Логическое программирование высшего порядка

Екатерина Вербицкая

Лаборатория языков инструментов JetBrains

14 сентября 2020

Логическое программирование

Декларативное программирование, основанное на формальной логике

Логическое программирование: пример

$$\forall x.human(x) \rightarrow mortal(x)$$

$$human(Socrates)$$

Смертен ли Сократ?

Логическое программирование: пример

$$\forall xz.\exists y.child(x, y) \wedge child(y, z) \rightarrow grandchild(x, z)$$

Логическое программирование: пример

$$\forall hxyz.(append\ x\ y\ x \vee (append\ x\ y\ z \rightarrow append\ (h :: x)\ y\ (h :: z)))$$

Как это работает: дизъюнкты Хорна

$$G ::= \top \mid A \mid G \wedge G \mid G \vee G \mid \exists_\tau x.G$$
$$D ::= A \mid G \to D \mid D \wedge D \mid \forall_\tau x.D$$

- $\top$ — истина
- A — атом
- G — цель
- D — дизъюнкт Хорна (правило логического вывода)

Как это работает: логический вывод (backchaining)

$$\frac{}{\Sigma; \mathcal{P} \longrightarrow \top}$$

$$\frac{\Sigma; \mathcal{P} \longrightarrow B_1}{\Sigma; \mathcal{P} \longrightarrow B_1 \vee B_2} \qquad \frac{\Sigma; \mathcal{P} \longrightarrow B_2}{\Sigma; \mathcal{P} \longrightarrow B_1 \vee B_2}$$

$$\frac{\Sigma; \mathcal{P} \longrightarrow B_1 \qquad \Sigma; \mathcal{P} \longrightarrow B_2}{\Sigma; \mathcal{P} \longrightarrow B_1 \wedge B_2} \qquad \frac{\Sigma; \mathcal{P}, B_1 \longrightarrow B_2}{\Sigma; \mathcal{P} \longrightarrow B_1 \to B_2}$$

$$\frac{\Sigma; \mathcal{P} \longrightarrow B[t/x] \qquad \Sigma; \varnothing \vdash t : \tau}{\Sigma; \mathcal{P} \longrightarrow \exists_\tau x.B} \qquad \frac{c : \tau, \Sigma; \mathcal{P} \longrightarrow B[c/x]}{\Sigma; \mathcal{P} \longrightarrow \forall_\tau x.B}$$

Как это работает: логический вывод (backchaining)

$$\frac{\Sigma; \mathcal{P} \xrightarrow{D} A}{\Sigma; \mathcal{P} \longrightarrow A} \qquad \frac{}{\Sigma; \mathcal{P} \xrightarrow{A} A} \qquad \frac{\Sigma; \mathcal{P} \xrightarrow{D} A \qquad \Sigma; \mathcal{P} \longrightarrow G}{\Sigma; \mathcal{P} \xrightarrow{G \to D} A}$$

$$\frac{\Sigma; \mathcal{P} \xrightarrow{D_1} A}{\Sigma; \mathcal{P} \xrightarrow{D_1 \wedge D_2} A} \qquad \frac{\Sigma; \mathcal{P} \xrightarrow{D_2} A}{\Sigma; \mathcal{P} \xrightarrow{D_1 \wedge D_2} A} \qquad \frac{\Sigma; \mathcal{P} \xrightarrow{D[t/x]} A \qquad \Sigma; \varnothing \vdash t : \tau}{\Sigma; \mathcal{P} \xrightarrow{\forall_\tau x. D} A}$$

Или если D имеет вид $\forall_{\tau_1} x_1, \ldots, \forall_{\tau_m} x_m.\ A_1 \wedge \cdots \wedge A_n \to A_0$:

$$\frac{\Sigma; \mathcal{P} \longrightarrow A_1\theta \qquad \ldots \qquad \Sigma; \mathcal{P} \longrightarrow A_n\theta}{\Sigma; \mathcal{P} \xrightarrow{D} A}\ , A = A_0\theta$$

