягає в тому, що вони надають можливість передбачити стан автомата в $(n + 1)$ -й момент часу на основі інформації про стан і значення входів в n -й момент часу.

Оскільки внутрішній стан автомата визначається значеннями виходів двох тригерів, то можна стверджувати, що існують лише чотири можливі стани автомата: $(q_1, q_2) = 00, 01, 10, 11$. Для кожного стану в приладі, що розглядається, можливі лише два значення вхідного сигналу X : 0, 1.

В результаті маємо лише вісім комбінацій стан-вхід (у загальному плані автомат з m станами і p вхідними сигналами матиме 2^{m+p} комбінацій стан-вхід).

Перехідні рівняння (4.5) – (4.6) можна задати в табличній формі (див. табл. 4.1).

Таблиця 4.1. Таблиця представлення перехідних рівнянь

C_n	X_n	q_{2n}	q_{1n}	$q_{2(n+1)}$	$q_{1(n+1)}$
1	0	0	0	0	0
2	0	0	1	0	1
3	0	1	0	1	0
4	0	1	1	1	1
1	1	0	0	0	1
2	1	0	1	1	0
3	1	1	0	1	1
4	1	1	1	0	0

З табл. 4.1 наглядно витікає особливість роботи автомата. Вона полягає в тому, що при $X_n = 0$ автомат не працює, тобто значення вихідних станів в $(n + 1)$ -й момент часу повторюють відповідні значення в n -й момент часу. При $X_n = 1$ він виконує функцію двійкового лічильника з максимальним фіксованим значенням $q_2 q_1 = 11_2 = 3_{10}$.

Побудуємо тепер на основі табл. 4.1 таблицю переходів автомата. Для цього кожен з можливих станів автомата позначимо відповідним значенням з множини Q , починаючи з нульового:

$$Q_0 = \overline{q_{2n}} \overline{q_{1n}} = 00; \quad Q_1 = \overline{q_{2n}} q_{1n} = 01; \quad Q_2 = q_{2n} \overline{q_{1n}} = 10; \quad Q_3 = q_{2n} q_{1n} = 11.$$

В результаті отримуємо таблицю переходів у вигляді табл. 4.2 або її скорочену форму – табл. 4.3.

Таблиця 4.2. Таблиця переходів

C_n	X_n	Q_n	Q_{n+1}
1	0	Q_0	Q_0
2	0	Q_1	Q_1
3	0	Q_2	Q_2
4	0	Q_3	Q_3
1	1	Q_0	Q_1
2	1	Q_1	Q_2
3	1	Q_2	Q_3
4	1	Q_3	Q_0

Таблиця 4.3. Скорочена таблиця переходів

$X \cdot C_n \backslash Q$	Q_0	Q_1	Q_2	Q_3
0	Q_0	Q_1	Q_2	Q_3
1	Q_1	Q_2	Q_2	Q_0

Граф-схема автомата приведена на рис. 4.2.

Послідовність переходів з одного стану до іншого забезпечується при наявності динамічного сигналу C_n від зовнішнього тактового генератора і потенційного (дозволяючого) сигналу x .

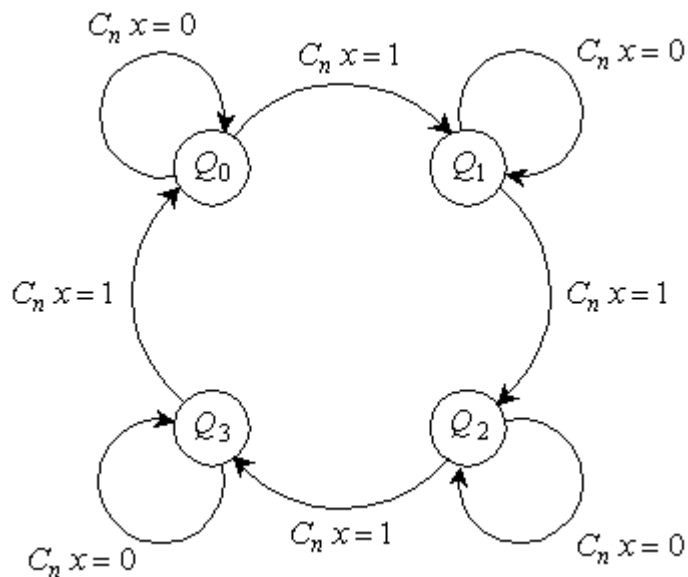


Рис. 4.2. Граф-схема автомата

Робота автомата може бути описана і за допомогою часових діаграм (рис. 4.3), які відображають зміну станів тригерів автомата (q_{1n} , q_{2n}) у відповідні тактові моменти часу, і відповідні стани автомата ($Q_0 \div Q_3$) на кожному тактовому інтервалі.

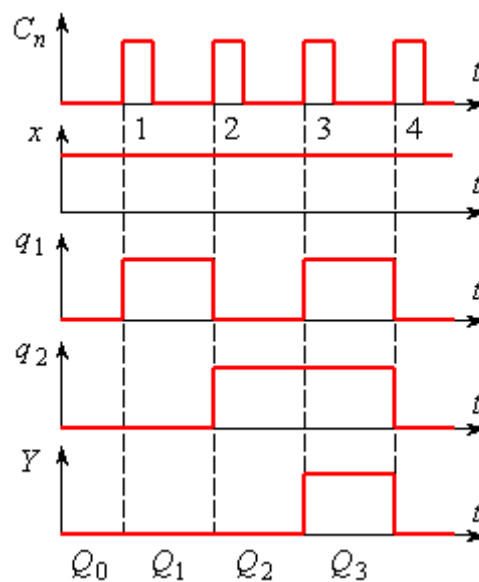


Рис. 4.3. Часові діаграми, що описують роботу схеми

Розглянуту схему автомата можливо спростити за рахунок видалення інверторів, оскільки інверсні значення виходів тригерів можна отримати безпосередньо з тригерів.

Вихідний сигнал автомата визначається внутрішнім його станом і значенням вхідного сигналу, тому функція L_2 описується рівнянням:

$$Y = x \cdot q_{1n} \cdot q_{2n}.$$

Приклад 4.1. Проаналізувати роботу скінченного автомата, принципова схема якого приведена на рис. 4.3. Побудувати його таблицю станів та часові діаграми роботи.

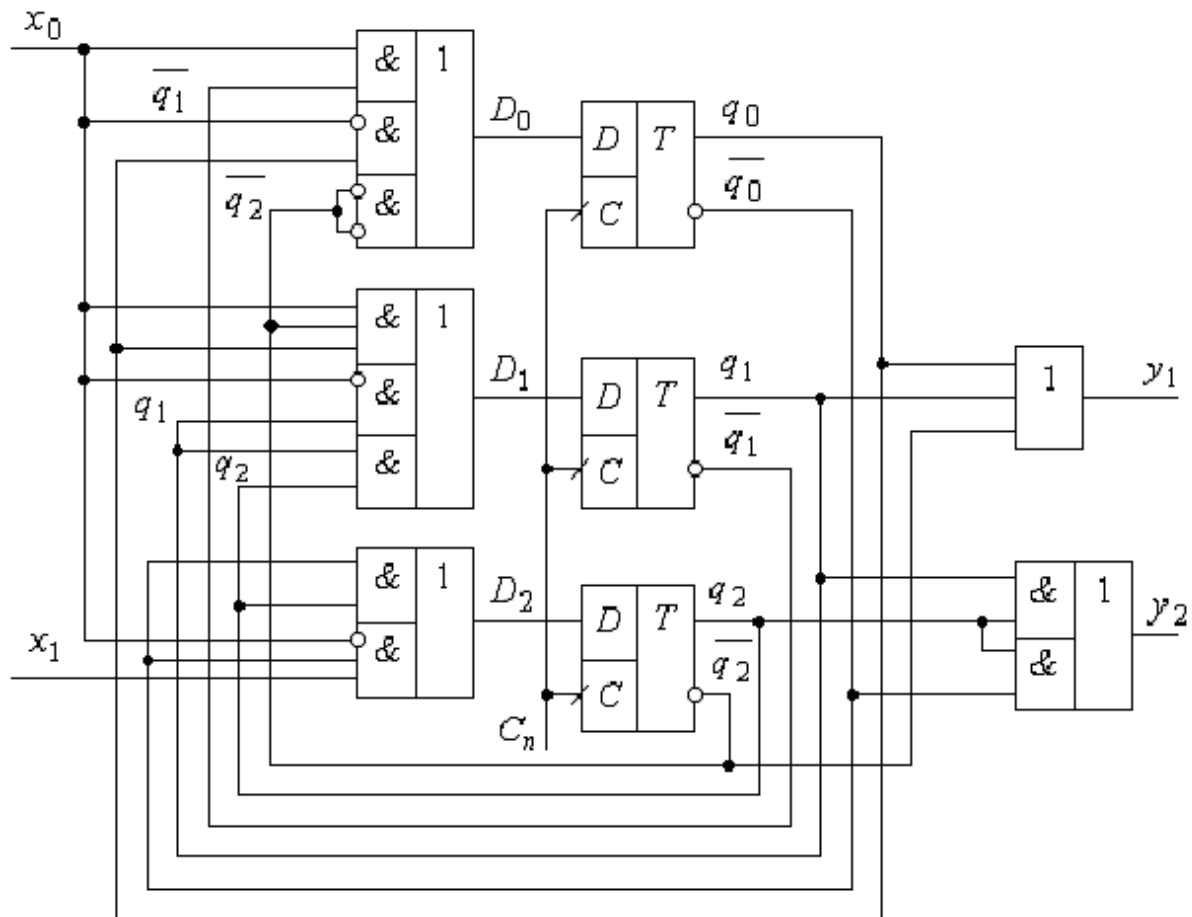


Рис. 4.3. Принципова схема скінченного автомата

Розв'язання. Із зовнішнього вигляду принципової схеми бачимо, що аналізований автомат відноситься до класу машин Мура, а рівняння для вихідних сигналів y_1 та y_2 мають вигляд:

$$y_1 = \overline{q_2} + q_1 + q_0; \quad (4.7)$$

$$y_2 = q_2 \cdot (q_1 + \overline{q_0}). \quad (4.8)$$

Аналізуючи роботу логічної схеми, можемо записати рівняння збудження тригерів:

$$D_{0n} = x_{0n} \overline{q_{1n}} + \overline{x_{0n}} q_{0n} + q_{2n} ;$$

$$D_{1n} = x_{0n} \overline{q_{2n}} q_{0n} + \overline{x_{0n}} q_{1n} + q_{2n} q_{1n} ;$$

$$D_{2n} = q_{2n} \overline{q_{0n}} + x_{1n} \overline{x_{0n}} \overline{q_{0n}} .$$

Підставляючи рівняння збудження в характеристичне рівняння D -тригерів, отримаємо:

$$q_{0(n+1)} = C_n D_{0n} = C_n (x_{0n} \overline{q_{1n}} + \overline{x_{0n}} q_{0n} + q_{2n}) ;$$

$$q_{1(n+1)} = C_n D_{1n} = C_n (x_{0n} \overline{q_{2n}} q_{0n} + \overline{x_{0n}} q_{1n} + q_{2n} q_{1n}) ;$$

$$q_{2(n+1)} = C_n D_{2n} = C_n (q_{2n} \overline{q_{0n}} + x_{1n} \overline{x_{0n}} \overline{q_{0n}})$$

Функції $q_{0(n+1)}$, $q_{1(n+1)}$, $q_{2(n+1)}$ зобразимо в табличній формі (див. табл. 4.4).

Таблиця 4.4. Зображення функцій у табличній формі

C_n	x_0	x_1	q_{2n}	q_{1n}	q_{0n}	$q_{2(n+1)}$	$q_{1(n+1)}$	$q_{0(n+1)}$
1	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	1
3	0	0	0	1	0	0	1	0
4	0	0	0	1	1	0	1	1
5	0	0	1	0	0	1	0	1
6	0	0	1	0	1	0	0	1
7	0	0	1	1	0	1	1	1
8	0	0	1	1	1	0	1	1
9	0	1	0	0	0	1	0	0
10	0	1	0	0	1	0	0	1
11	0	1	0	1	0	1	1	0
12	0	1	0	1	1	0	1	1
13	0	1	1	0	0	1	0	1
14	0	1	1	0	1	0	0	1
15	0	1	1	1	0	1	1	1
16	0	1	1	1	1	0	1	1
17	1	0	0	0	0	0	0	1
18	1	0	0	0	1	0	1	1
19	1	0	0	1	0	0	0	0
20	1	0	0	1	1	0	1	0
21	1	0	1	0	0	1	0	1
22	1	0	1	0	1	0	0	1
23	1	0	1	1	0	1	1	1
24	1	0	1	1	1	0	1	1
25	1	1	0	0	0	0	0	1
26	1	1	0	0	1	0	1	1
27	1	1	0	1	0	0	0	0
28	1	1	0	1	1	0	1	0
29	1	1	1	0	0	1	0	1
30	1	1	1	0	1	0	0	1
31	1	1	1	1	0	1	1	1
32	1	1	1	1	1	0	1	1

Введемо умовні позначення станів у відповідності до табл. 4.5.

Таблиця 4.5. Умовне позначення станів

№ п/п	q_{2n}	q_{1n}	q_{0n}	Q_m
0	0	0	0	Q_0
1	0	0	1	Q_1
2	0	1	0	Q_2
3	0	1	1	Q_3
4	1	0	0	Q_4
5	1	0	1	Q_5
6	1	1	0	Q_6
7	1	1	1	Q_7

Кодування станів дає можливість побудувати таблицю переходів автомата на основі табл. 4.4 (див. табл. 4.6).

Таблиця 4.6. Таблиця переходів

Q_n	Q_{n+1}				$y_1 y_2$
	$C_n \ x_0 \ x_1$				
	00	01	10	11	
Q_0	Q_0	Q_4	Q_1	Q_1	10
Q_1	Q_1	Q_1	Q_3	Q_3	10
Q_2	Q_2	Q_6	Q_0	Q_0	10
Q_3	Q_3	Q_3	Q_2	Q_2	00
Q_4	Q_5	Q_5	Q_5	Q_5	11
Q_5	Q_1	Q_1	Q_1	Q_1	10
Q_6	Q_7	Q_7	Q_7	Q_7	11
Q_7	Q_3	Q_3	Q_3	Q_3	11

На основі таблиці переходів будується граф-схема автомата (рис. 4.4) і аналізуються умови переходів.

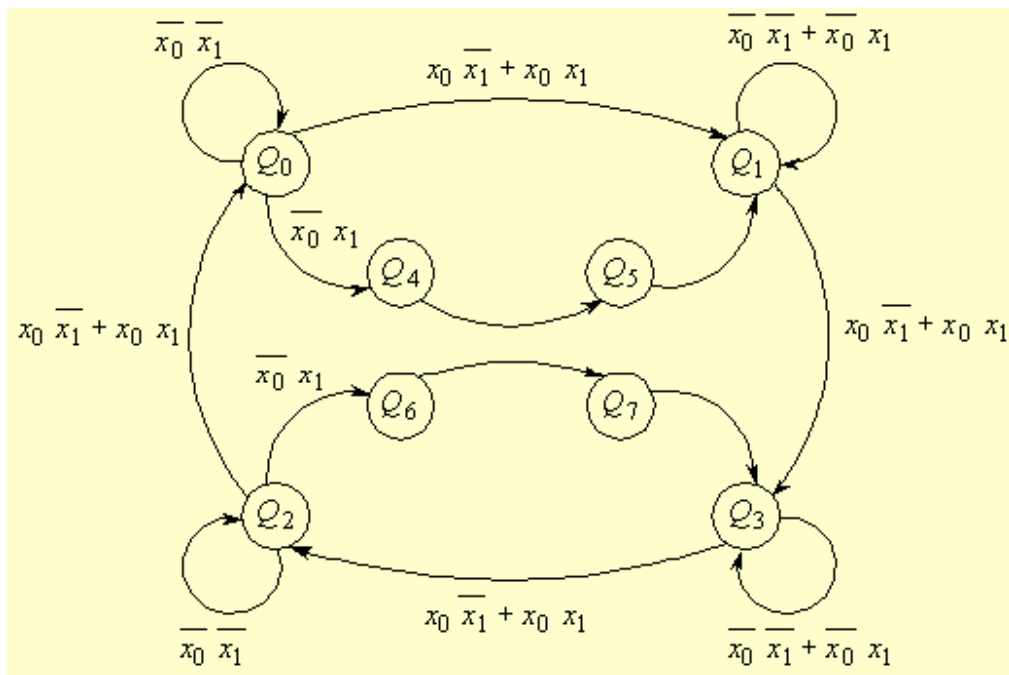


Рис. 4.4. Граф-схема автомата

Аналіз роботи автомата на основі граф-схеми з урахуванням керуючих сигналів дає можливість виявити деякі аспекти його роботи. Наприклад, перехід зі стану Q_4 в стан Q_5 забезпечується при всіх значеннях $x_0 x_1$ і є безумовним. Аналогічно, маємо переходи з Q_5 в Q_1 ; з Q_6 в Q_7 ; з Q_7 в Q_3 .

Звідси витікає, що перехід з Q_0 в Q_4 можна замінити на перехід з Q_0 в Q_1 , а перехід з Q_2 в Q_6 – на перехід з Q_2 в Q_3 . Таблиця переходів при цьому може бути спрощена (див. табл. 4.7), а вигляд граф-схеми відповідатиме рис. 4.5.

Таблиця 4.7. Таблиця переходів

Q_n	Q_{n+1}				$y_1 y_2$
	$C_n \ x_0 \ x_1$				
	00	01	10	11	
Q_0	Q_0	Q_1	Q_1	Q_1	10
Q_1	Q_1	Q_1	Q_3	Q_3	10
Q_2	Q_2	Q_3	Q_0	Q_0	10
Q_3	Q_3	Q_3	Q_2	Q_2	00

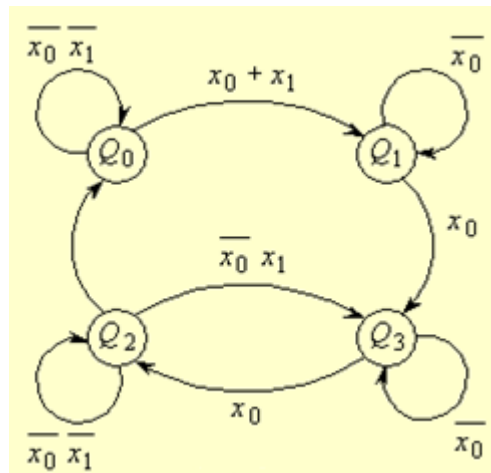


Рис. 4.5. Граф-схема автомата

Приведений аналіз схеми автомата показує, що при $x_1 = 0$ матимемо лічильник, який працює у відповідності до коду Грея. При $x_1 = 1$ виникає можливість забезпечувати своєрідний тригерний режим між станами Q_2 і Q_3 .

Отримані результати дають можливість побудувати часові діаграми роботи автомата. При цьому слід враховувати, що при станах $Q_4 \div Q_7$ має місце $y_1 = 1$, а y_2 переходить в нуль у стані Q_5 , який є нестійким. Тому можна вважати, що $y_2 = 1$.

Часові діаграми роботи автомата приведені на рис. 4.6. Часові діаграми виходів y_1 та y_2 не приведені, оскільки вони зрозумілі з формул (4.7), (4.8).

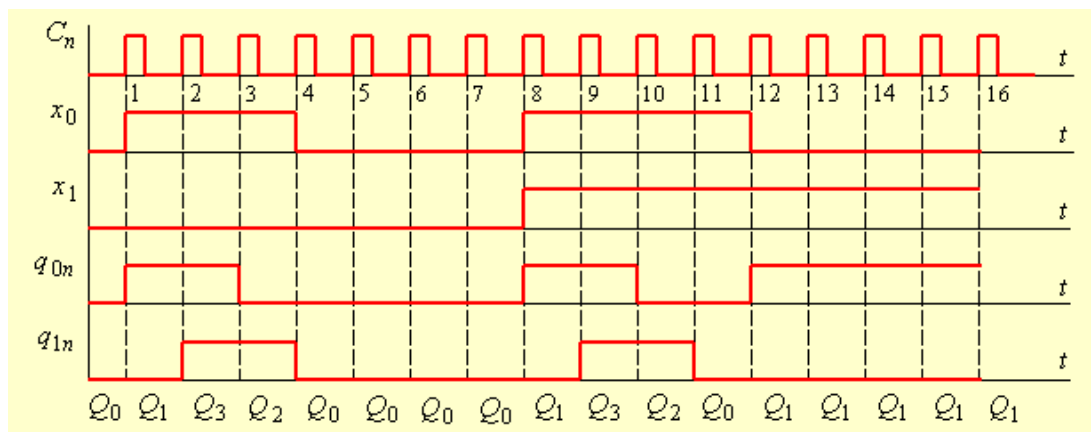


Рис. 4.6. Часові діаграми роботи автомата

Особливості аналізу скінченних автоматів з JK-тригерами

Синхронізовані скінченні автомати, що будуються на базі JK-тригерів, можуть також синхронізуватись у відповідності до процедури, яка була описана

вище. Різниця полягає лише в тому, що необхідно мати два збуджуючих рівняння (*Excitation equations*) для кожного тригера – одне для J -входу, друге – для входу K .

Для отримання рівнянь переходів (*Transition equations*) кожне з них повинно використовувати характеристичне рівняння JK -тригера:

$$Q_{n+1} = (J_n \overline{Q_n} + \overline{K_n} Q_n) C_n.$$

На рис. 4.7 приводиться приклад скінченного автомата з використанням JK -тригерів.

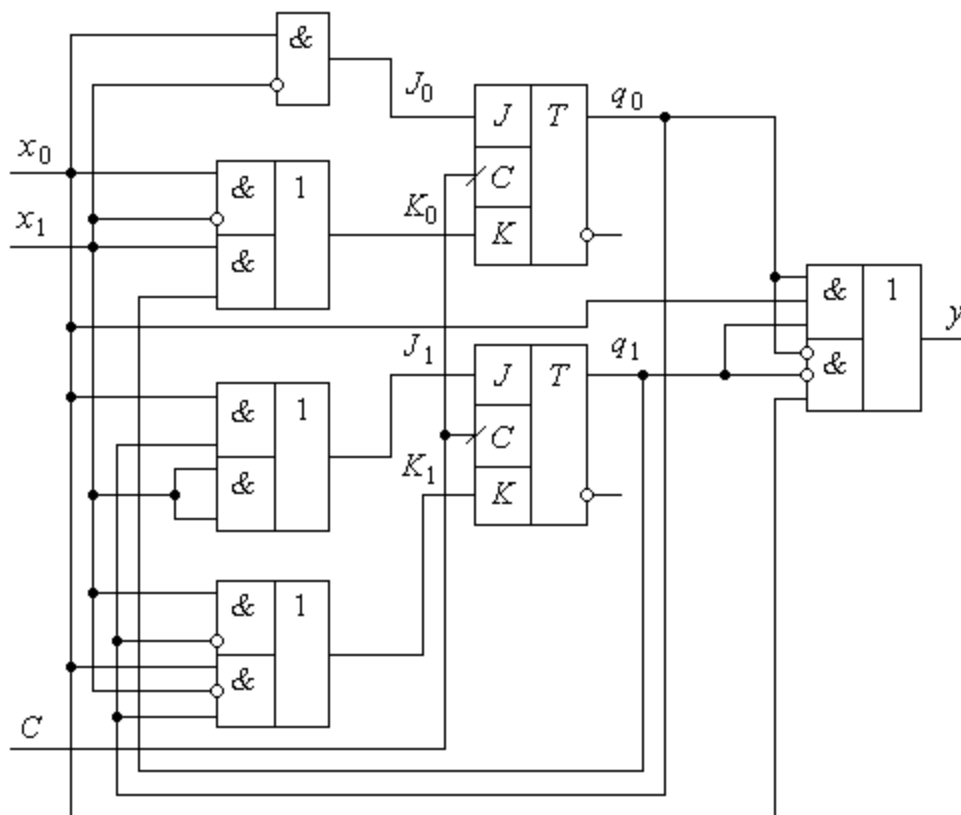


Рис. 4.7. Приклад скінченного автомата з використанням JK -тригерів

З рис. 4.7 можемо записати наступні рівняння збудження для входів JK -тригерів:

$$J_{0n} = x_0 \overline{x_1} ; \quad K_{0n} = x_0 \overline{x_1} + x_1 q_{1n} ;$$

$$J_{1n} = x_0 q_{0n} + x_1 ; \quad K_{1n} = x_1 \overline{q_{0n}} + x_0 \overline{x_1} q_{0n} .$$

Підставляючи в характеристичне рівняння тригера, отримуємо рівняння переходів:

$$q_{0(n+1)} = C_n (\overline{x_0} \overline{x_1} \overline{q_{0n}} + (\overline{x_0} \overline{x_1} + x_1 \overline{q_{1n}}) \overline{q_{0n}}); \quad (4.9)$$

$$q_{1(n+1)} = C_n ((x_0 \overline{q_{0n}} + x_1) \overline{q_{1n}} + (\overline{x_1} \overline{q_{0n}} + x_0 \overline{x_1} \overline{q_{0n}}) \overline{q_{1n}}). \quad (4.10)$$

Таблиця переходів, що базується на цих рівняннях, легко може бути побудована.

Рівняння (4.9) і (4.10) приведемо до досконалої форми:

$$q_{0(n+1)} = C_n (\overline{x_{0n}} \overline{x_{1n}} \overline{q_{0n}} \overline{q_{1n}} + \overline{x_{0n}} \overline{x_{1n}} \overline{q_{0n}} \overline{q_{1n}} + \overline{x_{0n}} \overline{x_{1n}} \overline{q_{0n}} \overline{q_{1n}} + \overline{x_{0n}} \overline{x_{1n}} \overline{q_{0n}} \overline{q_{1n}} + \overline{x_{0n}} \overline{x_{1n}} \overline{q_{0n}} \overline{q_{1n}} + \overline{x_{0n}} \overline{x_{1n}} \overline{q_{0n}} \overline{q_{1n}}) \\ q_{1(n+1)} = C_n (\overline{x_0} \overline{x_1} \overline{q_{0n}} \overline{q_{1n}} + \overline{x_0} \overline{x_1} \overline{q_{0n}} \overline{q_{1n}} + \overline{x_0} \overline{x_1} \overline{q_{0n}} \overline{q_{1n}} + \overline{x_0} \overline{x_1} \overline{q_{0n}} \overline{q_{1n}} + \\ + \overline{x_0} \overline{x_1} \overline{q_{0n}} \overline{q_{1n}} + \overline{x_0} \overline{x_1} \overline{q_{0n}} \overline{q_{1n}} + \overline{x_0} \overline{x_1} \overline{q_{0n}} \overline{q_{1n}} + \overline{x_0} \overline{x_1} \overline{q_{0n}} \overline{q_{1n}}) .$$

Таблиця станів для обох функцій $q_{0(n+1)}$ і $q_{1(n+1)}$ має вигляд табл. 4.8, а карти Карно приведені на рис. 4.8.

Таблиця 4.8. Таблиця станів для функцій $q_{0(n+1)}$ і $q_{1(n+1)}$

N	x_0	x_1	q_{0n}	q_{1n}	$q_{0(n+1)}$	$q_{1(n+1)}$
0	0	0	0	0	0	0
1	0	0	0	1	0	0
2	0	0	1	0	1	0
3	0	0	1	1	0	0
4	0	1	0	0	0	1
5	0	1	0	1	0	1
6	0	1	1	0	0	1
7	0	1	1	1	1	0
8	1	0	0	0	0	0
9	1	0	0	1	1	0
10	1	0	1	0	0	1
11	1	0	1	1	1	1
12	1	1	0	0	0	1
13	1	1	0	1	1	1
14	1	1	1	0	1	1
15	1	1	1	1	0	0

		$Q_{0(n+1)}$				$Q_{1(n+1)}$			
x_0	x_1	$q_{0n} q_{1n}$				$q_{0n} q_{1n}$			
		00	01	11	10	00	01	11	10
00	0	0	1	0	2	0	1	0	2
01	4	0	5	7	6	1	1	0	1
11	12	0	13	15	14	1	1	0	1
10	8	0	9	11	10	0	0	1	1

Рис. 4.8. Карти Карно

Складаємо таблицю кодування станів (табл. 4.9). Складаємо таблицю переходів (табл. 4.10).

Таблиця 4.9. Таблиця кодування станів

N	q_{0n}	q_{1n}	Q
0	0	0	Q_0
1	0	1	Q_1
2	1	0	Q_2
3	1	1	Q_3

Таблиця 4.10. Таблиця переходів

Q_n	Q_{n+1}			
	$x_0 x_1$			
	00	01	10	11
Q_0	Q_0	Q_1	Q_0	Q_1
Q_1	Q_0	Q_1	Q_2	Q_3
Q_2	Q_2	Q_1	Q_1	Q_3
Q_3	Q_0	Q_2	Q_3	Q_0

Будуємо граф-схему переходів (рис. 4.9).

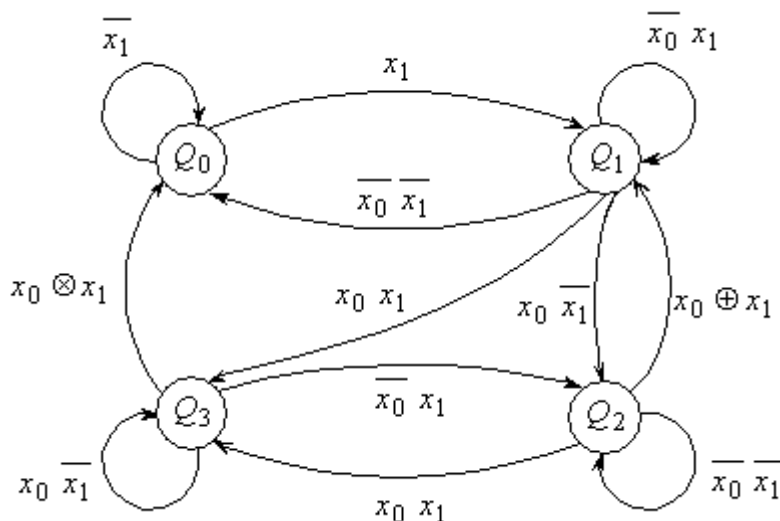


Рис. 4.9. Граф-схема переходів

Рівняння, що описує залежність вихідного сигналу, має вигляд:

$$y = x_0 q_{0n} q_{1n} + x_1 \overline{q_{0n}} \overline{q_{1n}} \quad .$$

Слід відмітити, що при використанні *JK*-тригерів можливо використати і інший шлях побудови таблиці переходів. Він полягає в тому, що створюються рівняння збудження і таблиця збудження відносно входів тригерів J_0, K_0, J_1, K_1 , які потім використовуються в характеристичних рівняннях кожного тригера при побудові таблиці переходів.

Звернемося знову до прикладу схеми двійкового лічильника, що приведена на рис. 4.1, і розглянемо інший варіант його побудови з використання *JK*-тригерів. Для цього побудуємо таблицю станів з урахуванням сигналів на входах *J*- і *K*- кожного тригера і визначимося з логічною функцією L_1 , що задаватиме сигнали *J* і *K*.

Оскільки нам необхідно мати два тригери, то позначимо їх, як і в схемі-прототипі, – через DD3 і DD4. Відповідні входи тригерів позначимо як J_3, K_3 і J_4, K_4 , а виходи – q_3 і q_4 .

Таблиця будується в такій послідовності. В стовбці C_n задаємо послідовність тактових сигналів. У стовбцях q_3 і q_4 задаємо початкові

значення виходів тригерів до початку дії синхросигналу і відповідних інформаційних сигналів. Формально в цих стовпцях записується таблиця двійкового коду для двох змінних. У стовбцях $q_{3(n+1)}$ і $q_{4(n+1)}$ записуються значення виходів q_3 і q_4 після дії синхросигналу в відповідності до табл. 3.4 (с. 1). В стовбцях J_3 , K_3 , J_4 , K_4 записуються ті значення входів, які приведуть до відповідної зміни кодових значень в стовбцях q_3 і q_4 . Наприклад: у стовбці q_{3n} до початку дії синхросигналу маємо “0”, а після його дії – “1”. До такої зміни станів тригерів можуть привести сигнали $J_4 = 0$, $J_3 = 1$, $K_3 = 0$. Значення входу K_4 – байдуже.

Внаслідок побудови таблиці отримуємо залежності між значеннями виходів q_3 і q_4 від J_i , K_i (див. табл. 4.11).

Таблиця 4.11. Таблиця збудження тригерів

C_n	q_{4n}	q_{3n}	$q_{4(n+1)}$	$q_{3(n+1)}$	J_4	K_4	J_3	K_3
1	0	0	0	1	0	x	1	x
2	0	1	1	0	1	x	x	1
3	1	0	1	1	x	0	1	x
4	1	1	0	0	x	1	x	1

Для першого тригера бачимо, що значення J_3 і K_3 мають значення входів, або байдуже, або рівня “1”. Звідси знаходимо, що на його входах постійно може бути присутньою “1”, а, відповідно до алгоритму його роботи, це режим T -тригера.

Для визначення значень входів J_4 і K_4 також бачимо, що вони можуть бути однаковими, тобто $J_4 = K_4$, а подальший аналіз зв'язку J_4 з q_3 приводить до того, що знаходимо $J_4 = K_4 = q_{3n}$.

Звідси витікає схема лічильника на JK -тригерах (рис. 4.10).

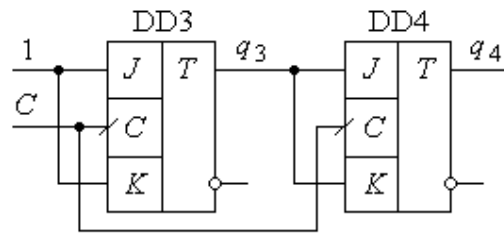


Рис. 4.10. Схема лічильника на JK -тригерах

Проаналізуємо його роботу. При $q_3 = q_4 = 0$ поява першого синхроімпульсу приведе до $q_3 = 1$; $q_4 = 0$. Другий синхроімпульс приведе до зміни стану DD4, тобто $q_4 = 1$, а також до зміни стану DD3: q_3 зміниться в нуль. Третій синхроімпульс знову приведе до зміни стану q_3 з “0” в “1” і, нарешті, четвертий імпульс приведе до обнуління обох тригерів.

