# Оптимізація обчислення дискретних функцій за допомогою машини станів

Андрій Севастьянов (mustitz)

31 травня 2016 р.

**Анотація**

Часто на практиці виникає задача швидкого обчислення деякої дискретної функції. В цій статті буде показано яким чином можна обчислювати значення таких функцій за допомогою машини станів. Для цього буде визначено новий тип машини станів, та операції над ним. Робота машини станів проілюструється на прикладі обчислення сили руки у покерних іграх.

## 1 Вступ

Як можна швидко обчислити дискретну функцію $f : X \to Y$? Відразу приходить до розуму використання таблиці. Що може бути швидше та простіше? Переводимо аргумент функції до індексу, та дивимося у таблиці значення функції. Але тут виникають дві проблеми. По-перше, перевід функції до індексу може зайняти певний час. За іронією долі, це також дискретна функція. По-друге, розмір таблиці може бути дуже великим.

Припустимо нам треба обчислити силу руки у покері. Якщо $C$ це множина карт, $|C| = 52$, $R$ це множина можливих рангів рук, $|R| = 7462$, то нам треба швидко обчислити функцію $f^{(5)} : C^5 \to R$, яка за п'ятьма картами повертає ранг руки. Звичайно ця функція визначена не для усіх елементів множини $C^5$, а лише для тих, де відповідні карти не повторюються.

Якщо будувати таблицю для обчислення функції у лоб, во вона буде містити $52^5 = 380\ 204\ 032$ елементів, що забагато. Якщо цей метод використати для техаського холдему, то таблиця вже буде складатися з $1\ 028\ 071\ 702\ 528 \approx 1.0 \cdot 10^{12}$, а якщо брати омаху, то отримаємо цілих $2\ 779\ 905\ 883\ 635\ 712 \approx 2.8 \cdot 10^{15}$ елементів, що трохи менше ніж безліч.

Звісно, це не єдиний можливий спосіб перевести множину карт до індексу. Якщо розглядати різні п'ятиелементні підмножини множини $C$, то їх кількість буде значно менше: $\binom{52}{5} = 2\ 598\ 960$. Щоб перевести елемент цієї множини у відповідний індекс, можна скористатися комбінаторную системою числення, яка ретельно описана у четвертому томі Дональда Кнута [1]. Якщо $(c_5, c_4, c_3, c_2, c_1) \in C^5$, та $c_5 > c_4 > c_3 > c_2 > c_1$, то номер цієї комбінації можна обчислити за формулою:

$$N = \binom{c_5}{5} + \binom{c_4}{4} + \binom{c_3}{3} + \binom{c_2}{2} + \binom{c_1}{1}.$$

Ми бачимо, що розмір таблиці суттєво скоротився. Але функція переводу комбінації до індексу ускладнилася: тепер вона вимагає сортування та обчислення кількості комбінацій. Для сортування ми можемо перевести масив карт до бітової маски та назад, це дозволить нам уникнути умов. Кількості комбінацій також можна прорахувати заздалегідь, та зберегти у масиві. Все це робить обчислення цієї функції досить швидким. Але є сумніви щодо оптимальності використання пам'яті. Якщо таблиця на два с половиною мільйона елементів прийнятна для більшості сучасних обчислювальних архітектур, то вже у разі семи карт ми отримаємо $\binom{52}{7} = 133\,784\,560$ елементів. У разі омахи все набагато сумніше, бо перші п'ять карт та останні чотири не взаємозамінні: сила руки A♣K♣Q♣J♣2♦ + 10♣3♦3♠3♥ відрізняється від сили руки A♣K♣Q♣3♥2♦ + J♣10♣3♦3♠. Тому в омахі кількість варіантів дорівнює $\binom{52}{5}\binom{47}{4} = 463\,563\,500\,400$.

## 2 Використання машини станів

Розглянемо інші шляхи швидкого обчислення дискретної функції. Якщо аргумент функції може бути поділено на декілька незалежних частин, то напрошується використання машини станів. Наприклад, якщо розглядати обчислення сили руки за п'ятьма картами, руку можна природно поділити на п'ять карт. Кожна карта буде послідовно посилатися на вхід машини станів, яка закінчить свою роботу в одному зі завершальних станів, по якому можна буде отримати силу руки.

