Автоматическое доказательство теорем (SAT и SMT-решатели)

Занятие №13

Со времен греков говорить «математика» – значит говорить «доказательство».

Н. Бурбаки. Элементы математики

Александр Сергеевич Камкин

kamkin@ispras.ru

Инструменты: пруверы, солверы и т.п.

- Системы доказательства теорем (provers)
- Системы помощи в доказательстве (proof assistants)
- Системы проверки доказательств (proof checkers)
- Системы разрешения ограничений (solvers)

Теория (аксиоматика, дедуктивная система)

$$T = \langle \Sigma, \Phi, A, R \rangle$$

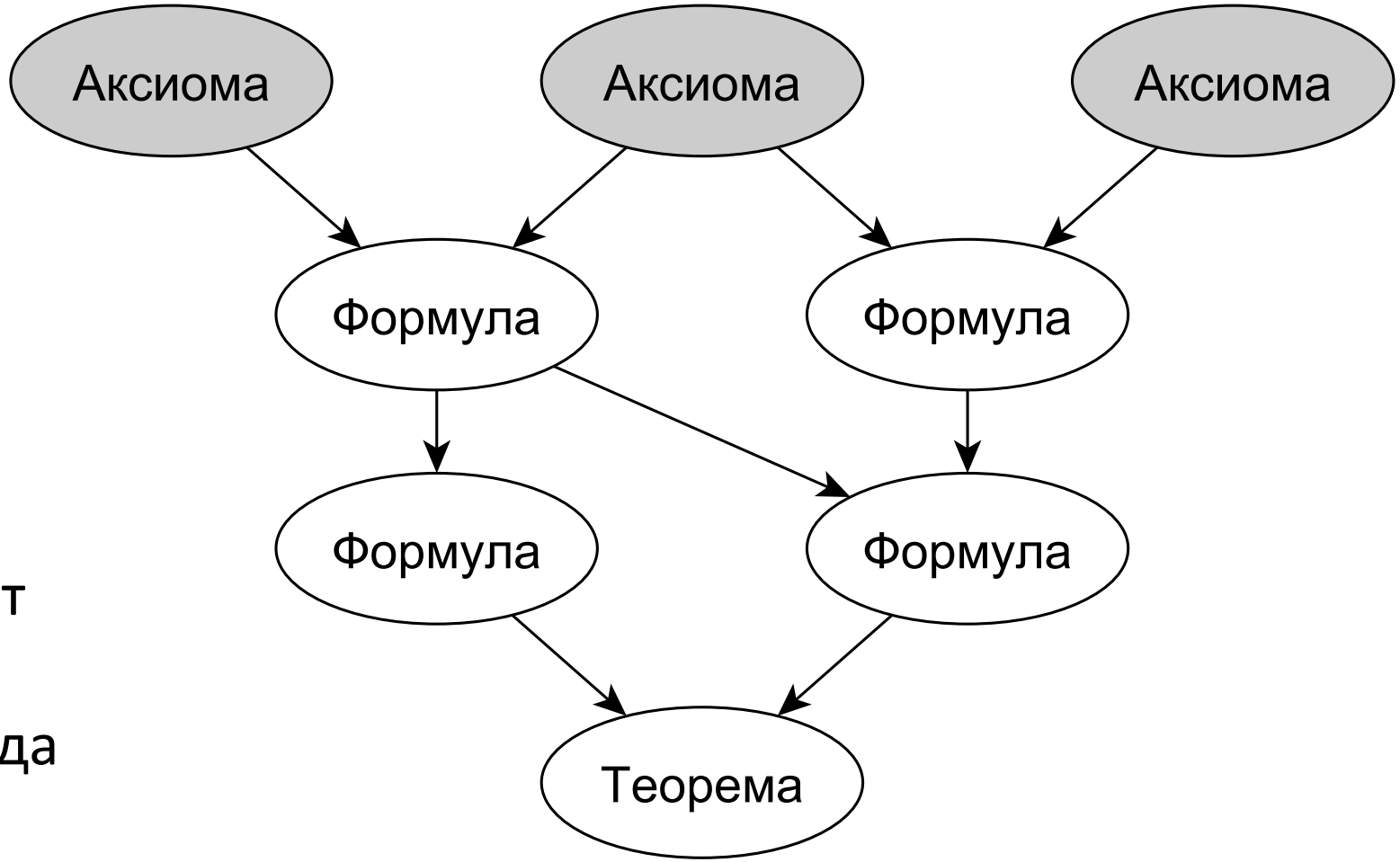
- Σ — конечный или счетный *алфавит* символов
- $\Phi \subseteq \Sigma^*$ — множество *формул*
- $A \subseteq \Phi$ — конечное или счетное множество *аксиом*
- $R = \left\{ R^i \subseteq (\Phi \times \cdots \times \Phi) \times \Phi \right\}_{i=1}^{n}$ — конечное множество *правил вывода*