Отриманий результат говорить про те, що складність комбінаційної схемотехніки скінченного автомата суттєво залежить від типу використовуваних тригерів. Якщо ж говорити більш точно, то складність комбінаційних схем залежить від уміння розробника використати логічні властивості використовуваних тригерів.

Контрольні запитання

1. Яка кількість вхідних слів є дозволеною для RS -тригера, а яка – заборонена?
2. Поясніть принципову різницю між аналізом та синтезом цифрового автомата.

Перелік посилань

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.
2. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.К. Прикладна теорія цифрових автоматів. – К. НАУ, 2007. – 364 с.

Лекція №5. Синтез скінченних автоматів

Задача синтезу (проектування) скінченних автоматів майже діаметрально протилежна задачі аналізу [1 – 4].

Приблизно такі кроки можна описати як етапи проектування:

1. Задається закон функціонування автомата;
2. Мінімізується кількість внутрішніх станів автомата;
3. Кодуються стани автомата;
4. Визначаються функції збудження елементів пам'яті і функції виходів, а також забезпечується їх мінімізація;
5. Складається функціональна схема автомата в вибраному елементному базисі.

Перші три етапи відносяться до *абстрактного синтезу* автомата, при якому по значеннях вхідних і вихідних сигналів визначаються і кодуються його стани.

При абстрактному синтезі на основі словесного, табличного, за допомогою часових діаграм і інших засобів опису роботи автомату виявляється закон його функціонування і задаються функції переходів і виходів. На цьому етапі забезпечується мінімізація кількості станів і визначається кількість елементів пам'яті, необхідних для побудови автомата.

Решта станів забезпечують процедуру одержання структурних схем автоматів, що задаються кодованими таблицями входів і виходів. Така процедура називається *структурним синтезом*.

Задача структурного синтезу полягає у виборі типів елементарних автоматів (тригерів) і знаходження такого способу їх з'єднання між собою, який забезпечить функціонування автомата у відповідності до заданих кодованими таблицями переходів і виходів.

Основною задачею цього етапу є синтез комбінаційних схем з урахуванням вибраних тригерів, знаходження мінімальних форм функцій збудження і виходів. Етап закінчується побудовою функціональної схеми автомата.

Розглянемо змістовну частину кожного з етапів на конкретних прикладах.

Етап 1. *Задання закону функціонування асинхронного автомата.*

Початковою інформацією для першого етапу може служити словесний опис алгоритму функціонування автомата або аналіз часових діаграм його роботи. Результатом цього етапу повинно бути отримання формалізованого опису закону функціонування в вигляді таблиць переходів, графів переходів і т. п. Практичний досвід показує, що зручно задавати автомат за допомогою *початкових таблиць переходів* (ПТП), оскільки їхня структура є найбільш простою.

ПТП характеризується тим, що в кожному її рядку є тільки один стійкий стан, а всі переходи між станами прості.

На основі аналізу зміни вхідних і вихідних сигналів можна вводити стани автомата, які характеризуються станом входу (сигналами x_0 і x_1 з вхідного алфавіту X) і станом виходу Y (сигналом y).

Різноразмірними комбінаціями вхідних та вихідних сигналів припишемо відповідні номери, які характеризують всі можливі стани автомата. За початковий приймемо стан $Q_{00} = 0$, який характеризується значеннями вхідних сигналів $x_1 x_0 = 00$ і значенням вихідного сигналу $y = 0$. Стан $Q_{00} = 0$ є *стійким*, тобто таким, який не зміниться без дії вхідних сигналів. В таблиці ПТП, що будується, стійкі стани будемо виділяти дужками. *Нестійкі* стани автомата – це такі, які можуть змінюватися на стійкі без зміни вхідних сигналів. Такі стани в ПТП будемо позначати їх номерами без дужок.

Якщо на автомат, що знаходиться в стійкому стані Q_{00} , діятиме вхідний сигнал $x_1 x_0 = 01$, то він перейде в новий, але нестійкий, стан, який потім перейде в стійкий $Q_1 = (1)$. Нестійкий стан відповідає ситуації, коли в результаті зміни вхідного сигналу X змінились сигнали збудження тригерів, але ще не змінився вихідний сигнал, тобто y залишився рівним нулю. По

закінченні перехідних процесів автомат перейде в стійкий стан $Q_1 = (1)$, який характеризується вхідними сигналами $x_1 x_0 = 01$ і вихідним $y = 1$.

Переходу зі стійкого стану (0) в нестійкий 1 в ПТП відповідає переміщення в межах першого рядка з клітинки з координатами $x_1 x_0 = 00$ в клітинку з координатами $x_1 x_0 = 01$. Вихідний сигнал при цьому залишається незмінним: $y = 0$.

Переходу з нестійкого стану 1 в стійкий стан (1) в ПТП відповідає переміщення в межах одного стовпця з координатами $x_1 x_0 = 01$ з першого рядка в другий. Вихідний сигнал при цьому змінюється і приймає значення $y = 1$ (табл. 5.1).

Таблиця 5.1. Початкова таблиця переходів

$Q \backslash X$	$x_1 x_0$				y
	00	01	11	10	
0	(0)	1			0
1		(1)			1

Звідси маємо, що перехід з одного стійкого стану (0) в інший стійкий стан (1) відповідає двом фрагментам переходу: спочатку у межах рядка, а потім у межах стовпця.

З часової діаграми, приведеної на рис. 10.1, видно, що стійкий стан (1) при $x_1 x_0 = 00$ змінюється на стан (0), а потім під дією сигналу $x_1 x_0 = 10$ переходить в стан (2), і т.д.

Номери станів на часовій діаграмі зображають так, щоб кожна змінювана ситуація в сигналах X , Y мала свій номер і щоб були враховані всі зовнішні однакові ситуації, які вимагають спеціального запам'ятовування. Наприклад, якщо автомат згенерує третій імпульс, то кожен з трьох імпульсів і кожен з пауз необхідно позначити своїм номером.

Аналіз всієї часової діаграми дозволяє заповнити ПТП (див. табл. 5.2), але не повністю – частина клітинок залишається вільними.

Таблиця 5.2. Неповністю заповнена початкова таблиця переходів

$Q \backslash X$	$x_1 x_0$				y
	00	01	11	10	
0	(0)	1	—	2	0
1	0	(1)	4	—	1
2	0	—	3	(2)	0
3	—	5	(3)	2	0
4	—	1	(4)	2	1
5	0	(5)	3	—	0

Наприклад, у першому рядку ПТП залишилась незаповненою клітка з $X = 11$. Оскільки стан $Q = 0$ є стійким при $X = 00$, то перехід в новий стан, що відповідає $X = 11$, вимагає одночасної зміни обох вхідних сигналів x_1 і x_0 , що практично неможливо. Оскільки перехід $00 \rightarrow 11$ виконати неможливо, в ПТП в відповідній клітинці проставляється риска, яка інформує про наявність невизначеного стану. Аналогічно виключаються інші переходи типу $00 \leftrightarrow 11$, а також $01 \leftrightarrow 10$.

Етап 2. Мінімізація кількості внутрішніх станів автомата.

Ідея формальної процедури мінімізації кількості станів полягає в визначенні еквівалентних станів, де два стани є *еквівалентними*, якщо неможливо визначитись з різницею в їх функціонуванні, розглядаючи лише поточні і майбутні стани виходів автомата. Пара еквівалентних станів може бути замінена одним.

Існують детальні процедури, які дозволяють формально розв'язувати задачу мінімізації кількості станів, але на практиці вони рідко використовуються. Реально це обумовлено тим, що досвідчені інженери з електроніки проектують цифрові автомати з мінімальною кількістю станів без будь-яких проблем, не використовуючи формальні процедури. Більш того, часто мають місце ситуації, коли збільшення кількості станів не ускладнює проект, а, навпаки, спрощує його, і мінімізуючи процедури не надають необхідної користі.

Як правило, ПТП містить ряд збиткових внутрішніх станів, усунення яких не порушить алгоритму роботи автомата, але, в той же час, дозволить

спростити його функціональну схему. В ряді випадків наявність збиткових станів не є очевидним фактом, тому для їх виявлення необхідно проводити детальний аналіз ПТП.

Скорочення частини станів асинхронного автомата базується на виявленні еквівалентних і псевдоеквівалентних станів, а також на виявленні і об'єднанні сумісних внутрішніх станів автомата (рядків ПТП).

Еквівалентними станами автомата називаються стійкі стани, які задовольняють наступним умовам:

1. Їм відповідає один і той же стан входу (тобто вони знаходяться в одному і тому ж стовбці ПТП);
2. Їм відповідає один і той же стан виходу (однакові вихідні сигнали автомата);
3. Будь-якій послідовності станів входу автомата відповідає одна і та ж сама послідовність станів його виходу, незалежно від того, який зі стійких станів автомата прийнято за початковий.

Для неповністю визначеного автомата, окрім поняття еквівалентності, існує і використовується поняття псевдоеквівалентності станів. *Псевдоеквівалентними* називаються такі два стійкі стани автомата, яким:

- відповідає один і той самий стан входу автомата;
- відповідають стани виходів автомата, між якими немає протиріччя;
- будь-якій послідовності станів входу автомата відповідають непротирічні послідовності станів його виходу, незалежно від того, який з цих стійких станів взятий за початковий. Іншими словами, серед пар послідовностей станів входу і виходу, що починаються з цих стійких станів, не повинно бути таких, що мають протиріччя.

Після об'єднання еквівалентних і псевдоеквівалентних станів завжди отримується таблиця, в кожному рядку якої, як і в ПТП, знаходиться лише один стійкий стан. Скорочення кількості рядків, порівняно з ПТП, відбувається за рахунок зменшення кількості стійких станів. Але після об'єднання стійких

станів можливо виконати подальше спрощення таблиці переходів за рахунок об'єднання сумісних внутрішніх станів автомата.

Під *сумісним внутрішнім станом* розуміють два або більше станів, яким відповідають рядки з розміщенням цифр в них, яке не має протиріччя, тобто такі рядки, в одному і тому ж стовбці яких знаходяться однакові цифри (або в одному рядку цифра, а в іншому – риска).

Рядки таблиці переходів, що відповідають сумісним станам, можуть бути об'єднані в одну. В об'єднаному рядку повинні бути стійкими стани у будь-якій з об'єднаних рядків. Тобто, при об'єднанні сумісних станів отримуємо таблицю, в кожному рядку якої може бути декілька стійких станів. При об'єднанні сумісних станів буде отриманий автомат, еквівалентний вихідному.

Об'єднання рядків з поєднаними стійкими станами виконується на основі наступних правил:

- два або більше рядків можуть бути об'єднані, якщо значення вихідних змінних (станів виходу) для цих рядків не мають протиріччя для автомата Мура і можуть бути будь-якими для автомата Мілі, а номери станів, що записані в одних і тих же стовпцях, співпадають між собою, або з прочерком;
- при об'єднанні станів з однаковими номерами в дужках і без них результируючий стан повинен бути у дужках;
- якщо при об'єднанні станів в одному з рядків знаходиться прочерк, а в інший – номер стану, то в скороченій таблиці пишеться номер стану.

Слід відмітити, що виявляти і об'єднувати еквівалентні і псевдоеквівалентні стани перед виявлення і об'єднанням сумісних станів немає потреби. Можна починати з виявлення і об'єднання сумісних внутрішніх станів. При цьому об'єднуються, якщо вони є, еквівалентні і псевдоеквівалентні стани. Більш того, у ряді випадків першочергове об'єднання псевдоеквівалентних станів недовизначеного автомата може ускладнити остаточне рішення, тобто може бути отриманий автомат з більшою кількістю внутрішніх станів, порівняно з автоматом, для якого зразу об'єднувались сумісні внутрішні стани.

Суттєву допомогу при аналізі сумісності рядків ПТП можуть надати діаграми (графи) сумісності внутрішніх станів, на яких сумісні стани розміщуються у вузлах, з'єднаних ненаправленими лініями. Множина внутрішніх станів є *сумісною*, якщо всі її стани є попарно сумісними. Таку множину можна замінити одним станом. Вибираючи мінімальну кількість таких множин, які охоплюють всі стани без їх повторення, отримують мінімальну кількість станів, яка достатня для реалізації автомата.

Для забезпечення знаходження максимальних множин сумісних станів іноді використовується трикутна таблиця сумісності.

Етап 3. Кодування станів автомата.

Найважливішою задачею, що має місце при кодуванні станів автомата є виключення значень його елементів пам'яті при заданих переходах. Таке кодування називається протигоночним. Найпростішим способом протигоночного кодування є сусіднє кодування, при якому два стани, що пов'язані між собою простими переходами кодуються наборами двійкових чисел, що відрізняються станом лише одного елемента пам'яті (ЕП). Основними вимогами до автомата, в якому забезпечується сусіднє кодування станів є:

- в графі автомата не повинно бути замкнутих контурів, що містять непарну кількість вершин;

- два сусідні стани другого порядку не повинні мати більше двох станів, що лежать між ними. При цьому під станами другого порядку маються на увазі два стани, шлях між якими по графу автомата складається з двох ребер (незалежно від орієнтації). На рис. 5.1 зображені два графа, які не задовольняють вказаним вимогам і не можуть бути закодовані сусідніми кодами.

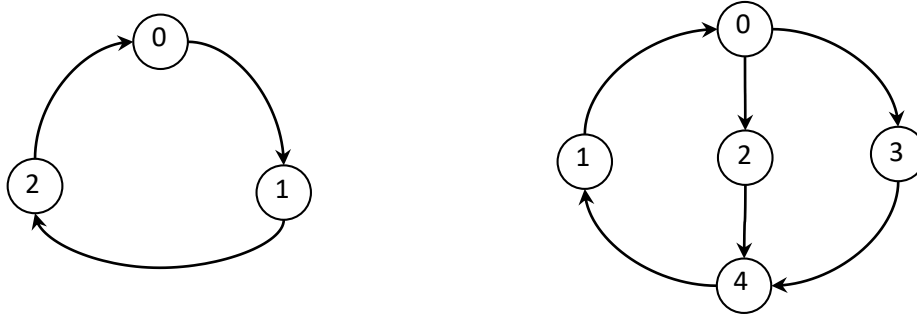


Рис. 5.1. Два графа, які не можуть бути закодовані сусідніми кодами

Граф переходів для генератора з заборною зображено на рис. 5.2. В розглянутому випадку граф має непарну кількість вершин, але контур незамкнений. Один з варіантів кодування станів приводить до табл. 5.3.

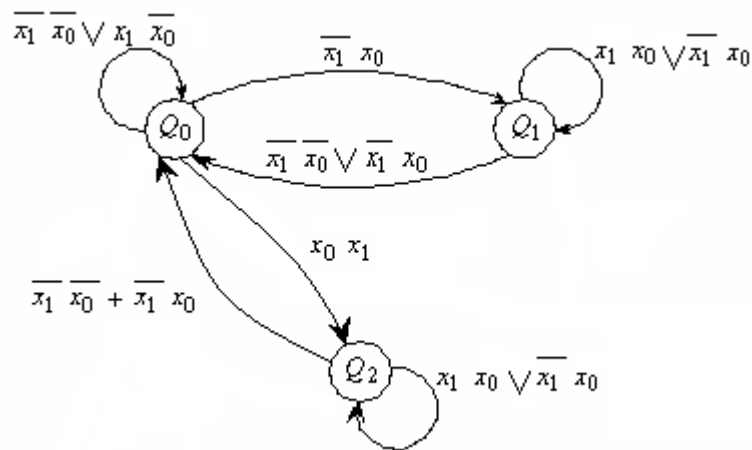


Рис. 5.2. Граф переходів для генератора з заборною

Таблиця 5.3. Один з варіантів кодування станів

Стани	Код	
	q ₁	q ₂
Q ₀	0	0
Q ₁	0	1
Q ₂	1	0

Після закінчення кодування станів будується закодована таблиця переходів автомата, що проектується. Для цього в скороченій таблиці переходів замінюються назви переходів їх кодовими значеннями.

Етап 4. Визначення функцій збудження елементів пам'яті і функцій виходів автомата.

Функції збудження автомата залежать від використовуваних типів тригерів, що вибираються в якості ЕП.

Розглянемо послідовно на конкретних прикладах особливості визначення функцій збудження ЕП для кожного з типів тригерів.

Характеристична таблиця RS -тригера має вигляд, зображений в табл. 5.4, де зірочкою ($*$) зображені дозвільні значення відповідної змінної.

Таблиця 5.4. Характеристична таблиця RS -тригера

q_n	q_{n+1}	R_n	S_n
0	0	*	0
0	1	0	1
1	0	1	0
1	1	0	*

Визначимо функції збудження RS -тригерів. Для цього скористаємось закодованою таблицею переходів автомата, в якій замість переходів $Q_n \rightarrow Q_{n+1}$ підставимо значення сигналів збудження відповідних входів тригерів. Таблиця прийме вигляд, зображений у табл. 5.5.

Таблиця 5.5. Закодована таблиця переходів автомата з RS -тригерами

$\begin{matrix} X \\ Q_n \end{matrix}$		$x_1 x_0$															
		00				01				11				10			
$q_1 q_0$		R_1	S_1	R_0	S_0	R_1	S_1	R_0	S_0	R_1	S_1	R_0	S_0	R_1	S_1	R_0	S_0
0 0		*	0	*	0	*	0	0	1	0	1	*	0	*	0	*	0
0 1		*	0	1	0	*	0	0	*	0	0	0	*	*	0	1	0
1 1		—				—				—				—			
1 0		1	0	*	0	0	*	*	0	0	*	*	0	1	0	*	0

Заповнення клітинок табл. 5.5 забезпечується безпосереднім використанням табл. 5.4. Для отримання функції збудження для кожного з входів тригерів переносимо їх значення в карти Карно (рис. 5.3, $a - z$).

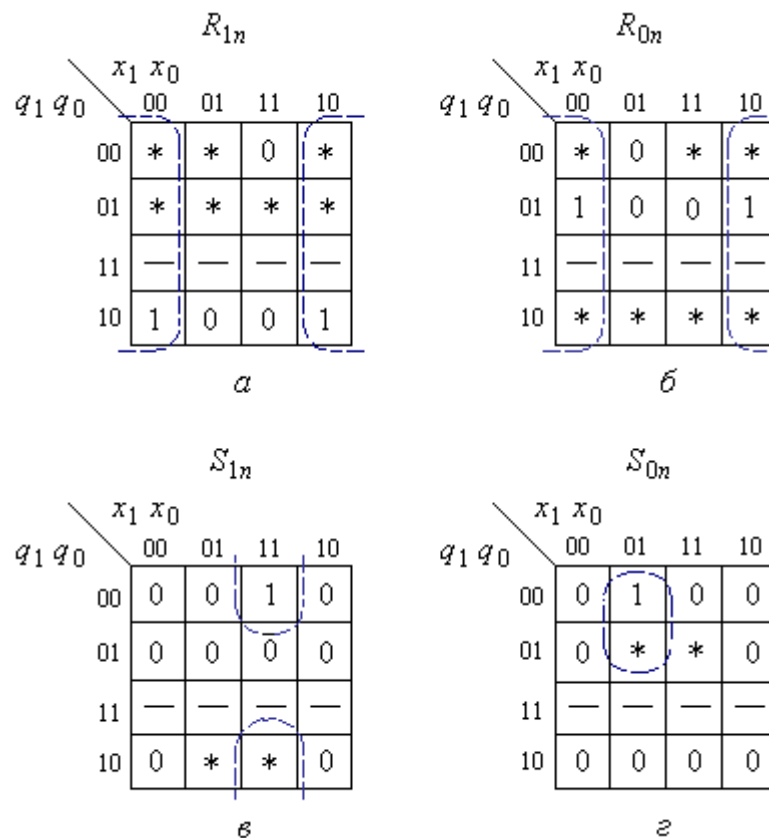


Рис. 5.3. Карти Карно

Мінімізуючи кожен з функцій збудження входів, отримуємо:

$$R_{1n} = \overline{x_0};$$

$$R_{0n} = \overline{x_0};$$

$$S_{1n} = \overline{q_0} x_1 x_0;$$

$$S_{0n} = \overline{q_1} \overline{x_1} x_0.$$

Для визначення функції виходу карта Карно будується на основі скороченої таблиці переходів, в якій стійкі стани замінюються значеннями у відповідному рядку, а нестійкі – символом * – довільного значення функції. В результаті отримаємо карту, зображену на рис. 5.4.

		$x_1 x_0$			
$q_1 q_0$		00	01	11	10
00	00	0	*	*	0
01	01	*	1	1	*
11	11	—	—	—	—
10	10	*	0	*	0

Рис. 5.4. Карта Карно функції виходу

З карти отримуємо мінімізоване значення функції виходу: $y = q_0$.

Етап 5. *Складання функціональної (принципової) схеми автомата в вибраному елементному базисі.*

Отримані в попередньому етапі вирази функцій збудження ЕП і функції виходу дають можливість легко розробити функціональну і принципову схеми автомата. Така схема приведена на рис. 5.5.

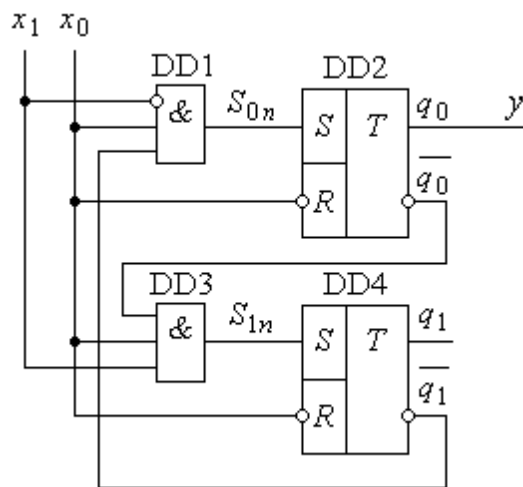


Рис. 5.5. Принципова схема автомата

Розглянемо її роботу. У нульовому стані при відсутності вхідних сигналів значення $q_0 = q_1 = 0$. В першому стані $x_1 = 0$, а $x_0 = 1$, тому, враховуючи, що $\overline{q_1} = 1$, сигнал S_{0n} з виходу DD1 буде поданий на тригер DD2 і встановить його вихід q в 1. При повторенні нульового стану тригер DD1 установиться в початковий стан нульовим значенням вхідного сигналу x_0 . В другому –

третьому стану при $x_1 = 1$ на вході DD1 формується сигнал заборони для сигналу x_0 . З цього короткого опису бачимо, що функціональна схема повністю відтворює часову діаграму, за якою спроектований автомат.

Приклад 5.1. Для автомата, заданого закодованою таблицею переходів (табл. 5.5), розробити функціональну схему з використанням JK -тригерів асинхронного типу.

Розв'язання. На основі таблиці переходів JK -тригера побудуємо характеристичну таблицю (табл. 5.6).

Таблиця 5.6. Характеристична таблиця

$q_n \rightarrow q_{n+1}$		J_n	K_n
0	0	0	*
0	1	1	*
1	0	*	1
1	1	*	0

Розробимо таблицю для визначення функції збудження тригерів автомата. Після підстановки в табл. 5.5 значень входів тригерів з характеристичної таблиці (табл. 5.6) отримаємо табл. 5.7.

Таблиця 5.7. Таблиця для визначення функції збудження тригерів автомата

$\begin{matrix} X \\ \hline Q_n \end{matrix}$		$x_1 \ x_0$															
		00				01				11				10			
$q_1 \ q_0$		J_1	K_1	J_0	K_0	J_1	K_1	J_0	K_0	J_1	K_1	J_0	K_0	J_1	K_1	J_0	K_0
0 0		0	*	0	*	0	*	0	*	0	*	1	*	0	*	0	*
0 1		1	*	*	0	0	*	*	0	0	*	*	0	0	*	*	0
1 1		*	0	*	0	*	0	*	1	*	0	*	1	*	0	*	0
1 0		—	—	—	—	—	—	—	—	*	0	0	*	0	1	0	*

Карта функції виходу будується на основі табл. 5.3 (рис. 5.6, а).

Будуємо карти Карно для функцій збудження JK -тригерів (рис. 5.6, б – д).

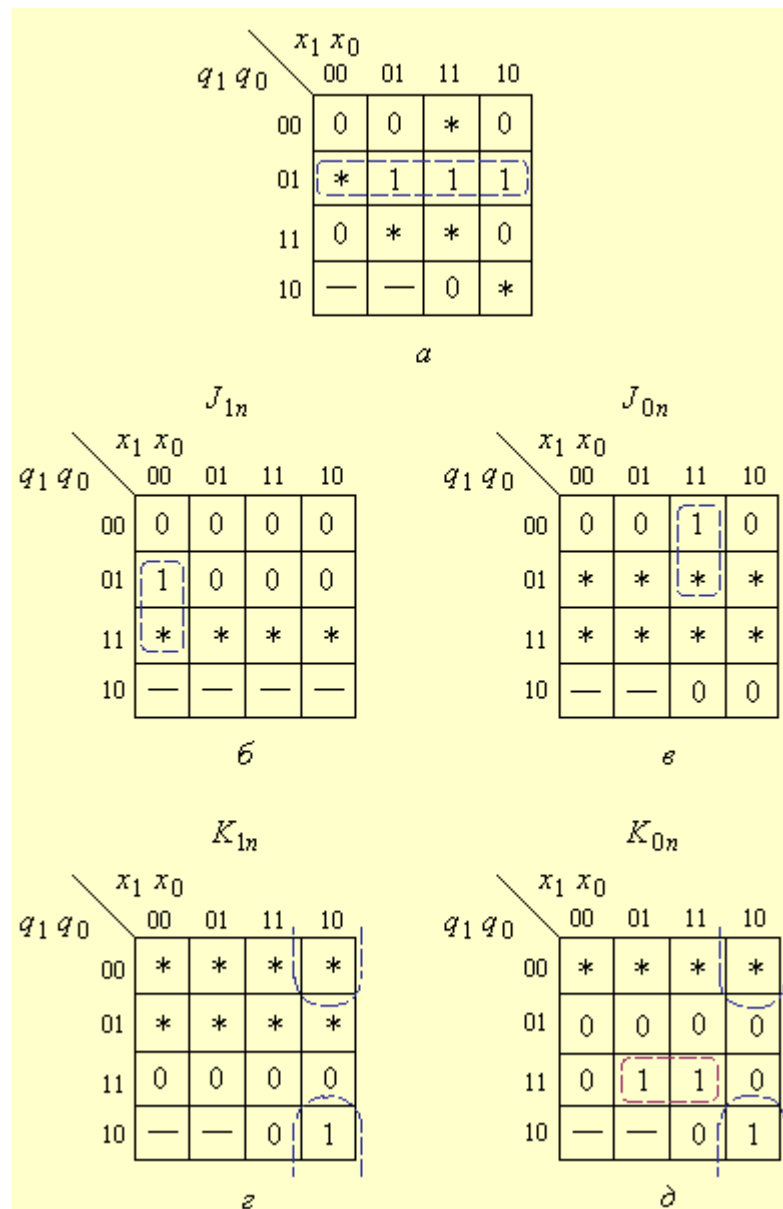


Рис. 5.6. Карти Карно

Мінімізовані функції збудження тригерів мають вигляд:

$$J_{1n} = \overline{q_0} \overline{x_1} \overline{x_0};$$

$$J_{0n} = \overline{q_1} x_1 x_0;$$

$$K_{1n} = \overline{q_0} x_1 x_0;$$

$$K_{0n} = q_1 q_0 x_0 + \overline{q_0} x_1 x_0.$$

З карти Карно функції виходу отримуємо: $y = \overline{q_1} q_0$.

За отриманими функціями будується комбінаційна схема автомата і його функціональна схема (рис. 5.7).

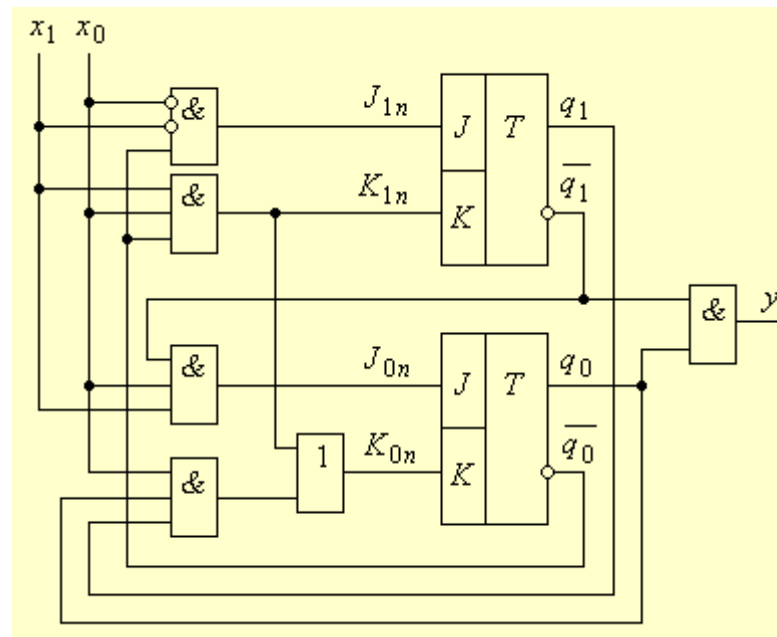


Рис. 5.7. Функціональна схема автомата

Контрольні запитання

1. Назвіть особливості початкових таблиць переходів.
2. Які внутрішні стани цифрових автоматів називаються еквівалентними?
3. Які внутрішні стани цифрових автоматів є сумісними?
4. Чи є можливість здійснити сусіднє кодування станів цифрового автомата, якщо в його графі переходів присутній замкнений контур із непарною кількістю вершин?
5. Після мінімізації внутрішніх станів автомата їх кількість зменшилась до 12. Яка кількість тригерів достатня для їх реалізації?
6. Поясніть різницю між еквівалентними та псевдоеквівалентними станами автомата.
7. Що є умовою для об'єднання трьох (чотирьох) сумісних внутрішніх станів автомата?
8. Поясніть принципову різницю між аналізом та синтезом цифрового автомата.

Перелік посилань

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.
2. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.К. Прикладна теорія цифрових автоматів. – К. НАУ, 2007. – 364 с.

Лекція №6. Синтез асинхронних імпульсних автоматів

Асинхронними імпульсними автоматами називаються такі, на які вхідні сигнали впливають лише на короткі інтервали часу зміни їх рівня з “0” в “1” або навпаки [1, 2]. Строге описання законів функціонування даних автоматів можливо лише на основі їх логічних властивостей, що визначаються функціями переходів. Для аналітичного формування закону функціонування асинхронних імпульсних автоматів вводиться оператор переходу d . Використання оператора переходу до потенційних сигналів $x(t)$ дає імпульсний сигнал $dx(t)$ (рис. 6.1, а, б), який приймає одиничне значення в момент переходу $x(t)$ з “0” в “1”. Для опису асинхронних імпульсних автоматів, що спрацьовують по передньому фронту, слід використовувати вираз для імпульсного сигналу у вигляді $d\bar{x}(t)$ (рис. 6.1, в).

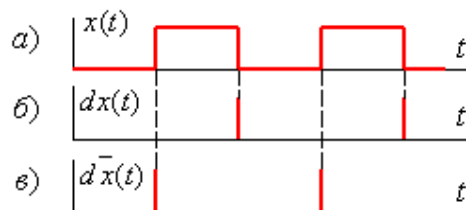


Рис. 6.1. Використання оператора переходу до потенційних сигналів

Задача синтезу асинхронних імпульсних автоматів може бути зведена до задачі синтезу асинхронних потенційних автоматів шляхом розробки методики перетворення функцій переходів, в які входять оператори d , в функції переходів виходів, що не містять операторів d . Таке перетворення називається *інтегруванням функції переходів імпульсного автомата* або *інтегруванням імпульсного автомата*.

Найбільш просто імпульсні автомати інтегруються табличним методом, оскільки є можливість побудувати таблицю переходів асинхронного потенційного автомата безпосередньо за функцією переходу імпульсного автомата.

Розглянемо це на прикладі синтезу імпульсного JK -тригера, повна функція переходів якого задається табл. 6.1.

Таблиця 6.1. Повна функція переходів

dJ	dK	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Вона практично не відрізняється від відповідної функції потенційного тригера, тому аналітичний вираз для закону функціонування (характеристичне рівняння) має вигляд:

$$Q_{n+1} = dJ\bar{Q}_n + d\bar{K}Q_n.$$

Виконуємо інтегрування характеристичного імпульсного JK -тригеа побудовою ПТП асинхронного потенційного автомата, еквівалентного синтезованому імпульсному тригеру.

Якщо сигнали J - і K - не змінюються, то $dJ = dK = 0$, тому в стійких станах, відповідно до характеристичного рівняння тригера, має місце рівність $Q_{n+1} = Q_n$. З цього витікає, що ПТП повинна мати в кожному стовбці, що відповідає стану входу, по два стійкі стани, які відрізняються значенням функції виходу $Q = 0$ або $Q = 1$. Оскільки можливих станів входу чотири, то число стійких станів асинхронного потенційного автомата дорівнює 8. Пронумерувавши їх у ряд, починаючи з нуля, заповнимо діагональ в ПТП (табл. 6.2). Потім, використовуючи характеристичне рівняння тригера, знайдемо переходи між стійкими станами.

