Математическая логика

mk.cs.msu.ru $\rightarrow$ Лекционные курсы $\rightarrow$ Математическая логика (318, 319/2, 241, 242)

Блок 33

Задачи и проблемы
Алгоритмы
Разрешимость
М-сводимость

Лектор:
Подымов Владислав Васильевич

E-mail:
valdus@yandex.ru

ВМК МГУ, 2020/2021, весенний семестр

Вступление

Корректность и полнота метода семантических таблиц:
$\models \varphi \Leftrightarrow$ для таблицы $\langle \mid \varphi \rangle$ **существует** успешный вывод

Корректность и полнота метода резолюций:
$\models \varphi \Leftrightarrow$ для системы дизъюнктов … **существует** успешный вывод

Корректность и полнота НИП:
$\models \varphi \Leftrightarrow$ **существует** доказательство секвенции $\vdash \varphi$

А можно ли написать программу, которая сможет автоматически проверить общезначимость **любой** формулы φ логики предикатов?

Оказывается, что написать такую программу **невозможно**,
и теперь можно это строго сформулировать и обосновать
(**теорема Чёрча**)

Аналогичные утверждения для других задач, как и бо́льшая часть сопутствующих определений, известны вам из других курсов,
но всё же полезно будет всё это повторить ещё раз

Задачи и проблемы

У Васи есть 5 яблок. 2 яблока он отдал Коле.
Сколько яблок осталось у Васи?

Это пример того, что принято называть задачей

Более точно, это индивидуальная задача:
задача с конкретными входными данными и конкретным ответом (3)

У васи есть N яблок. K яблок ($K \leq N$) он отдал Коле.
Сколько яблок осталось у Васи?

Это также пример того, что принято называть задачей

Ответ к этой задаче определяется
значениями параметров (входными данными) N и K

Такие (*параметризованные*) задачи принято называть
массовыми задачами, или, по-другому, проблемами

Задачи и проблемы

Массовую задачу $\mathfrak{T}$
можно понимать как отображение $\mathfrak{T} : \mathfrak{I} \to \mathfrak{O}$, где

- $\mathfrak{I}$ — множество всевозможных входных данных (входов)
- $\mathfrak{O}$ — множество всевозможных выходных данных (выходов; ответов)
- значение $\mathfrak{T}(\mathfrak{i})$ — правильный ответ для входа $\mathfrak{i}$

Например, последняя задача о яблоках — это отображение
$\mathfrak{T} : \{(n, k) \mid n, k \in \mathbb{N}_0,\ k \leq n\} \to \mathbb{N}_0$, где

- $\mathbb{N}_0$ — множество всех неотрицательных целых чисел
- $\mathfrak{T}(n, k) = n - k$

Алгоритмы и разрешимость

Алгоритм — это особая совокупность действий,[1] согласно которой *входы* заданного множества $\mathfrak{I}$ преобразуются в *ответы* заданного множества $\mathfrak{O}$ (или не преобразуются, если применяется бесконечно много действий)[2]

Алгоритмом $\mathcal{A}$ реализуется *частично определённое* отображение $\overline{\mathcal{A}} : \mathfrak{I} \to \mathfrak{O}$ следующего вида:

- если к входу $\mathfrak{i}$ применяется конечное число действий, то $\overline{\mathcal{A}}(\mathfrak{i})$ — ответ, вычисляемый алгоритмом
  - и алгоритм останавливается (завершается) на входе $\mathfrak{i}$
- иначе значение $\overline{\mathcal{A}}(\mathfrak{i})$ не определено
  - и алгоритм не останавливается (не завершается) на входе $\mathfrak{i}$

[1] Вспоминайте из других курсов, какая именно совокупность каких действий

[2] На самом деле бывают и другие алгоритмы, согласно которым выходные данные получаются многократно, или постоянно, или понятие выходных данных отсутствует — но не будем всё переусложнять

Алгоритмы и разрешимость

Алгоритмы[1] придумываются для того,
чтобы **решать** *массовые задачи*

Алгоритм $\mathcal{A}$ решает задачу $\mathfrak{T} : \mathfrak{I} \to \mathfrak{O}$, если $\overline{\mathcal{A}} = \mathfrak{T}$, то есть

- $\mathcal{A}$ и $\mathfrak{T}$ определены
  для одинаковых множеств входных и выходных данных, и
- $\mathcal{A}$ завершается на любом входе
  и всегда вычисляет правильный ответ к задаче $\mathfrak{T}$

Массовая задача (алгоритмически) разрешима,
если существует алгоритм, решающий эту задачу,
и неразрешима, если такого алгоритма не существует