Выводимость и теоремы

Теорема: $\vdash_T \varphi$

Существует $\{\varphi_i\}_{i=1}^{n}$:

- $\varphi_k = \varphi$
- φ_i – это
  - либо аксиома
  - либо результат применения правила вывода



Разрешимые теории и разрешающие процедуры

- Разрешающая процедура $\Rightarrow$ разрешимая теория
  - $\langle \varphi \in \Phi \rangle \; DP \; \langle (verdict = true) \leftrightarrow (\vdash \varphi) \rangle$
- Частичная разрешающая процедура
  - $\{ \varphi \in \Phi \} \; DP \; \{ (verdict = true) \leftrightarrow (\vdash \varphi) \}$
- Полуразрешимая теория
  - $\langle \varphi \in \Phi \wedge (\vdash \varphi) \rangle \; DP \; \langle verdict = true \rangle$
  - $\{ \varphi \in \Phi \wedge (\nvdash \varphi) \} \; DP \; \{ verdict = false \}$

Примеры разрешимых теорий

Разрешимая теория	Кем и когда доказана разрешимость
Булева алгебра	Альфред Тарский (Alfred Tarski), 1949 г.
Евклидова геометрия	Альфред Тарский (Alfred Tarski), 1949 г.
Гиперболическая геометрия	Вольфрам Швабхаузер (Wolfram Schwabhäuser), 1959 г.
Теория первого порядка с равенством	Леопольд Лёвенгейм (Leopold Löwenheim), 1915 г.
Теория первого порядка с равенством и одним унарным функциональным символом	Анджей Эренфойхт (Andrzej Ehrenfeucht), 1959 г.
Арифметика Пресбургера (теория первого порядка, описывающая натуральные числа с равенством, сложением, но без умножения)	Мойжеш Пресбургер (Mojżesz Presburger), 1929 г.
Теория равенства неинтерпретируемых функциональных символов (бесквантерная теория первого порядка с равенством и произвольным числом функциональных символов)	Вильгельм Фридрих Аккерман (Wilhelm Friedrich Ackermann), 1954 г.
Бесквантерная теория массивов (теория первого порядка с равенством, операцией получения элемента по индексу и операцией модификации элемента)	Джон Маккарти (John McCarthy), 1962 г.