Таблиця 6.2. Початкова таблиця переходів

$Q \backslash X(J, K)$	JK			
	00	01	11	10
0	(0), 0	2	4	6
1	(1), 1	3	5	7
2	0	(2), 0	5	6
3	0	(3), 1	5	7
4	1	3	(4), 0	6
5	1	3	(5), 1	7
6	1	3	4	(6), 0
7	0	3	4	(7), 1

Нехай, наприклад, початковим є стан входу $x=1,0$, якому відповідають стійкі стани Q_2 і Q_3 . Допустимо, що стан входу змінюється на $x=1,0$. Оскільки при цьому J змінюється в напрямку $0 \rightarrow 1$, а K $1 \rightarrow 0$, то $dJ=0$, а $d\bar{K}=1$, то автомат повинен перейти в той внутрішній стан, якому відповідає стан виходу, який визначається характеристичним рівнянням тригера $Q_{n+1} = 0 \cdot \bar{Q}_n + \bar{1} \cdot Q_n = 0$, тобто, незалежно від вихідного стійкого стану (Q_2 або Q_3), він повинен перейти в стійкий стан Q_4 . Аналогічно заповнюється решта клітинок ПТП, які відповідають нестійким станам у табл. 6.2.

Наступним етапом проектування є абстрактний і структурний синтез за методикою синтезу асинхронних потенційних автоматів.

Особливості синтезу синхронних автоматів

У синхронних автоматах, на відміну від асинхронних, використовується тактовий сигнал C , що задає моменти часу, в які можуть мати місце зміни внутрішніх станів автомата [3, 4]. У синхронних автоматах здебільшого використовуються D - або JK -тригери. З особливостей роботи синхронних тригерів витікає, що зміна інформаційних сигналів не може викликати зміни стану ЕП. Звідси витікає, що особливість синхронних тригерів, яка полягає в тому, що інформаційні сигнали лише керують змінами станів тригерів, переноситься і на синхронні автомати. Тому, оскільки на всі тригери автомата

синхросигнал подається одночасно, то вони і спрацьовують одночасно, незалежно від моменту подачі інформаційних сигналів.

У той же час, для синхронних автоматів, як відмічалось раніше, характерним є дискретний час, і стан автомата можна розглядати лише в дискретні моменти часу. Це дозволяє виключити з розгляду перехідні процеси, що протікають у комбінаційній схемі при зміні її вхідних сигналів.

Оскільки таблиці станів синхронних і асинхронних D - і JK -тригерів у цілому співпадають, то методи структурного синтезу синхронних автоматів ідентичні методам, використовуваним для синтезу асинхронних автоматів. У той же час, наявність синхронного режиму роботи всіх ЕП приводить до появи ряду особливостей проектування:

- комбінаційні схеми можуть проектуватись не вільними від гонок;
- допускаються несусідні зміни станів входу;
- допускаються критичні змагання ЕП, тобто допускається несусіднє кодування внутрішніх станів.

У великій кількості практичних задач із синтезу синхронних автоматів перехід від словесного опису до стандартних способів задання функцій переходів і виходів буває дуже простим. Тому для синхронних автоматів часто можливо, минаючи етап абстрактного синтезу, зразу ж перейти до етапу структурного синтезу. Таке спрощення процедури синтезу є наслідком використання синхронних ЕП, які виконують значно складніші функції, ніж асинхронні потенційні тригери.

Приклад 6.1. Синтезувати лічильник за модулем 5, що задається таблицею переходів (табл. 6.2) з використанням D - і JK -тригерів. Характеристичні таблиці переходів для D - і JK -тригерів приведені в табл. 6.3, 6.4

Таблиця 6.3. Характеристична таблиця переходів для D -тригера

N	Q_n			Q_{n+1}		
	q_2	q_1	q_0	q_2	q_1	q_0
Q_0	0	0	0	0	0	1
Q_1	0	0	1	0	1	0
Q_2	0	1	0	0	1	1
Q_3	0	1	1	1	0	0
Q_4	1	0	0	0	0	0

Таблиця 6.4. Характеристична таблиця переходів для JK -тригера

Q_n	Q_{n-1}	D_n	J_n	K_n
0	0	0	0	*
0	1	1	1	*
1	0	0	*	1
1	1	1	*	0

Розв'язання. Будуємо таблицю збудження для D -входів в D -тригерах (табл. 6.5).

Таблиця 6.5. Таблиця збудження для D -входів в D -тригерах

$Q_n \rightarrow Q_{n+1}$	Q_n			D_n		
	q_2	q_1	q_0	D_2	D_1	D_0
$Q_0 \rightarrow Q_1$	0	0	0	0	0	1
$Q_1 \rightarrow Q_2$	0	0	1	0	1	0
$Q_2 \rightarrow Q_3$	0	1	0	0	1	1
$Q_3 \rightarrow Q_4$	0	1	1	1	0	0
$Q_4 \rightarrow Q_0$	1	0	0	0	0	0

З цієї таблиці отримуємо функціональні залежності входів D -тригерів $D_0 \dots D_2$ від попереднього стану виходів тих же тригерів (функція збудження):

$$D_0 = \overline{q_2} \overline{q_0};$$

$$D_1 = \overline{q_2} (q_1 \oplus q_0);$$

$$D_2 = \overline{q_2} q_1 q_0.$$

За отриманими функціями збудження розробляємо функціональну схему лічильника (рис. 6.2).

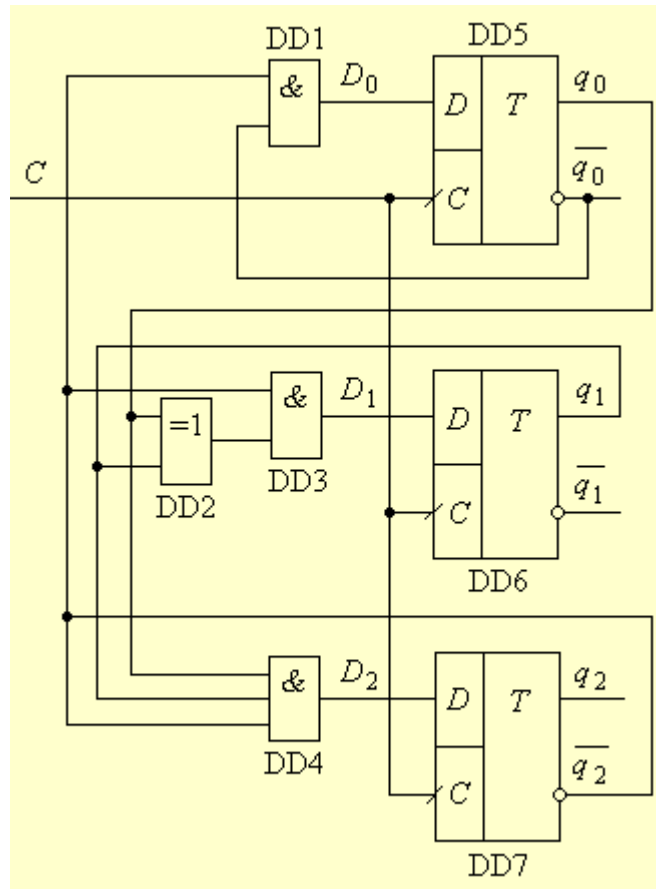


Рис. 6.2. Функціональна схема лічильника

У початковому стані Q_0 всі прямі виходи лічильника $q_2 q_1 q_0 = 000$, а їх інверсні еквіваленти $\overline{q_2} \overline{q_1} \overline{q_0} = 111$. Тому на вході тригера DD5 маємо $D_0 = 1$, а на входах тригерів DD6 і DD7, відповідно, $D_1 = 0$ і $D_2 = 0$. Тому при подачі синхросигналу DD5 записує на своєму прямому вході $q_0 = 1$. Це приводить до зміни значень входів перед другим тактовим імпульсом. Тепер ми маємо, що $D_0 = 0$, $D_1 = 1$, $D_2 = 0$ і по фронту C обнуляється q_0 , а $q_1 = 1$. Схема циклічно змінюватиме свої стани у відповідності до алгоритма роботи.

Тепер розглянемо особливості синтезу з використанням JK -тригерів. Будуємо таблицю збудження для входів JK -тригерів (табл. 6.6).

Таблиця 6.6. Таблиця збудження для для входів JK -тригерів

$Q_n \rightarrow Q_{n+1}$	Q_n			$J_n K_n$					
	q_2	q_1	q_0	J_2	K_2	J_1	K_1	J_0	K_0
$Q_0 \rightarrow Q_1$	0	0	0	0	*	0	*	1	*
$Q_1 \rightarrow Q_2$	0	0	1	0	*	1	*	*	1
$Q_2 \rightarrow Q_3$	0	1	0	0	*	*	0	1	*
$Q_3 \rightarrow Q_4$	0	1	1	1	*	*	1	*	1
$Q_4 \rightarrow Q_0$	1	0	0	*	1	0	*	0	*

Після мінімізації знаходимо:

$$J_0 = \overline{q_2}; \quad K_0 = 1;$$

$$J_1 = q_0; \quad K_1 = q_0;$$

$$J_2 = q_1 q_0; \quad K_2 = 1.$$

Схема лічильника з використанням синхронних тригерів приведена на рис. 6.3.

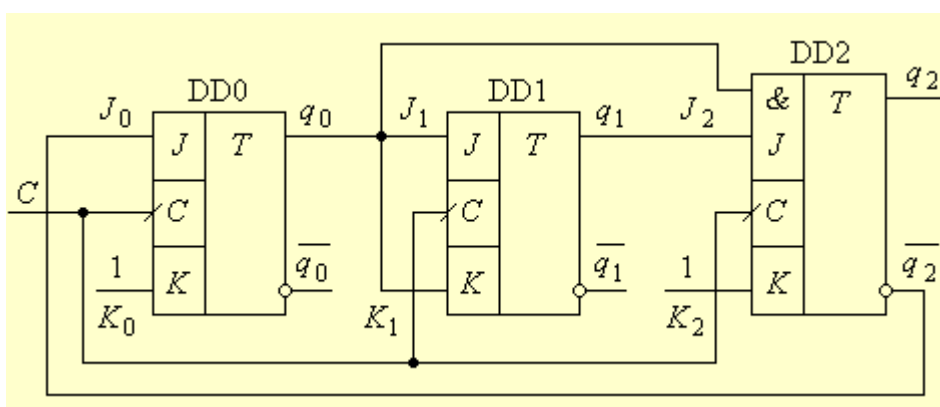


Рис. 6.3. Схема лічильника з використанням синхронних тригерів

Робота схеми повністю відповідає заданому алгоритму.

Використання теореми Шенона при синтезі скінченних автоматів на основі JK -тригерів

Як відомо, характеристичне рівняння JK -тригера має вигляд [1, 2]:

$$Q_{n+1} = J_n \overline{Q_n} + \overline{K_n} Q_n.$$

Звернемо тепер увагу на теорему Шенона:

$$f(x_{n-1}, \dots, x_i, \dots, x_0) = \overline{x_i} \cdot f(x_{n-1}, \dots, 0, \dots, x_0) + x_i \cdot f(x_{n-1}, \dots, 1, \dots, x_0). \quad (6.1)$$

Зрозуміло, що функції $f(x_{n-1}, \dots, 0, \dots, x_0)$ і $f(x_{n-1}, \dots, 1, \dots, x_0)$ – це різні функції, тому позначимо їх відповідно як функції f_1 та f_2 , а формулу (6.1) перепишемо у вигляді:

$$f(x_{n-1}, \dots, x_i, \dots, x_0) = \overline{x_i} \cdot f_1 + x_i \cdot f_2.$$

Тепер звернемо увагу на той факт, що функцію переходів будь-якого тригера скінченного автомата можемо розглядати як функцію вигляду:

$$Q_{(n+1)i} = f(x_0 \dots x_n; Q_0 \dots Q_n).$$

тому, розглядаючи її в відповідності до теореми Шенона, відносно стану заданого i -го тригера, отримуємо:

$$Q_{(n+1)i} = f_{1i} \cdot \overline{Q_i} + f_{2i} \cdot Q_i, \quad (6.2)$$

де, зрозуміло, функції f_{1i} та f_{2i} вже не містять відповідно змінних Q_i та $\overline{Q_i}$.

Порівнюючи тепер характеристичне рівняння JK -тригера з формулою (6.2), бачимо, що:

$$f_{1i} = J \text{ і } \overline{f_{2i}} = K.$$

Звідси витікає, що, прийнявши в функції переходів тригера $Q_i = 0$, отримуємо функцію збудження для входу J -:

$$J_i = f_{1i} = Q_{(n+1)i} \Big|_{Q_i = 0},$$

а, прийнявши $Q_i = 1$, отримуємо функцію збудження для входу K -:

$$\overline{K_i} = \overline{f_{2i}} = \overline{Q_{(n+1)i}} \Big|_{Q_i = 1} \text{ або } K_i = \overline{\overline{f_{2i}}}.$$

Приклад 6.2. Розробити автомат, який би в залежності від керуючого сигналу G міг працювати як двійковий лічильник ($G = 0$) або як лічильник з відліком в відповідності до коду Грея з $M = 8$ ($G = 1$). Автомат розробити з використанням JK -тригерів.

Розв'язання. Таблиця переходів автомата має вигляд табл. 6.7.

Таблиця 6.7. Таблиця переходів автомата

M	Q_{2n}	Q_{1n}	Q_{0n}	$Q_{2(n+1)}$	$Q_{1(n+1)}$	$Q_{0(n+1)}$
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	1
0	0	1	1	1	0	0
0	1	0	0	1	0	1
0	1	0	1	1	1	0
0	1	1	0	1	1	1
0	1	1	1	0	0	0
1	0	0	0	0	0	1
1	0	0	1	0	1	1
1	0	1	1	0	1	0
1	0	1	0	1	1	0
1	1	1	0	1	1	1
1	1	1	1	1	0	1
1	1	0	1	1	0	0
1	1	0	0	0	0	0

Рівняння переходів кожного з тригерів:

$$Q_{2(n+1)} = \overline{M} \cdot (\overline{Q_{2n}} Q_{1n} Q_{0n} + Q_{2n} \overline{Q_{1n}} \overline{Q_{0n}}) + M \cdot (\overline{Q_{2n}} Q_{1n} \overline{Q_{0n}} + Q_{2n} Q_{1n} Q_{0n});$$

$$Q_{1(n+1)} = \overline{M} Q_{2n} \overline{Q_{1n}} Q_{0n} + M \overline{Q_{2n}} Q_{1n} Q_{0n};$$

$$Q_{0(n+1)} = \overline{M} \cdot (\overline{Q_{2n}} Q_{1n} \overline{Q_{0n}} + Q_{2n} \overline{Q_{1n}} \overline{Q_{0n}}) + M \cdot (\overline{Q_{2n}} \overline{Q_{1n}} Q_{0n} + Q_{2n} Q_{1n} Q_{0n}).$$

Функції збудження для входів J - і K -:

$$J_2 = \overline{M} Q_{1n} Q_{0n} + M Q_{1n} \overline{Q_{0n}}; \quad K_2 = \overline{M} \overline{Q_{1n}} \overline{Q_{0n}} + M Q_{1n} Q_{0n};$$

$$J_1 = \overline{M} Q_{2n} Q_{0n}; \quad K_1 = M \overline{Q_{1n}} Q_{0n};$$

$$J_0 = \overline{M} \overline{Q_{2n}} Q_{1n} + M Q_{2n} \overline{Q_{1n}}; \quad K_0 = M (\overline{Q_{2n}} \overline{Q_{1n}} + Q_{2n} Q_{1n}).$$

Контрольні запитання

1. Охарактеризуйте різницю між асинхронними та синхронними цифровими автоматами.
2. Назвіть особливості синтезу асинхронних імпульсних автоматів.

Перелік посилань

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.

2. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.К. Прикладна теорія цифрових автоматів. – К. НАУ, 2007. – 364 с.

Лекція №7. Машина Тюрінга

У 1936 році англійський математик Алан Тюрінг вперше формалізував поняття алгоритму та з використанням теорії скінченних автоматів показав, що можна побудувати такий автомат, який може розв'язувати складні обчислювальні задачі. По суті, машина Тюрінга відрізняється від моделі скінченного автомата лише наявністю стрічки нескінченної пам'яті, на якій фіксуються стани автомату, та зчитувальної головки, яка може вільно рухатись вперед та назад вздовж всієї стрічки. Тобто, якщо скінчений автомат пам'ятає лише свій останній стан, у машині Тюрінга до нескінченної стрічки пам'яті записуються всі стани автомату. Наочне порівняння моделей скінченного автомату та машини Тюрінга наочно показано на рис. 7.1.

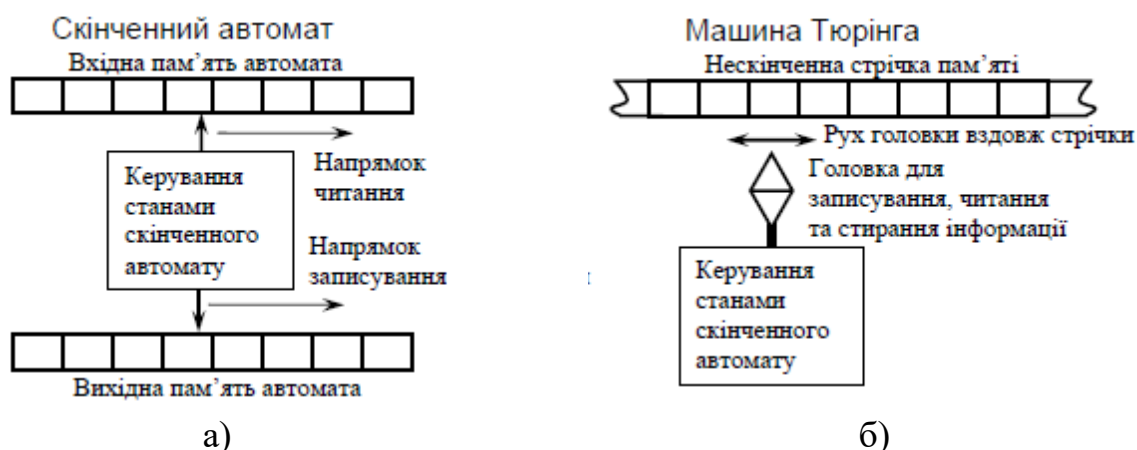


Рис. 7.1. Порівняння моделей скінченного автомату (а) та машини Тюрінга (б)

Величезна заслуга Тюрінга як вченого-математика полягає в тому, що він побудував формальні моделі для процесу розв'язування людським мозком математичних задач і показав, що за допомогою автомату, наведеного на рис. 7.1, б, такі задачі також можуть бути розв'язані. Дійсно, під час розв'язування навіть найскладніших математичних проблем людина виконує лише дві елементарні дії: логічно та несуперечливо мислить та робить відповідні висновки і записує результати своїх логічних міркувань на аркушах паперу, які можна вважати найпростішою моделлю нескінченної пам'яті. Головна відмінна риса цього процесу мислення полягає у тому, що людина постійно повертається до записів, які вона зробила раніше, і може або відновляти їх і використовувати

повторно, або знищувати. Тобто, якщо у машині Тюрінга використовується магнітна стрічка із можливістю стирання інформації, такий автомат повністю відтворює процес мислення людини, яка розв'язує математичну задачу. Скінченний автомат моделює процес логічного міркування, а магнітна стрічка використовується для запам'ятовування необхідної інформації. Зайва інформації в ході обчислювального процесу періодично стирається [1].

Важливим також є те, що Тюрінг ретельно продумав організацію роботи свого обчислювального автомату з технічної точки зору. Машина Тюрінга працює за часовими тактами, і на кожному такті виконуються такі основні дії [2]:

1. Читання символу із поточної комірки робочої стрічки.
2. Змінює свій стан залежно від поточного стану та прочитаного символу.
3. Виконує дії відповідно до поточного стану.
4. Записує свій стан на стрічку.
5. Переміщує зчитувальну головку праворуч або ліворуч на відповідну позицію стрічки.

Необхідно відзначити, що і сьогодні цей базовий принцип роботи комп'ютерів залишається незмінним. Всі сучасні комп'ютерні, мікропроцесорні та мікроконтролерні системи працюють за машинними тактами. Це дозволяє забезпечити синхронне виконання всього обчислювального процесу та уникнути помилок, пов'язаних із несинхронним виконанням окремих команд. Також слід відзначити, що машина Тюрінга з самого початку знаходить широке впровадження у системах криптографічного кодування та декодування інформації. Перші випробування запропонованої та розробленої ним машини Тюрінг провів під час другої світової війни за завданням англійської розвідки, реалізуючи алгоритми дешифрування для складних кодів, створених німецькою кодувальною машиною «Enigma». Цими кодами передавалися повідомлення для німецької армії і до робіт Тюрінга вважалося, що їх неможливо розшифрувати, проте з використанням машини Тюрінга англійській розвідці це

вдалося зробити. На основі моделі машини Тюрінга англійцями була створена декодувальна машина «Bombe».

Формально машина Тюрінга, модель якої наведена на рис. 7.1, б, описується сім'ю змінними, які характеризують роботу скінченного автомата із пам'яттю:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F), \quad (7.1)$$

де Q – повна множина станів керувального скінченного автомату, Σ – множина вхідних символів, Γ – множина символів, які записуються на стрічку ($\Sigma \subset \Gamma$), δ – функція переходу, q_0 – початковий стан, B – порожній символ, F – множина заключних станів автомата, які допускаються ($F \subset Q$).

Для того, щоб формально, на математичній мові, описати послідовність операцій, які виконує машина Тюрінга, необхідно побудувати повну та несуперечливу систему записування станів такої машини. Оскільки довжина стрічки машини Тюрінга є нескінченною, з першого погляду може здатися, що скінченне описання принципу роботи такого пристрою є неможливим. Проте слід мати на увазі, що після будь-якої скінченної кількості кроків роботи машина Тюрінга може проглядати лише скінченну кількість позицій стрічки. У теорії скінченних автоматів доведено, що у будь-якому описанні логіки роботи машини Тюрінга завжди існує нескінченна послідовність правих та лівих елементів стрічки, які не обробляються. Дійсно, для більшої кількості реальних задач програмування кількість комірок пам'яті, які використовуються для їхнього розв'язування, є скінченною. Тому машина Тюрінга часто використовується на практиці для перевірки можливості програмної реалізації розроблених алгоритмів. Якщо тест на машині Тюрінга показує, що для виконання алгоритму керувальному пристрою необхідна нескінченна кількість циклів, такий алгоритм неможна реалізувати. Якщо алгоритм роботи машини Тюрінга передбачає наявність на стрічці певної кількості пробілів B , це припускається, але у цьому разі кількості пробілів має також бути скінченною величиною.

Згідно із принципом роботи машини Тюрінга, який показаний на рис. 7.1, для позначення операцій, пов'язаних із роботою керувального автомату із стрічкою, необхідно вказувати також напрямок руху стрічки [3-5]. Зазвичай він позначається або стрілками ($\rightarrow, \leftarrow$), або літерами R та L (аббревіатура англійських слів *Right* – праворуч та *Left* – ліворуч). Для описання послідовності символів керування використовують послідовність:

$$(X_1, X_2, \dots, X_{i-1}, q, X_i, \dots, X_n, F, \{R, L\}), \quad (7.2)$$

де q – стан машини, $X_1, \dots, X_n$ – скінченна послідовність символів, записана на стрічці. Переходи між станами у машині Тюрінга позначаються символом $\perp_M$.

Наприклад, якщо розглядається перехід із зміщенням головки ліворуч, $\delta(q, X) = (p, Y, L)$, співвідношення (7.2) переписується у вигляді:

$$(X_1, X_2, \dots, X_{i-1}, q, X_i, \dots, X_n, L) \perp_M (X_1, X_2, \dots, X_{i-2}, p, X_{i-1}, Y, X_{i+1}, \dots, X_n, L).$$

Відповідно, для зміщення головки праворуч, тобто $\delta(q, X) = (p, Y, R)$, маємо:

$$(X_1, X_2, \dots, X_{i-1}, q, X_i, \dots, X_n, L) \perp_M (X_1, X_2, \dots, X_{i-1}, Y, p, X_{i+1}, \dots, X_n, L).$$

Розглянемо простий приклад побудови машини Тюрінга.

Приклад 7.1. Побудувати машину Тюрінга, вхідною мовою якої є регулярна мова $\{0^n, 1^n \mid n \geq 1\}$. Після послідовності нулів та одиниць на стрічці стоять пробіли. Машина повинна послідовно замінювати «0» на «X» та «1» на «Y», доки всі символи не будуть відповідати один одному.

Розглянемо алгоритм роботи машини більш детально. Аналіз введеного рядка починається зліва, з останнього символу введеної послідовності. Тут повинні стояти нулі. Послідовністю операцій, які машина має виконати на цьому етапі роботи, є заміна «0» на «X» та зміщення головки праворуч. Ці операції здійснюються, поки машина не знаходить символ «1». Машина замінює «1» на «Y» разом з рухом головки праворуч, вважаючи символи «1», «Y» та «0» припустимими, доки не знаходить символ «X». У цей момент

машина шукає «0», розташований праворуч, змінює його на «X» та продовжує роботу. Якщо праворуч після останнього символу «Y» або ліворуч після останнього символу «X» машина не знаходить порожній символ «B», або за умови, що ліворуч за останнім символом «Y» не стоїть символ «X», машина виходить на аварійне припинення, що свідчить про помилку введення вхідного рядку. Блок-схема спрощеного алгоритму роботи такої машини показана на рис. 7.2.

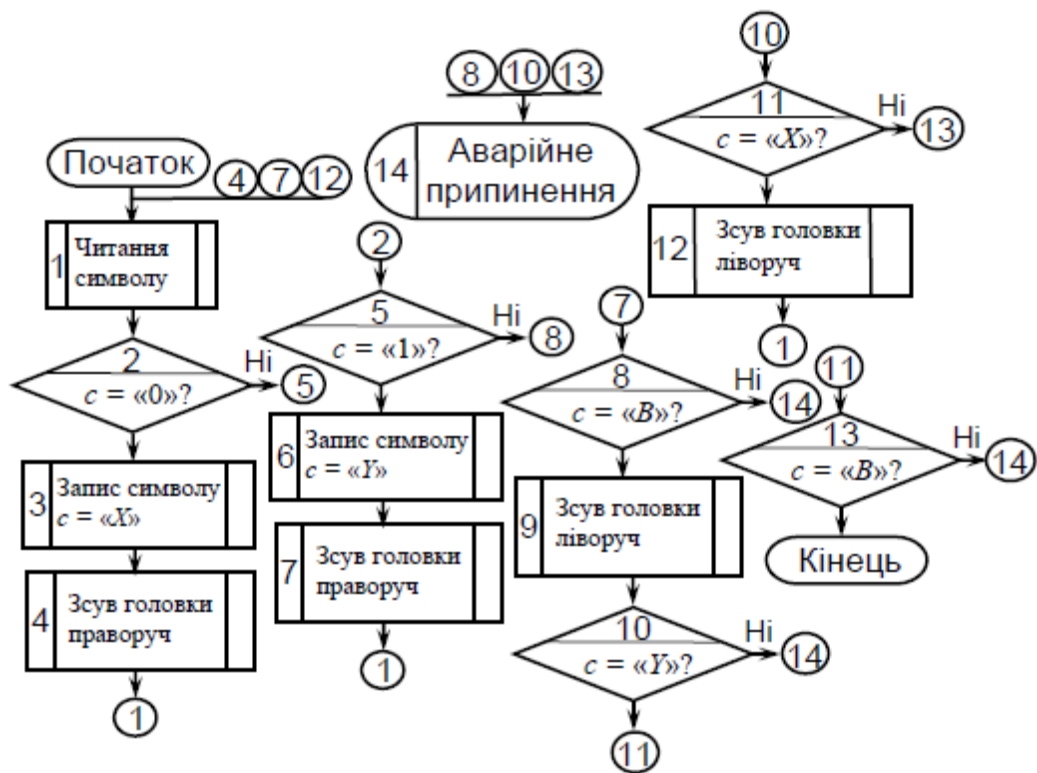


Рис. 7.2. Блок-схема алгоритму роботи машини Тюрінга для прикладу 7.1

Математичний опис алгоритму роботи машини Тюрінга, яка аналізується, згідно із співвідношенням (7.2) має наступний вигляд:

$$M = (\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, X, Y, B\}, \delta, q_0, B, \{q_4\}). \quad (7.3)$$

Функція δ для співвідношення (7.3) наведена у таблиці 7.1.

Розглянемо функцію переходів машини Тюрінга, яка наведена у таблиці 7.1, більш детально. Під час роботи машини частина стрічки, вздовж якої пройшла головка, завжди містить послідовність символів, яку можна описати

регулярним виразом $X^*0^*Y^*1^*$. Проаналізуємо послідовно всі можливі стани машини.

Таблиця 7.1. Таблиця функції переходів δ для машини Тюрінга, яка розглядається у прикладі 7.1

Стани машини Тюрінга	Символи на стрічці				
	0	1	X	Y	B
q_0	(q_1, X, R)	-	-	(q_3, X, R)	-
q_1	$(q_1, 0, R)$	(q_2, Y, L)	-	(q_1, Y, R)	-
q_2	$(q_1, 0, L)$	-	(q_0, X, R)	(q_2, Y, L)	-
q_3	-	-	-	(q_3, Y, R)	(q_4, B, R)
q_4	-	-	-	-	-

1. Стан машини q_0 є початковим і машина повертається до нього, коли головка читання рухається зправа-наліво та доходить до останнього нуля. Згідно із функцією переходів, яка наведена у таблиці 7.1, знаходячись у стані q_0 машина переходить до стану q_1 , записує у поточну комірку символ «X», а головка рухається праворуч. Якщо машина знаходиться у стані q_1 , згідно із функцією переходів, коли машина читає символи «0» та «Y», головка рухається праворуч, але якщо вона читає «1», головка рухається ліворуч. Окремим випадком для стану q_1 є читання символу «B». Як видно з таблиці 7.1, у цьому випадку стан машини не визначений, оскільки регулярна вхідна мова його не дозволяє. Тобто, за такої умови машина Тюрінга переходить до стану аварійної зупинки.

2. Розглянемо стан q_2 , який є більш складним, ніж q_1 . Коли машина знаходиться у цьому стані, головка читання-запису рухається ліворуч через всі символи «0» та «Y», доки не досягає крайнього лівого символу «X». Ця подія означає, що головка дійшла до правого кінця блоку нулів, які замінені на «X». У цьому разі можливі два випадки. Розглянемо їх окремо.

3. Якщо після «X» стоїть «0», машина пересуває головку ліворуч та повторює цикл стану q_2 , описаний у пункті 2.

4. Якщо після «X» стоїть «Y», це означає, що всі символи «0» вже замінені на «X». У цьому випадку машина переходить до наступного стану q_3 .

5. У стані q_3 машина починає рух головки читання праворуч через символи «Y». Якщо в результаті цієї перевірки першим з'являється невизначений символ «B», вважається, що послідовність $0^n 1^n$ є вірною. Тобто, машина допускає вхідну регулярно мову, працює без помилок і переходить до кінцевого стану q_4 . Якщо в результаті такого аналізу замість символу «B» з'являється «0», це свідчить про надмірну кількість нулів, а якщо «1» – про надмірну кількість одиниць. У цих двох випадках виникає помилка читання вхідної послідовності і машина переходить до стану аварійної зупинки.

З використанням виразу (7.3) на формальній мові математичної логіки описаний вище алгоритм для послідовності «0011» можна переписати наступним чином:

$$\begin{aligned} & (q_0 0011) \perp (Xq_1 011) \perp (X 0q_1 11) \perp (Xq_2 0Y1) \perp (q_2 X 0Y1) \perp (Xq_0 0Y1) \perp \\ & \perp (XXq_1 Y1) \perp (XXq_2 YY) \perp (Xq_2 XY Y) \perp (XXq_0 YY) \perp (XXYq_3 Y) \perp \\ & \perp (XXYYq_3 B) \perp (XXYYBq_4 B). \end{aligned} \quad (7.4)$$

Для іншої послідовності «0010», яка не відповідає вхідній мові машини Тюрінга, що розглядається, математичний вираз має вигляд:

$$\begin{aligned} & (q_0 0010) \perp (Xq_1 010) \perp (X 0q_1 10) \perp (Xq_2 0Y0) \perp (q_2 X 0Y0) \perp (Xq_0 0Y0) \perp \\ & \perp (XXq_1 Y0) \perp (XXYq_1 0) \perp (XXY0q_1 B). \end{aligned} \quad (7.5)$$

Отримані математичні вирази (7.4) та (7.5) є схожими, але їхня відмінність полягає у тому, що, згідно із останнім виразом співвідношення (7.5), коли машина знаходиться у стані q_1 ланцюжок заміненних символів містить «0» перед пробілом. Коли головка машини переміщується праворуч і знаходить пробіл, вона переходить до стану аварійного припинення, оскільки, згідно із таблицею 7., для стану q_1 символ «B» є невизначеним.

Зрозуміло, що описання принципу роботи машини Тюрінга з використанням методів теорії скінченних автоматів є більш наочним та зрозумілим, ніж блок-схема алгоритму, яка наведена на рис. 7.2. Діаграма станів скінченного автомату, функція переходів якого описується таблицею 7.1, наведена на рис. 7.3.

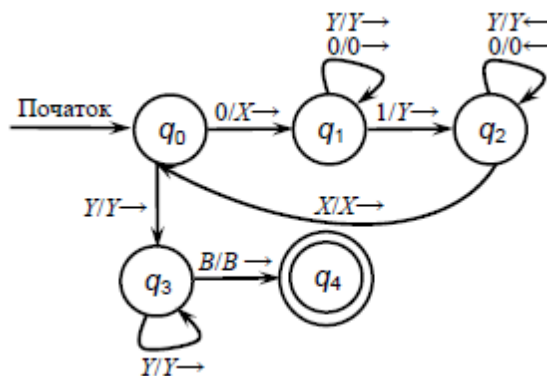


Рис. 7.3. Діаграма станів машини Тюрінга для прикладу 7.1

Як було відмічено раніше, сьогодні машина Тюрінга часто використовується саме для перевірки працездатності та несуперечливості способів кодування повідомлень для різних регулярних мов, які розробляються. Проте початково ця модель використовувалась лише для обробки натуральних чисел, і для організації роботи машини Тюрінга була розроблена спеціальна вхідна мова. Натуральні числа кодувались як послідовності нулів або одиниць, наприклад, число 5 записувалось як «11111» або «00000». Результати обчислень зберігались через зміну кількості одиниць, або через запис нових блоків одиниць в іншому місці стрічки. Така система кодування натуральних чисел називається одиничною. Зараз в комп'ютерних системах такий спосіб кодування майже не використовується, його недолік полягає у тому, що код числа займає велику кількість розрядів, проте на початковому етапі розвитку обчислювальної техніки саме одиничні коди здавалися дослідникам найбільш ефективними. Розглянемо приклади розв'язування задач теорії чисел на машині Тюрінга за умови їхнього подання в одиничному коді.

