ТЕОРІЯ КОМПІЛЯТОРІВ

Лекція 12

УНІВЕРСАЛЬНА МОВА ПРОГРАМУВАННЯ. МАШИНА МІНСЬКОГО

2020

Повний текст лекції буде розміщений на сайті baklaniv.at.ua

Алгоритмічні властивості відносин, що програмуються

Нехай задано деяку множину S. Бінарне відношення R на S називається

1. частково обчислюваним(або рекурсивно перелічуваним) тоді і тільки тоді, коли існує алгоритм, який для довільної пари s' і s'' з S працює наступним чином:

1.1. якщо $(s', s'')\in R$, то алгоритм завершує свою роботу з відповіддю «ТАК»,

1.2. якщо $(s', s'')\notin R$, то алгоритм не завершує свою роботу;

2. обчислюваних (або рекурсивним) тоді і тільки тоді, коли існує алгоритм, який для довільної пари s' і s'' з S працює наступним чином:

2.1. якщо $(s', s'') \in R$, то алгоритм завершує свою роботу з відповіддю «ТАК»,

2.2. якщо $(s', s'') \notin R$, то алгоритм завершує свою роботу з відповіддю «НІ».

Будь-яка синтаксично правильна обчислювальна програма на мові УМ (універсальна мова) складається з описів і тіла. Якщо обрана семантика спадково-кінцевих таблиць над кільцем цілих чисел (або будь-яким кільцем лишків за модулем фіксованого числа) для типів даних, то описи дозволяють визначити безліч станів, а тіло - бінарне відношення вхід-вихід на цій множині.

Для будь-якої програми α мови УМ її обчислювальна операційна семантика $[\alpha]$ є частково обчислюваних бінарним відношенням на просторі станів $SP(\alpha)$.

Виникають дві проблеми:

1) для будь-якого чи частково обчислюваного бінарного відношення R на множині SP, яке є простором станів будь-якої кінцевої сукупністю описів, існує така УМ-програма α, що $R = [\alpha]$?

2) чи є операційна обчислювальна семантика $[\alpha]$ довільної УМ-програми α рекурсивним відношенням на просторі станів $SP(\alpha)$?

Машини Марвіна Мінського

Принципове утруднення, яке не дозволяє відразу відповісти на поставлені вище два питання, полягає в тому, що поняття функції обчислюваною алгоритмічно належить до числа невизначених понять, воно носить такий же фундаментальний характер як, наприклад, поняття безлічі. Обидва ці поняття не можна визначити через інші, а можливо тільки постулювати деякі з їхніх властивостей.

Постулати (аксіоми) неформальній теорії множин ми «згадували», а тепер готові «згадати» основний постулат теорії алгоритмів - т.зв. «Теза Алонзо Черча». Для того, щоб цей постулат сформулювати, нам доведеться ввести деякі допоміжне поняття машин Марвіна Мінського.

Марвін Лі Мінський *— американський дослідник в галузі штучного інтелекту, співзасновник лабораторії штучного інтелекту Массачусетського Технологічного Інституту, автор праць з штучного інтелекту та філософії.*

У наших термінах машини М. Мінського можуть бути визначені наступним чином.

Машина М. Мінського - це будь-яка програма віртуальної УМ-машини, яка задовольняє двом легко перевіряється синтаксичним обмеженням, легко перевіряється:

- будь-яка мітка мітить не більше одного оператора, а будь-яке множина міток після «goto», «then» або «else» складається рівно з однією мітки,

- всі змінні мають цілий тип, всі вирази в операторах присвоювання мають вигляд (<змінна> +1) або (<змінна> -1), а всі умови в умовних операторах - виду <змінна> > 0.

Так як будь-яка машина М. Мінського - це програма віртуальної УМ-машини, то для кожної такої машини визначено поняття простору станів, конфігурації, спрацьовування, траси, семантичних дерев і обчислювальної семантики. Тому ми можемо визначити семантику машин М. Мінського наступним чином.

Нехай C - машина М. Мінського. Семантика [C] - це як обчислювальна семантика [C] програми віртуальної УМ-машини C.