Примеры неразрешимых теорий

Неразрешимая теория	Кем и когда доказана неразрешимость
Теория первого порядка с равенством и либо одним предикатным символом арности не меньше двух, либо одним функциональным символом арности не меньше двух, либо двумя унарными функциональными символами	Борис Абрамович Трахтенброт, 1953 г.
Арифметика натуральных чисел	Джон Баркли Россер (John Barkley Rosser), 1936 г.
Арифметика целых чисел	Альфред Тарский (Alfred Tarski), Анджей Мостовский (Andrzej Mostowski), 1949 г.
Арифметика рациональных чисел	Джулия Холл Робинсон (Julia Hall Robinson), 1949 г.
Теории групп и колец	Альфред Тарский (Alfred Tarski), 1949 г.
Теория полей	Джулия Холл Робинсон (Julia Hall Robinson), 1949 г.
Арифметика Робинсона (фрагмент арифметики Пеано без схемы индукции)	Рафаэль Митчел Робинсон (Raphael Mitchel Robinson), 1950 г.
Общая теория массивов (теория первого порядка с равенством, операцией получения элемента по индексу и операцией модификации элемента)	Джон Маккарти (John McCarthy), 1962 г.

Общезначимость и выполнимость

- Выполнимость – существование модели
- Общезначимость – истинность во всех интерпретациях
- Формула φ общезначима $\Leftrightarrow \neg\varphi$ невыполнима (UNSAT)
- Формула φ выполнима (SAT) $\Leftrightarrow \neg\varphi$ необщезначима

Логика высказываний: КНФ-выполнимость

- **Вход: КНФ (клаузальная форма)**
  - Формула – множество клауз
  - Клауза (дизъюнкт) – множество литералов
  - Литерал (буква) – атом или его отрицание
  - Атом – элементарное высказывание
- **Примеры клаузальных форм**
  - $\emptyset \equiv true$ — пустая формула
  - $\{\square\} \equiv false$ — формула, состоящая из пустой клаузы
  - $\{pr, \bar{q}\bar{p}q, p\bar{p}q\} \equiv \{\{p, r\}, \{\neg q, \neg p, q\}, \{p, \neg p, q\}\}$

Кодировка Цейтина: формула → КНФ

- Эквивалентные преобразования к КНФ – тупик!
  - Размер КНФ экспоненциально зависит от размера ДНФ
- Кодировка Цейтина (1968)
  - Получаемая КНФ равносильна исходной формуле
  - Могут потребоваться дополнительные переменные

Г.С. Цейтин
(род. 1936)

$$encode(\varphi) \equiv p_\varphi \wedge link(\varphi, p_\varphi)$$

$$link(\psi \circ \chi, p_{\psi\circ\chi}) \equiv cnf\left(p_{\psi\circ\chi} \leftrightarrow (p_\psi \circ p_\chi)\right) \wedge link(\psi, p_\psi) \wedge link(\chi, p_\chi)$$

$link(\gamma, p_\gamma) \equiv true$, если γ — литерал (в этом случае $p_\gamma \equiv \gamma$)

Кодировка Цейтина: базовые преобразования

Type	Operation	CNF Sub-expression
 AND	$C = A \cdot B$	$(\overline{A} \vee \overline{B} \vee C) \wedge (A \vee \overline{C}) \wedge (B \vee \overline{C})$
 NAND	$C = \overline{A \cdot B}$	$(\overline{A} \vee \overline{B} \vee \overline{C}) \wedge (A \vee C) \wedge (B \vee C)$
 OR	$C = A + B$	$(A \vee B \vee \overline{C}) \wedge (\overline{A} \vee C) \wedge (\overline{B} \vee C)$
 NOR	$C = \overline{A + B}$	$(A \vee B \vee C) \wedge (\overline{A} \vee \overline{C}) \wedge (\overline{B} \vee \overline{C})$
 NOT	$C = \overline{A}$	$(\overline{A} \vee \overline{C}) \wedge (A \vee C)$
 XOR	$C = A \oplus B$	$(\overline{A} \vee \overline{B} \vee \overline{C}) \wedge (A \vee B \vee \overline{C}) \wedge (A \vee \overline{B} \vee C) \wedge (\overline{A} \vee B \vee C)$
 XNOR	$C = \overline{A \oplus B}$	$(\overline{A} \vee \overline{B} \vee C) \wedge (A \vee B \vee C) \wedge (A \vee \overline{B} \vee \overline{C}) \wedge (\overline{A} \vee B \vee \overline{C})$