В цій статті наша увага буде зосереджена на дискретних функціях, аргументи яких діляться на фіксовану кількість частин, зокрема на покерних руках. Наша задача полягає у тому, щоб для кожної функції побудувати машину станів для її обчислення. Оскільки аргумент функції поділяється на фіксовану кількість частин, то відповідні машини також будуть мати певні особливості.

Зробимо загальні визначення, взявши за основу визначення з книги [2]:

**Визначення 1.** *Детермінованою машиною станів ми будемо називати п'ятірку $M = \{Q, \Sigma, \delta, q_0, F\}$, де*

*(1) $Q$ — скінчена множина станів;*
*(2) $\Sigma$ — скінчена множина допустимих вхідних символів;*
*(3) $\delta : Q \times \Sigma \to Q$ — функція переходів;*
*(4) $q_0 \in Q$ — початковий стан;*
*(5) $F \subseteq Q$ — множина заключних станів.*

Детермінована машина станів приймає на вхід ланцюг символів, а на виході ми отримуємо поточний стан $q \in Q$. Якщо цей стан входить у множину заключних станів ($q \in F$), то ми кажемо, що детермінована машина станів допускає цей ланцюг символів, значення $q$ може розглядатися як *результат* роботи машини станів.

Якщо розглядати задачу обчислення сили руки у покері за п'ятьма картами, то $\Sigma = C$, $F = R$, машина станів повинна приймати усі послідовності з п'яти карт без повторень, результат роботи машини станів—сила комбінації.

Проте, у разі вхідних ланцюгів фіксованої довжини ми можемо накласти на машину станів додаткові обмеження, що може зробити алгоритми синтезу та оптимізації машини станів більш простішими.

**Визначення 2.** *Суворою цибулевою машиною станів ми будемо називати таку детерміновану машину станів $M = \{Q, \Sigma, \delta, q_0, F\}$, де множина станів $Q$ може бути розбита на неперетинні частини $S_0 = \{q_0\}$, $S_1$, $S_2$, …, $S_N = F$ таким чином, що для усіх $i = 0, 1, \ldots, N-1$, та $q \in S_i$ буде виконуватися $\delta(q, \sigma) \in S_{i+1}$.*

Сувора цибулева машина для обробки кожного $i$-го символу використовує окремий шар $S_i$. Для оптимізації суворої цибулевої машині станів нам треба оптимізувати окремо кожний шар, починаючи з останнього до першого.

Повертаємося до обчислення дискретної функції $f : X \to Y$. Щоб обчислити її за допомогою суворої цибулевої машини станів, кожному елементу множини $X$ треба поставити у відповідність ланцюг символів фіксованої довжини. Це можна зробити за допомогою функції $\phi : X \to \Sigma^N$.

# 3 Ігнорування помилок

Якщо ми розглянемо функцію $\phi$, то ми може помітити, що не всі значення $\Sigma^N$ мають прообраз у $X$. Візьмемо наш улюблений приклад—обчислення сили руки у покері за п'ятьма картами. Комбінація 2♦A♦K♦Q♣2♦ входить до масиву $C^5$, але не є припустимою комбінацією з п'яти карт.

Як себе має вести машина станів у разі, коли на вхід подається неприпустима комбінація? Є два варіанти розумної поведінки. По-перше, у результаті обробці ланцюга символів машина станів може зупинитися у спеціальному стані, котрий буде індикатором помилки. По-друге, ми можемо припустити, що на вхід машини станів можуть подаватися лише припустимі ланцюги. У разі неприпустимого ланцюга машина станів може повернути будь яке значення.

Наприклад, у разі комбінації 2♦A♦K♦Q♣2♦ результатом машини станів може бути «пара двійок», а не «помилкові вхідні дані».

**Визначення 3.** *Цибулевою машиною станів ми будемо називати таку детерміновану машину станів $M = \{Q, \Sigma, \delta, q_0, F\}$, де множина станів $Q$ може бути розбита на неперетинні частини $\{\epsilon\}, S_0 = \{q_0\}$, $S_1$, $S_2$, …, $S_N = F$ таким чином, що*

*(1) $\delta(\epsilon, \sigma) = \epsilon$;*

*(2) для усіх $i = 0, 1, \ldots, N-1$, та $q \in S_i$ буде виконуватися $\delta(q, \sigma) \in S_{i+1} \cup \{\epsilon\}$.*

Стан $\epsilon$ називається *помилковим*.

## Література


[1] Дональд Кнут. Искусство программирования, том 4, A. *2015*.

[2] Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Том 1. Синтаксический анализ. Москва, Мир, *1978*.