Таким чином, кожна машина М. Мінського - це одна програма для віртуальної УМ-машини. Різниця між віртуальною УМ-машиною і машиною М. Мінського полягає в тому, що віртуальна УМ-машина - це універсальний обчислювальний пристрій, здатний виконати будь-яку програму на будь-яких вхідних даних, а машина М. Мінського - це спеціалізоване обчислювальний пристрій, що виконує тільки одну програму , але для будь-яких вхідних даних.

Нехай X і Y - деякі множини, а $f: X \to Y$ - (часткова) функція; тоді графік функції f - це множина $\{(x, f(x)): x \in X\}$. В силу того, що в машинах М. Мінського всяка мітка мітить не більше одного оператора, а після «goto», «then» або «else» складається рівно з однією мітки, будь-яка машина М. Мінського C є детермінованою в наступному сенсі: для будь-якого стану $s \in SP(C)$ розширене семантичне дерево $ET\ C(s)$ складається з єдиної траси, яка є або повної, або нескінченної початкової.

Тому обчислювальна семантика $[C]$ будь-якої машини М.Мінського C є графіком (часткової) функції $f\ C\colon (INT)\ VAR \to (INT)\ VAR$, де VAR — множина змінних в C, визначеної в такий спосіб:

$fC = \lambda\ s' \in (INT)^{VAR} = s'' \in (INT)^{VAR}$, якщо $(s', s'') \in [C]$,

$=$ не визначено в протилежному випадку.

Нехай C - машина М. Мінського, а VAR - множина змінних C.

1. Графік функції fC є частково обчислюваним бінарним відношенням на $(INT)^{VAR}$.

2. Якщо $fC\colon (INT)^{VAR} \to (INT)^{VAR}$ є тотальною функцією, то її графік є обчислюваних бінарним відношенням на $(INT)^{VAR}$.

Взагалі кажучи, для машин М. Мінського тип INT інтерпретується тільки натуральними числами (тобто всіма невід'ємними цілими числами), причому операція віднімання інтерпретується «відніманням без переходу через нуль»: λ x. λ y. (Якщо (x-y)> 0 то (x-y) інакше 0). Однак ми будемо інтерпретувати тип INT звичайним чином, тобто увесь натовп цілих чисел (що не змінює обчислювальної потужності, тому що у нас під рукою є предикат «> 0»).

Нехай VAR - кінцеве множина змінних, а $f: (INT^{VAR}) \to (INT^{VAR})$ часткова функція на цілих числах $n = |VAR|$ аргументів.

Будемо говорити, що ця функція реалізується машиною М. Мінського C з безліччю змінних VAR тоді і тільки тоді, коли $f = fC$.

Машини М. Мінського і УМ-програми

Для формулювання наступної пари тверджень нам знадобиться додаткове наступне допоміжне поняття: нехай X і Y - довільні множини, а $x \in X$ і $f: X \to Y$ - довільні елемент X і (часткова) функція; тоді $f \setminus x: (X \setminus \{x\}) \to Y$ - це така (часткова) функція, яка на всіх елементах $(X \setminus \{x\})$ збігається з f. Це поняття, зокрема, може бути застосовано до станів УМ-програм і програм віртуальної УМ-машини, які, як відомо є відображеннями з множини змінних в множини значень відповідно до типів змінних.

Існує алгоритм, який по довільній програмі β віртуальної УМ-машини будує УМ-програму $\alpha\beta$ таку, що сукупність описів $\alpha\beta$ відрізняється від описів β тільки однією додатковою цілою змінною label, якої немає в β, і

$$[\beta] = \{(s'\backslash label , s''\backslash label): s', s''\in SP (\alpha\beta) \text{ і } (s', s'')\in [\alpha\beta]\}.$$

Звідси отримуємо, що існує алгоритм, який по довільній машині М.Мінського C будує УМ-програму αC таку, що сукупність описів αC відрізняється від описів C тільки однією додатковою цілою змінною label, якої немає в C, і

$$[C] = \{(s'\backslash label, s''\backslash label): s', s'' \in SP(\alpha C) \text{ і } (s', s'') \in [\alpha C]\}.$$

Відзначимо, що останні твердження можна умовно назвати лемами про трансляцію програм віртуальної УМ-машини і машин М. Мінського в УМ-програми. Крім того їх можна було б назвати лемами про трансляцію неструктурованих (і детермінованих в разі машин М.Минская) програм в структуровані.