Приклад 7.2. Побудувати машину Тюрінга для обчислення функції власного віднімання (англійський термін – proper subtraction), яка також іноді називається усіченою різницею. Ця функція у математиці позначається символом $\div$ та визначається наступним чином:

$$m \div n = \max(m - n, 0) = \begin{cases} m - n, & m > n; \\ 0, & m \leq n. \end{cases} \quad (7.6)$$

Машину Тюрінга, яка обчислює функцію власного віднімання натуральних чисел, згідно із співвідношенням (7.2) можна описати наступним математичним виразом:

$$M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B). \quad (7.7)$$

Оскільки у даному прикладі машина Тюрінга не використовується для контролю вхідної мови, множина станів машини, які допускаються, також не розглядається.

Алгоритм роботи машини можна описати наступним чином. Машина читає зі стрічки послідовності « $B0^m10^nB$ », а закінчує свою роботу записом послідовності « $B0^k B$ », де $k = m \div n$. Розглянемо цей алгоритм більш детально. Рух головки починається з кінця послідовності, зправа наліво. Машина знаходить перший нуль, який стоїть після «1», замінює його пробілом, після чого головка починає рухатися зліва направо. Коли під час цього руху машина знаходить «0», вона замінює його на «1», після чого головка змінює напрямок руху і знов починає рухатися зправа наліво у пошуках нуля, що стоїть після одиниць. Для цієї циклічної процедури існує два можливих кінцевих результати.

1. Під час руху головки зправа наліво після одиниць машина замість нуля знаходить порожній символ. Це означає, що всі n нулів, які стояли після одиниці у початковій послідовності « $B0^m10^nB$ », замінені на «1», а n із m нулів, які стояли до 1 – на пробіли. Тобто, у даному випадку $m > n$ і отримана шукана послідовність « $B0^k B$ », де $k = m - n$.

2. Під час руху головки зліва направо машина перед одиницями не знаходить нуля. Це означає, що $m < n$ і у даному випадку функція власного віднімання $k = m \div n = 0$.

Наочно алгоритм роботи машини Тюрінга для першого випадку показаний на рис. 7.4, а, а для другого – на рис. 7.4, б.

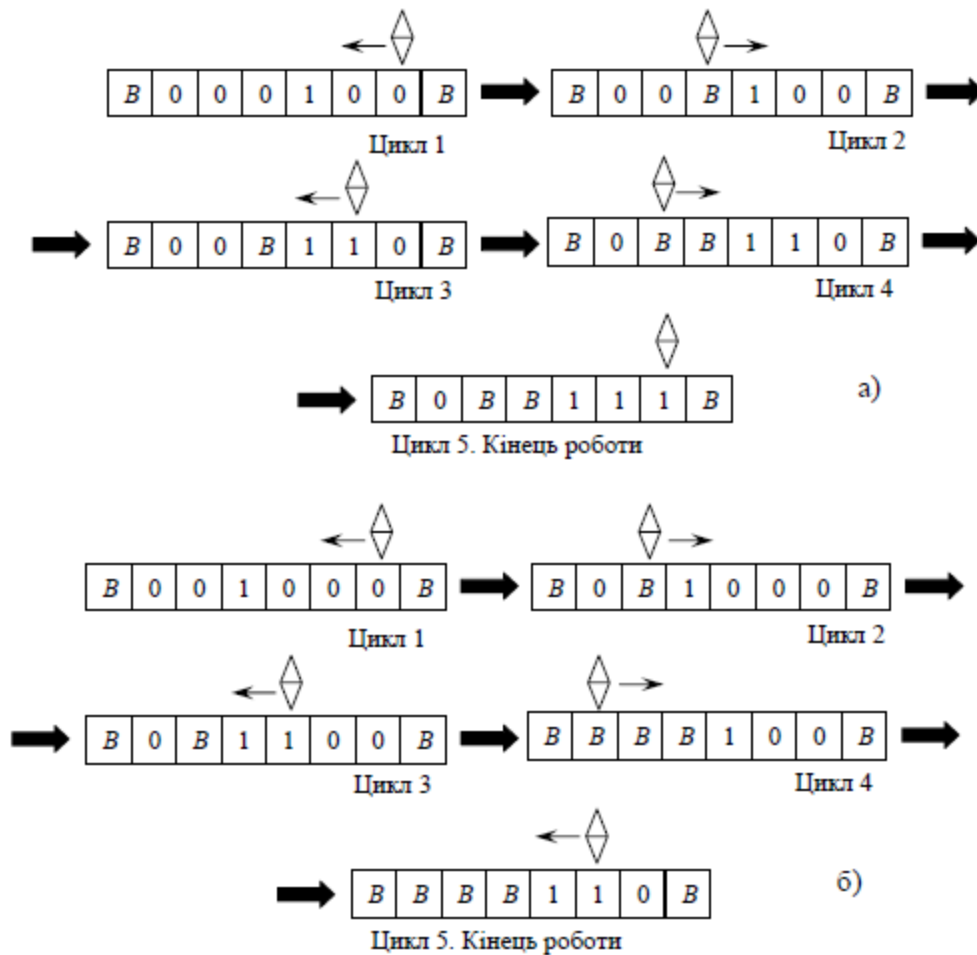


Рис. 7.4. Ілюстрація принципу роботи машини Тюрінга для прикладу 7.2 за умови вхідних послідовностей «B000100B» та «B001000B»

Алгоритм роботи машини Тюрінга для прикладу, який розглядається, наведений на рис. 7.5 а діаграма станів скінченного автомату, який описує принцип роботи цієї машини – на рис. 7.6.

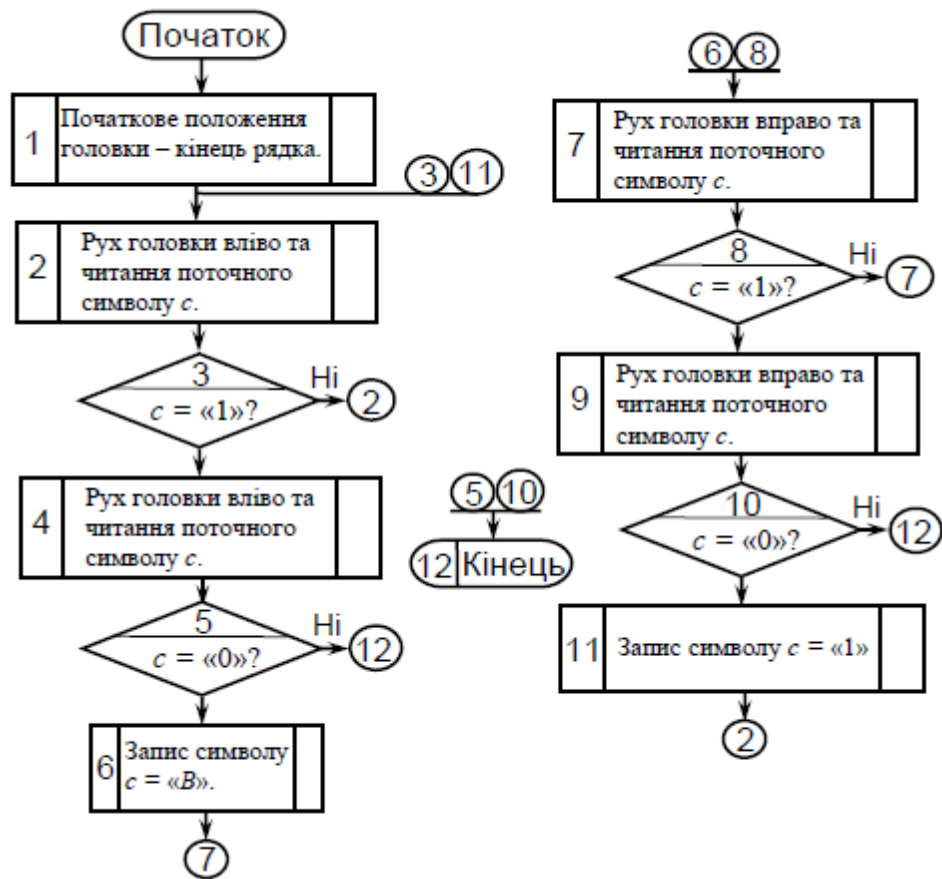


Рис. 7.5. Алгоритм роботи машини Тюрінга для прикладу 7.2

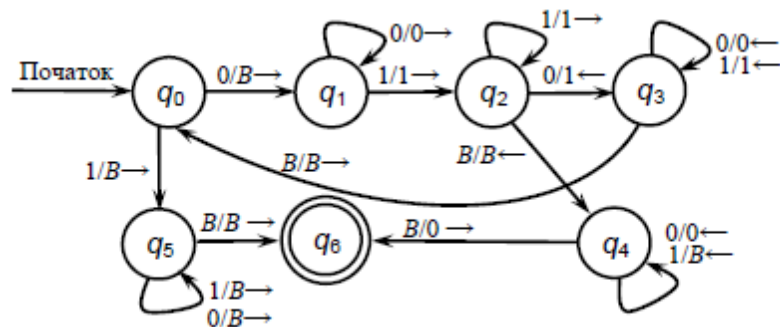


Рис. 7.6. Діаграма станів машини Тюрінга для прикладу 7.2

Функція переходів для моделі скінченного автомату, діаграма станів якого представлена на рис. 7.6, наведена у таблиці 7.2.

Таблиця 7.2. Таблиця функції переходів δ для машини Тюрінга, яка розглядається у прикладі 7.2

Стани машини Тюрінга	Символи на стрічці		
	0	1	B
q_0	(q_1, B, R)	(q_5, B, R)	-
q_1	$(q_1, 0, R)$	$(q_2, 1, R)$	-
q_2	$(q_3, 1, L)$	$(q_2, 1, R)$	(q_4, B, L)
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$	(q_0, B, R)
q_4	$(q_4, 0, L)$	(q_4, B, L)	$(q_6, 0, R)$
q_5	(q_5, B, R)	(q_5, B, R)	(q_6, B, R)
q_6	-	-	-

Розглянемо характеристику станів машини Тюрінга $q_0 - q_6$, із якими пов'язана функція переходів δ .

q_0 – початковий стан машини, у якому починається перший цикл її роботи. Якщо головка читає символ «0» – цикл продовжується, а якщо символ «B» – переривається. Машина переходить до стану q_5 .

q_1 – читання початкового блока нулів, пошук одиниці. Якщо одиниця знайдена – машина переходить до стану q_2 .

q_2 – рух головки праворуч, читання символів «1» до появи символу «0». Зчитаний символ «0» міняється на «1» і машина переходить до стану q_3 . Якщо всі нулі вже замінені і замість символу «0» знайдений символ «B», тоді машина переходить до стану q_4 .

q_3 – головка рухається праворуч, читає символи «0» та «1» та шукає пробіл. Коли знаходить – машина переходить до стану q_0 .

q_4 – головка рухається праворуч, читає символи «1» та шукає пробіл. Пробіл міняється на символ «0». Машина переходить до стану q_6 та зупиняється.

q_5 – в цей стан машина переходить із стану q_0 , якщо всі нулі із першого блоку вже замінені на пробіли. У цьому випадку машина замінює всі символи «0» та «1», які залишились, на пробіли, та переходить до стану q_6 .

q_6 – завершення роботи за заданою програмою.

Діаграма станів машини Тюрінга, призначеної для обчислення функції власного віднімання, яка наведена на рис. 7.6, дозволяє реалізувати цю математичну функцію на базі електронного пристрою, моделлю якого є розглянутий нами скінченний автомат. Використовуючи функції розбиття автоматів, можна засобами програмування перейти від виконання елементарних математичних операцій до проведення більш складних обчислень. Саме таким чином у комп'ютерній техніці реалізований виклик підпрограм, за рахунок чого на базі простих програмованих операцій створюються складні програмні комплекси. Наприклад, на рис. 7.7 наведена діаграма станів машини Тюрінга, яка реалізує алгоритм Евкліда для пошуку найбільшого спільного дільника двох натуральних чисел a та b .

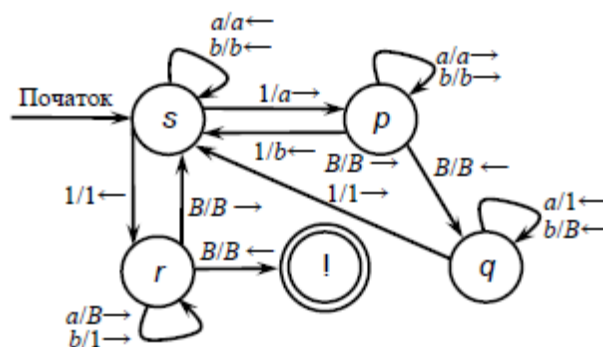


Рис. 7.7. Діаграма станів машини Тюрінга, яка обчислює найбільший спільний дільник двох натуральних чисел a та b

Розглянутий приклад 7.2 показує, що із-за використання одиначної системи числення структура обчислювального автомату, побудованого на основі класичної машини Тюрінга, у значній мірі ускладнюється. Зрозуміло, що пізніше були розроблені більш ефективні архітектури комп'ютерів, орієнтованих на обробку двійкових чисел, наприклад, архітектура фон Неймана

і гарвардська архітектура, головною відмінною рисою яких є прямий вільний доступ у будь-який момент часу до всіх комірок пам'яті. Проте у тридцятих роках XX сторіччя, на початковому етапі розвитку обчислювальної техніки, спочатку необхідно було проаналізувати можливості автоматизації простих розрахунків на простих моделях і довести необхідні теореми теорії автоматів, що і було зроблено на базі моделі машини Тюрінга. Такий шлях розвитку обчислювальної техніки дещо нагадує історичний розвиток всієї математики як науки. Дійсно, спочатку люди знали лише натуральні числа і поступово вчилися рахувати на пальцях, на палицях та на інших предметах, які були під рукою. Таким же чином і зараз вчать рахувати дітей у дитячих садках. Вже пізніше, із розвитком торгівлі та товарних відносин, з'явилися від'ємні, дрібні та раціональні, ірраціональні та комплексні числа та були визначені способи проведення математичних операцій з ними.

Виникає важливе питання. Як програмувати Машину Тюрінга і який набір елементарних команд є достатнім для реалізації будь-якого алгоритму? Тюрінг показав, що для програмування його машини необхідною та достатньою є сукупність із п'яти інструкцій, список яких наведений у таблиці 7.3.

Таблиця 7.3. Таблиця елементарних інструкцій для машини Тюрінга

№	Команда	Призначення
1.	зсув $\{R, L\}$	Зсув головки на одну позицію праворуч або ліворуч
2.	перехід p	Безумовний перехід до команди із номером p
3.	якщо s, p	Перехід до команди із номером p за умови, що у комірці, яка читається, знаходиться символ s
4.	запис s	Запис символу s до поточної комірки
5.	стоп	Припинення роботи машини

Наприклад, початок програми, яка реалізує алгоритм роботи машини Тюрінга для пошуку найбільшого спільного дільника двох натуральних чисел, можна записати так:

```
00: якщо a, 02;
01: перехід 05;
02: запис a;
03: зсув L;
```

04: перехід 00;
 05: якщо b , 07;
 06: перехід 10;
 07: запис b ;
 08: зсув L ;

Пізніше італійські математики Бом та Якопіні, а також Дайкстра у своїх працях, на основі теорії предикатів показали, що якщо машина має оператор циклу, тоді оператор переходу за номером команди є зайвим. Саме цей теоретичний підхід став підґрунтям для створення методів структурного програмування та відповідних мов програмування. Слід також відзначити, що одночасно із Аланом Тюрінгом до концепції обчислювальних машин із стрічкою прийшли відомий американський математик Еміль Пост та російський математик Андрій Андрійович Марков (молодший).

Розглянемо також дві важливі теореми, пов'язані із можливістю декомпозиції машини Тюрінга та спрощення її логічної структури.

Теорема 7.1. Для будь-якої машини Тюрінга існує еквівалентна машина, така, що всім ребрам, які входять до її вершини, відповідає один напрямок переміщення головки.

Відповідний спосіб еквівалентного перетворення машини Тюрінга пов'язаний із паралельною декомпозицією автоматів, приклад такого розбиття показаний на рис. 7.8.

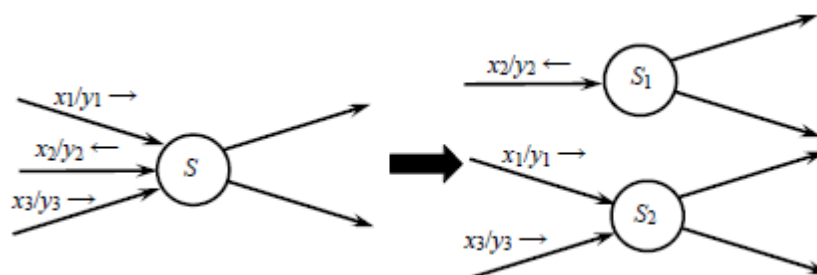


Рис. 7.8. Ілюстрація декомпозиції машини Тюрінга згідно із теоремою 7.1

Теорема 7.2. Для будь-якої машини Тюрінга існує еквівалентна машина, яка працює на напівнескінченній стрічці.

Ця теорема є досить простою і вона тісно пов'язана із визначенням поняття напівнескінченної стрічки. Припустимо, що половина стрічки має марковані комірки та заштрихуємо їх. Відповідна модель стрічки показана на рис. 7.9, а. Тепер марковані та немарковані комірки можна розташувати поряд, а символ «*», який не міститься у вхідному словнику регулярної мови машини, будемо використовувати для розділення заштрихованої та незаштрихованої зон. Модель такої перетвореної напівнескінченної стрічки показана на рис. 7.9, б, символ «*» тут вважається границею двох зон.

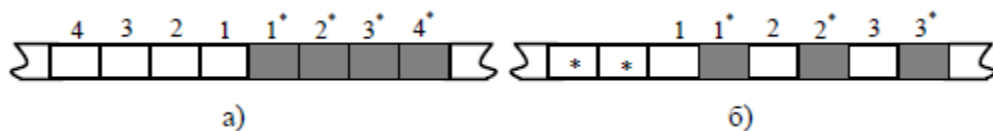


Рис. 7.9. Приклад напівнескінченної стрічки (а) та її перетворення (б)

Спосіб декомпозиції машини Тюрінга, яка працює на напівнескінченній стрічці, показаний на рис. 7.10. Таке перетворення відповідає послідовній декомпозиції скінченних автоматів.

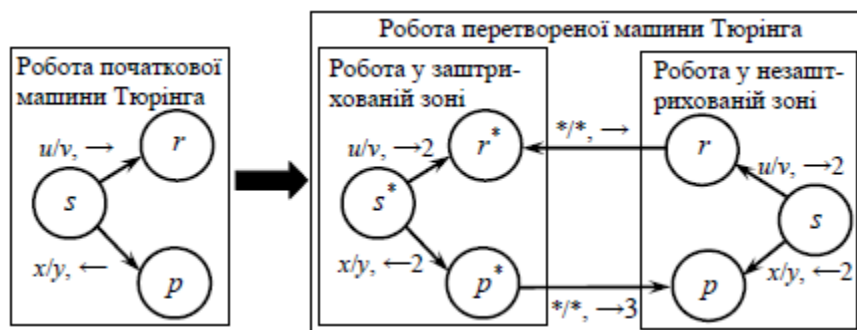


Рис. 7.10. Спосіб еквівалентного перетворення машини Тюрінга через моделювання її роботи на напівнескінченній стрічці

Перетворення машини Тюрінга, наведені на рис. 7.9 та рис. 7.10, є теоретичним підґрунтям для аналізу можливостей розпаралелювання алгоритмів.

Як розвиток математичних методів алгебри Буля, теорії множин, теорії предикатів, теорії скінченних автоматів та теорії ймовірностей у шістдесятих-

сімдесятих роках XX сторіччя був сформований один із найновітніших розділів сучасної математики – теорія нечіткої логіки. Бурхливий розвиток методів моделювання, пов’язаних із нечіткою логікою, та практична спрямованість цих математичних досліджень призвели до того, що сьогодні алгоритми нечіткої логіки також складають теоретичне підґрунтя для розробки сучасної складної програмованої електронної апаратури та стандартів зв’язку.

Контрольні запитання

1. Що являє собою машина Тюрінга і чим вона суттєво відрізняється від скінченного автомату?
2. Сформулюйте головні принципи роботи машини Тюрінга. Чи відповідають вони принципам роботи сучасних ЕОМ? Свою відповідь обґрунтуйте.
3. Наведіть власні приклади побудови машини Тюрінга, призначеної для автоматизації виконання розрахунків за алгоритмами теорії чисел та побудуйте діаграму станів такої машини.
4. Які елементарні операції використовуються для організації роботи машини Тюрінга? Наведіть власні приклади програм, написаних з використанням таких операцій.
5. Яка із елементарних інструкцій класичної машини Тюрінга вважається зайвою з точки зору сучасної теорії програмування? Свою відповідь обґрунтуйте та наведіть доречні приклади.

Перелік посилань

1. Hartmanis J., Stearns R. Algebraic Structure Theory of Sequential Machines. – N.Y.: Printice Hall Inc. – 1966, – 211 p.
2. Turing A.M. On Computable Numbers, with an Application to the Entscheidungsproblem. – Proceedings of the London Mathematical Society, no. 42, 1937, p. 230–265.
3. Turing A.M. On Computable Numbers, with an Application to the Entscheidungsproblem: A correction. – Proceedings of the London Mathematical Society, no. 43, 1937, p. 544–566.

4. Turing A.M. Intelligent Machinery. – Baltimore: University Park Press, 1968, –31 p.

5. Hennie F.C., Stearns R.E. Two-tape simulation of multitape Turing machines. – JACM, no. 13(4), 1966, p. 533–546.

Лекція №8. Методологія нечіткого моделювання

Першою відомою публікацією з теорії нечіткої логіки (англійський термін – fuzzy logic) вважається робота американського математика азербайджанського походження, професор університету Берклі в Каліфорнії Лотфі Заде, яку він оприлюднив у 1965 році. У цій роботі Л. Заде вперше ввів поняття нечіткої множини та теоретично показав, який математичний апарат алгебри логіки та теорії ймовірностей можна використовувати для аналізу нечітких множин. Математикам ця теорія здалася дуже цікавою і вони відразу почали займатися проблемами нечіткої логіки, вагомий внесок у розвиток цього розділу математики зробили Л. Заде, Д. Дюбуа, А. Прада, М. Сугено, Д. Беждека та Р. Ягер. Спочатку вважалося, що це суто теоретичні пошуки, а їхнє практичне значення аналітики вважали дуже сумнівним. Ситуація ретельно змінилася у 1975 році, коли видатний англійський математик Ібрагім Мамдані вперше застосував розроблений їм алгоритм нечіткої логіки для керування паровим генератором. Оригінальність алгоритму Мамдані полягала у тому, що йому вдалося уникнути надмірно великого об'єму обчислень за рахунок дуже чіткої організації введення даних. Саме після робіт Мамдані теорія нечіткої логіки була належно оцінена не лише математиками, але й інженерами [1-4].

Щоб зрозуміти сутність теорії нечіткої логіки та відповідну методологію побудови нечітких моделей, визначимо спочатку предметну область цієї науки як окремої галузі математики.

Нечітка модель розглядається як інформаційно-логічна модель системи, яка базується на теорії нечітких множин та на нечіткій логіці.

Згідно із теорією системного аналізу, окремі етапи формування моделей нечіткої логіки визначаються наступним чином.

1. Аналіз проблемної ситуації.
2. Визначення структури предметної області, яка аналізується, та формування для неї нечіткої моделі.
3. Проведення розрахунків та обчислювальних експериментів з використанням нечіткої моделі.

4. Впровадження результатів обчислювальних експериментів для реальної ситуації, яка моделювалася, або для реального приладу.

5. Корекція та доробка математичної моделі з урахуванням результатів натурального експерименту.

Найбільш складним у формуванні моделей нечіткої логіки є невизначеність подання структури або поведінки системи, яка розглядається, і у цьому разі сама категорія невизначеності може бути розглянутою з різних точок зору. У сучасній методології системного моделювання розглядаються такі різновиди невизначеності модельних уявлень.

1. Неясність та нечіткість границь системи. Прикладом такої невизначеності є використання двійкової логіки алгебри Буля для ознак та характеристик предметів, як-то: «світлий – темний», «яскравий – блідний», «високий – низький», «довгий – короткий», «швидкий – повільний» «рідкий – густий», «простий – складний» «молодий – старий», «дорогий – дешевий» тощо.

2. Невизначеність та багатозначність окремих термінів, які використовуються під час побудови концептуальних моделей системи. Це пов'язано з тим, що граматики звичайних людських мов, таких як українська, англійська, німецька, французька або інша, згідно із класифікацією за Хомським є не регулярними, а контекстно-залежними. Це означає, що деякі слова, які ми використовуємо, є багатозначними і зрозуміти їхнє значення можна лише за контекстом. Наприклад, слово «автомат» в українській мові має багато принципово різних значень: автомат як пристрій, який виконує відповідну роботу, автомат як математична модель електронної цифрової системи, гральний автомат або автомат як вогнепальна зброя.

3. Неповнота модельних уявлень про деякі складні системи, особливо у зв'язку із розв'язуванням складних проблем, які погано формалізуються. У цьому випадку будь-яка спроба побудувати адекватну модель складної системи зіштовхується із принциповою неможливістю врахувати всі важливі аспекти її функціонування.

4. Суперечливість окремих компонентів, а також окремих уявлень або вимог, яким має задовольняти модель складної системи. Наприклад, зазвичай суперечливими є вимоги високої чутливості вимірювальної електронної системи та широкого діапазону для вхідної фізичної величини, яку вона вимірює. Високочастотні системи мають високу швидкодію, але зазвичай вони є досить дорогими та споживаються багато електроенергії. Суперечливими оптимальними факторами в економіці є мінімальний час виконання роботи та мінімальні фінансові витрати. У теорії моделювання такі задачі називають задачами, які носять суперечливо-компромісний характер.

5. Невизначеність появи тих або інших подій, які описують можливість знаходження системи у тому або іншому стані у майбутньому.

Щодо методології нечіткого моделювання, вона стосується насамперед розробки адекватних моделей для систем із першим аспектом невизначеності, тобто, для систем із нечітким описанням границі та окремих станів системи. Проте слід сказати, що введення та розвинення концепції нечіткої міри та теорії невизначених можливостей дозволяє стверджувати, що і інші перелічені аспекти невизначеності також можуть бути вивчені з використанням методів теорії нечіткого аналізу. Графічна модель системи із нечіткою границею, основана на теорії множин, наведена на рис. 8.1.

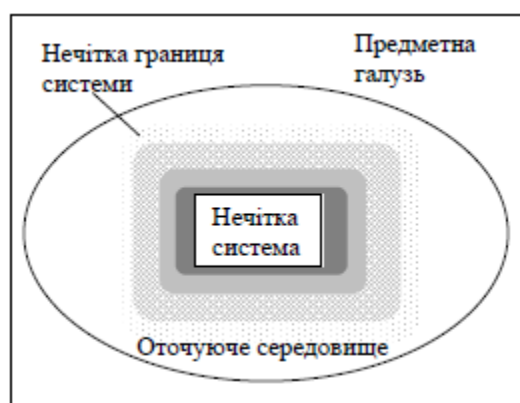


Рис. 8.1. Графічна ілюстрація системи із нечіткою границею

Деякі фахівці іноді плутають моделі нечіткої логіки із моделями теорії ймовірностей та математичної статистики. Такі підходи у нечіткій логіці дійсно

використовуються, але зазичай вони описують різні аспекти станів нечіткої системи, які були сформульовані вище. Розглянемо відповідний приклад.

Приклад 8.1. Мандрівник, якому після довгого шляху в пустелі дуже хочеться пити, знаходить дві пляшки із невідомою рідиною. Проблема полягає у тому, що мандрівник не знає, чи можна пити рідину із першої та другої пляшки. Знайти розв'язок цієї задачі з використанням методів теорії ймовірностей та нечіткої логіки.

Використаємо спочатку для розв'язування цієї задачі апарат теорії ймовірностей. Припустимо, що і в пляшці 1, і в пляшці 2 із імовірністю 0,91 міститься рідина, яку можна пити. Після аналізу цієї інформації мандрівник може подумати, що у пляшці 1 знаходиться рідина, яка не зовсім придатна для пиття, наприклад, болотна вода. У цьому разі для питної води коефіцієнт придатності для пиття дорівнював би 1. Навпаки, якщо у пляшці міститься отруйна рідина, наприклад, сіркова кислота, можна вважати, що придатність до пиття дорівнює 0. Перейдемо до статистичного визначення імовірності та частоти появи події. Згідно із цими міркуваннями, можна передбачити, що приблизно у 9 випадках з 10 мандрівник міг би напиться з будь-якої із пляшок. У цьому разі можна передбачити, що і в пляшці 1, і в пляшці 2 міститься чиста питна вода. Проте, що може статися в один з десяти випадків, є невідомим. Однак, якщо у пляшці знаходиться пиво або лимонад, зрозуміло, що з неї можна пити. Тому, згідно із імовірнісним аналізом, можна передбачити, що і у пляшці може бути непридатна для пиття рідина, наприклад сіркова кислота. Чи варто пити мандрівнику? Імовірнісний аналіз не дає відповіді на це питання.

Тепер припустимо, що з'являється додаткова інформація щодо рідини у пляшках. Наприклад, за визначених початкових умов стає відомим, що у пляшці 1 налитий лимонад, а у пляшці 2 – розчинений оцет. Тоді мандрівнику відразу стає зрозуміло, що краще пити з пляшки 1. У цьому випадку імовірнісні оцінки також змінюються. Якщо для пляшки 1 імовірність придатності води для пиття залишається майже незмінною, для пляшки 2 вона стає близькою до 0. У пошуках такої інформації і у зміні оцінок імовірності за різних умов і

полягає головна сутність методів аналізу, пов'язаних із нечіткою логікою. Розглянемо головні поняття та визначення цього розділу математики.

Нечіткі множини та функції належності

Нечіткою множиною (англійський термін – fuzzy set) називається сукупність елементів будь-якої природи, відносно яких неможна чітко стверджувати, чи належать вони заданій множині.

Зрозуміло, що визначення є неповним, оскільки в ньому не описується метод, відповідно з яким визначається, що заданий об'єкт належить або не належить до визначеної множини. Насправді, існує багато варіантів для визначення такої відповідності, але найбільш поширеним із них є математичне визначення нечіткої множини.

Нечіткою множиною A називається множина упорядкованих пар, які мають вигляд $\langle x, \mu_A(x) \rangle$, де x є елементом універсу X відповідної чіткої множини A , а $\mu_A(x)$ розглядається як функція належності.

Функцією належності $\mu_A(x)$ (англійський термін – membership function) називається функція, яка визначає відповідність між множиною X та дійсним числом із інтервалу $[0, 1]$, або у математичній формі:

$$\mu_A : X \rightarrow [0,1]. \quad (8.1)$$

Розглянемо головні властивості функції належності μ_A .

Властивість 8.1. Елемент x остаточно належить нечіткій множині X за умови $\mu_A = 1$.

Властивість 8.2. Елемент x остаточно не належить нечіткій множині X за умови $\mu_A = 0$.

Крім прийнятого позначення $\langle x, \mu_A(x) \rangle$ у математичній літературі використовують інші позначення нечіткої множини A , як-то:

$$\begin{aligned}
A &= \{(\mu_A(x_1), x_1), (\mu_A(x_2), x_2), \dots, (\mu_A(x_n), x_n)\}, \\
A &= \{(x_1/\mu_A(x_1)) + (x_2/\mu_A(x_2)) + \dots + (x_n/\mu_A(x_n))\}, \\
A &= \left\{ \frac{x_1}{\mu_A(x_1)} + \frac{x_2}{\mu_A(x_2)} + \dots + \frac{x_n}{\mu_A(x_n)} \right\}.
\end{aligned} \tag{8.2}$$

У наведених співвідношеннях (8.2) знак ділення означає роздільну риску між елементом та функцією належності, а знак «+» – об'єднання елементів множини.

Порожньою нечіткою множиною називається нечітка множина, яка не містить жодного елементу.

Універсом називається множина, яка містить в межах визначеного контенту всі відомі елементи.

Носієм нечіткої множини A називається звичайна множина A_s , яка містить ті і лише ті елементи, для яких значення функції належності є відмінні від нуля.

У математичній формі визначення для носія нечіткої множини можна записати наступним чином:

$$A_s = \{\forall x \in X \mid \mu_A(x) > 0\}. \tag{8.1}$$

Відповідно до кількості елементів розрізняють скінченні та нескінченні нечіткі множини.

Розрізняють два головні способи описання елементів нечіткої множини.

1. У вигляді списку з перерахуванням всіх елементів множини.
2. У вигляді математичного співвідношення для відповідної функції належності.

Розглянемо приклади визначення деяких нечітких множин.

Приклад 8.2. Побудувати нечітку множину, яка описує робочі та вихідні дні тижня, та визначити відповідну функцію належності.