Кодировка Цейтина: пример $(p \oplus q) \to \bar{r}$

- Новая переменная $f \leftrightarrow \left((p \oplus q) \to \bar{r}\right)$
- Новая переменная $h \leftrightarrow (p \oplus q)$
- $h \leftrightarrow (p \oplus q)$
  - $\left(\bar{p} \vee \bar{q} \vee \bar{h}\right) \wedge (\bar{p} \vee q \vee h) \wedge (p \vee \bar{q} \vee h) \wedge \left(p \vee q \vee \bar{h}\right) \equiv \{\bar{p}\bar{q}\bar{h}, \bar{p}qh, p\bar{q}h, pq\bar{h}\}$
- $f \leftrightarrow (h \to \bar{r}) \Leftrightarrow f \leftrightarrow \left(\bar{h} \vee \bar{r}\right)$
  - $\left(\bar{h} \vee \bar{r} \vee \bar{f}\right) \wedge (h \vee f) \wedge (r \vee f) \equiv \{\bar{h}\bar{r}\bar{f}, hf, rf\}$
- $\boldsymbol{\{f, \bar{h}\bar{r}\bar{f}, hf, rf, \bar{p}\bar{q}\bar{h}, \bar{p}qh, p\bar{q}h, pq\bar{h}\}}$

Метод резолюций для логики высказываний

Вход: клаузальная форма F, не содержащая клаузы □
Выход: *true* (F выполнима) или *false* (F невыполнима)

```
while F содержит «неспаренные» сталкивающиеся клаузы do
  выбрать из F новую пару сталкивающихся клауз C1 и C2;
  C := Res(C1, C2);
  if C = □ then
    return false
  end;
  F := F ∪ {C};
end;

return true
```

$$Res(C_1, C_2) = (C_1 \setminus \{l\}) \cup (C_2 \setminus \{l^C\})$$

Метод резолюций: пример

$$p \wedge (\neg p \vee q) \wedge \neg r \wedge (\neg p \vee \neg q \vee r) \equiv \{p, \bar{p}q, \bar{r}, \bar{p}\bar{q}r\}$$

- $C_1 = p$
- $C_2 = \bar{p}q$
- $C_3 = \bar{r}$
- $C_4 = \bar{p}\bar{q}r$
- $C_5 = Res(C_3, C_4) = Res(\bar{r}, \bar{p}\bar{q}r) = \bar{p}\bar{q}$
- $C_6 = Res(C_2, C_5) = Res(\bar{p}q, \bar{p}\bar{q}) = \bar{p}$
- $C_7 = Res(C_1, C_6) = Res(p, \bar{p}) = \square$

UNSAT

Алгоритм DPLL (1962)

- Удаление тавтологий
  - Удаляются все клаузы, содержащие контрарные пары
- Распространение единицы
  - Если есть единичная клауза $\{l\}$
    - Удаляются все клаузы, содержащие литерал l
    - Из оставшихся удаляются вхождения контрарных литералов l^c
- Исключение чистых литералов
  - Если l – чистый литерал, т.е. литерал, входящий с одним «знаком»
    - Удаляются все клаузы, содержащие литерал l

Алгоритм DPLL: псевдокод

```
Функция DPLL
Вход:  КНФ F
Выход: true ⇔ F выполнима
  F′ := удалить_тавтологии(F);
  DPLL′(F′, ∅)

Функция DPLL′
Вход:  КНФ F, частичная интерпретация I
Выход: true ⇔ F выполнима

F′ := F;
I′ := I;

while F′ содержит единичные клаузы do
  выбрать единичную клаузу {p^x};
  F′ := распространить_единицу(F′, p^x);
  I′ := I′[p := x]
end;

while F′ содержит чистые литералы do
  выбрать чистый литерал p^x;
  F′ := исключить_чистый_литерал(F′, p^x);
  I′ := I′[p := x]
end;

// конфликт
if □ ∈ F′ (F′ содержит клаузу, ложную в I′) then
  return false;
end;

// решение
if F′ = ∅ (F′ истинна в I′) then
  return true;
end;

выбрать элементарное высказывание p, входящее в F′;
выбрать значение истинности x ∈ {true, false};

// вычисления выполняются по правилам короткой логики
return DPLL′(F′[p := x],  I′[p := x]) ∨
       DPLL′(F′[p := ¬x], I′[p := ¬x]);
```