Теза А.Черча і повнота за А. Т'юрінгом

Теза А. Черча є постулатом теорії алгоритмів. Для машин М. Мінського його можна сформулювати наступним чином:

Нехай VAR - скінченна множина цілочисельних змінних, а $f\colon INT^{VAR} \to INT^{VAR}$ часткова функція на цілих числах. Графік функції f є частково обчислюваних ставленням тоді і тільки тоді, коли f реалізується певною машиною М. Мінського.

Зокрема, з цього постулату випливає, що тотальна функція на цілих числах має обчислюваності графік тоді і тільки тоді, коли вона реалізується певною машиною М. Мінського.

Для тих, хто цікавиться аргументацією на користь достовірності тези А. Черча, відзначимо, що найвагомішим аргументом на його користь є те, що всі варіанти формалізації поняття часткової обчислюваності на цілих числах, які були запропоновані різними математиками в 30-70 роках XX століття виявилися еквівалентними один одному і, зокрема, машинам М. Мінського.

Відзначимо так само, що формалізація М. Мінського не дивлячись на свою «програмістську» простоту не є найвідомішою формалізацією, а однією з найвідоміших формалізацій для поняття часткової обчислюваності є апарат машин А. Тьюринга. Крім того, спочатку тезу А. Черча був сформульований не для машин М. Мінського, і навіть не для машин А. Тьюринга, а для зовсім іншого формалізму для поняття часткової обчислюваності на цілих числах - т.зв. λ-числення.

Для будь-якої кінцевої сукупності описів змінних δ мовою УМ, для будь-якого частково обчислюваного бінарного відношення $R \subseteq SP(\delta) \times SP(\delta)$ існує така УМ-програма αR, яка використовує кілька додаткових цілих змінних new1, ... newn (що не зустрічаються в δ), така, що

$R = \{((s' \backslash new1 \ldots \backslash newn), (s'' \backslash new1 \ldots \backslash newn)):$

$s', s'' \in SP(\alpha R)$ і $(s', s'') \in [\alpha R]\}$.

Мова програмування називається універсальною алгоритмічною мовою, якщо

- цілі числа представимо (або можуть бути промоделювати) в термінах типів даних цієї мови,
- будь-яка його програма реалізує певний «обчислювальний» алгоритм,
- будь-який «обчислювальний» алгоритм над будь-якими типами даних цієї мови реалізуємо у вигляді програми на цій мові.

Іноді замість «універсальний алгоритмічний» говорять «повний по Тьюрингу», так як машини А. Тьюринга є одним з найбільш загальноприйнятих формалізмів для поняття алгоритму.

УМ є універсальним алгоритмічною мовою програмування тому (згідно з визначенням обчислювальної операційної семантики) цілі числа є дозволеним типом даних мови, і для будь-якої кінцевої сукупності описів змінних δ на УМ, для будь-якого бінарного відносини $R \subseteq SP(\delta) \times SP(\delta)$ маємо: R є частково обчислюваних тоді і тільки тоді, коли існує така УМ-програма α, яка використовує кілька допоміжних змінних new1, ... newn, що

$R = \{((s '\backslash\ new1\ ...\ \backslash\ newn), (s' '\backslash\ new1\ ...\ \backslash\ newn\ )): s ', s'' \in SP(\alpha)$ і $(s', s'') \in [\alpha]\}$.

Тим самим ми показали, що обчислювальні УМ-програми можуть служити формалізмом для поняття алгоритм на типах даних мови УМ і, одночасно, відповіли на перше запитання, поставлене на початку лекції. Відповідь на друге питання вже не входить в рамки даного курсу, а швидше ставитися до загальної теорії алгоритмів.

На наступній лекції ми розглянемо детерміноване програмування на універсальній мові (УМ).