P. Urzyczyn: Materiały do wykładu z semantyki

Logika Hoare'a

Rozważamy najprostszy model imperatywnego języka programowania z jednym typem danych. Wartości tego typu moga być elementami dowolnej struktury algebraicznej o ustalonej sygnaturze. Poniżej, symbol c oznacza dowolną stałą tej sygnatury, a r (odpowiednio f) jest dowolnym k-argumentowym symbolem relacyjnym (funkcyjnym). Są trzy kategorie wyrażeń:[1]

Wyrażenia logiczne (B): $\mathit{false},\ \mathit{true},\ E_1 = E_2,\ r(E_1, E_2, \ldots, E_k),\ \neg B,\ B_1 \wedge B_2,\ B_1 \vee B_2.$

Wyrażenia algebraiczne (E): $c,\ x,\ f(E_1, E_2, \ldots, E_k).$

Programy (P): $\mathit{skip},\ \mathit{loop},\ x := E,\ \texttt{begin}\ P_1; P_2\ \texttt{end},\ \texttt{if}\ B\ \texttt{then}\ P_1\ \texttt{else}\ P_2,\ \texttt{while}\ B\ \texttt{do}\ P_1.$

Semantyka naturalna (dużych kroków)

Znaczenie $[\![W]\!]^{\mathcal{A}}_s$ wyrażenia algebraicznego lub logicznego W w strukturze $\mathcal{A}$ przy wartościowaniu s (lub: *w stanie* s) definiujemy jak zwykle. Górny indeks często pomijamy. Znaczenie programów określimy metodą dużych kroków,[2] tj. zdefiniujemy relację $\langle P, s\rangle \Downarrow s'$, którą odczytujemy tak: program P uruchomiony w stanie s kończy obliczenie w stanie s'.

$$\langle \mathit{skip}, s\rangle \Downarrow s \qquad \frac{[\![E]\!]_s = a}{\langle x := E, s\rangle \Downarrow s[x \mapsto a]} \qquad \frac{\langle P_1, s\rangle \Downarrow s' \quad \langle P_2, s'\rangle \Downarrow s''}{\langle \texttt{begin}\ P_1; P_2\ \texttt{end}, s\rangle \Downarrow s''}$$

$$\frac{[\![B]\!]_s = \mathit{true} \quad \langle P_1, s\rangle \Downarrow s'}{\langle \texttt{if}\ B\ \texttt{then}\ P_1\ \texttt{else}\ P_2, s\rangle \Downarrow s'} \qquad \frac{[\![B]\!]_s = \mathit{false} \quad \langle P_2, s\rangle \Downarrow s'}{\langle \texttt{if}\ B\ \texttt{then}\ P_1\ \texttt{else}\ P_2, s\rangle \Downarrow s'}$$

$$\frac{[\![B]\!]_s = \mathit{true} \quad \langle P, s\rangle \Downarrow s' \quad \langle \texttt{while}\ B\ \texttt{do}\ P, s'\rangle \Downarrow s''}{\langle \texttt{while}\ B\ \texttt{do}\ P, s\rangle \Downarrow s''} \qquad \frac{[\![B]\!]_s = \mathit{false}}{\langle \texttt{while}\ B\ \texttt{do}\ P, s\rangle \Downarrow s}$$

Ponieważ każdy program ma tylko skończenie wiele zmiennych, więc stany można uważać za skończone krotki wartości. *Dziedziną* while-programu P w strukturze $\mathcal{A}$ nazywamy zbiór tych stanów s, dla których $\langle P, s\rangle \Downarrow$. Nietrudno pokazać, że każdy rekurencyjnie przeliczalny podzbiór $\mathbb{N}$ jest w standardowym modelu arytmetyki $\mathcal{N} = \langle \mathbb{N}, +, \cdot, s, 0\rangle$ rzutem dziedziny pewnego programu na pierwszą współrzędną, tj. ma postać $\{s(x_0) \mid \langle P, s\rangle \Downarrow\}$.