Метод резолюций: формула → клаузы

- Приведение к предваренной нормальной форме (ПНФ)
  - переименование связанных переменных
  - перемещение кванторов в начало формулы
- Преобразование матрицы (бескванторной части) к КНФ
- Приведение к скулемовской стандартной форме (ССФ)
  - элиминация кванторов существования (скулемовские функции)
- Запись матрицы ССФ в клаузальной форме

Метод резолюций: унификация, резольвента

- Подстановка θ – *унификатор* формул $\{F_1, \dots, F_n\}$,
  - если $F_1\theta = \cdots = F_n\theta$
- Унификатор θ – *наиболее общий унификатор* (*НОУ*),
  - если любой другой унификатор $\eta = \theta\mu$, где μ – некоторая подстановка
- $Res(C_1, C_2) = (C_1\theta \setminus L_1\theta) \cup (C_2\theta \setminus L_2\theta)$ – резольвента
  - C_1 и C_2 — клаузы, не имеющие общих переменных
  - $L_1 = \{l_{1,i}\}_{i=1}^{n}$ и $L_2 = \{l_{2,i}\}_{i=1}^{m}$ — их подмножества
  - $L_1 \cup \left(L_2^C = \left\{l_{2,i}^C\right\}_{i=1}^{m}\right)$ могут быть унифицированы
  - θ — НОУ $L_1 \cup L_2^C$

Алгоритм унификации Мартелли-Монтанари

1. Преобразовать уравнение $\boldsymbol{t = x}$, где $\boldsymbol{t}$ не переменная, а $\boldsymbol{x}$ — переменная, к виду $\boldsymbol{x = t}$
2. Удалить уравнение $\boldsymbol{x = x}$, где $\boldsymbol{x}$ — переменная
3. Для уравнения $\boldsymbol{t_1 = t_2}$, где $\boldsymbol{t_1}$ и $\boldsymbol{t_2}$ — термы, не являющиеся переменными
    a. Если внешние функциональные символы $\boldsymbol{t_1}$ и $\boldsymbol{t_2}$ различаются
        i. **система не имеет решения**
    b. Иначе, то есть если уравнение имеет вид $\boldsymbol{f\left(t_1^1, \dots, t_1^k\right) = f(t_2^1, \dots, t_2^k)}$,
        i. заменить его на систему уравнений $\boldsymbol{\{t_1^i = t_2^i\}_{i=1}^k}$
4. Для уравнения $\boldsymbol{x = t}$, где $\boldsymbol{x}$ — переменная, имеющая другие вхождения
    a. Если переменная $\boldsymbol{x}$ входит в терм $\boldsymbol{t}$
        i. **система не имеет решения**
    b. Иначе — заменить все другие вхождения переменной $\boldsymbol{x}$ на терм $\boldsymbol{t}$

Теория равенства

- Без кванторов
- Произвольные функциональные символы
- Один двуместный предикатный символ $=$
  - Рефлексивность
  - Симметричность
  - Транзитивность
  - Подстановочность

$$\left(\bigwedge_{i=1}^{k}\left(t_1^i = t_2^i\right)\right) \rightarrow f\left(t_1^1, \ldots, t_1^k\right) = f\left(t_2^1, \ldots, t_2^k\right)$$

- Разрешимость (Ackermann, 1954)
- Разрешающая процедура (Nelson & Oppen, 1976)