Унификация

Даны два терма t, s

Задача: найти подстановку на свободных переменных термов (унификатор) θ, такую что

$$t\theta = s\theta$$

Алгоритм унификации

Будем искать подстановку как множество уравнений $\mathcal{E} = \{t_i = s_i\}$

- Упрощение термов: $(f\ t_1 \dots t_n = g\ s_1 \dots s_m) \in \mathcal{E}$
  - Если f, g — различные константы, то $\mathcal{E} = \bot$
  - Иначе заменяем уравнение в $\mathcal{E}$ на множество $t_1 = s_1, \dots, t_n = s_n$
- Переориентация: $(t = x) \in \mathcal{E}$
  - Если t — терм, x — переменная, заменяем в $\mathcal{E}$ уравнение на $x = t$
- Элиминация переменных: $(x = t) \in \mathcal{E}$, x входит в какое-то уравнение
  - Если x входит в t, $t \equiv x$, то удаляем уравнение из $\mathcal{E}$
  - Иначе, если x входит в t, то $\mathcal{E} = \bot$
  - Иначе, подставляем t вместо x во всех уравнениях в $\mathcal{E}$

Унификация: пример

$$\{node\ El\ T\ T = node\ 1\ (node\ 2\ emp\ emp)\ (node\ 2\ emp\ emp)\}$$

$$\{El = 1, T = node\ 2\ emp\ emp, T = node\ 2\ emp\ emp\}$$

$$\{El = 1, T = node\ 2\ emp\ emp, node\ 2\ emp\ emp = node\ 2\ emp\ emp\}$$

$$\{El = 1, T = node\ 2\ emp\ emp, 2 = 2, emp = emp, emp = emp\}$$

$$\{El = 1, T = node\ 2\ emp\ emp\}$$

Унификация: пример

$$\{node\ El\ T\ \ T = node\ 1\ (node\ 2\ emp\ emp)\ (node\ 3\ emp\ emp)\}$$

$$\{El = 1, T = node\ 2\ emp\ emp, T = node\ 3\ emp\ emp\}$$

$$\{El = 1, T = node\ 2\ emp\ emp, node\ 2\ emp\ emp = node\ 3\ emp\ emp\}$$

$$\{El = 1, T = node\ 2\ emp\ emp, 2 = 3, emp = emp, emp = emp\}$$

$$\bot$$

Логическое программирование над абстрактным синтаксом

$$\frac{\langle x, \tau \rangle \in \Gamma}{\Gamma \vdash x : \tau} \qquad \frac{\langle x, \tau_1 \rangle, \Gamma \vdash B : \tau}{\Gamma \vdash \lambda x.B : \tau_1 \to \tau} \qquad \frac{\Gamma \vdash m : \tau_1 \to \tau \qquad \Gamma \vdash n : \tau_1}{\Gamma \vdash m\ n : \tau}$$

FOAS vs HOAS

- First-order abstract syntax
  - Структурное представление термов
  - Переменные представляются конкретными значениями (строками, числами, ...)
  - Необходимо реализовывать capture-avoiding substitution
    - $\lambda x.y[x/y] \to \lambda x.x$ — неправильно
    - $\lambda x.y[x/y] \to \lambda z.x$ — сложно
  - Не очень сложно реализовать сравнение термов на равенство
- Higher-order abstract syntax
  - Структурное представление термов
  - Переменные и связывания представляются силами метаязыка
  - Нет необходимости в capture-avoiding substitution: об этом позаботится метаязык
  - Проверка термов на равенство затруднена

HOAS в логическом программировании

- Использование предикатов высшего порядка
- Использование функций высшего порядка

Миграция связываний

```
type app tm -> (tm -> tm)
type abs (tm -> tm) -> tm
```

$$\forall M.term(M) \rightarrow \forall N.term(N) \rightarrow term(app(M, N))$$
$$\forall B.(\forall x.term(x) \rightarrow term(B\ x)) \rightarrow term(abs(B))$$

```
?- term (abs y\ app y y).
?- pi x\ term x => term ((y \ app y y) x).
?- pi x \ term x => term (app x x)
?- term (app c c).
```

