

АЛГОРИТМ ТА ПРОГРАМА МІНІМІЗАЦІЇ ЛОГІКОВИХ ФУНКЦІЙ У ПОЛІНОМНОМУ ФОРМАТІ

Проблема мінімізації залишається актуальною і сьогодні, особливо для розробників цифрових пристроїв, адже зараз мінімізацію застосовують зазвичай при проектуванні мікросхем (мікропроцесорів, складних логікових елементів, програмованих логікових матриць – PLA, і т. д.) і для тестування та виявлення помилок у виготовлених мікросхемах. Ефективно проведена мінімізація дає змогу зменшити кількість логікових елементів, в базисі яких складається мікросхема, відтак ціна продукту знижується без зміни обчислювальної потужності. Тому головним застосовуваним критерієм ефективності мінімізації є кошт реалізації логікової функції. Він визначається як кількість кон'юнктерів і літералів до мінімізації віднесена до цієї ж кількості після мінімізації. Тут літерал – це аргумент функції, а кон'юнктер – це кон'юнкція літералів. Слайд презентації з порівнянням схеми до мінімізації та схеми після мінімізації, що реалізують одну і ту ж логікову функцію, показує скільки елементів дає змогу заощадити ефективна мінімізація.

Логікові функції можна представляти у різних форматах, наприклад: класично у диз'юнктивному форматі, у поліномному, поліномом Жегалкіна, і т. п. Кожен формат має свої характерні особливості, корисні для деяких логікових мікросхем. До прикладу, для реалізації функції, заданої поліномом Жегалкіна, не потрібно використовувати інвертори. Більше того, з одного формату можна перейти в інший, використовуючи певні математичні перетворення. Основною відмінністю поліномного формату від диз'юнктивного є використання суми за модулем 2 кон'юнктерів (виняткове ЧИ) замість диз'юнкції кон'юнктерів (логікове ЧИ). Тому для більшості функцій перехід між цими двома форматами елементарний і полягає у заміні диз'юнкції на суму за модулем 2 і навпаки. Найчастіше використовується саме диз'юнктивний формат представлення логікових функцій, історично склалося, що абсолютна більшість методів мінімізації працюють із ним (класичний метод карт Карно, метод Квайна Мак-Класкі, ...). Проте, через властивості операції виняткового ЧИ, поліномний формат дає набагато компактніші результати мінімізації із кращим коштом реалізації. В презентації представлено порівняння результатів процедури розчеплення кон'юнктерів однакової функції у диз'юнктивному форматі та

у поліномному форматі. Поліномний формат дає набагато більш компактний результат із майже удвічі кращим коштом реалізації.

Незважаючи на таку перспективність, усі розроблені дотепер точні розв'язки задачі мінімізації у поліномному форматі придатні хіба-що для функцій від невеликої кількості змінних (до 10) і лише для спеціальних класів функцій. Тому відносно ширше практичне застосування мають евристичні методи, але вони дають лише наближений результат. При цьому усі методи мають спільний недолік – вони не функціонують для відстані Гемінга (кількість відмінних позицій) між кон'юнктерами більшої від 2, максимум 3. Новий метод мінімізації на основі розчеплення кон'юнктерів і узагальнених правил їх спрощення (теорем) вирішує ці проблеми. Наприклад, як показано у презентації, метод *xlinking* зупиняється на проміжному результаті, але насправді, використовуючи теореми мінімізації, можна далі спростити два кон'юнктери з відстанню Гемінга 3 і отримати ще кращий результат. Окрім того, метод працює також для неповних і недовизначених логікових функцій, а також їх систем.

Щоб зреалізувати новий метод мінімізації програма має циклічно здійснювати процедуру розчеплення кон'юнктерів поки є можливість, інакше перейти до спрощення результату на основі теорем, далі вертатись до розчеплення, і так по колу. Якщо з результатом не вдається виконати жодних дій по спрощенню, то результат і є змінімізованою функцією. Але на певних етапах програми виникають неоднозначності. Так, наприклад, є багато варіантів покриття матриці розчеплення, і не завжди найочевидніший спосіб дасть кращий результат. Аналогічно з теоремами спрощення. Тому це складна комбінаторна задача, де потрібно перебирати багато варіантів і обирати найкращий. Знову ж таки, кількість цих варіантів росте степеневно з кількістю змінних і кон'юнктерів, тому потрібно буде задати певні обмеження на кількість ітерацій і кількість варіантів до розгляду на кожній ітерації. Все ж, отримані результати підтверджують, що такий алгоритм можна реалізувати програмно, і мабуть скоро найефективніший досі алгоритм мінімізації буде загальнодоступний.