Призначимо робочому дню коефіцієнт 0, а вихідному дню – коефіцієнт 1. Згідно із класичним визначенням чіткої множини можна вважати, що дні з понеділка до п'ятниці є робочими, а субота та неділя – вихідними. Характеристична функція для такої множини показана на рис. 8.2, а. Проте

зазвичай люди оцінюють робочі та вихідні дні не за календарем, а за емоційним забарвленням. Багато хто вважає, що п'ятниця вже напіввихідний день, бо робота закінчується, проте ввечері у неділю більшість людей вже сумують і готуються до наступного робочого тижня. Таку функцію належності можна визначити двома способами: для скінченної множини та для нескінченної множини. Наприклад, для скінченної множини будемо вважати, що дні тижня мають такі коефіцієнти: $A = \{ \langle \text{Пн.}, 0 \rangle, \langle \text{Вт.}, 0 \rangle, \langle \text{Ср.}, 0 \rangle, \langle \text{Чт.}, 0 \rangle, \langle \text{Пт.}, 0,5 \rangle, \langle \text{Сб.}, 1 \rangle, \langle \text{Вс.}, 0,8 \rangle \}$. Для нескінченної нечіткої множини ці значення можуть бути апроксимовані як часова залежність $\mu(x)$. Відповідні графічні залежності наведені на рис. 8.2, б, в.

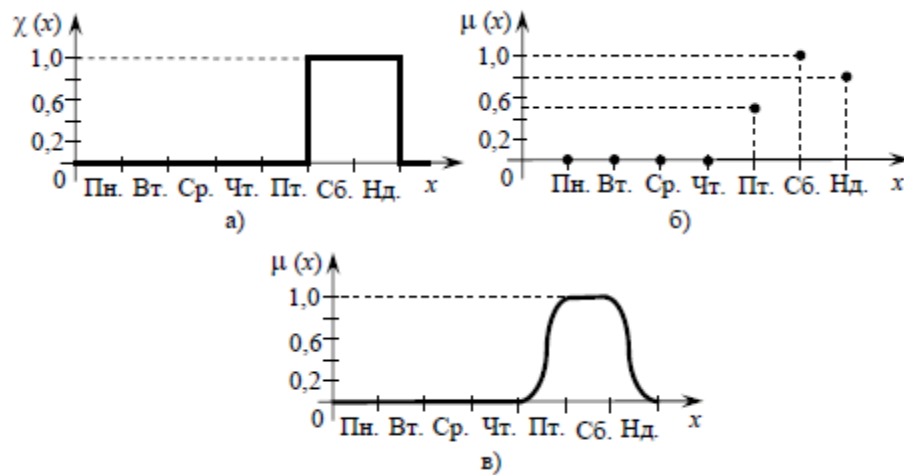


Рис. 8.2. Характеристична функція для звичайної множини днів тижня (а) та функція належності для відповідної скінченної (б) та нескінченної (в) множини

Із розглянутого простого прикладу зрозуміло, що для задач нечіткої логіки функція належності не може мати однозначного описання. Наприклад, значення цієї функції для елементів множини $\langle \text{Пт.}, 0,2 \rangle$ та $\langle \text{Вс.}, 0,9 \rangle$ також є можливими. З формальної точки зору існує лише єдина вимога: ці значення мають задовольняти визначенню нечіткої множини та визначенню функції належності.

Приклад 8.3. Побудувати нечітку множину для визначення усередненої температури кави. Тобто, необхідно з'ясувати, є кава гарячою чи холодною, та побудувати відповідну функцію належності.

Для даного прикладу можна вважати, що діапазон зміни температури кави становить від 0°C до 100°C . Тоді універсум нечіткої множини B можна записати наступним чином: $X = \{x | 0^{\circ}\text{C} < x < 100^{\circ}\text{C}\}$. Тепер необхідно визначити інтервали нульового та одиничного значення функції належності μ_B . Зазвичай, як і у попередній задачі, ці діапазони аргументу функції належності визначаються емпірично, із практичного досвіду. Для прикладу, який розглядається можна вважати, що:

$$\forall (x \leq 10^{\circ}\text{C}) \mu_B(x) = 0, \forall (x \geq 90^{\circ}\text{C}) \mu_B(x) = 1. \quad (8.2)$$

Тобто, якщо температура кави є меншою за 10°C , вона вважається холодною, а у випадку, коли її температура є вищою за 90°C – гарячою. Щодо діапазону температур, який лежить між цими граничними значеннями, тут неможна зробити чітких оцінок. Наприклад, каву, температура якої становить 55°C , одна людина може вважати гарячою, а інша – холодною. Саме у цьому і полягає нечіткість визначення заданої нескінченної множини температури кави X . Функція належності μ_B для множини X може бути як кусково-лінійною, так і нелінійною. Перевага кусково-лінійної функції полягає у простоті її аналітичного описання на лінійних відрізках, але для такої функції не існує похідної у точках згибу, що іноді ускладнює аналітичний аналіз під час розв'язування задач нечіткої логіки. Приклади визначення таких функцій для задачі, яка розглядається, наведені на рис. 8.3, а.

Із наведених залежностей видно, що функції належності дійсно пов'язані із теорією ймовірностей. Функцію належності можна характеризувати ймовірністю того, що властивості об'єкту за заданих умов відповідають його визначеній характеристиці. Наприклад, за мови $T = 30^{\circ}\text{C}$ каву однозначно не можна вважати гарячою, а за умови $T = 70^{\circ}\text{C}$ – однозначно можна. А якщо температура кави лежить в діапазоні від 30°C до 70°C , однозначного висновку зробити не можна, він залежить від власного смаку того, хто п'є каву. Але у будь-якому разі можна зробити висновок щодо характеру зміни функції

залежності. Зрозуміло, що для нечіткої множини «гаряча кава» ця функція є монотонно зростаючою від 0 до 1 в діапазоні зміни аргументу $[0^{\circ}\text{C}, 100^{\circ}\text{C}]$.

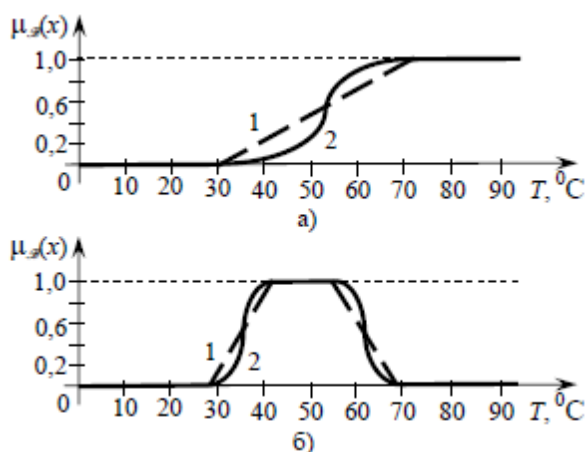


Рис. 8.3. Приклади визначення функції належності для гарячої (а) та для теплої (б) кави. 1 – кусково-лінійні функції, 2 – неперервні функції

Як було відмічено, спосіб побудови функції належності суттєво залежить від способу описання характеристик об'єкту. Наприклад, якщо необхідно визначити умови, за яких кава є теплою, функція належності буде мати інший характер. У цьому випадку в діапазоні температур $[0^{\circ}\text{C}, 30^{\circ}\text{C}]$ та $[70^{\circ}\text{C}, 100^{\circ}\text{C}]$ каву теплою однозначно вважати неможна, а в діапазоні температур $[40^{\circ}\text{C}, 60^{\circ}\text{C}]$ – однозначно можна. Відповідні графіки для функції належності наведені на рис. 8.3, б. Із наведених графіків зрозуміло, що характеристика «тепла кава» є невизначеною в діапазонах температур $[30^{\circ}\text{C}, 40^{\circ}\text{C}]$ та $[60^{\circ}\text{C}, 70^{\circ}\text{C}]$.

Розглянутий приклад є важливим з тієї точки зору, що запропоновану універсальну методику оцінки характеристик об'єкту можна розповсюдити і на інші ситуації, пов'язані із аналізом нечіткої інформації. Наприклад, такі характеристики об'єктів, як «швидкодійний прилад», «високонадійний прилад», «робота, яка добре оплачується», «швидкісний автомобіль», «коштовні прикраси», «охайна людина», «пухнаста кішка», «престижне житло», завжди є неточно визначеними, тому їх можна описати з використанням нечітких множин та функцій належності, які цим множинам відповідають.

Різновидом та узагальненням поняття нечіткої множини є нечітка множина α -рівня. Надамо відповідне визначення.

Нечіткою множиною α -рівня називається множина, для якої $\mu_A(x) \geq \alpha$, де α – деяке число з інтервалу $[0, 1]$.

Розглянемо приклади побудови нечітких множин для натуральних та раціональних чисел.

Приклад 8.4. Побудувати нечітку множину, яка визначена характеристикою «невелике натуральна число».

Задамо множину A , яка шукається, наступним чином:

$$A = \{ \langle 1, 1.0 \rangle, \langle 2, 1.0 \rangle, \langle 3, 0.9 \rangle, \langle 4, 0.8 \rangle, \langle 5, 0.6 \rangle, \langle 6, 0.5 \rangle, \langle 7, 0.4 \rangle, \langle 8, 0.2 \rangle, \langle 9, 0.1 \rangle \}. \quad (8.3)$$

Нечіткі множини α -рівня у математичній літературі часто описуються з використанням діаграм Венна. Діаграма Венна для множини A , заданої співвідношенням (8.3), наведена на рис. 8.4.

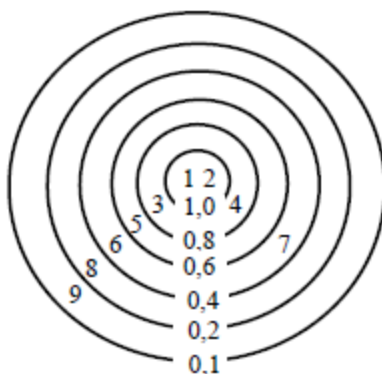


Рис. 8.4. Діаграма Венна для прикладу 8.3

Діаграми Венна для нечіткої множини відрізняються від діаграм Вейля тим, що діаграми Венна будуються як сукупність кількох кіл, які не перетинаються.

Максимальне значення функції належності називається *висотою* нечіткої множини.

У математичній формі висота нечіткої множини визначається наступним чином:

$$h_A = \sup(\mu_A(x)), \quad (8.4)$$

де $\sup(\mu_A(x))$ – функція супремума.

Властивість 8.3. Якщо задані нечітка множина A та два раціональних числа α_1 та α_2 , для яких виконується умова

$$\alpha_1 \in [0,1]; \alpha_2 \in [0,1]; \alpha_1 > \alpha_2,$$

тоді завжди є вірним співвідношення:

$$(\alpha_1 > \alpha_2) \Rightarrow (A_{\alpha_1} \subseteq A_{\alpha_2}), \quad (8.5)$$

де A_{α_1} – підмножина A рівня α_1 , A_{α_2} – підмножина A рівня α_2 .

Властивість 8.4. Наслідок властивості 8.3. Якщо визначена нечітка множина рівня 0, тоді множина рівня α є її підмножиною.

Розглянемо приклад визначення підмножини заданого рівня.

Приклад 8.5. Побудувати функцію належності для нечіткої множини A , яка визначена характеристикою «дійсне число, наближене до нуля», а також функцію належності підмножини $A_{0,5}$ множини A , яка відповідає рівню 0,5.

Для даного прикладу можна вважати, що функція належності $\mu_A(x) = 1$ за умови $x = 0$ і $\mu_A(x) = 0$ за умови $|x| \geq 1$. Якщо припустити, що як і для прикладу 8.2, функція належності є кусково-лінійною, така функція буде мати вигляд, наведений на рис. 8.5, а. На рис. 8.5, б, наведений графік цієї функції за для підмножини $A_{0,5}$. Зрозуміло, що у цьому випадку область визначення функції $\mu_A(x)$ є обмеженою і складає $[-0,5, 0,5]$.

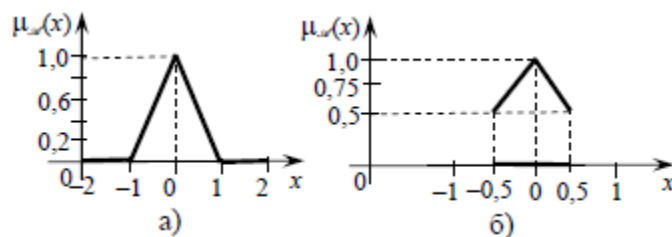


Рис. 8.5. Функції належності для прикладу 8.4 для множини A (а) та підмножини $A_{0,5}$ (б)

Приклад 8.6. Дослідити функцію $y(x) = \frac{x-1}{x}$ та побудувати її графік.

Визначити, за яких умов цю функцію можна вважати функцією належності для нескінченної нечіткої множини A , яка визначена характеристикою «велике дійсне число».

Дослідження функції $y(x) = \frac{x-1}{x}$ показують, що:

$$\lim_{x \rightarrow 0} y(x) = -\infty, \quad \lim_{x \rightarrow \infty} y(x) = 1, \quad y(1) = 0, \quad y'(x) = \frac{1}{x^2}.$$

Графіки функцій $y(x)$ та $y'(x)$ в діапазоні значень $x \in [0,5;10]$ наведені на рис. 8.6. Враховуючи те, що функція $y(x)$ в діапазоні значень $x \in [1;\infty]$ є монотонно зростаючою та змінює своє значення в діапазоні $[0;1]$, саме в цьому діапазоні значень x функцію $y(x)$ можна розглядати як функцію належності для визначеної заданої нечіткої множини A .

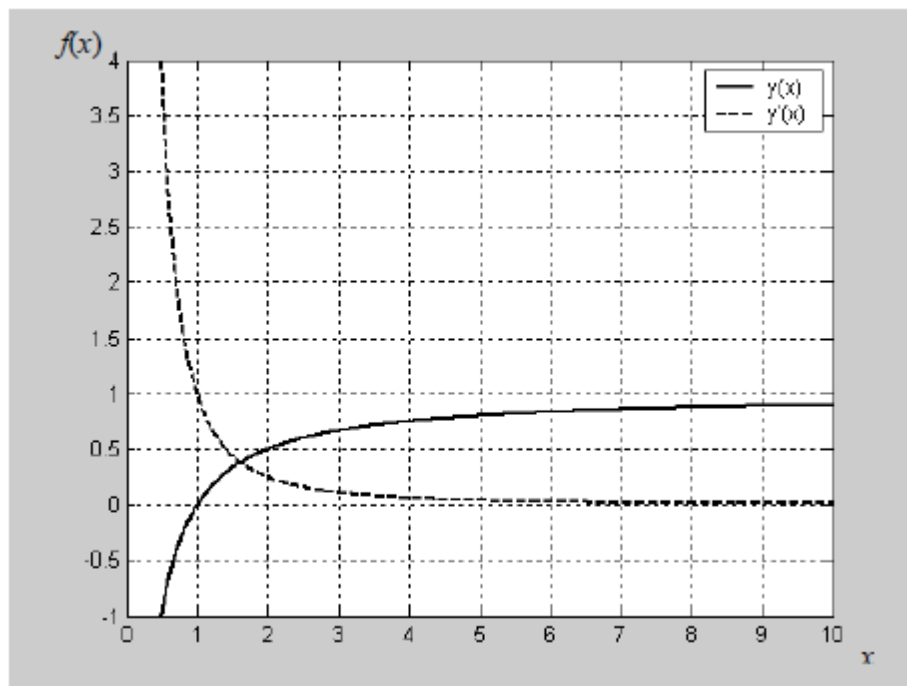


Рис. 8.6. Графіки функцій $y(x)$ та $y'(x)$ для прикладу 8.5

Надамо інші важливі поняття теорії нечіткої логіки, пов'язані із функцією належності.

Границями нечіткої множини називаються такі елементи універсума, для яких значення функції належності відрізняються від 0 та 1.

Ядром нечіткої множин називаються такі елементи універсума, для яких значення функції належності дорівнюють 1.

Носієм нечіткої множин називаються такі елементи універсума, для яких значення функції належності не дорівнюють 0.

Властивість 8.5. Носій нечіткої множини визначається як об'єднання її ядра із її границями.

Графічний спосіб визначення ядра, границь та носія для неперервних функцій належності, які були отримані у прикладі 8.2, показаний на рис. 8.7.

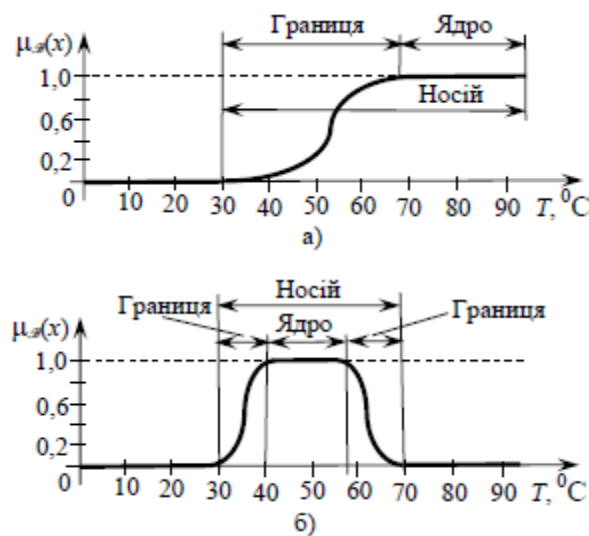


Рис. 8.7. Визначення ядра, границь та носія для функцій належності гарячої (а) та теплої (б) кави, які були розглянуті у прикладі 8.2

Контрольні запитання

1. Що являє собою теорія нечіткої логіки і яке вона має практичне значення для розвитку сучасної електроніки? Наведіть власні приклади використання теорії нечіткої логіки для формування алгоритмів роботи програмованих електронних пристроїв.

2. У чому полягає методологія нечіткого моделювання? Наведіть приклади використання цієї методології для формалізації задач нечіткої логіки.

3. Що являє собою нечітка множина як базове поняття теорії нечіткої логіки? Наведіть власні приклади нечітких множин.
4. Що являє собою функція належності нечіткої множини? Наведіть власні приклади формування функції належності.
5. Які властивості має функція належності?
6. Що являє собою порожня нечітка множина? Наведіть власні приклади визначення порожньої нечіткої множини.
7. Що являє собою універс нечіткої множини? Наведіть власні приклади визначення універсу нечіткої множини.
8. Що являє собою носій нечіткої множини? Наведіть власні приклади визначення носія нечіткої множини.
9. Які головні способи описання елементів нечіткої множини Вам відомі? Наведіть власні приклади таких описань.
10. Що являє собою висота нечіткої множини?
11. Що являють собою границі нечіткої множини? Наведіть власні приклади визначення границь нечіткої множини.
12. Що являє собою ядро нечіткої множини? Наведіть власні приклади визначення ядра нечіткої множини.

Перелік посилань

1. Zadeh, L. A. Fuzzy sets. – Information and Control, no. 8 (3), 1965, p. 338.
2. Zadeh, L. A. Fuzzy logic = computing with words. – IEEE Transactions on Fuzzy Systems, no. 4 (2), 1996, p. 103.
3. Zadeh, L. A. Fuzzy algorithms. – Information and Control, no. 12 (2), 1968, p. 94.
4. Потій О.В., Леншин А.В. Основні положення математичного апарату суб'єктивної логіки та його застосування для оцінки рівня зрілості систем забезпечення безпеки інформації. – Радіотехніка. Тематичний випуск “Інформаційна безпека”. Харків: Харківський національний університет радіоелектроніки, 2005. с. 144–160.

Лекція №9. Операції над нечіткими множинами та нечіткі відношення

Як і у класичній теорії множин, для нечітких множин визначаються відповідні алгебраїчні операції, які, разом із функціями належності, є теоретичним підґрунтям для формулювання та розв'язування задач нечіткої логіки. Розглянемо головні із цих операцій та надамо відповідні визначення [1-2].

Перетином двох нечітких множин A та B із однаковим універсумом X називається третя нечітка множина, яка визначається для кожного елемента x універсума X як мінімальне значення двох функцій належності $\mu_A(x_i)$ та $\mu_B(x_i)$, або у математичній формі:

$$A \cap B \Rightarrow \mu_E(x_i) = \min[\mu_A(x_i), \mu_B(x_i)], \forall x_i \in X. \quad (9.1)$$

Об'єднанням двох нечітких множин A та B із однаковим універсумом X називається третя нечітка множина, яка визначається для кожного елемента x універсума X як максимальне значення двох функцій належності $\mu_A(x_i)$ та $\mu_B(x_i)$, або у математичній формі:

$$A \cup B \Rightarrow \mu_E(x_i) = \max[\mu_A(x_i), \mu_B(x_i)], \forall x_i \in X. \quad (9.2)$$

Графічна інтерпретація визначення операцій перетину та об'єднання нечітких множин наведена на рис. 9.1.

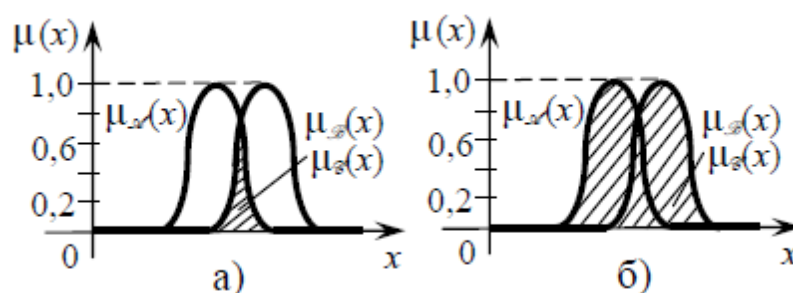


Рис. 9.1. Ілюстрація операцій перетину (а) та об'єднання (б) нечітких множин

Різницею двох нечітких множин A та B із однаковим універсумом X називається третя нечітка множина E , яка визначається для кожного елемента універсума x як максимальне значення різниці двох функцій належності $\mu_A(x_i)$ та $\mu_B(x_i)$ та нуля, або у математичній формі:

$$E = A \setminus B \Rightarrow \mu_E(x_i) = \max[\mu_A(x_i) - \mu_B(x_i), 0], \forall x_i \in X. \quad (9.3)$$

Наприклад, на рис. 9.2 наведений графік функції різниці двох нечітких множин, які описуються через функції залежності наступним чином:

$$\begin{aligned} f_{\Pi_1}(x; 1, 2, 4, 5) &= f_{S_1}(x; 1, 2) \cdot f_{Z_1}(x; 4, 5); \\ f_{\Pi_2}(x; 3, 5, 6, 8) &= f_{S_1}(x; 3, 5) \cdot f_{Z_1}(x; 6, 8); \\ \mu_A(x_i) &= f_{\Pi_1}(x); \mu_B(x_i) = f_{\Pi_2}(x); E = A \setminus B. \end{aligned} \quad (9.4)$$

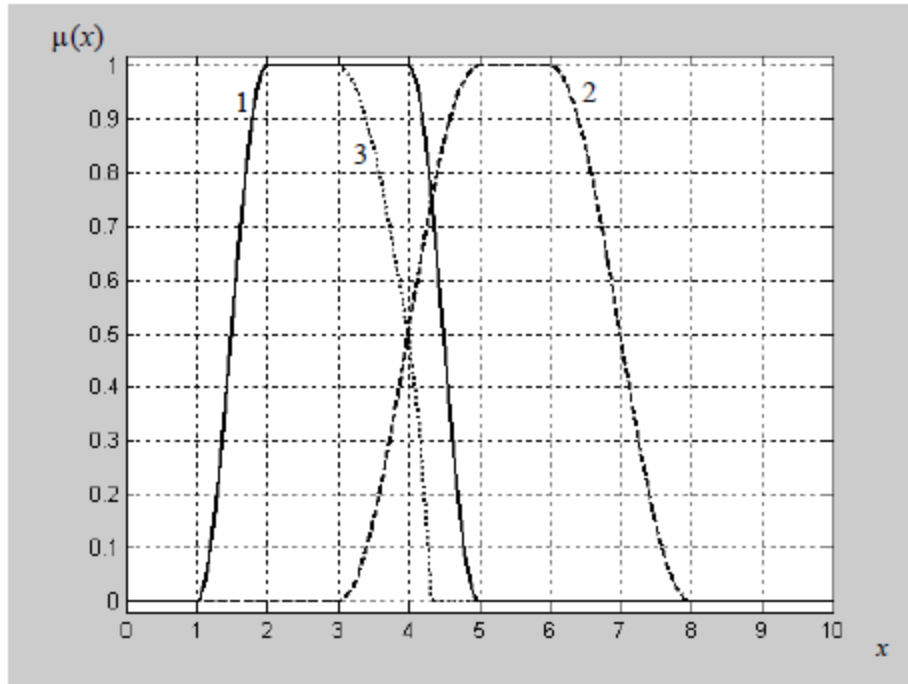


Рис. 9.2. Функція різниці нечітких множин, заданих співвідношеннями (9.4). 1 – $\mu_A(x)$; 2 – $\mu_B(x)$; 3 – $\mu_E(x)$

Слід відзначити, що, на відміну від перетину та об'єднання множин, операція різниці множин не є комутативною, тобто $A \setminus B \neq B \setminus A$. У зв'язку з цим у теорії нечіткої логіки введена також операція симетричної різниці множин.

Симетричною різницею двох нечітких множин A та B із однаковим універсумом X називається третя нечітка множина E , яка визначається для кожного елемента універсума x як модуль різниці двох функцій належності $\mu_A(x_i)$ та $\mu_B(x_i)$, або у математичній формі:

$$E = A \circ B \Rightarrow \mu_E(x_i) = \max|\mu_A(x_i) - \mu_B(x_i)|, \forall x_i \in X. \quad (9.5)$$

На рис. 9.3 наведений графік функції симетричної різниці двох нечітких множин, заданих співвідношеннями (9.4).

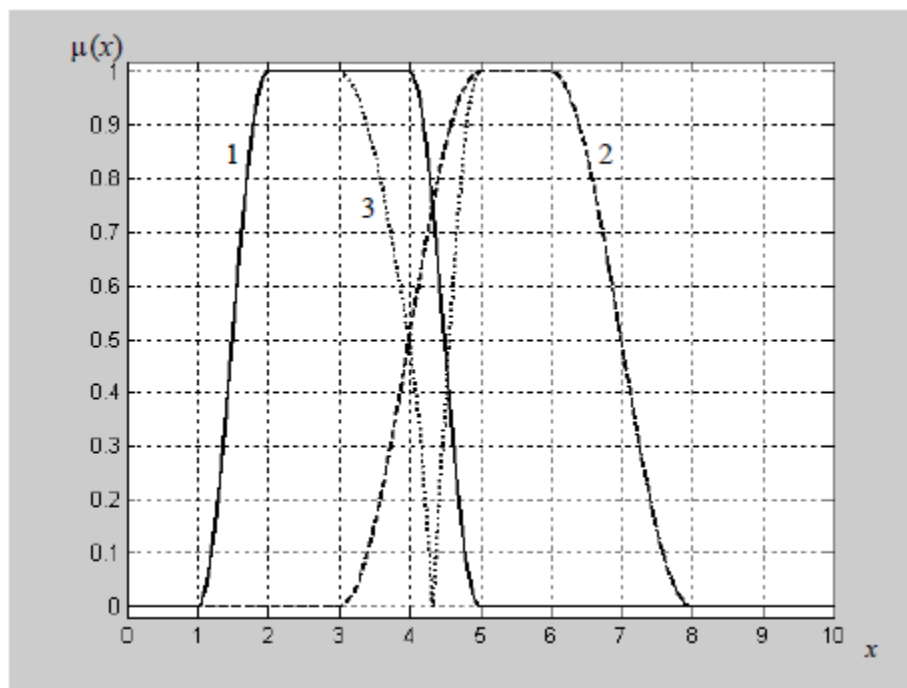


Рис. 9.3. Функція симетричної різниці нечітких множин, заданих співвідношеннями (9.4). 1 – $\mu_A(x)$; 2 – $\mu_B(x)$; 3 – $\mu_{E^c}(x)$, $E^c = A \circ B$

Крім описаних вище основних операцій над нечіткими множинами, у теорії нечіткої логіки визначені також відповідні альтернативні операції. Надамо відповідні визначення.

Алгебраїчним перетином, або *добутком* двох нечітких множин A та B із однаковим універсумом X називається третя нечітка множина D , яка визначається для кожного елемента універсума x як добуток двох функцій належності $\mu_A(x_i)$ та $\mu_B(x_i)$, або у математичній формі:

$$D = A \cdot B \Rightarrow \mu_D(x_i) = \mu_A(x_i) \cdot \mu_B(x_i), \forall x_i \in X. \quad (9.6)$$

Графік функції алгебраїчного перетину двох нечітких множин, заданих співвідношеннями (9.4), наведений на рис. 9.4.

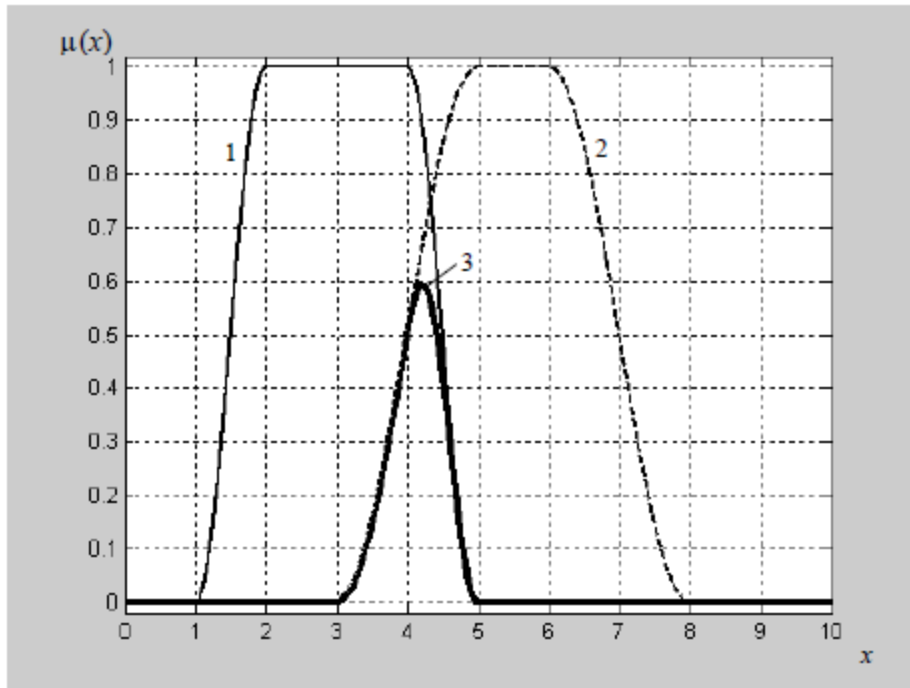


Рис. 9.4. Функція алгебраїчного перетину нечітких множин, заданих співвідношеннями (9.4). 1 – $\mu_A(x)$; 2 – $\mu_B(x)$; 3 – $\mu_D(x)$, $D = A \cdot B$

Алгебраїчним об'єднанням, або сумою двох нечітких множин A та B із однаковим універсумом X називається третя нечітка множина D , яка визначається для кожного елемента універсума x як сума двох функцій належності $\mu_A(x_i)$ та $\mu_B(x_i)$ із відніманням їхнього добутку, або у математичній формі:

$$D = A + B \Rightarrow \mu_D(x_i) = \mu_A(x_i) + \mu_B(x_i) - \mu_A(x_i) \cdot \mu_B(x_i), \forall x_i \in X. \quad (9.7)$$

Графік функції алгебраїчного об'єднання двох нечітких множин, заданих співвідношеннями (9.4), наведений на рис. 9.5.

Зрозуміло, що алгебраїчні операції перетину та об'єднання нечітких множин відповідають теоремам про імовірність сумісних та несумісних подій. Це також свідчить про тісний зв'язок теорії нечітких множин із теорією ймовірностей.

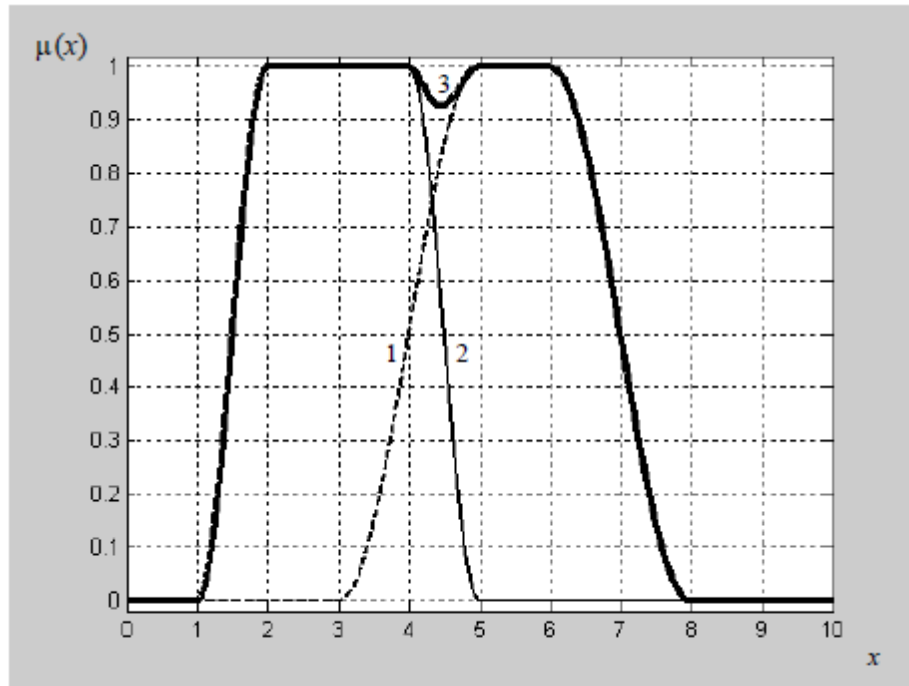


Рис. 9.5. Функція алгебраїчного об'єднання нечітких множин, заданих співвідношеннями (9.4). 1 – $\mu_A(x)$; 2 – $\mu_B(x)$; 3 – $\mu_D(x)$, $D = A + B$

Розглянуті операції суми та множення нечітких множин мають наступні властивості, які аналогічні властивостям звичайних множин.