Идиома миграции связываний

Чтобы продолжить анализировать под связыванием:

1. Примени связывание к новой переменной под квантором всеобщности
2. Используй импликацию, в которой посылка использует эту новую переменную, чтобы сделать выводы о терме

Higher-order hereditary Harrop formulas

$$G ::= \top \mid A \mid G \wedge G \mid G \vee G \mid \exists_\tau x.G \mid D \to G \mid \forall_\tau x.G$$
$$D ::= A_r \mid G \to D \mid D \wedge D \mid \forall_\tau x.D$$

- $\top$ — истина
- A — атом
- A_r — жесткий (rigid) атом
  - Rigid atom: $h\ t_1 \ldots t_n$, где h — (не логическая) константа
  - Flexible atom: $h\ t_1 \ldots t_n$, где h — переменная
- G — цель
- D — дизъюнкт Хорна (правило логического вывода)

Унификация высшего порядка: пример

$$\forall a\ \exists F\ [F\ a = g\ a\ a]$$

$$F \in \{\lambda x.g\ a\ a,\ \lambda x.g\ a\ x,\ \lambda x.g\ x\ a,\ \lambda x.g\ x\ x\}$$

Унификация высшего порядка: пример

$$\forall a\ \exists F\ [F\ a = g\ a\ a]$$

$$F \in \{\lambda x.g\ a\ a,\ \lambda x.g\ a\ x,\ \lambda x.g\ x\ a,\ \lambda x.g\ x\ x\}$$

$$\exists F\ \forall a\ [F\ a = g\ a\ a]$$

$$F \in \{\lambda a.g\ a\ a\}$$

Унификация высшего порядка: пример

Предположим, в мире есть только $u : i \to i$, $v : i \to i$
$\lambda w.w$ соответствует пустой строке, $\lambda w.u(v(u\ w))$ — строке “uvu”

$$\exists F\ \exists G[\lambda w.F(Gw) = \lambda w.u(v(u\ w))]$$

$$F = \lambda w.u(v(u\ w)),\ G = \lambda w.w$$
$$F = \lambda w.u(v\ w),\ G = \lambda w.u\ w$$
$$F = \lambda w.u\ w,\ G = \lambda w.v(u\ w)$$
$$F = \lambda w.w,\ G = u(v(u\ w))$$

Унификация высшего порядка: пример

Предположим, в мире есть только $u : i \to i$, $v : i \to i$
$\lambda w.w$ соответствует пустой строке, $\lambda w.u(v(u\ w))$ — строке “uvu”

$$\exists F[\lambda w.u(F(uw)) = \lambda w.u(v(v(u\ w)))]$$

$$F = \lambda w.v(v\ w)$$

Унификация высшего порядка: пример

Предположим, в мире есть только $u : i \to i$, $v : i \to i$
$\lambda w.w$ соответствует пустой строке, $\lambda w.u(v(u\ w))$ — строке “uvu”

$$\exists F[\lambda w.u(F\ w) = \lambda w.F(u\ w)]$$

$$F \in \{\lambda w.w, \lambda w.u\ w, \lambda w.u(u\ w), \dots\}$$

Унификация высшего порядка: пример

У следующих задач унификаторов нет
c, d — константы, F, G — переменные

$$\lambda x.d(c(F\ x)) = \lambda x.c(d(G\ x))$$

$$\lambda x.x(F\ x) = \lambda x.(c(G\ x))$$

$$\lambda x.\lambda y.x(F\ x\ y) = \lambda x.\lambda y.y(G\ x\ y)$$

$$\lambda x.\lambda y.x(F\ x\ y) = \lambda x.\lambda y.G\ y\ y$$

Унификация высшего порядка: неразрешимость

Унификация высшего порядка неразрешима в общем случае: задача соответствия Поста может быть сведена к унификации