Ориентированный помеченный граф

$$G = \langle V, \Sigma, L, \delta, [] \rangle$$

- V — конечное множество вершин
- Σ — алфавит пометок
- $L: V \to \Sigma$ — функция разметки вершин
- $\delta: V \to \mathbb{N}$ — степени исхода вершин
- $[]: V \times \mathbb{N} \to V$ — занумерованное множество дуг
  - если $v \in V$ и $1 \leq i \leq \delta(v)$:
    - $v[i]$ — конец i-ой дуги, исходящей из v

Конгруэнтность и конгруэнтное замыкание

- Вершины v и u *конгруэнтны* по отношению R, если:
  - $L(v) = L(u)$ и $\delta(v) = \delta(u)$
  - для всех $1 \leq i \leq \delta(v)$:
    - либо $v[i] = u[i]$
    - либо $(v[i], u[i]) \in R$
- Отношение R *замкнуто относительно конгруэнтности*, если для всех вершин v и u, конгруэнтных по отношению R,
  - имеет место принадлежность $(v, u) \in R$
- *Конгруэнтное замыкание* отношения R – минимальное отношение эквивалентности, содержащее R и замкнутое относительно конгруэнтности

Теория равенства: разрешающая процедура

$$(t_1 = t_1' \ ) \wedge \cdots \wedge \left(t_p = t_p'\right) \wedge (s_1 \neq s_1') \wedge \cdots \wedge (s_q \neq s_q') \ \text{SAT?}$$

- Построить граф G над множеством термов
  - Для каждого элементарного терма v, где v — переменная или константа
    - Положить $L(v) = v$
  - Для каждого терма $f(v_1, \ldots, v_k)$, где $v_1, \ldots, v_k$ — некоторые термы
    - Положить $L(f(v_1, \ldots, v_k)) = f$
    - Добавить дуги $f(v_1, \ldots, v_k)[i] = v_i$, для всех $i \in \{1, \ldots, k\}$
- Построить конгруэнтное замыкание R^* отношения $R = \{ (t_i, t_i') \mid i \in \{1, \ldots, p\} \}$:
  - Положить R^* равным R
  - Пока возможно, добавить в R^*пару $(v, u) \notin R^*$, если истинно хотя бы одно из условий
    - v и u конгруэнтны по отношению R^*
    - $(u, v) \in R^*$
    - $(v, w) \in R^*$ и $(w, u) \in R^*$ для некоторой вершины w
- SAT $\Leftrightarrow$ для всех $i \in \{1, \ldots, q\}$ $(s_i, s'_i) \notin R^*$

Комбинирование теорий

- $F \equiv (x \leq y) \wedge (y + z \leq x) \wedge (z \geq 0) \wedge \Big(f\big(f(x) - f(y)\big) \neq f(z)\Big)$
- $F_1 \equiv (x \leq y) \wedge (y + z \leq x) \wedge (z \geq 0) \wedge (w_1 = w_2 - w_3)$
- $F_2 \equiv (w_2 = f(x)) \wedge (w_3 = f(y)) \wedge \big(f(w_1) \neq f(z)\big)$
- $F_2' \equiv (w_2 = f(x)) \wedge (w_3 = f(y)) \wedge \big(f(w_1) \neq f(z)\big) \wedge (x = y)$
- $F_1' \equiv (x \leq y) \wedge (y + z \leq x) \wedge (z \geq 0) \wedge (w_1 = w_2 - w_3) \wedge (w_2 = w_3)$
- $F_2'' \equiv (w_2 = f(x)) \wedge (w_3 = f(y)) \wedge \big(f(w_1) \neq f(z)\big) \wedge (x = y) \wedge (w_1 = z)$

Практикум №3

- Реализуйте на языке программирования C/C++ алгоритм DPLL
  - Без рекурсии
  - Вход – файл с КНФ в формате DIMACS
  - https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html