Властивість 9.1. *Властивість комутативності.* Операції сумування та множення нечітких множин є комутативними:

$$A + B = B + A, A \cdot B = B \cdot A. \quad (9.8)$$

Властивість 9.2. *Властивість асоціативності.* Операції сумування та множення нечітких множин є асоціативними:

$$(A + B) + C = A + (B + C), (A \cdot B) \cdot C = A \cdot (B \cdot C). \quad (9.9)$$

Властивість 9.3. *Універсальні верхні та нижні границі нечіткої множини.* Порожня множина та універсум X є нижньою та верхньою границями для операції сумування та множення нечітких множин:

$$(A + \emptyset) = A, (A + X) = X, (A \cdot \emptyset) = \emptyset, (A \cdot X) = A.$$

Властивість 9.4. *Закони де Моргана.*

$$\overline{(A + B)} = \bar{A} + \bar{B}; \overline{(A \cdot B)} = \bar{A} \cdot \bar{B}. \quad (9.10)$$

Серед додаткових операцій нечіткої логіки, які широко застосовуються для побудови моделей складних систем, окремо розглядаються операції множення множини на число та піднесення множини до степені. Надамо відповідні визначення.

Для нечіткої множини $A = \{x, \mu_A(x)\}$, яка задана на універсумі X , та додатного дійсного числа a , для якого виконується умова $a \cdot h_A \leq 1$, операція множення нечіткої множини на число визначається на тому ж як універсумі X як добуток $a \cdot \mu_A$, або у математичній формі:

$$B = a \cdot A \Rightarrow B = \{x, \mu_B(x)\}, \mu_B(x) = a \cdot \mu_A(x). \quad (9.11)$$

Наприклад, для множини $A = \{<1, 1,0>, <3, 0,9>, <4, 0,8>, <9, 0,1>\}$ множина $B = a \cdot A$ за умови $a = 0,7$ визначається як $B = \{<1, 0,7>, <3, 0,63>, <4, 0,56>, <9, 0,07>\}$.

Для нечіткої множини $A = \{x, \mu_A(x)\}$, яка задана на універсумі X , та для додатного дійсного числа k , операція піднесення нечіткої множини до степені визначається на тому ж як універсумі X як піднесення до степені k функції належності $\mu_A(x)$, або у математичній формі:

$$B = A^k \Rightarrow B = \{x, \mu_B(x)\}, \mu_B(x) = (\mu_A(x))^k. \quad (9.12)$$

Графічні залежності для функції належності $\mu_A(x)$, заданої співвідношенням

$$\mu_A(x) = f_{\Pi}(x;1,4,5,9) = f_{S_1}(x;1,4) \cdot f_{Z_1}(x;5,9), \quad (9.13)$$

де функції $f_{S_1}(x;a,b)$ та $f_{Z_1}(x;c,d)$ визначаються співвідношеннями

$$f_{S_1}(x;a,b) = \begin{cases} 0, & x \leq a \\ \frac{1 + \cos\left(\frac{\pi(x-b)}{b-a}\right)}{2}, & a < x < b \\ 1, & x \geq b \end{cases} \quad \text{та}$$

$$f_{Z_1}(x; c, d) = \begin{cases} 0, & x \leq a \\ \frac{1 + \cos\left(\frac{\pi(x-b)}{b-a}\right)}{2}, & a < x < b, \\ 1, & x \geq b \end{cases}$$

а також для функцій $0,7 \cdot \mu_A(x)$,

$(\mu_A(x))^2$ та $(\mu_A(x))^{0,5}$, наведені на рис. 9.6.

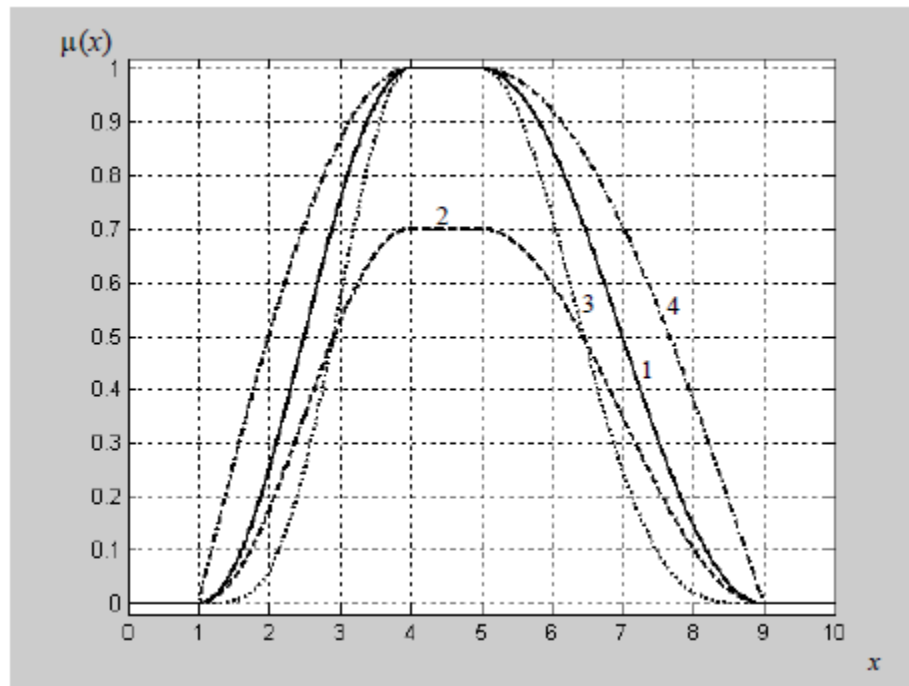


Рис. 9.6. Графічні залежності для функції належності $\mu_A(x)$, заданої співвідношенням (9.13) (1), а також для функцій $0,7 \cdot \mu_A(x)$ (2), $(\mu_A(x))^2$ (3) та $(\mu_A(x))^{0,5}$ (4)

На основі описаної вище операції піднесення множини до степені визначаються операції концентрування та розтягування нечітких множин. Надамо відповідні визначення.

Концентруванням нечіткої множини $A = \{x, \mu_A(x)\}$, яка задана на універсумі X , називається множина B , функція належності якої обчислюється як

$$\mu_B(x) = (\mu_A(x))^2. \quad (9.14)$$

Розтягуванням нечіткої множини $A = \{x, \mu_A(x)\}$, яка задана на універсумі X , називається множина B , функція належності якої обчислюється як

$$\mu_B(x) = (\mu_A(x))^{0,5}. \quad (9.15)$$

У математичній літературі функція концентрування позначається як $\text{CON}(A)$, а функція розтягування – як $\text{DIV}(A)$.

Практичне застосування операцій концентрування та розтягування функції належності тісно пов'язане із поняттями ентропія та кількості інформації про об'єкт. Зокрема, концентрування функції належності нечіткої множини означає зменшення характеристики її нечіткості і зазвичай причиною цього є надходження додаткової інформації про об'єкт моделювання. Навпаки, розтягування функції належності зазвичай є наслідком або часткової втрати інформації, або надходження інформації про додаткові впливові фактори, які не були початково враховані під час побудови моделі. Операції концентрування та розтягування нечіткої множині A , заданої співвідношенням (9.13), наочно показані на рис. 9.6.

Опуклою комбінацією нечітких множин $A_1, A_2, \dots, A_n$, називається нечітка множина D , функція належності якої визначається як лінійна комбінація функцій належності множин $A_1, A_2, \dots, A_n$, або у математичній формі:

$$\mu_D(x) = \lambda_1 \cdot \mu_{A_1}(x) + \lambda_2 \cdot \mu_{A_2}(x) + \dots + \lambda_n \cdot \mu_{A_n}(x). \quad (9.15)$$

Диз'юнктивною сумою двох нечітких множин A і B називається множина D , задана на тому ж універсумі X , функція належності якої $\mu_D(x)$ визначається через співвідношення:

$$\mu_D(x) = \max \{ \min \{ \mu_A(x), 1 - \mu_B(x) \}, \min \{ 1 - \mu_A(x), \mu_B(x) \} \}. \quad (9.16)$$

Результат операції диз'юнктивної суми функцій належності f_{Π_1} та f_{Π_2} , заданих співвідношеннями (9.4), показаний на рис. 9.7. Слід зазначити, що операція диз'юнктивної суми нечітких множин у практичних задачах використовується не часто.

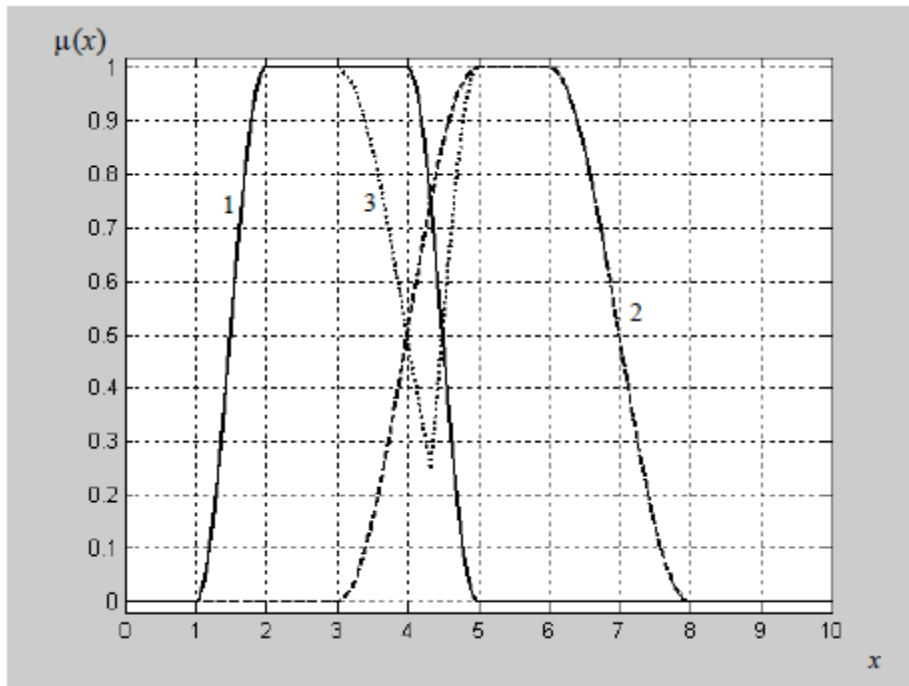


Рис. 9.7. Графіки функцій належності f_{Π_1} (1) та f_{Π_2} (2), заданих співвідношеннями (9.4), та результат операції диз'юнктивної суми цих двох функцій (3)

Нечіткими відношеннями називаються відношення між сукупностями елементів декількох нечітких множин, які визначаються як функції належності між окремими їхніми елементами.

Слід відзначити, що визначення дає найпростіше поняття нечіткого відношення, існують інші альтернативні визначення цього поняття. Наприклад, можна розглядати нечіткі відношення як добуток елементів нечітких множин в евклідовому просторі. Серед нечітких відношень окремо виділяють бінарні відношення, які найбільш широко використовуються для розв'язування практичних задач. Надамо відповідне визначення.

Бінарними нечіткими відношеннями називаються відношення між елементами двох множин.

Існують різні форми подання нечітких відношень, і найпростішою із них є форма списку із явним переліченням елементів множини. Припустимо, що задані дві множини із загальним універсумом $\{1, 2, 3\}$. У формі списку зв'язок між елементами цих множин записується наступним чином:

$$Q = \{(<1, 1>, \mu_{11}), (<1, 2>, \mu_{12}), (<1, 3>, \mu_{13}), (<2, 1>, \mu_{21}), (<2, 2>, \mu_{22}), (<2, 3>, \mu_{23}), (<3, 1>, \mu_{31}), (<3, 2>, \mu_{32}), (<3, 3>, \mu_{33})\}. \quad (9.17)$$

де μ_{ij} – значення функції відношення між елементами двох множин.

Часто вважається, що для коефіцієнтів μ_{ij} виконуються наступні співвідношення:

$$\mu_{ij} = 1, \forall i = j; \mu_{ij} < 1, \forall i \neq j. \quad (9.18)$$

Іншими важливими формами подання відношень між нечіткими множинами є наступні [3, 4].

1. Аналітична форма подання у вигляді математичного співвідношення.
2. Графічна форма подання.
3. Форма подання у вигляді матриці нечіткого співвідношення.
4. Форма подання у вигляді направленого графа.
5. Форма подання у вигляді таблиці відношень.

Розглянемо способи формування відношень між нечіткими множинами та зв'язок між різними формами їхнього подання на прикладі задачі пошуку несправності автомобіля.

Приклад 9.1. Побудувати систему нечітких відношень, яка описує зв'язок можливих несправностей автомобіля із зовнішніми проявами цих несправностей за наступних умов.

1. Множина причин несправності $X = \{x_1, x_2, x_3, x_4\}$, де: x_1 – несправність акумулятора; x_2 – несправність карбюратора; x_3 – погана якість бензину; x_4 – несправність системи запалювання.

2. Множина наслідків несправності $Y = \{y_1, y_2, y_3\}$, де: y_1 – не запускається двигун; y_2 – двигун працює нестабільно; y_3 – двигун працює не на повну потужність.

Припустимо, що у вигляді списку відношення між елементами множин X та Y задані наступним чином:

$$Q = \{(<x_1, y_1>, 1), (<x_1, y_2>, 0,9), (<x_1, y_3>, 0,2), (<x_2, y_1>, 0,8), (<x_2, y_2>, 0,9), (<x_2, y_3>, 1), (<x_3, y_1>, 0,7), (<x_3, y_2>, 0,8), (<x_3, y_3>, 0,5), (<x_4, y_1>, 1), (<x_4, y_2>, 0,5), (<x_4, y_3>, 0,2)\}. \quad (9.19)$$

Тоді таблична форма функції відношення, заданої у вигляді списку співвідношенням (9.19), буде мати наступний вигляд.

Таблиця 9.1. Таблична форма подання співвідношення (9.19)

Множина причин несправності	Множина наслідків несправності		
	y_1	y_2	y_3
x_1	1	0,1	0,2
x_2	0,8	0,9	1
x_3	0,7	0,8	0,5
x_4	1	0,5	0,2

У матричній формі співвідношення (9.19) записується наступним чином:

$$M = \begin{bmatrix} 1 & 0,1 & 0,2 \\ 0,8 & 0,9 & 1 \\ 0,7 & 0,8 & 0,5 \\ 1 & 0,5 & 0,2 \end{bmatrix}. \quad (9.20)$$

Зважений орієнтований граф, який відповідає співвідношенню (9.19), наведений на рис. 9.8.

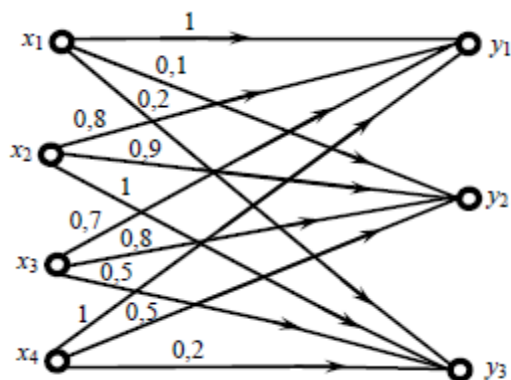


Рис. 9.8. Орієнтований граф, який описує нечіткі відношення для прикладу 9.1

Слід відзначити що для матриць відношень між елементами нечітких множин, аналогічних матриці (9.20), можна також використовувати матричні операції алгебри Буля.

Контрольні запитання

1. Що являє собою операція перетину двох нечітких множин? Наведіть власні приклади визначення перетину двох нечітких множин.

2. Що являє собою операція об'єднання двох нечітких множин? Наведіть власні приклади визначення об'єднання двох нечітких множин.

3. Що являє собою операція різниці двох нечітких множин? Наведіть власні приклади визначення різниці двох нечітких множин.

4. Що являє собою операція симетричної різниці двох нечітких множин? Наведіть власні приклади визначення симетричної різниці двох нечітких множин.

5. Що являє собою операція алгебраїчного перетину двох нечітких множин? Наведіть власні приклади визначення алгебраїчного перетину двох нечітких множин.

6. Що являє собою операція алгебраїчного об'єднання двох нечітких множин? Наведіть власні приклади визначення алгебраїчного об'єднання двох нечітких множин.

7. Які властивості операцій над нечіткими множинами Вам відомі?

8. Що являють собою алгебраїчні операції множення нечіткої множини на число та піднесення нечіткої множини до степені?

9. Що являє собою операція опуклої комбінації нечітких множин?

10. Що являє собою операція диз'юнктивної суми нечітких множин?

Перелік посилань

1. Zadeh, L. A. Fuzzy sets. – Information and Control, no. 8 (3), 1965, p. 338.
2. Zadeh, L. A. Fuzzy logic = computing with words. – IEEE Transactions on Fuzzy Systems, no. 4 (2), 1996, p. 103.
3. Zadeh, L. A. Fuzzy algorithms. – Information and Control, no. 12 (2), 1968, p. 94.
4. Потій О.В., Леншин А.В. Основні положення математичного апарату суб'єктивної логіки та його застосування для оцінки рівня зрілості систем забезпечення безпеки інформації. – Радіотехніка. Тематичний випуск “Інформаційна безпека”. Харків: Харківський національний університет радіоелектроніки, 2005. с. 144–160.

Лекція №10. Загальні положення нечіткої логіки та нечіткі висновки

Загалом теорія нечіткої логіки базується класичній математичній логіці, зокрема на алгебрі Буля та на теорії предикатів, але у значній мірі розширює та доповнює їх. Незважаючи на те, що класична логіка та теорія предикатів стали теоретичним підґрунтям для створення теорії скінченних автоматів та моделі обчислювального процесу на основі машини Тюрінга, все ж таки класична математична логіка є дещо обмеженою порівняно із логікою мислення людини. Ця обмеженість насамперед полягає в тому, що у класичній логіці не розглядається проблема невизначеності, яка завжди притаманна людському мисленню. Дійсно, коли людина мислить над серйозною проблемою, вона спочатку завжди сумнівається, міркує над нею, і лише потім, після серйозних міркувань, приймає остаточне рішення. Зрозуміло, що такий процес не притаманний класичній математичній логіці та теорії предикатів, де розглядається лише два можливих варіанти висловлення – істинний та хибний [1-3].

Елементарним нечітким висловленням називається висловлення, про яке попередньо, без додаткової інформації, не можна сказати, є воно істинним чи хибним.

Нечітким предикатом називається відображення простору універсумів висловлень $X_1, X_2, X_3, \dots, X_n$ на впорядковану множину значень істинності цих висловлень, яка є дійсним числом і належить інтервалу значень $[0, 1]$.

Декартовий добуток висловлень $X_1 \times X_2 \times X_3 \times \dots \times X_n$ називається їхньою *предметною областю*.

Слід відзначити, що нечіткі предикати цілком відповідають нечітким відношенням.

Припустимо, що U – непорожня множина нечітких висловлень, а відображення цих висловлень на множину $[0, 1]$ позначено як $T: U \rightarrow [0, 1]$.

Розглянемо основні логічні операції із нечіткими висловленнями.

Операцією *заперечення* нечіткого висловлення A називається логічна операція, результат якої обчислюється як різниця між одиницею та значенням степені істинності відповідного висловлення, тобто:

$$T(\bar{A}) = 1 - T(A). \quad (10.1)$$

Сутність співвідношення (10.1) полягає у тому, що якщо функція належності для висловлення A складає a , то для протилежного висловлення $\bar{A}$ вона буде складати $1 - a$. Наприклад, якщо висловлення «Тиск у вакуумній системі є високим» є істинним із степінню істинності 0,8, тоді протилежне висловлення «Тиск у вакуумній системі є низьким» є істинним із степінню істинності 0,2.

Операцією *кон'юнкції* двох нечітких логічних висловлень називається логічна операція, результат якої обчислюється як мінімальне із двох значень степені істинності цих висловлень, тобто:

$$T(A \wedge B) = \min\{T(A), T(B)\}. \quad (10.2)$$

У теорії нечіткої логіки існують альтернативні способи обчислення кон'юнкції двох нечітких висловлень, які називаються алгебраїчним добутком, граничним добутком та драстичним добутком.

Операцією *алгебраїчного добутку* двох нечітких логічних висловлень називається логічна операція, результат якої обчислюється як добуток значень степені істинності цих висловлень, тобто:

$$T(A \wedge B) = T(A) \cdot T(B). \quad (10.3)$$

Операцією *граничного добутку* двох нечітких логічних висловлень називається логічна операція, результат якої обчислюється як максимальне значення між різницею суми значень степені істинності цих висловлень та одиниці та нульовим значенням, тобто:

$$T(A \wedge B) = \max\{T(A) + T(B) - 1, 0\}. \quad (10.4)$$

Операцією *драстичного добутку* двох нечітких логічних висловлень називається логічна операція, результат якої дорівнює 1, якщо степінь

істинності хоча б одного із цих висловлень складає 1, або 0 в іншому випадку, тобто:

$$T(A \wedge B) = \begin{cases} T(A), T(A) = 1; \\ T(B), T(B) = 1; \\ 0, \text{ за інших умов.} \end{cases} \quad (10.5)$$

Розглянемо приклад обчислення операції диз'юнкції нечітких висловлень за формулами (10.2) – (10.5).

Приклад 10.1. З використанням співвідношень (10.2) – (10.5) різними способами знайти логічну кон'юнкцію для висловлення «Комунікаційний пристрій є надвисокочастотним і він нагрівається до високої температури», якщо степінь істинності висловлення «Комунікаційний пристрій є надвисокочастотним» складає 0,8, а степінь істинності висловлення «Комунікаційний пристрій нагрівається до високої температури» – 0,6.

За формулою (10.2) маємо:

$$T(A \wedge B) = \min \{T(A), T(B)\} = \min \{0,8, 0,6\} = 0,6.$$

Для алгебраїчного добутку, згідно із формулою (10.3), можна записати:

$$T(A \wedge B) = T(A) \cdot T(B) = 0,8 \cdot 0,6 = 0,48.$$

Відповідно, для граничного добутку:

$$T(A \wedge B) = \max \{T(A) + T(B) - 1, 0\} = \max \{0,8 + 0,6 - 1, 0\} = \max \{0,4 + 0\} = 0,4.$$

І в решті решт, зрозуміло, що драстичний добуток для двох заданих висловлень згідно із формулою (10.5) дорівнює 0.

Операцією *диз'юнкції* двох нечітких логічних висловлень називається логічна операція, результат якої обчислюється як максимальне із двох значень степені істинності цих висловлень, тобто:

$$T(A \vee B) = \max \{T(A), T(B)\}. \quad (10.6)$$

Як і для операції кон'юнкції, для операції диз'юнкції у теорії нечіткої логіки також існують інші способи обчислення, крім співвідношення (10.6), а саме операції алгебраїчної, граничної та драстичної суми. Надамо визначення цих операцій.

Операцією *алгебраїчної суми* двох нечітких логічних висловлень називається логічна операція, результат якої обчислюється як сума значень степені істинності цих висловлень без їхнього добутку, тобто:

$$T(A \vee B) = T(A) + T(B) - T(A) \cdot T(B). \quad (10.7)$$

Операцією *граничної суми* двох нечітких логічних висловлень називається логічна операція, результат якої обчислюється як мінімальне значення між сумою значень степені істинності цих висловлень та одиницею, тобто:

$$T(A \vee B) = \min\{T(A) + T(B), 1\}. \quad (10.8)$$

Операцією *драстичної суми* двох нечітких логічних висловлень називається логічна операція, результат якої дорівнює 0, якщо степінь істинності хоча б одного із цих висловлень складає 0, або 1 в іншому випадку, тобто:

$$T(A \vee B) = \begin{cases} T(A), & T(A) = 0; \\ T(B), & T(B) = 0; \\ 1, & \text{за інших умов.} \end{cases} \quad (10.9)$$

Розглянемо приклад обчислення операції диз'юнкції нечітких висловлень за формулами (10.6) – (10.9).

Приклад 10.2. З використанням співвідношень (10.6) – (10.9) різними способами знайти логічну диз'юнкцію для висловлення «Комунікаційний пристрій є надвисокочастотним або високонадійним», якщо степінь істинності висловлення «Комунікаційний пристрій є надвисокочастотним» складає 0,8, а степінь істинності висловлювання «Комунікаційний пристрій є високонадійним» – 0,9.

За формулою (10.6) маємо:

$$T(A \vee B) = \max\{T(A), T(B)\} = \max\{0,8, 0,9\} = 0,9.$$

Для алгебраїчної суми, згідно із формулою (10.7), можна записати:

$$T(A \vee B) = T(A) + T(B) - T(A) \cdot T(B) = 0,8 + 0,9 - 0,8 \cdot 0,9 = 0,98.$$

Відповідно, для граничної суми, згідно із співвідношенням (10.8):

$$T(A \vee B) = \min\{T(A) + T(B), 1\} = \min\{0,8 + 0,9, 1\} = 1$$

Зрозуміло також, що драстичний добуток для двох заданих висловлювань згідно із формулою (10.9) дорівнює 1.

Крім операцій кон'юнкції та диз'юнкції у нечіткій логіці розглядається операція імплікації, яка також має декілька альтернативних визначень.

Операція *імплікації* двох висловлень A та B у математичній літературі позначається як $T(A \supset B)$, цей математичний вираз читається як «Із A слідує B », або «Якщо вірно A , тоді вірно також B ». Існують наступні рівноцінні способи обчислення імплікації.

Операцією *класичної нечіткої імплікації*, або *нечіткою імплікацією за Заде*, називається величина, яка обчислюється через значення степені істинності висловлень $T(A)$ і $T(B)$ наступним чином:

$$T(A \supset B) = \max\{\min\{T(A), T(B), 1 - T(A)\}\}. \quad (10.10)$$

Операцією *класичної нечіткої імплікації* за умови $T(A) \geq T(B)$, або *імплікацією за Гьоделем*, називається величина, яка обчислюється через значення степені істинності висловлень $T(A)$ і $T(B)$ наступним чином:

$$T(A \supset B) = \max\{T(\bar{A}), T(B)\} = \max\{1 - T(A), T(B)\}. \quad (10.11)$$

Операцією *нечіткої імплікації мінімуму кореляції*, або *імплікацією за Мамдані*, називається величина, яка обчислюється через значення степені істинності висловлень $T(A)$ і $T(B)$ наступним чином:

$$T(A \supset B) = \min\{T(A), T(B)\}. \quad (10.12)$$

Слід відзначити, що формула для імплікації за Мамдані (10.12) є наслідком формули (10.10) за умови $T(A) \geq 0,5$, $T(B) \geq 0,5$.

Операцією *нечіткої імплікації за Лукашевичем* називається величина, яка обчислюється через значення степені істинності висловлень $T(A)$ і $T(B)$ наступним чином:

$$T(A \supset B) = \min\{1, 1 - T(A) + T(B)\}. \quad (10.13)$$

Операцією *нечіткої імплікації за Гогеном* називається величина, яка обчислюється через значення степені істинності висловлень $T(A)$ і $T(B)$ наступним чином:

$$T(A \supset B) = \min \left\{ 1, \frac{T(B)}{T(A)} \right\}. \quad (10.14)$$

Операцією *нечіткої імплікації за формулою граничної суми* називається величина, яка обчислюється через значення степені істинності висловлень $T(A)$ і $T(B)$ наступним чином:

$$T(A \supset B) = \min \{1, T(A) + T(B)\}. \quad (10.15)$$

Операцією *нечіткої імплікації за формулою добутку* називається величина, яка обчислюється через значення степені істинності висловлень $T(A)$ і $T(B)$ наступним чином:

$$T(A \supset B) = T(A) \cdot T(B). \quad (10.16)$$

Операцією *нечіткої імплікації за Ваді* називається величина, яка обчислюється через значення степені істинності висловлень $T(A)$ і $T(B)$ наступним чином:

$$T(A \supset B) = \max \{T(A) \cdot T(B), 1 - T(A)\}. \quad (10.17)$$

Операцією *нечіткої імплікації за Брауером* називається величина, яка обчислюється через значення степені істинності висловлень $T(A)$ і $T(B)$ наступним чином:

$$T(A \supset B) = \begin{cases} 1, & T(A) \leq T(B); \\ T(B), & \text{за інших умов.} \end{cases} \quad (10.18)$$

Операцією *нечіткої імплікації для стандартної RSEQ послідовності* називається величина, яка обчислюється через значення степені істинності висловлень $T(A)$ і $T(B)$ наступним чином:

$$T(A \supset B) = \begin{cases} 1, & T(A) \leq T(B); \\ 0, & \text{за інших умов.} \end{cases} \quad (10.19)$$

Наведені формули нечіткої імплікації є важливими у теорії нечіткої логіки для побудови логічних міркувань. Розглянемо відповідний приклад.

Приклад 10.3. З використанням співвідношень (10.10) – (10.19) різними способами знайти імплікацію для висловлення «Якщо комунікаційний пристрій є надвисокочастотним, тоді він є високонадійним», якщо степінь істинності висловлення «Комунікаційний пристрій є надвисокочастотним» складає 0,8, а степінь істинності висловлювання «Комунікаційний пристрій є високонадійним» – 0,4.

За формулою (10.10):

$$T(A \supset B) = \max \{ \min \{ T(A), T(B), 1 - T(A) \} \} = \max \{ \min \{ 0,8, 0,4, 1 - 0,8 \} \} = \max \{ 0,4, 0,2 \} = 0,4.$$

За формулою (10.11):

$$T(A \supset B) = \max \{ T(\bar{A}), T(B) \} = \max \{ 1 - T(A), T(B) \} = \max \{ 1 - 0,8, 0,4 \} = \max \{ 0,2, 0,4 \} = 0,4.$$

За формулою (10.12): $T(A \supset B) = \min \{ T(A), T(B) \} = \min \{ 0,8, 0,4 \} = 0,4.$

За формулою (10.13):

$$T(A \supset B) = \min \{ 1, 1 - T(A) + T(B) \} = \min \{ 1, 1 - 0,8 + 0,4 \} = \min \{ 1, 0,6 \} = 0,6.$$

За формулою (10.14): $T(A \supset B) = \min \left\{ 1, \frac{T(B)}{T(A)} \right\} = \min \left\{ 1, \frac{0,4}{0,8} \right\} = 0,5.$

За формулою (10.15): $T(A \supset B) = \min \{ 1, T(A) + T(B) \} = \min \{ 1, 0,8 + 0,4 \} = 1.$

За формулою (10.16): $T(A \supset B) = T(A) \cdot T(B) = 0,8 \cdot 0,4 = 0,32.$

За формулою (10.17):

$$T(A \supset B) = \max \{ T(A) \cdot T(B), 1 - T(A) \} = \max \{ 0,8 \cdot 0,4, 1 - 0,8 \} = \max \{ 0,32, 0,2 \} = 0,32.$$

За формулою (10.18):

$$T(A \supset B) = \begin{cases} 1, & T(A) \leq T(B) \\ T(B), & \text{за інших умов} \end{cases} \Rightarrow T(A \supset B) = T(B) = 0,4.$$

За формулою (10.19): $T(A \supset B) = \begin{cases} 1, & T(A) \leq T(B) \\ 0, & \text{за інших умов} \end{cases} \Rightarrow T(A \supset B) = 0.$

Нечіткою еквівалентністю називається логічна операція, результат якої описується нечітким логічним висловленням $T(A \equiv B)$, істинність якого визначається за формулою:

$$T(A \equiv B) = \min \left\{ \max \{T(\bar{A}), T(B)\}, \max \{T(A), T(\bar{B})\} \right\}. \quad (10.20)$$

Розглянемо приклад визначення еквівалентності висловлень.

Приклад 10.4. З використанням співвідношення (10.20) встановити степінь еквівалентності висловлень «Комунікаційний пристрій є надвисокочастотним» та «Комунікаційний пристрій є швидкодійним», якщо степінь істинності першого та другого висловлень складає 0,9.

Згідно із формулою (10.20) маємо:

$$\begin{aligned} T(A \equiv B) &= \min \left\{ \max \{T(\bar{A}), T(B)\}, \max \{T(A), T(\bar{B})\} \right\} = \\ &= \min \left\{ \max \{0, 1, 0, 9\}, \max \{0, 9, 0, 1\} \right\} = \min \{0, 9, 0, 9\} = 0, 9. \end{aligned}$$

Слід відзначити, що згідно із визначенням у нечіткій логіці еквівалентність висловлень не є чітко встановленим фактом, а оцінюється через істинність цих висловлень за шкалою $[0, 1]$.

Іншим важливим поняттям теорії нечіткої логіки є правила нечіткої продукції. Надамо відповідні визначення.

Правилом нечіткої продукції називається логічний вираз, який має вигляд:

$$(i): \quad Q; P; A \Rightarrow B; S; F; N, \quad (10.21)$$

де i – ім'я, або номер продукції, Q – сфера її застосування, P – умова застосування ядра продукції, $A \Rightarrow B$ – ядро продукції, у якому A – умова ядра, або антецедент, B – висновок ядра, або консеквент, S – метод визначення кількісного значення степені істинності висновку ядра, F – коефіцієнт визначеності нечіткої продукції, N – післяумова продукції.

Зазвичай логічний вираз для ядра нечіткої продукції (10.21) записується у вигляді правила «Якщо A , тоді B », або в англійському варіанті, згідно із правилами міжнародного стандарту IEC 1331-7 «IF A , THEN B ».

Складність формування тверджень нечіткої логіки з використанням продукцій полягає в тому, що необхідно на основі початкових нечітких висловлень, або умов, визначити степінь істинності висновків. Взаємозв'язок між умовою та висновком визначається через відношення між нечіткими множинами A та B , де A – множина умов, задана на універсумі X , а B – множина висновків, задана на універсумі Y . Тоді відношення між цими множинами визначається як $Q\{<x, y>, \mu_Q(<x, y>)\}$. Якщо відома функція належності $\mu_A(<x>)$ функція нечіткої композиції $\mu_Q(<x, y>)$, тоді існують відповідні методи для обчислення функції належності множини висновків $\mu_B(<y>)$.