Унификация высшего порядка: rigid-rigid уравнения

Rigid терм — $\lambda x_1 \ldots \lambda x_n.h\ t_1 \ldots t_m$, где $h = x_i$ или h находится под квантором всеобщности

- Если $c\ t_1 \ldots t_m = d\ s_1 \ldots s_n$, где $c \neq d$, то унификация невозможна
- Если $c = c$, то заменяем это выражение на $\top$
- Если $c\ t_1 \ldots t_n = c\ s_1 \ldots s_n$, то заменяем уравнение на $t_1 = s_1 \wedge \cdots \wedge t_n = s_n$

Унификация высшего порядка: flexible-rigid уравнения

Flexible терм — $\lambda x_1 \ldots \lambda x_n.F\ t_1 \ldots t_m$, где F находится под квантором существования

$$F\ t_1 \ldots t_n = c\ s_1 \ldots s_m$$

- Подстановка-имитация
  - F связан в скоупе, где связан c
  - $F = \lambda x_1 \ldots \lambda x_n.c\ (H_1\ x_1 \ldots x_n) \ldots (H_m\ x_1 \ldots x_n)$
  - Добавляем кванторы для $H_1 \ldots H_m$ на место квантора для F
- Подстановка-проекция
  - $F = \lambda x_1 \ldots \lambda x_n.x_i\ (H_1\ x_1 \ldots x_n) \ldots (H_m\ x_1 \ldots x_n)$

Унификация высшего порядка: пример

$$\forall a.\forall g.\exists F[F\ a = g\ a\ a]$$

$$F \in \{\lambda x.g\ (H_1\ x)\ (H_2\ x),\ \lambda x.x\}$$

Подставляем второй вариант, нормализуем, получается неунифицирующееся уравнение

$$\forall a.\forall g.[a = g\ a\ a]$$

Унификация высшего порядка: пример

$$\forall a.\forall g.\exists F[F\ a = g\ a\ a]$$

$$F \in \{\lambda x.g\ (H_1\ x)\ (H_2\ x),\ \lambda x.x\}$$

Подставляем первый вариант

$$\forall a.\forall g.\exists H_1.\exists H_2.[g\ (H_1\ a)\ (H_2\ a) = g\ a\ a]$$

Упрощаем

$$\forall a.\forall g.\exists H_1.\exists H_2.[H_1\ a = a \wedge H_2\ a = a]$$

Унификация высшего порядка: пример

$$\forall a.\forall g.\exists F[F\ a = g\ a\ a]$$

$$\forall a.\forall g.\exists H_1.\exists H_2.[H_1\ a = a \wedge H_2\ a = a]$$

$$H_1 \in \{\lambda x.x,\ \lambda x.a\}$$

Подставляем любой из двух вариантов, получаем

$$\forall a.\forall g.\exists H_2.[\top \wedge H_2\ a = a]$$

Собираем все вместе, получаем

$$F \in \{\ \lambda x.g\ a\ a,\ \lambda x.g\ a\ x,\ \lambda x.g\ x\ a,\ \lambda x.g\ x\ x\}$$

Унификация высшего порядка: flexible-flexible

$$\forall a.\forall g.\exists X.\exists Y.[X\ a = g\ (F\ a)]$$

$$X = \lambda x.g\ (H\ x)$$

$$\forall a.\forall g.\exists H.\exists Y.[H\ a = Y\ a]$$

- Для примитивного типа τ создать уникальную переменную H^{τ}
- Для каждого типа $\tau_1 \to \cdots \to \tau_n \to \sigma$ назовем каноническим термом $\lambda x_1 \ldots \lambda x_n.H^{\sigma}$
- Используем канонический терм подходящего типа для flexible термов

Унификация высшего порядка: нетерминируемость

$$\forall g.\exists F.\forall x.[F\ (g\ x) = g\ (F\ x)]$$

$$\forall g.\exists H.\forall x.[H\ (g\ x) = g\ (H\ x)]$$

$$F \in \{\lambda x.x,\ \lambda x.g\ x,\ \lambda x.g\ (g\ x), \dots\}$$

Язык L_λ: кошмарные определения

Вхождение подформулы C в формулу B называется *положительным*, если оно находится слева от четного количества импликаций в B. Иначе — *негативным*