Reguły Hoare'a

Formułą logiki Hoare'a nazywamy trójkę postaci $\{\varphi\}\, P\, \{\psi\}$, gdzie φ i ψ są formułami pierwszego rzędu, a P jest programem. Formuła taka jest *prawdziwa* w strukturze $\mathcal{A}$, gdy dla dowolnych stanów s i s' o własności $\mathcal{A}, s \models \varphi$ oraz $\langle P, s\rangle \Downarrow s'$ zachodzi $\mathcal{A}, s' \models \psi$. Mówimy wtedy, że program P jest *częściowo poprawny ze względu na warunek początkowy* φ *i warunek końcowy* ψ i piszemy $\mathcal{A} \models \{\varphi\}\, P\, \{\psi\}$. Jeśli jest tak dla każdej struktury, to piszemy $\models \{\varphi\}\, P\, \{\psi\}$ (trójka jest *prawdziwa*).

[1] Nawiasowanie jest domyślne.

[2] W istocie nasza definicja jest „mieszana", bo dla wyrażeń stosujemy metodę denotacyjną.

Wnioskowanie w logice Hoare'a (dla while-programów) opiera się na następujących aksjomatach i regułach. Implikacje pierwszego rzędu, występujące w ostatniej regule (zwanej regułą konsekwencji) mają być uniwersalnie prawdziwe w rozważanej strukturze.[3]

$$\{\varphi\}\ \mathit{skip}\ \{\varphi\} \qquad \{\varphi[E/x]\}\ x := E\ \{\varphi\} \qquad \{\mathit{true}\}\ \mathit{loop}\ \{\mathit{false}\}$$

$$\frac{\{\varphi\}\ P_1\ \{\psi\} \quad \{\psi\}\ P_2\ \{\vartheta\}}{\{\varphi\}\ \texttt{begin}\ P_1; P_2\ \texttt{end}\ \{\vartheta\}}$$

$$\frac{\{\varphi \wedge B\}\ P_1\ \{\psi\} \quad \{\varphi \wedge \neg B\}\ P_1\ \{\psi\}}{\{\varphi\}\ \texttt{if}\ B\ \texttt{then}\ P_1\ \texttt{else}\ P_2\ \{\psi\}}$$

$$\frac{\{\varphi \wedge B\}\ P\ \{\varphi\}}{\{\varphi\}\ \texttt{while}\ B\ \texttt{do}\ P\ \{\varphi \wedge \neg B\}}$$

$$\frac{\varphi' \Rightarrow \varphi \quad \{\varphi\}\ P\ \{\psi\} \quad \psi \Rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

Piszemy $\mathrm{Th}(\mathcal{A}) \vdash \{\varphi\}\ P\ \{\psi\}$, lub po prostu $\mathcal{A} \vdash \{\varphi\}\ P\ \{\psi\}$, gdy trójka $\{\varphi\}\ P\ \{\psi\}$ ma wyprowadzenie w logice Hoare'a, przy czym wszystkie formuły postaci $\alpha \Rightarrow \beta$, użyte w tym wyprowadzeniu, są prawdziwe w strukturze $\mathcal{A}$. Jeśli zaś wszystkie te formuły są tautologiami logiki pierwszego rzędu, to możemy napisać po prostu $\vdash \{\varphi\}\ P\ \{\psi\}$.

Twierdzenie 0.1 (Poprawność) *Jeśli* $\mathcal{A} \vdash \{\varphi\}\ P\ \{\psi\}$, *to* $\mathcal{A} \models \{\varphi\}\ P\ \{\psi\}$.

Dowód: Indukcja ze względu na dowód. Mamy tyle przypadków ile aksjomatów i reguł. Większość z nich jest łatwa. Na przykład, dla aksjomatu $\{\varphi[E/x]\}\ x := E\ \{\varphi\}$ należy pokazać, że jeśli $\mathcal{A}, s \models \varphi[E/x]$ to $\mathcal{A}, s[x \mapsto [\![E]\!]_s] \models \varphi$. W przypadku `while` zakładamy, że $\mathcal{A}, s \models \varphi$ oraz $\langle \texttt{while}\ B\ \texttt{do}\ P, s\rangle \Downarrow s''$ i chcemy pokazać $\mathcal{A}, s'' \models \varphi \wedge \neg B$. Robimy to przez pomocniczą indukcję ze względu na wyprowadzenie $\langle \texttt{while}\ B\ \texttt{do}\ P, s\rangle \Downarrow s''$. Mamy tu dwa przypadki: pierwszy oczywisty, gdy $[\![B]\!]_s = \mathit{false}$, drugi gdy $[\![B]\!]_s = \mathit{true}$ oraz $\langle P, s\rangle \Downarrow s'$ i $\langle \texttt{while}\ B\ \texttt{do}\ P, s'\rangle \Downarrow s''$. W tym drugim przypadku z głównego założenia indukcyjnego mamy $\mathcal{A}, s' \models \varphi$, a z pomocniczego $\mathcal{A}, s'' \models \varphi \wedge \neg B$. Resztę dowodu pomijamy. ∎