1. Мах-мін-композиція, або максимінна нечітка згортка:

$$\mu_B(y) = \max_{x \in X} \left\{ \min \left\{ \mu_B(y), \mu_Q(x, y) \right\} \right\}. \quad (10.22)$$

2. Мін-макс-композиція:

$$\mu_B(y) = \min_{x \in X} \left\{ \max \left\{ \mu_B(y), \mu_Q(x, y) \right\} \right\}. \quad (10.23)$$

3. Мах-макс-композиція:

$$\mu_B(y) = \max_{x \in X} \left\{ \max \left\{ \mu_B(y), \mu_Q(x, y) \right\} \right\}. \quad (10.24)$$

4. Мін-мін-композиція:

$$\mu_B(y) = \min_{x \in X} \left\{ \min \left\{ \mu_B(y), \mu_Q(x, y) \right\} \right\}. \quad (10.25)$$

5. Мах-average-композиція:

$$\mu_B(y) = \max_{x \in X} \left\{ \mu_B(y), \mu_Q(x, y) \right\}. \quad (10.26)$$

6. Sum-prod-композиція:

$$\mu_B(y) = f \left(\sum_{x \in X} \left(\mu_B(y) \cdot \mu_Q(x, y) \right) \right). \quad (10.27)$$

де f – S -функція, аналогічна функціям, заданим співвідношеннями

$$f_{S/I}(x, a, b) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b, \\ 1, & x > b \end{cases}$$

$$F_{S/I}(x, a, b) = 0 \cdot (x \leq a) + \frac{x-a}{b-a} \cdot (a < x \leq b) + 1 \cdot (x > b) \text{ із областю значень } [0, 1].$$

Взагалі у теорії нечіткої логіки розрізняють наступні три типи висловлень [1].

1. Висловлення « $\beta \in \alpha$ », де β – лінгвістична змінна, α – значення цієї змінної. Значення α відповідає терм-множині T , до якої належить змінна β .

2. Висловлення « $\beta \in \Delta \alpha$ », де Δ – модифікатор висловлення, який можна замінити словами: «ДУЖЕ», «НАБАГАТО БІЛЬШЕ», «БІЛЬШ-МЕНШ» або іншими словами та словосполученнями, які входять до процедур G та M даної лінгвістичної змінної.

3. Складні висловлювання, які формуються із висловлювань першого та другого типу з використанням операторів заперечення, кон'юнкції, диз'юнкції та імплікації, описаних вище.

Наведемо приклади таких висловлювань.

1. «Комунікаційний пристрій є надвисокочастотним та високонадійним».

2. «Комунікаційний пристрій є не надвисокочастотним, але високовольтним».

3. «Комунікаційний пристрій є високонадійним, але не надвисокочастотним».

У разі використання модифікаторів часто застосовують операції концентрування та розтягування функції належності нечіткої множини через піднесення до степені згідно із співвідношенням (9.12). Наприклад, якщо функція належності $\mu(x)$ описує лінгвістичну змінну «СЕРЕДНЯ ШВИДКІСТЬ АВТОМОБІЛЯ», тоді змінну «ШВИДКІСТЬ АВТОМОБІЛЯ, ЯКА НАБЛИЖАЄТЬСЯ ДО СЕРЕДНЬОЇ» можна описати функцією $(\mu(x))^{0.5}$, а

змінну «ШВИДКІСТЬ АВТОМОБІЛЯ, ЯКА ДУЖЕ БЛИЗЬКА ДО СЕРЕДНЬОЇ» – функцією $(\mu(x))^2$.

У разі формування продукцій нечіткої логіки третього типу використовують операції об'єднання та перетину нечітких множин. Наприклад, якщо функції залежності $\mu_1(x)$, $\mu_2(x)$ та $\mu_3(x)$ для лінгвістичних змінних $\alpha_1 = \text{«ШВИДКІСТЬ АВТОМОБІЛЯ Є НИЗЬКОЮ»}$, $\alpha_2 = \text{«ШВИДКІСТЬ АВТОМОБІЛЯ Є СЕРЕДНЬОЮ»}$ та $\alpha_3 = \text{«ШВИДКІСТЬ АВТОМОБІЛЯ Є ВИСОКОЮ»}$, тоді функціям належності $\mu_4(x)$ для змінної $\alpha_4 = \text{«ШВИДКІСТЬ АВТОМОБІЛЯ Є ВИСОКОЮ ТА СЕРЕДНЬОЮ»}$ та $\mu_5(x)$ для змінної $\alpha_5 = \text{«ШВИДКІСТЬ АВТОМОБІЛЯ Є ВИСОКОЮ АБО СЕРЕДНЬОЮ»}$ відповідають графіки, які наведені на рис. 10.1.

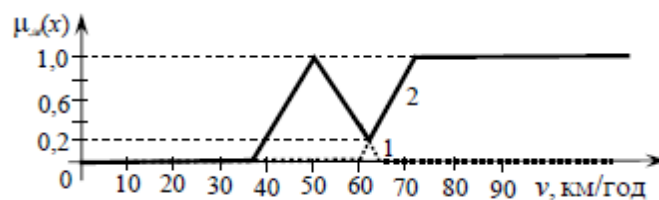


Рис. 10.1. Типові лінійні функції належності для нечітких змінних α_4 (1), та α_5 (2)

На графіках функцій $\mu_4(x)$ та $\mu_5(x)$, наведених на рис. 10.1, для лінійних функцій належності $\mu_2(x)$ та $\mu_3(x)$ використані операції кон'юнкції нечітких логічних висловлювань за мінімумом (10.2) та диз'юнкції за максимумом (10.6). Зрозуміло, що у разі використання інших форм подання кон'юнкції та диз'юнкції, буде отриманий близький, але дещо інший результат.

Основними етапами алгоритмів формування нечітких висновків у експертних електронних системах є наступні.

1. Формування бази нечітких правил згідно із теорією продукцій.
2. Фаззифікація вхідних лінгвістичних змінних.
3. Агрегування нечітких умов.
4. Активізація проміжних нечітких висновків.

5. Акумуляція нечітких висновків.

Розглянемо ці процедури теорії нечіткої логіки окремо та надамо відповідні визначення.

Фаззифікацією нечіткої змінної (англійський термін – fuzzyfication) називається процес визначення значень функції належності для вхідних параметрів системи, яка моделюється.

Фаззифікація є дуже важливим процесом початкового етапу аналізу нечітких систем, оскільки саме із цим процесом пов'язаний перехід від конкретних значень параметрів системи, яка моделюється, до правил нечіткої логіки, які описують поведінку системи, і такий перехід здійснюється саме через множину функцій належності. Наприклад, розглянемо нечітку систему контролю швидкості автомобіля, функції належності для якої наведені на рис. 10.2, і будемо вважати, що автомобіль рухається із швидкістю 40 км/год.

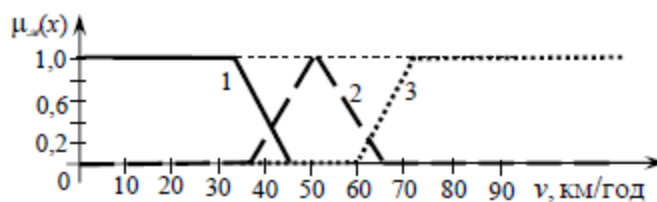


Рис. 10.2. Функція належності нечіткої системи контролю швидкості автомобіля

Процес фаззифікації швидкості автомобіля для заданого значення швидкості наочно показаний на рис. 10.3.

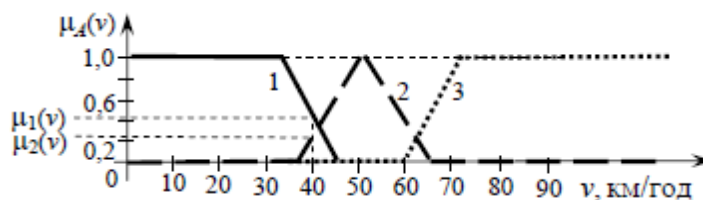


Рис. 10.3. Фаззифікація змінної α = «ШВИДКІСТЬ АВТОМОБІЛЯ» для значення $v = 40$ км/год

Із наведених на рис. 10.3 графічних залежностей зрозуміло, що за умови визначеного простору лінгвістичних змінних α_1 , α_2 та α_3 у разі, якщо швидкість складає $v = 40$ км/год, значення функцій належності становлять: $\mu_1(v) \approx 0,41$, $\mu_2(v) \approx 0,21$, $\mu_3(v) = 0$. Якщо такий результат фаззифікації змінної описати через модифікатор Δ , таку швидкість можна характеризувати як «НЕВИСОКА, БЛИЗЬКА ДО СЕРЕДНЬОЇ».

Агрегуванням нечіткої змінної (англійський термін – aggregation) називається процедура визначення степені істинності умов за кожним із правил системи нечіткого висновку.

Для агрегування нечітких даних використовуються правила нечіткої логіки для заперечення, кон'юнкції, диз'юнкції та імплікації, які описуються наведеними вище визначеннями та співвідношеннями (10.1) – (10.19).

Зазвичай агрегування проводиться для розглянутих вище умов третього типу, які під час моделювання реальних систем є найбільш поширеними та складними.

Розглянемо приклад агрегування нечітких висловлень за умови відомих функцій належності.

Приклад 10.5. Відомо, що комунікаційний пристрій працює на частоті $f = 3$ ГГц. Знайти функції належності $\mu_3(x)$ для нечіткого висловлення «Комунікаційний пристрій є надвисокочастотним та високонадійним» та $\mu_4(x)$ для нечіткого висловлення «Комунікаційний пристрій є надвисокочастотним або високонадійним», якщо функція належності $\mu_1(x)$ для висловлення «Комунікаційний пристрій є надвисокочастотним» описується

співвідношенням
$$f_{S_1}(x; a, b) = \begin{cases} 0, & x \leq a \\ \frac{1 + \cos\left(\frac{\pi(x-b)}{b-a}\right)}{2}, & a < x < b \\ 1, & x \geq b \end{cases} \quad \text{з параметрами}$$

$a = 1$ ГГц та $b = 10$ ГГц, а функція належності для висловлення

«Комунікаційний пристрій є високонадійним» – співвідношенням

$$f_{S_I}(x, a, b) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \text{ з такими ж параметрами } a \text{ та } b. \\ 1, & x > b \end{cases}$$

Для розв'язування поставленої задачі скористаємось програмами для обчислення функцій належності, написаних мовою програмування системи MatLab. Нижче наведені командні рядки системи, призначені для виклику цих функцій та для побудови відповідних графіків.

```
>> x=0:0.01:10;
>> c=fz1(x,1,10);
>> s=fs1(x,1,10);
>> andcs=min(c,s);
>> orcs=max(c,s);
>> andcsalg=c.*s;
>> orcsalg=c+s-andcsalg;
>> subplot(1,1,2);
>> subplot(1,2,1);
>> plot(x,c)
>> hold on
>> plot(x,s)
>> grid on
>> subplot(1,2,2);
>> plot(x,andcs);
>> hold on
>> plot(x,orcs);
>> plot(x,andcsalg)
>> plot(x,orcsalg);
>> grid on
```

Отримані результати розрахунків значень функцій належності $\mu_1(f) = f_{S1}(f)$ та $\mu_2(f) = f_{Z1}(f)$ наведені на рис. 10.4, а, а результати кон'юнкції та диз'юнкції цих функцій – на рис. 10.4, б.

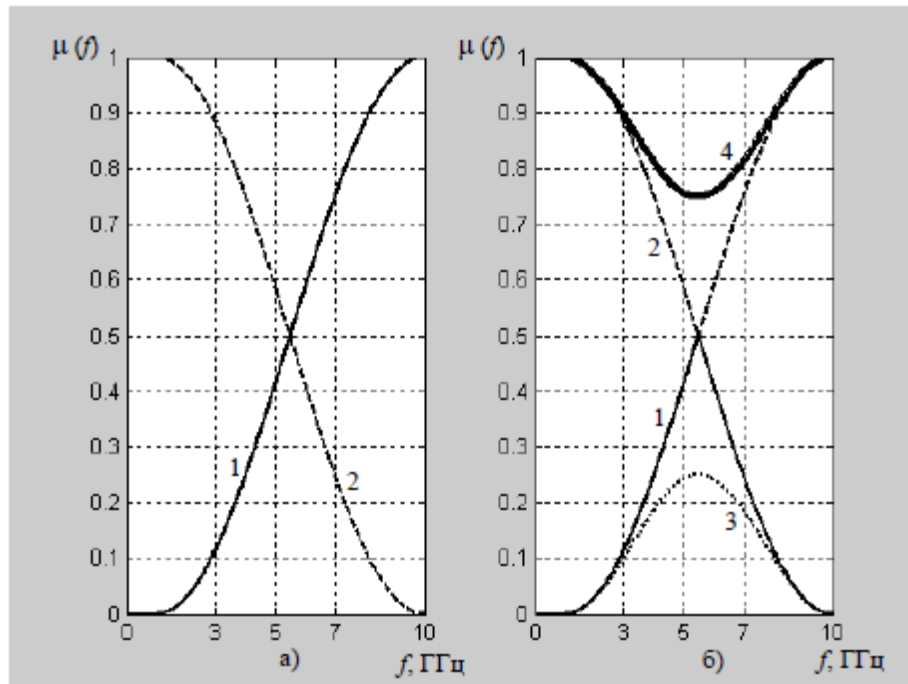


Рис. 10.4. Залежності $\mu_1(f) = f_{S1}(f)$ (1) та $\mu_2(f) = f_{Z1}(f)$ (2) (а), а також операцій кон'юнкції (1), диз'юнкції (2), алгебраїчного добутку (3) та алгебраїчної суми (4) цих функцій (б)

Зрозуміло, що згідно із наведеними вище визначеннями кон'юнкція та диз'юнкція функцій належності $\mu_1(f)$ та $\mu_2(f)$ можуть бути розраховані різними способами. У даному прикладі кон'юнкція розраховувалась класичним способом за формулою (10.2) та як алгебраїчний добуток за формулою (10.3), а диз'юнкція – класичним способом за формулою (10.6) та через алгебраїчну суму за формулою (10.7). Із отриманих результатів розрахунків видно, що за умови частоти роботи пристрою 3 ГГц його із степінню істинності $\mu_1(f) \approx 0,1$ можна вважати високочастотним, а із степінню істинності $\mu_2(f) \approx 0,9$ – високонадійним. Тому результатом кон'юнкції цих висловлень є мінімальна із цих величин $\mu_3(f) = 0,1$, а результатом диз'юнкції – $\mu_4(f) = 0,9$. Слід також відзначити, що на частоті $f = 3$ ГГц результат класичної операції кон'юнкції майже не відрізняється від результату операції алгебраїчного добутку, а результат операції диз'юнкції – від результату операції алгебраїчної суми. Тобто, результати проведених розрахунків показують, що за заданих умов роботи комунікаційний пристрій майже достовірно можна вважати високонадійним, проте наврядчи його можна вважати надвисокочастотним.

Приклад 10.6. За заданих у попередньому прикладі умов з використанням співвідношення (10.12) та (10.16) знайти степінь істинності висловлення: «Якщо пристрій є надвисокочастотним, тоді він є високонадійним».

Зрозуміло, що формула (10.12) для імплікації двох висловлювань відповідає формулі (10.2) для кон'юнкції двох функцій належності, а формула (10.16) – формулі (10.3) для їхнього алгебраїчного добутку. Тому легко зрозуміти, що розв'язку поставленої задачі відповідають криві 1 та 2, які наведені на рис. 10.4, б.

Активацією нечіткої змінної (англійський термін – activation) називається процедура або процес визначення степені істинності кожного із висновків правил нечіткої продукції.

Загалом процес активації є аналогічним процесу композиції нечітких відношень, який був розглянутий вище та виконується з використанням співвідношень (10.22) – (10.27), але оскільки у системі нечітких висновків, яка аналізується, введені лінгвістичні змінні, процес активації ускладнюється і співвідношення (10.22) – (10.27) втрачають своє значення. З іншого боку, як видно із співвідношення (10.21), кожне правило нечіткої логіки містить ваговий коефіцієнт F_i , який завжди лежить в інтервалі $[0,1]$ і за замовченням дорівнює 1. З урахуванням цього процедура активації правил, яка є найважливішою у системах моделювання, пов'язаних із нечіткою логікою, виконується наступним чином. До початку етапу активації вважаються відомими всі значення степені істинності умов для кожного правила, задані через функцію належності змінних $\mu_B(b_i)$. Далі необхідно розглянути кожне правило нечітких висновків. Якщо висновок формується з використанням висловлень першого або другого типу, тобто не є складним і не використовує правила кон'юнкції або диз'юнкції лінгвістичних змінних, тоді степені істинності початкової умови c_i обчислюється просто як добуток

$$c_i = \mu_B(b_i) \cdot F_i. \quad (10.28)$$

У випадку складних правил нечіткої логіки, коли вони формуються з використанням висловлень третього типу, необхідно спочатку згідно із співвідношенням (10.28) обчислити коефіцієнти c_i для кожної елементарної умови, а потім використати для формування остаточного значення степені істинності висловлення описаний вище процес агрегування змінних. Після цього, за умови відомих значень степенів істинності початкових нечітких тверджень c_i , функція належності $\mu'(y)$ для вихідної змінної y за умови відомої функції належності $\mu(y)$ формується за одним із наступних правил нечіткої логіки, які аналогічні формулам (10.22) – (10.27).

1. Правило мінімальної активації, або min-активація:

$$\mu'(y) = \min \{c_i, \mu(y)\}. \quad (10.29)$$

2. Правило продукції, або правило множення, prod-активація:

$$\mu'(y) = c_i \cdot \mu(y). \quad (10.30)$$

3. Правило усереднення, average-активація:

$$\mu'(y) = \frac{c_i + \mu(y)}{2}. \quad (10.31)$$

Коли згідно з одним із сукупності співвідношень (10.29) – (10.31) визначені всі лінгвістичні змінні, які входять до правил нечіткої продукції, процес активації вважається завершеним.

Розглянемо приклад активації вихідної змінної за визначеним правилом нечіткої продукції.

Приклад 10.7. Активувати вихідну змінну y для правила $b_i =$ «Якщо сигнал готовності до роботи надходить від електронного пристрою частіше, ніж кожні п'ять секунд, але не частіше, ніж кожні дві секунди, такий пристрій можна вважати справним» за наступних умов.

1. Степінь істинності початкової умови $x =$ «Сигнал готовності до роботи надходить частіше, ніж кожні п'ять секунд, але не частіше, ніж кожні дві секунди» задається П-функцією залежності $f_{S_1}(x, a, b) \cdot f_{Z_1}(x, c, d)$ з параметрами $a = 2$ с, $b = 2,5$ с, $c = 4,5$ с, $d = 5$ с.

2. Ваговий коефіцієнт F_i для правила b_i складає 0,9.

3. Степінь істинності висновку $y = \text{«Пристрій можна вважати справним»}$

задається П-функцією залежності $f_{\Pi_2}(x, a, b, c) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}}$ з параметрами

$a = 2$ с, $b = 3,5$ с, $c = 5$ с.

Числові розрахунки провести за формулою prod-активації (10.30) для значень $t = 2,1$ с, $t = 3,5$ с та $t = 4,8$ с.

Задачу цього прикладу, як і попереднього, зручно розв'язувати з використанням засобів програмування системи MatLab. Нижчі наведені відповідні командні рядки, а отримані результати розрахунків у графічному вигляді – на рис. 10.5.

```
>> x=1.5:0.001:7;
>> c=0.9.*fs1(x,2,2.5).*fz1(x,4.5,5);
>> plot(x,c)
>> c3=c.*fp2(x,2,3.5,5);
>> hold on
>> plot(x,c3)
>> c4=min(c,fp2(x,2,3.5,5));
271
>> plot(x,c4)
>> grid on
>> fc21=0.9.*fs1(2.1,2,2.5).*fz1(2.1,4.5,5);
>> fc35=0.9.*fs1(3.5,2,2.5).*fz1(3.5,4.5,5);
>> fc48=0.9.*fs1(4.8,2,2.5).*fz1(4.8,4.5,5);
>> ffc21_prod=fc21.*fp2(2.1,2,3.5,5)
ffc21_prod =
0.0059
>> ffc21_min=min(fc21,fp2(2.1,2,3.5,5))
ffc21_min =
0.0691
>> ffc35_prod=fc35.*fp2(3.5,2,3.5,5)
ffc35_prod =
0.7940
>> ffc35_min=min(fc35,fp2(3.5,2,3.5,5))
ffc35_min =
0.8822
>> ffc48_prod=fc48.*fp2(4.8,2,3.5,5)
ffc48_prod =
0.3109
>> ffc48_min=min(fc48,fp2(4.8,2,3.5,5))
```

$$\text{ffc48_min} = 0.3109$$

Із наведених командних рядків видно, що значення $\mu'_{\min}(y)$, обчислені за співвідношенням (10.29), складають $\mu'_{\min}(2,1) \approx 0,0691$, $\mu'_{\min}(3,4) \approx 0,8822$ та $\mu'_{\min}(4,8) \approx 0,3109$. Значення $\mu'_{\text{prod}}(y)$, обчислені за співвідношенням (10.29), складають $\mu'_{\text{prod}}(2,1) = 0,0059$, $\mu'_{\text{prod}}(3,5) = 0,7940$ та $\mu'_{\text{prod}}(4,8) = 0,3109$.

Зрозуміло, що за визначених умов найбільша степінь істинності того, що прилад є справним відповідає значенню $\mu'(3,5)$, а спосіб обчислення цього параметру не має суттєвого значення. Такий же результат дають графічні залежності, наведені на рис. 10.5.

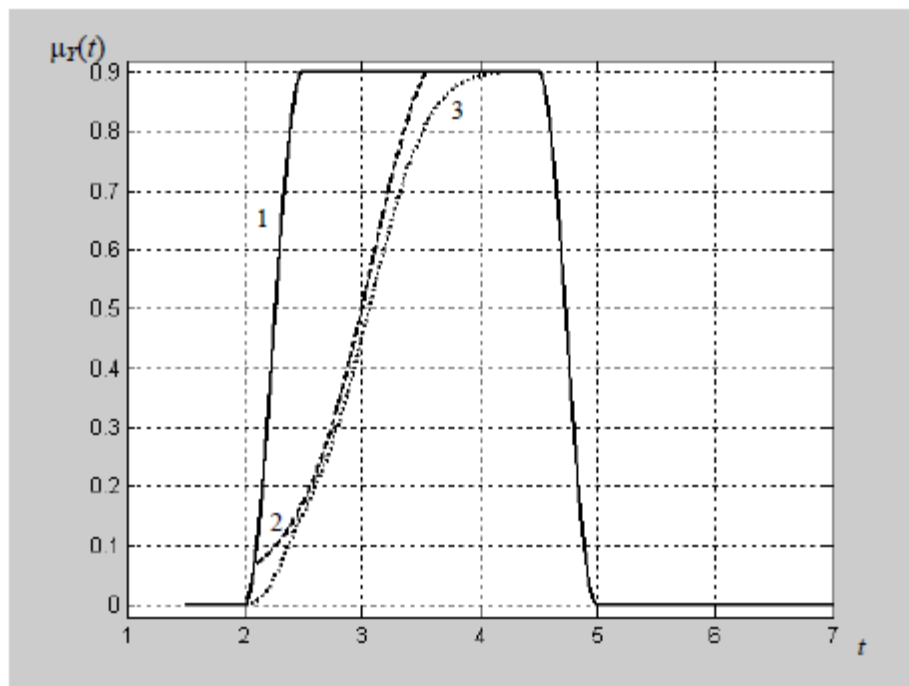


Рис. 10.5. Результати розрахунків функції $\mu_T(t)$ (1), а також результатів активації змінної у $\mu'(y)$ з використанням співвідношень (10.29 (2) та 10.30 (3)

Обрання критерію для обчислення функції активації залежить від характеру та складності розрахунків, які проводяться. Зазвичай розрахунки з використанням правила мінімальної активації є простішими з обчислювальної точки зору, проте недолік такого подання полягає у тому, що розраховані таким чином функції належності зазвичай не є гладкими і можуть мати точки зламу, в яких не існує похідної. Ця відмінність спостерігається також на графічних

залежностях, наведених на рис. 10.5. Обрання найбільш ефективного методу активації змінної також залежить від особливостей постановки задачі нечіткої логіки, яка розв'язується.

Акумуляцією нечіткої змінної (англійський термін – accumulation) називається процедура визначення функції належності для кожної із вихідних лінгвістичних змінних із множини $W = \{w_1, w_2, \dots, w_n\}$. Інакше кажучи, акумуляція пов'язана із процесом об'єднання всіх змінних із множини W з використанням співвідношень (9.2) або (9.6).

Припустимо, якщо аналізується комплексна нечітка змінна $A = \text{«ШВИДКІСТЬ АВТОМОБІЛЯ»}$. Згідно із підходом, який розглядається у цьому підрозділі, необхідно об'єднати три функції залежності для змінної α : $\alpha_1 = \text{«НИЗЬКА ШВИДКІСТЬ»}$, $\alpha_2 = \text{«СЕРЕДНЯ ШВИДКІСТЬ»}$ та $\alpha_3 = \text{«ВИСОКА ШВИДКІСТЬ»}$. Тільки у цьому випадку можна через функцію належності встановити відповідність між значенням змінної та характеристикою об'єкта за умов, які задаються нечіткими правилами, або продукціями. Зрозуміло, що в ході цієї процедури враховується вага кожного із правил та результат попереднього етапу активації.

Розглянемо приклад акумуляції нечітких множин.

Приклад 10.8. Здійснити процес акумуляції нечіткої множини X з прикладу 10.7, якщо вона складається з трьох таких лінгвістичних змінних.

1. $x_2 = \text{«МАЛИЙ ЧАС НАДХОДЖЕННЯ СИГНАЛУ ГОТОВНОСТІ ПРИСТРОЮ»}$. Для цієї змінної вага відповідного правила $R_1 = \text{«У разі, якщо сигнал готовності до роботи надходить частіше, ніж кожні дві секунди, пристрій можна вважати справним»}$ складає $F_1 = 0,1$. Для цього правила функція належності вихідної змінної Y описується співвідношенням $F_1 \cdot f_{Z_1}(x, a, b)$, $a = 1,8$, $b = 2,2$.

2. $x_3 = \text{«ВЕЛИКИЙ ЧАС НАДХОДЖЕННЯ СИГНАЛУ ГОТОВНОСТІ ПРИСТРОЮ»}$. Для цієї змінної вага відповідного правила $R_3 = \text{«У разі, якщо сигнал готовності до роботи надходить рідше, ніж кожні п'ять секунд, пристрій можна вважати справним»}$ складає $F_3 = 0,2$. Для цього правила функція

належності вихідної змінної Y описується співвідношенням $F_1 \cdot f_{S_1}(x, a, b)$,
 $a = 4,8, b = 5,2$.

3. Для змінної $x_1 = \text{«СЕРЕДНІЙ ЧАС НАДХОДЖЕННЯ СИГНАЛУ ГОТОВНОСТІ ПРИСТРОЮ»}$ правило та функції належності відповідають умовам, описаним у прикладі 10.7.

Нижче наведені командні рядки системи MatLab для розв'язування цієї задачі, а отримані графічні залежності показані на рис. 10.6.

```
>> x=1:0.001:7;
>> c=0.9.*fs1(x,2,2.5).*fz1(x,4.5,5);
>> c2=0.1.*fz1(x,1.8,2.2);
>> c3=0.2.*fs1(x,4.8,5.2);
>> c4=c+c2+c3-c.*c2-c.*c3-c2.*c3+c.*c2.*c3;
>> subplot(1,2,1)
>> plot(x,c)
>> hold on
>> plot(x,c2)
>> plot(x,c3)
>> grid on
>> subplot(1,2,2)
>> plot(x,c4)
>> grid on
```

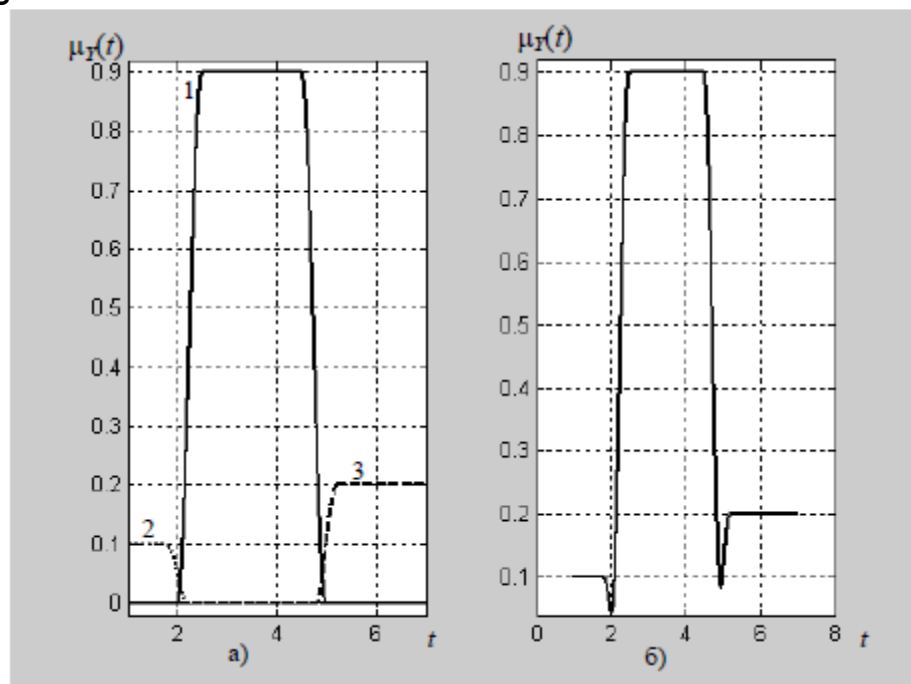


Рис. 10.6. Функції належності $\mu_T(t)$ (а) для змінних y_1 (1), y_2 (2), y_3 (3) та результат їхньої акумуляції (б)

Дефаззифікацією нечіткої змінної (англійський термін – defuzzification) називається процедура визначення звичайного значення вихідної змінної через її функцію належності.

У теорії нечіткої логіки існує декілька альтернативних методів дефаззифікації. Розглянемо головні з них [1, 2].

1. Метод центра ваги.

Згідно із цим методом значення вихідної змінної у визначається як положення центра ваги функції належності після її акумуляції згідно із співвідношенням:

$$y = \frac{\int_{x_{\min}}^{x_{\max}} x \cdot \mu(x) dx}{\int_{x_{\min}}^{x_{\max}} \mu(x) dx}. \quad (10.32)$$

Розглянемо приклад дефаззифікації змінної за методом центра ваги.

Приклад 10.9. Здійснити процедуру дефаззифікації для акумульованої функції належності, побудованої для часу надходження сигналу про готовність пристрою до роботи у прикладі 10.8.

Командні рядки системи MatLab для чисельного розрахунку визначених інтегралів методом прямокутників можна пристосувати для обчислення дробу (10.32), з урахуванням результатів, отриманих у прикладі, 10.8, наступним чином.

```
>> y1=sum(x.*c4.*0.001)./sum(c4.*0.001)
y1 =
    3.7908
```

Тобто, згідно із співвідношенням (10.32), наближене значення чіткої змінної у складає $y \approx 3,8$.

2. Метод центра ваги одноточкових моментів.

Якщо всі значення вихідної лінгвістичної змінної є рівноправними та однаковими, тоді усереднене значення цієї змінної можна розрахувати за формулою, аналогічною формулі (10.32) через суму за всіма інтервалами, на

яких значення функції належності є постійними або дискретними величинами. Для цього випадку узагальнена формула (10.32) суттєво спрощується та переписується у вигляді частки двох сум наступним чином:

$$y = \frac{\sum_{i=1}^n x_i \cdot \mu(x_i)}{\sum_{i=1}^n \mu(x_i)}. \quad (10.33)$$

де n – кількість інтервалів, на яких функція належності $\mu(x_i)$ приймає постійне значення, або кількість лінгвістичних змінних, які відповідають цим інтервалам. Слід відзначити, що дискретні функції належності для скінченних нечітких множин часто описуються не аналітично, а у вигляді таблиці або безпосередньо через перелічення елементів нечіткої множини. Такий метод дефазифікації часто використовуються для аналізу сигналів керування в цифрових електронних системах.

3. Метод центра площини.

Згідно із цим методом значення вихідної змінної y визначається як розв'язок інтегрального рівняння:

$$\int_{x_{\min}}^y \mu(x) dx = \int_y^{x_{\max}} \mu(x) dx, \quad x_{\min} < y < x_{\max}. \quad (10.34)$$

4. Метод лівого модального значення.

У разі використання цього методу значенню вихідної змінної y відповідає ліва мода функції належності $\mu_A(y)$ після акумуляції. Ліве модельне значення для нечіткої множини A розраховується через співвідношення:

$$y = \min \{x_m\}, \quad (10.35)$$

де x_m – модальне значення функції належності. *Модальним значенням* називається таке значення функції, яке є постійним вздовж певного інтервалу аргументу. Якщо функція має декілька модальних значень, тоді вибирається найменше з них. Спосіб обчислення лівого модального значення наочно показаний на рис. 10.7.

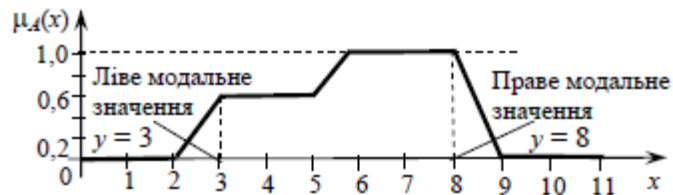


Рис. 10.7. Способи дефазифікації вихідної змінної x за методами лівого та правого модальних значень

5. Метод правого модального значення.

У разі використання цього методу значенню вихідної змінної y відповідає права мода функції належності $\mu_A(y)$ після акумуляції. Праве модельне значення для нечіткої множини A розраховується через співвідношення:

$$y = \max \{x_m\}, \quad (10.36)$$

Спосіб обчислення правого модального значення також показаний на рис. 10.7.

Алгоритми формування нечітких висновків

Розглянуті етапи формування нечітких висновків не можуть бути реалізовані без чіткої формалізації, оскільки вони містять окремі параметри, які мають бути відповідним чином специфіковані. Через вибір конкретних варіантів розрахунку цих параметрів і формуються формалізовані алгоритми, які у повному обсязі реалізують способи формування нечітких висновків в системах з нечіткими продукціями. Сьогодні у теорії нечіткої логіки існує велика кількість таких алгоритмів, тому розглянемо лише ті з них, які знайшли найбільше практичне впровадження у системах нечіткого моделювання [2, 4].