- Вхождение связанной переменной в цель G называется *essentially universal*, если оно связано позитивным квантором всеобщности, негативным квантором существования или λ-абстракцией
- Вхождение связанной переменной в цель G называется *essentially existential*, если оно не является essentially universal
- Вхождение связанной переменной в D называется *essentially universal*, если оно связано негативным квантором всеобщности, позитивным квантором существования или λ-абстракцией
- Вхождение связанной переменной в D называется *essentially existential*, если оно не является essentially universal

Язык L_λ: основная характеристика

Запрещена квантификация над предикатами, а также: в каждом подтерме $x\ t_1 \ldots t_n, n \geq 0$, в котором x является essentially existential, все t_i должны быть различными existential universal переменными, связанными внутри скоупа x

Это означает, что если x будет инстанциировано термом t, то результирующие β-редексы будут иметь вид $t\ y_1 \ldots y_n$, где все y_i будут связанными, а значит $t = \lambda y_1 \ldots y_n.t'$, и $(\lambda y_1 \ldots y_n.t')\ y_1 \ldots y_n = t'$

Higher-order pattern unification

Отслеживаем выполнение основной характеристики L_λ в процессе выполнения унификации

Работа с rigid-rigid уравнениями не изменяется

В случае flexible-rigid уравнений все, кроме одной, подстановки сразу же ломаются

Higher-order pattern unification: flexible-rigid

$$F\ c_1 \ldots c_n = c\ t_1 \ldots t_m$$

- Если F связано в скоупе квантора, который связывает c, то c не совпадает ни с одним из $c_1, \ldots, c_n$: ни одна подстановка-проекция не завершится успехом
- Если связано в скоупе квантора, который связывает F, тогда подстановка-имитация не завершится успехом, и только i-ая подстановка-проекция может выжить: когда $c = c_i$

Higher-order pattern unification: flexible-flexible

$$F\ c_1 \ldots c_n = G\ d_1 \ldots d_m$$

- c_i, d_j, все связаны в скоупе F, G
- Если F и G — разные переменные, то $F = \lambda c_1. \ldots \lambda c_n.H\ e_1 \ldots e_l$, $G = \lambda d_1. \ldots \lambda d_n.H\ e_1 \ldots e_l$, где $e_1, \ldots, e_l$ — общие переменные
- Если F и G — одинаковые переменные, то $n = m$, $F = \lambda c_1. \ldots \lambda c_n.H\ e_1 \ldots e_l$, $G = \lambda d_1. \ldots \lambda d_n.H\ e_1 \ldots e_l$, где $e_1, \ldots, e_l$ — такие переменные, что $c_i = d_i = e_i$

В результате получится mgu.

Higher-order pattern unification: терминируемость

$$\forall f.\exists X[X = f\ X]$$

$$\forall f.\exists H[H = f\ H]$$

В этом случае спасет occurs-check

Higher-order pattern unification: примеры

$$\forall f.\forall g.\exists U.\exists V.\forall w.\forall x.\forall y.[f\ (U\ x\ y) = f\ (g(V\ y\ w))]$$

$$\forall g.\exists U.\exists V.\forall w.\forall x.\forall y.[U\ x\ y = g(V\ y\ w)]$$

$$U = \lambda x.\lambda y.g\ (V'\ y), V = \lambda y.\lambda w.V'\ y$$

Следующая задача не имеет унификатора, потому что срабатывает occurs-check

$$\forall g.\exists U.\forall w.\forall x.\forall y.[U\ x\ y = g(U\ y\ w)]$$

Следующая задача не имеет унификатора, потому что w не встречается слева, а g — под квантором всеобщности

$$\forall g.\exists U.\forall w.\forall x.\forall y.[U\ x\ y = g\ w]$$