[1] Как минимум такие алгоритмы, как на предыдущем слайде

M-сводимость

Задача распознавания — это массовая задача
с множеством ответов $\{\textbf{да}, \textbf{нет}\}$ (оно же $\{1, 0\}$, оно же $\{\mathbb{t}, \mathbb{f}\}$)

Задача распознавания $\mathfrak{T}_1 : \mathfrak{I}_1 \to \{1, 0\}$ m-сводится[1]
к задаче распознавания $\mathfrak{T}_2 : \mathfrak{I}_2 \to \{1, 0\}$,
если существует алгоритм $\mathcal{A}$, такой что

- $\overline{\mathcal{A}} : \mathfrak{I}_1 \to \mathfrak{I}_2$ — всюду определённое отображение и
- для любого входа $\mathfrak{i}$ задачи $\mathfrak{T}_1$ верно $\mathfrak{T}_1(\mathfrak{i}) = \mathfrak{T}_2(\mathcal{A}(\mathfrak{i}))$

Обозначенный алгоритм $\mathcal{A}$ m-сводит задачу $\mathfrak{T}_1$ к задаче $\mathfrak{T}_2$

---

1 Many-one reducible: разным входным данным $\mathfrak{T}_1$ могут соответствовать одинаковые входные данные $\mathfrak{T}_2$. Этот вид сводимости (*должен быть*) вам известен из доказательства неразрешимости проблемы останова машин Тьюринга: эта проблема m-сводилась к проблеме самоприменимости

M-сводимость

Теорема(*об m-сводимости*). **Если задача $\mathfrak{T}_1$ m-сводима к разрешимой задаче $\mathfrak{T}_2$, то задача $\mathfrak{T}_1$ также разрешима**

Доказательство.[1]

Положим, что задача $\mathfrak{T}_2$ разрешима и что задача $\mathfrak{T}_1$ m-сводима к $\mathfrak{T}_2$

По *определению разрешимости*, существует алгоритм $\mathcal{A}$, решающий $\mathfrak{T}_2$

По *определению m-сводимости*,
существует алгоритм $\mathcal{B}$, m-сводящий $\mathfrak{T}_1$ к $\mathfrak{T}_2$

Тогда существует и алгоритм решения задачи $\mathfrak{T}_1$:
последовательно примéним $\mathcal{B}$ и $\mathcal{A}$ к входу $\mathfrak{i}$ задачи $\mathfrak{T}_1$ $\quad (\mathfrak{i} \xrightarrow{\mathcal{B}} \mathfrak{j} \xrightarrow{\mathcal{A}} \mathfrak{o})$

По выбору алгоритма $\mathcal{A}$, $\overline{\mathcal{A}}(\overline{\mathcal{B}}(\mathfrak{i})) = \mathfrak{T}_2(\overline{\mathcal{B}}(\mathfrak{i}))$

По выбору алгоритма $\mathcal{B}$, $\mathfrak{T}_2(\overline{\mathcal{B}}(\mathfrak{i})) = \mathfrak{T}_1(\mathfrak{i})$ ▼

Следствие. Если неразрешимая задача $\mathfrak{T}_1$ m-сводима к задаче $\mathfrak{T}_2$, то задача $\mathfrak{T}_2$ также неразрешима

---

[1] Это доказательство (*должно быть*) вам известно из курса, посвящённого алгоритмам. Повторим его “для профилактики”.

M-сводимость

Поясняющий пример

Представьте себе пекарню с двумя работниками:

- **Хозяин** пекарни хочет, чтобы его работники умели определять
  - по ингредиентам — можно ли из них испечь вкусный торт ($\mathfrak{T}_1$)
  - по торту — вкусный ли он ($\mathfrak{T}_2$)
- **Кондитер** $\mathcal{B}$ знает, для каких ингредиентов какой торт лучший

Положительный случай (теорема): в пекарню устроился эксперт $\mathcal{A}$, способный оценить вкус любого торта

При помощи $\mathcal{A}$ и $\mathcal{B}$ можно оценить и пригодность ингредиентов:

- Показать $\mathcal{B}$ ингредиенты ($\mathfrak{i}$) и узнать, какой торт ($\mathfrak{j}$) лучший
- Спросить у $\mathcal{A}$, вкусный ли этот торт $\mathfrak{j}$

Отрицательный случай (следствие): хозяин узнал, что пригодность ингредиентов достоверно оценить, вообще говоря, нельзя

Тогда и про оценку вкуса тортов можно забыть