Алгоритм Мамдані.

1. Формування бази правил системи нечітких висновків.
2. Фазифікація вхідних змінних.
3. Агрегація умов у нечітких правилах продукції. Для визначення степені істинності умов кожного правила використовуються логічні операції. Лише ті правила, ступінь істинності яких є відмінною від нуля, вважаються активними та використовуються для проведення подальших розрахунків.

4. Активація умов у правилах продукцій згідно із співвідношенням (10.29).

5. Акумуляція висновків нечітких правил згідно із співвідношенням (9.2).

6. Дефазифікація вихідних змінних з використанням методу центра ваги, співвідношення (10.32) – (10.33), або методу центра площини, співвідношення (10.34).

Алгоритм Цукамото.

1. Формування бази правил системи нечітких висновків.

2. Фазифікація вхідних змінних.

3. Агрегація умов у нечітких правилах продукцій.

4. Активація умов у правилах продукцій згідно із співвідношенням (10.29), після чого знаходять звичайні, не нечіткі значення всіх вихідних лінгвістичних змінних w_i у кожному із висновків правил нечітких продукцій через розв'язування рівнянь

$$c_i = \mu(w_i), \quad \forall i \in 1 \dots q, \quad (10.37)$$

де q – кількість умов у базі правил. У зв'язку з тим, що розрахунки проводяться для дійсних чисел w_i , процес акумуляції в алгоритмі Цукамото майже не використовується.

5. Дефазифікація вихідних змінних згідно із формулою (10.33) для нечітких множин.

Алгоритм Цукамото є найбільш ефективним для моделювання сучасних дискретних цифрових електронних систем.

Алгоритм Ларсена.

1. Формування бази правил системи нечітких висновків.

2. Фазифікація вхідних змінних.

3. Агрегація умов у нечітких правилах продукцій. На цьому етапі використовуються нечіткі логічні операції над множинами, зокрема $\max$ -диз'юнкція (10.6) та $\min$ -кон'юнкція (9.2).

Лише ті правила, степінь істинності яких є відмінною від нуля, вважаються активними та використовуються для проведення подальших розрахунків.

4. Активація умов у правилах продукцій з використанням формули prod-активації (10.30). В результаті цієї операції визначається сукупність нечітких множин $C_1, C_2, \dots, C_q$, де q – загальна кількість умов у базі правил.

5. Акумуляція висновків нечітких правил згідно із співвідношенням (10.2) для об'єднання нечітких множин.

6. Дефазифікація вихідних змінних з використанням будь-якого із розглянутих методів.

Алгоритм Сугено.

1. Формування бази правил системи нечітких висновків, для чого використовують лише стандартні лінгвістичні конструкції типу:

ПРАВИЛО#:

$$\text{ЯКЩО } \ll \beta_1 = \alpha_1 \gg \text{ТА } \ll \beta_2 = \alpha_2 \gg \text{ТОДІ } \ll w = \varepsilon_1 \cdot \alpha_1 + \varepsilon_2 \cdot \alpha_2 \gg, (10.38)$$

де $\varepsilon_1, \varepsilon_2$ – вагові коефіцієнти умов.

2. Фазифікація вхідних змінних.

3. Агрегація умов у нечітких правилах продукцій. Для визначення істинності умов у лінгвістичних конструкціях (10.38) використовується метод min-кон'юнкції (10.2). Лише ті правила, степінь істинності яких є відмінною від нуля, вважаються активними та використовуються для проведення подальших розрахунків.

4. Активація умов у правилах нечітких продукцій. Спочатку з використанням співвідношення для min-активації (10.29) визначають значення степені істинності для правил, записаних у вигляді (10.38). На другому етапі активації здійснюється розрахунок звичайних значень вихідних змінних для співвідношення (10.38), де значення змінних α_1 та α_2 розглядаються як значення вхідних змінних цього етапу дефазифікації. Таким чином визначаються нечіткі множини висновків $C = \{c_1, c_2, \dots, c_n\}$ та вихідні змінні $W = \{w_1, w_2, \dots, w_n\}$.

5. Акумуляція висновків нечітких правил продукцій.

6. Дефаззифікація вихідних змінних множини W з використанням співвідношення (10.33).

Спрощений алгоритм формування нечітких висновків.

1. Формування бази правил системи нечітких висновків, для чого використовують лише спрощені стандартні лінгвістичні конструкції типу:

$$\text{ПРАВИЛО\#}: \text{ЯКЩО } \ll \beta_1 = \alpha_1 \gg \text{ ТА } \ll \beta_2 = \alpha_2 \gg \text{ ТОДІ } \ll w = \varepsilon \gg, (10.39)$$

де ε – ціле число.

2. Фаззифікація вхідних змінних.

3. Агрегація умов у нечітких правилах продукцій. Для визначення степені істинності висловлень (10.39) зазвичай використовується $\min$ -кон'юнкція (10.2). Лише ті правила, степінь істинності яких є відмінною від нуля, вважаються активними та використовуються для проведення подальших розрахунків.

4. Активація умов у правилах нечітких продукцій (10.39), яка зазвичай виконується з використанням співвідношення для $\min$ -активації (10.29). В результаті визначаються степені істинності нечітких висновків $C = \{c_1, c_2, \dots, c_n\}$.

5. Акумуляція висновків нечітких правил продукцій. Оскільки c_i – дійсні числа, цей етап розрахунків значно спрощується.

6. Дефаззифікація вихідних змінних з використанням співвідношення для центра ваги дискретних множин (10.33).

Для розв'язування простих задач, які описують загальні принципи роботи електронних систем керування, а також для проведення ручних розрахунків, зазвичай використовується розглянутий вище спрощений алгоритм.

Контрольні запитання

1. Що являють собою нечіткі відношення між елементами нечітких множин? Наведіть власні приклади формування нечітких відношень між множинами для розв'язування задач нечіткої логіки.

2. Що являють собою нечіткі змінні та як вони описуються? Наведіть власні приклади описання нечіткої змінної.

3. Що являють собою лінгвістичні нечіткі змінні та як вони описуються? Наведіть власні приклади описання лінгвістичної нечіткої змінної.
4. Що являють собою нечіткі інтервали та як вони описуються? Наведіть власні приклади описання нечіткого інтервалу.
5. Що являють собою нечіткі числа та як вони описуються? Наведіть власні приклади описання нечіткого числа.
6. Які арифметичні операції над нечіткими числами та інтервалами Вам відомі? Наведіть власні приклади використання цих операцій для розв'язування задач нечіткої логіки.
7. Що являють собою функції L -типу та R -типу?
8. Що являють собою нечіткі висловлення та нечіткі предикати? Наведіть власні приклади нечітких висловлень.
9. Які логічні операції над нечіткими висловленнями Вам відомі?
10. Що являє собою імплікація нечітких висловлень та які види імплікації Вам відомі?
11. Що являє собою нечітка еквівалентність і чим це поняття відрізняється від класичного поняття еквівалентності?
12. Що являють собою правила нечіткої продукції?
13. Які три типи правил нечіткої продукції розрізняють у теорії нечіткої логіки?
14. Що являє собою фаззифікація нечітких змінних?
15. Що являє собою агрегування нечітких змінних?
16. Що являє собою активація нечітких змінних?
17. Які методи активації нечітких змінних вам відомі?
18. Що являє собою акумуляція нечітких змінних?
19. Що являє собою дефаззифікація нечітких змінних?
20. Які методи дефаззифікації нечітких змінних Вам відомі?
21. Які алгоритми формування нечітких висновків Вам відомі та чим ці алгоритми відрізняються один від одного?

22. Які головні послідовні кроки обробки нечітких даних використовуються в алгоритмі Мамдані?

23. Які головні послідовні кроки обробки нечітких даних використовуються в алгоритмі Цукамото?

24. Які головні послідовні кроки обробки нечітких даних використовуються в алгоритмі Ларсена?

25. Які головні послідовні кроки обробки нечітких даних використовуються в алгоритмі Сугено?

26. У чому полягає спрощений алгоритм формування нечітких висновків?

Перелік посилань

1. Zadeh, L. A. Fuzzy sets. – Information and Control, no. 8 (3), 1965, p. 338.
2. Zadeh, L. A. Fuzzy logic = computing with words. – IEEE Transactions on Fuzzy Systems, no. 4 (2), 1996, p. 103.
3. Zadeh, L. A. Fuzzy algorithms. – Information and Control, no. 12 (2), 1968, p. 94.
4. Потій О.В., Леншин А.В. Основні положення математичного апарату суб'єктивної логіки та його застосування для оцінки рівня зрілості систем забезпечення безпеки інформації. – Радіотехніка. Тематичний випуск “Інформаційна безпека”. Харків: Харківський національний університет радіоелектроніки, 2005. с. 144–160.

Лекція №11. Приклади використання алгоритмів нечіткої логіки в електронних системах керування складними об'єктами

Розглянемо декілька важливих прикладних інженерних задач, пов'язаних із організацією роботи складних електронних систем керування, розв'язування яких є можливим лише з використанням розглянутих вище алгоритмів нечіткої логіки [1-3].

Приклад 11.1. Електронна система автоматичного регулювання температури води у душі під час купання. Сутність цієї задачі полягає в тому, що необхідно розробити автоматичний регулятор постачання гарячої води у душі. Регулятор повинен працювати таким чином, щоб підтримувати середню температуру теплої води на виході системи через автоматичне збільшення та зменшення потоку гарячої води. Така задача керування є вкрай важливою як для забезпечення зручності тим людям, які миються під душем, так і з точки зору оптимізації витрат гарячої води та економії енергії.

Згідно із сформульованою постановкою задачі визначимо спочатку основні нечіткі правила, за якими має працювати система керування. Будемо записувати правила за спрощеною лінгвістичною конструкцією.

ПРАВИЛО 1: ЯКЩО «ВОДА Є ГАРЯЧОЮ» ТОДІ «ПОВЕРНУТИ ВЕНТИЛЬ КРАНУ ГАРЯЧОЇ ВОДИ НА ВЕЛИКИЙ КУТ ПРАВОРУЧ».

ПРАВИЛО 2: ЯКЩО «ВОДА Є НЕ ДУЖЕ ГАРЯЧОЮ» ТОДІ «ПОВЕРНУТИ ВЕНТИЛЬ КРАНУ ГАРЯЧОЇ ВОДИ НА НЕВЕЛИКИЙ КУТ ПРАВОРУЧ».

ПРАВИЛО 3: ЯКЩО «ВОДА Є ТЕПЛОЮ» ТОДІ «ЗАЛИШАТИ КУТ ПОВОРОТУ ВЕНТИЛЯ КРАНУ ГАРЯЧОЇ ВОДИ БЕЗ ЗМІНИ».

ПРАВИЛО 4: ЯКЩО «ВОДА Є ПРОХОЛОДНОЮ» ТОДІ «ПОВЕРНУТИ ВЕНТИЛЬ КРАНУ ГАРЯЧОЇ ВОДИ НА НЕВЕЛИКИЙ КУТ ЛІВОРУЧ».

ПРАВИЛО 5: ЯКЩО «ВОДА Є ХОЛОДНОЮ» ТОДІ «ПОВЕРНУТИ ВЕНТИЛЬ КРАНУ ГАРЯЧОЇ ВОДИ НА ВЕЛИКИЙ КУТ ЛІВОРУЧ».

Згідно із сформульованими правилами 1 – 5 легко визначити вхідну та вихідну лінгвістичні змінні та їхні значення. Зрозуміло, що вихідною змінною є температура води. Позначимо цю змінну як T_1 , і вона повинна мати таку множину значень: $T = \{t_1 = \text{«Гаряча»}, t_2 = \text{«Не дуже гаряча»}, t_3 = \text{«Тепла»}, t_4 = \text{«Прохолодна»}, t_5 = \text{«Холодна»}\}$.

Вхідною змінною для системи керування, яка розглядається, є кут обертання вентиля гарячої води. Позначимо цю змінну як W , а її значення, згідно із нечіткими правилами 1 – 5, будуть такими: $W = \{w_1 = \text{«Великий кут ліворуч»}, w_2 = \text{«Невеликий кут ліворуч»}, w_3 = \text{«Нуль»}, w_4 = \text{«Невеликий кут праворуч»}, w_5 = \text{«Великий кут праворуч»}\}$.

Кусково-лінійні функції належності для змінних T та W наведені на рис. 11.1.

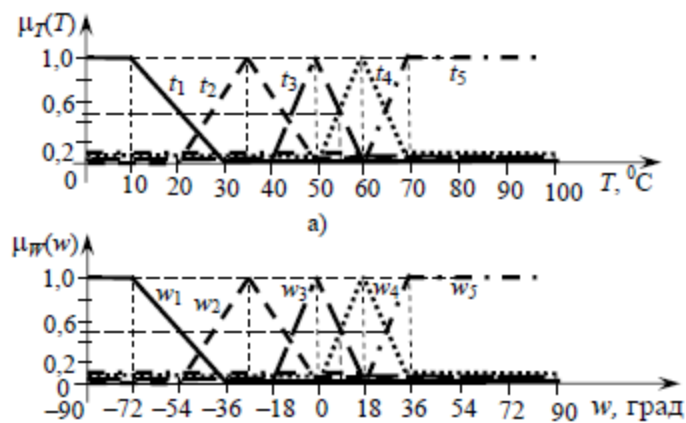


Рис. 11.1. Кусково-лінійні функції належності для лінгвістичних змінних T (а) та W (б)

Визначення функцій належності є найскладнішим завданням для практичних задач, пов'язаних із використанням теорії нечіткої логіки. Однак, зазвичай на першому етапі моделювання можна використовувати кусково-лінійні апроксимації функцій належності, які є досить простими, проте достовірно описують нечіткі правила, задані лінгвістичними конструкціями (10.38) та (10.39). Складність задач нечіткої логіки полягає у тому, що у правилах, аналогічних 1 – 5, вихідні змінні не визначені явним чином, тому саме функції належності встановлюють відповідність між вхідними і вихідними

змінними. Дійсно, що являє собою характеристика «Не дуже гаряча вода», визначена змінною t_2 , уявити собі не дуже важко, тому і легко побудувати відповідну функцію належності. А ось визначити «Невеликий кут обертання ліворуч», який відповідає вхідній змінній t_2 на виході і через функцію належності вихідної змінної w_2 описує реакцію системи керування температурою води на цю подію, вже набагато складніше. Для даного прикладу обрана проста лінійна залежність $W(T)$, але, на жаль, на практиці такі прості моделі керування не завжди є адекватними.

Розглянемо, як працює введена нами система нечітких правил на конкретному прикладі. Припустимо, що температура води складає 55°C та проаналізуємо реакцію системи на цю подію згідно з алгоритмом Мамдані. Процес фазифікації вхідної змінної показаний на рис. 11.1, а. Зрозуміло, що за такої умови активними вважаються лише правила 3 та 4, а істинність всіх інших правил дорівнює 0. Оскільки правила 1 – 5 є правилами першого роду, їхня агрегація не є складною процедурою, можна просто вважати, що степінь істинності цих правил 3 та 4 складає 0,5. Не є важким для поставленої задачі також процес активізації. Для правил першого типу можна вважати вагові коефіцієнти рівними 1, а у цьому випадку вихідні функції належності в ході активації не змінюються. Тоді акумуляція вихідних функцій належності w_3 та w_4 , згідно із графічними побудовами, які показані на рис. 11.2, приводить до остаточної функції належності для вихідної змінної W , яка показана на рис. 11.2. У цьому випадку акумуляція вихідної змінної здійснювалась через об'єднання нечітких множин за формулою (9.2).

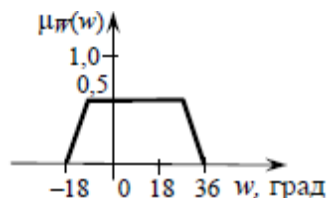


Рис. 11.2. Остаточна функція належності для вихідної змінної W після акумуляції

Дефазифікацію змінної W можна провести з урахуванням того, що кусково-лінійна функція належності, наведена на рис. 10.7, є симетричною та має вигляд трапеції. Тоді остаточно маємо: $w = \frac{36-18}{2} = 9$. Тобто, згідно із побудованою системою нечітких правил, якщо електронний датчик показує, що температура води в душі становить 55°C , вентиль крану гарячої води необхідно повернути на 9° праворуч.

Найбільш цікавим є те, що застосування методів нечіткої логіки до досить складної практичної задачі керування температурою води у душі приводить не до наближеної або інтервальної оцінки кута обертання вентиля крана, а до точного розв'язування поставленої нечіткої задачі.

Приклад 11.2. Автоматична електронна система керування кондиціонером. Іншою важливою задачею, пов'язаною із забезпеченням комфорту людей у приміщеннях та економією електроенергії, є автоматична система керування кондиціонуванням повітря. Сутність оптимізації роботи такої системи керування полягає у постійному підтриманні нормальної температури у приміщенні за умови мінімального енергоспоживання. Зрозуміло, що спосіб розв'язування такої задачі оптимізації у загальному вигляді також ґрунтується на використанні описаних вище алгоритмів нечіткої логіки.

Спочатку сформуємо для цієї задачі систему нечітких правил. Для цього визначимо вхідні та вихідні лінгвістичні змінні та задамо їхні можливі значення. Вхідною змінною, яка є контрольованим параметром для системи керування кондиціонером, можна вважати температуру повітря у приміщенні. Позначимо цю змінну як T і будемо вважати, що вона має таку терм-множину значень:

$$\begin{aligned} T &= \{\text{«Дуже холодно»}, \text{«Холодно»}, \text{«Нормально»}, \text{«Тепло»}, \text{«Жарко»}\} = \\ &= \{\text{«NB»}, \text{«NS»}, \text{«Z»}, \text{«PS»}, \text{«PB»}\}. \end{aligned}$$

Другою вхідною змінною D будемо вважати зміну температури у приміщенні, терм-множину значень якої запишемо наступним чином:

$$D = \{\text{«Від'ємна»}, \text{«Дорівнює нулю»}, \text{«Додатна»}\} = \{\text{«NS»}, \text{«Z»}, \text{«PS»}\}.$$

Вихідною змінною для формування системи нечітких правил керування кондиціонером, як і в попередньому прикладі, будемо вважати кут обертання регулятора. Визначимо для цієї лінгвістичної змінної W терм-множину аналогічно значенням цієї змінної в попередньому прикладі:

$$\begin{aligned} W = \{w_1 = \text{«Дуже великий кут праворуч»}, w_2 = \text{«Великий кут праворуч»}, \\ w_3 = \text{«Невеликий кут праворуч»}, w_4 = \text{«Вимкнути кондиціонер»}, \\ w_5 = \text{«Невеликий кут ліворуч»}, w_6 = \text{«Великий кут ліворуч»}, \\ w_7 = \text{«Дуже великий кут ліворуч»}\} = \\ = \{\text{«NB»}, \text{«NM»}, \text{«NS»}, \text{«Z»}, \text{«PS»}, \text{«PM»}, \text{«PB»}\}. \end{aligned}$$

Тепер, враховуючи введені нечіткі лінгвістичні змінні T , D , та W , можна записати 15 нечітких правил керування кондиціонером наступним чином.

ПРАВИЛО 1: ЯКЩО «« $T = \text{«PB»}$ » ТА « $D = \text{«PS»}$ »» ТОДІ « $W = \text{«NB»}$ ».

ПРАВИЛО 2: ЯКЩО «« $T = \text{«PB»}$ » ТА « $D = \text{«NS»}$ »» ТОДІ « $W = \text{«NS»}$ ».

ПРАВИЛО 3: ЯКЩО «« $T = \text{«PS»}$ » ТА « $D = \text{«PS»}$ »» ТОДІ « $W = \text{«NM»}$ ».

ПРАВИЛО 4: ЯКЩО «« $T = \text{«PS»}$ » ТА « $D = \text{«NS»}$ »» ТОДІ « $W = \text{«Z»}$ ».

ПРАВИЛО 5: ЯКЩО «« $T = \text{«NB»}$ » ТА « $D = \text{«NS»}$ »» ТОДІ « $W = \text{«PB»}$ ».

ПРАВИЛО 6: ЯКЩО «« $T = \text{«NB»}$ » ТА « $D = \text{«PS»}$ »» ТОДІ « $W = \text{«PS»}$ ».

ПРАВИЛО 7: ЯКЩО «« $T = \text{«NS»}$ » ТА « $D = \text{«NS»}$ »» ТОДІ « $W = \text{«PM»}$ ».

ПРАВИЛО 8: ЯКЩО «« $T = \text{«NS»}$ » ТА « $D = \text{«PS»}$ »» ТОДІ « $W = \text{«Z»}$ ».

ПРАВИЛО 9: ЯКЩО «« $T = \text{«PB»}$ » ТА « $D = \text{«Z»}$ »» ТОДІ « $W = \text{«NM»}$ ».

ПРАВИЛО 10: ЯКЩО «« $T = \text{«PS»}$ » ТА « $D = \text{«Z»}$ »» ТОДІ « $W = \text{«NS»}$ ».

ПРАВИЛО 11: ЯКЩО «« $T = \text{«NB»}$ » ТА « $D = \text{«Z»}$ »» ТОДІ « $W = \text{«PB»}$ ».

ПРАВИЛО 12: ЯКЩО «« $T = \text{«NS»}$ » ТА « $D = \text{«Z»}$ »» ТОДІ « $W = \text{«PS»}$ ».

ПРАВИЛО 13: ЯКЩО «« $T = \text{«Z»}$ » ТА « $D = \text{«PS»}$ »» ТОДІ « $W = \text{«NS»}$ ».

ПРАВИЛО 14: ЯКЩО «« $T = \text{«Z»}$ » ТА « $D = \text{«NS»}$ »» ТОДІ « $W = \text{«PS»}$ ».

ПРАВИЛО 15: ЯКЩО «« $T = \text{«Z»}$ » ТА « $D = \text{«Z»}$ »» ТОДІ « $W = \text{«Z»}$ ».

Один із можливих варіантів визначення кусково-лінійних функцій належності для введених лінгвістичних змінних T , D , та W наведений на рис. 11.3.

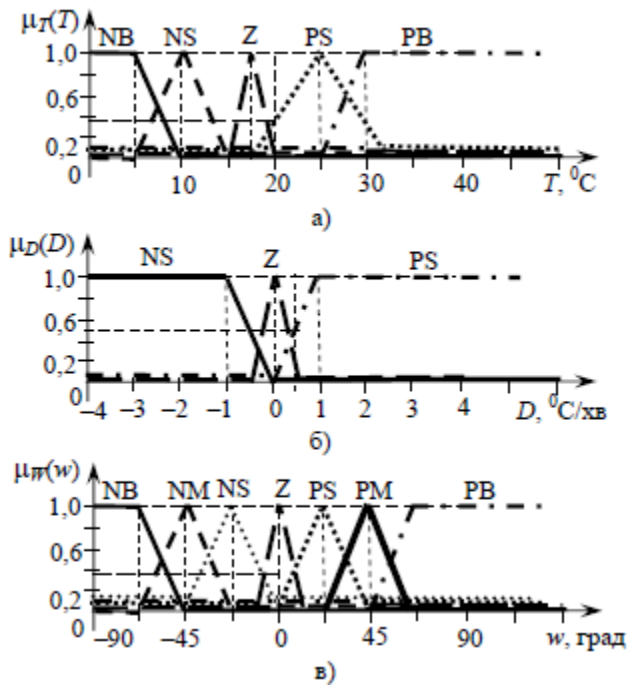


Рис. 11.3. Функції належності для вхідних лінгвістичних змінних T (а) та D (б), а також для вихідної лінгвістичної змінної для W (в)

На етапі формування нечітких висновків використаємо для введених функцій належності алгоритм Мамдані, вважаючи, що температура у приміщенні складає 20°C , а змінюється температура в сторону зростання на $0,5^{\circ}\text{C}/\text{хв}$. Для таких значень вхідних змінних $\mu_T(T) = \mu_{PS}(T) \approx 0,4$, $\mu_D(D) = \mu_{PS}(D) \approx 0,4$, $\mu_D(D) = \mu_Z(D) \approx 0,5$, відповідні графічні побудови показані на рис. 11.3, а, б. Значення лінгвістичних змінних $T = \text{«PS»}$ та $D = \text{«PS»}$ відповідають правилу 3, а значення $T = \text{«PS»}$ та $D = \text{«Z»}$ – правилу 10. Оскільки всі правила 1 – 15 є нечіткими висловленнями першого типу, згідно із рис. 11.3, в, для акумуляції двох вихідних значень змінної W «NM» та «NS» необхідно знайти об'єднання функцій належності $\mu_{NM}(w)$ та $\mu_{NS}(w)$ на рівні 0,4. Значення $W = \text{«NM»}$ відповідає правилу 3, а значення $W = \text{«NS»}$ – правилу 10. Результатом цієї алгебраїчної операції над нечіткими множинами є функція залежності $\mu_{\text{out}} = \min(0,4, \max(\mu_{NM}, \mu_{NS}))$, графік якої наведений на рис. 11.4.

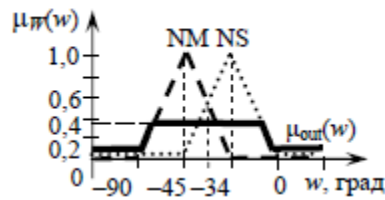


Рис. 11.4. Остаточна функція належності $\mu_{out}(w)$ для вихідної лінгвістичної змінної W після акумуляції

З геометричних побудов, наведених на рис. 11.4, видно, що процес дефазифікації вихідної змінної W , згідно із методом центра площини, приводить до остаточного значення кута обертання регулятора кондиціонера $w = -34^\circ$.

Приклад 11.3. Автоматична електронна система керування контейнерним краном. Розглянемо тепер нечітку модель керування контейнерним краном. Ця задача нечіткої логіки є важливою для організації вантажних робіт на будівництві, на складах, на вокзалах, в морських портах та аеропортах. Невірна організація роботи контейнерних кранів протягом довгого часу пов'язана із зайвими витратами часу на проведення вантажних робіт та перевитратами електроенергії, тому розв'язування цієї нечіткої задачі також має важливе практичне значення.

Для цієї задачі вхідними лінгвістичними змінними будемо вважати відстань від крана до вантажу, який необхідно перенести, та кут розташування крана відносно вантажу, а вихідної змінною – потужність, яку споживає кран, від якої залежить швидкість руху крана. Організація оптимальної роботи контейнерного крана полягає у тому, що, по мірі наближення крана до вантажу, потужність, яку споживає кран, постійно зменшується. Будемо розглядати наступні значення вхідних змінних та вихідної змінної.

Відстань від крана до вантажу: $R = \{r_1 = \text{«Дуже велика відстань»}, r_2 = \text{«Велика відстань»}, r_3 = \text{«Середня відстань»}, r_4 = \text{«Близька відстань»}, r_5 = \text{«Нульова відстань»}\}.$

Кут положення крана відносно вантажна: $W = \{w_1 = \text{«Від'ємний великий»}, w_2 = \text{«Від'ємний малий»}, w_3 = \text{«Нульовий»}, w_4 = \text{«Додатний малий»}, w_5 = \text{«Додатний великий»}\}$.

Потужність, яку споживає контейнерний кран: $P = \{P_1 = \text{«Від'ємна висока»}, P_2 = \text{«Від'ємна середня»}, P_3 = \text{«Нульова»}, P_4 = \text{«Додатна середня»}, P_5 = \text{«Додатна висока»}\}$.

Зрозуміло, що потужність, яка споживається краном, не може мати від'ємне значення. Для модельної задачі, яка розглядається, потужність вважається від'ємною, якщо відстань від крана до вантажу є від'ємною та зменшується під час руху крана. Крім того, будемо вважати, що потужність крана є дискретним значенням і описувати її як точкову дискретну множину.

З урахуванням введених лінгвістичних змінних нечіткі правила для задачі керування контейнерним краном можна записати наступним чином.

ПРАВИЛО 1: ЯКЩО « $\langle R = r_2 \rangle$ ТА « $\langle W = w_3 \rangle$ » ТОДІ « $\langle P = p_4 \rangle$ ».

ПРАВИЛО 2: ЯКЩО « $\langle R = r_1 \rangle$ ТА « $\langle W = w_2 \rangle$ » ТОДІ « $\langle P = p_5 \rangle$ ».

ПРАВИЛО 3: ЯКЩО « $\langle R = r_1 \rangle$ ТА « $\langle W = w_1 \rangle$ » ТОДІ « $\langle P = p_4 \rangle$ ».

ПРАВИЛО 4: ЯКЩО « $\langle R = r_3 \rangle$ ТА « $\langle W = w_2 \rangle$ » ТОДІ « $\langle P = p_2 \rangle$ ».

ПРАВИЛО 5: ЯКЩО « $\langle R = r_4 \rangle$ ТА « $\langle W = w_4 \rangle$ » ТОДІ « $\langle P = p_4 \rangle$ ».

ПРАВИЛО 6: ЯКЩО « $\langle R = r_5 \rangle$ ТА « $\langle W = w_3 \rangle$ » ТОДІ « $\langle P = p_3 \rangle$ ».

Слід відзначити, що правила 1 – 6 сформовані на основі досвіду робітників контейнерних кранів, тому їх можна використовувати для автоматичної електронної системи електронного керування краном.

Функції належності для визначених лінгвістичних змінних R , W та P наведені на рис. 11.5.

Подальше розв'язування задачі будемо проводити наступним чином.

1. Для активізації змінних використаємо метод $\min$ -активізації, формула (10.29).

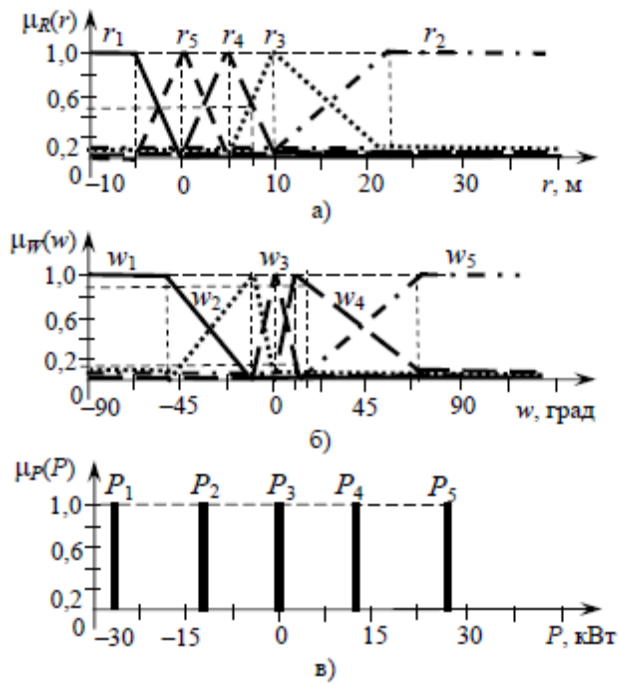


Рис. 11.5. Функції належності для відстані від крана до вантажу (а), кута між напрямком руху крана та вантажем (б) та потужності, яку споживає кран (в)

2. Для акумуляції змінних використаємо метод тах-диз'юнкції, формула (9.6).

3. Для дефазифікації змінних використаємо метод центра ваги для дискретних множин, формула (10.33).

З урахуванням вибраних методів аналізу проведемо числові розрахунки для нечітких правил 1 – 6 за умови $r = 7,5$ м, $w = 10^\circ$. Із графічних залежностей, наведених на рис. 11.5, а, зрозуміло, що у разі $r = 7,5$ м $\mu_{r3}(7,5) = \mu_{r4}(7,5) \approx 0,5$. Тобто, відстань від крана до контейнера $r = 7,5$ м із однаковою степінню істинності можна вважати як близькою, так і середньою. Фаззифікація змінної $w = 10^\circ$, згідно із рис. 11.5, б, дає значення $\mu_{w4}(10) \approx 0,9$, $\mu_{w5}(10) \approx 0,1$. Тобто, за умови визначених функцій належності кут між напрямком руху крана та контейнером із степінню істинності 0,9 можна вважати додатним малим, а із степінню істинності 0,1 – додатним великим. Аналіз нечітких правил 1 – 6 показує, що єдиним правилом, під яке підлягають ці значення вхідних змінних, є правило 5. Оскільки для нечіткої задачі, яка розв'язується, альтернативних рішень немає, згідно із правилом 5 остаточно отримуємо, що потужність

електричного двигуна має бути додатною середньою ($P = P_4$), що, згідно із графічними даними, наведеними на рис. 11.5, в, складає приблизно 12 кВт.

Наведені приклади показують, наскільки математичні моделі нечіткої логіки, які на перший погляд виглядають досить абстрактними, є корисними для розв'язування конкретних практичних задач, пов'язаних із формуванням алгоритмів роботи складних електронних систем, призначених для керування фізичними та технологічними об'єктами [4].

Контрольні запитання

1. Поясніть приклад 11.1 та графічні залежності, які наведені на рис. 11.1 та 11.2.
2. Поясніть приклад 11.2 та графічні залежності, які наведені на рис. 11.3 та 11.4.
3. Поясніть приклад 11.3 та графічні залежності, які наведені на рис. 11.5.

Перелік посилань

1. Zadeh, L. A. Fuzzy sets. – Information and Control, no. 8 (3), 1965, p. 338.
2. Zadeh, L. A. Fuzzy logic = computing with words. – IEEE Transactions on Fuzzy Systems, no. 4 (2), 1996, p. 103.
3. Zadeh, L. A. Fuzzy algorithms. – Information and Control, no. 12 (2), 1968, p. 94.
4. Потій О.В., Леншин А.В. Основні положення математичного апарату суб'єктивної логіки та його застосування для оцінки рівня зрілості систем забезпечення безпеки інформації. – Радіотехніка. Тематичний випуск “Інформаційна безпека”. Харків: Харківський національний університет радіоелектроніки, 2005. с. 144–160.