Własność odwrotną do twierdzenia 0.1 nazywamy *relatywną pełnością* logiki Hoare'a. Nie jest to bardzo efektywna własność, bo w dowodach dopuszczamy wszystkie implikacje prawdziwe w strukturze $\mathcal{A}$. Można ograniczyć stosowanie reguły konsekwencji do implikacji dowodliwych w jakimś systemie wnioskowania (twierdzenie o poprawności oczywiście pozostaje prawdziwe) i pytać o „prawdziwą" pełność: czy każdą trójke prawdziwą w $\mathcal{A}$ można udowodnić?

Niestety, odpowiedź na to pytanie jest negatywna. Zauważmy bowiem, że trójka $\{\mathit{true}\}\mathit{skip}\{\psi\}$ oznacza to samo co ψ, zatem gdybyśmy mieli efektywny system wnioskowania dla trójek prawdziwych w strukturze $\mathcal{A}$, to teoria $\mathrm{Th}(\mathcal{A})$ musiałaby być rekurencyjnie przeliczalna. Tak nie jest na przykład dla algebry $\mathcal{N}$ liczb naturalnych. Inne uzasadnienie jest takie: trójka

[3] Czasami jednak żąda się, aby były to tautologie, a czasami — twierdzenia jakiejś teorii.

$\{true\}P\{false\}$ wyraża pustość dziedziny programu P (a więc pustość odpowiedniego zbioru rekurencyjnie przeliczalnego). To też nie jest rekurencyjnie przeliczalna własność programów.

Wiadomo, że istnieją struktury (na przykład ciało liczb zespolonych $\mathcal{C} = \langle \mathbb{C}, +, \cdot, 0, 1\rangle$), dla których logika Hoare'a nie jest relatywnie pełna. Potrzebujemy dodatkowego założenia. Powiemy, że struktura $\mathcal{A}$ jest *wyrażalna*, gdy dziedzina każdego programu jest definiowalna w logice pierwszego rzędu, tj. gdy dla dowolnego P istnieje taka formuła $\mathrm{Halt}(P)$, że zachodzi równoważność: $\langle P, s\rangle \Downarrow \quad \Leftrightarrow \quad \mathcal{A}, s \models \mathrm{Halt}(P)$. Dla struktur wyrażalnych można zdefiniować w logice pierwszego rzędu wiele innych własności programów. Nas interesują głównie *najsłabsze prewarunki*, czyli zbiory postaci

$$wp(P, \alpha) = \{s \mid \text{jeśli } P, s \Downarrow s' \text{ to } \mathcal{A}, s' \models \alpha\}.$$

Lemat 0.2 *Jeśli $\mathcal{A}$ jest strukturą wyrażalną, to dla każdego P i α istnieje formuła ξ definiująca $wp(P, \alpha)$, tj. taka, że dla dowolnego stanu s:*

$$\mathcal{A}, s \models \xi \quad \textit{wtedy i tylko wtedy, gdy} \quad s \in wp(P, \alpha).$$

Dowód: Załóżmy, że $\vec{x}$ to wszystkie zmienne występujące w P i niech $\vec{y}$ będzie wektorem nowych zmiennych o tej samej długości. Rozpatrzmy program

$$Q = \texttt{begin } P; \texttt{if } \vec{x} = \vec{y} \texttt{ then } \textit{skip} \texttt{ else } \textit{loop} \texttt{ end}.$$

Jako ξ można przyjąć formułę $\forall \vec{y}(\mathrm{Halt}(Q) \to \alpha[\vec{x} := \vec{y}])$. ∎

Odtąd napis $wp(P, \alpha)$ oznacza też formułę definiującą zbiór $wp(P, \alpha)$. Zauważmy od razu, że

$$\mathcal{A} \models \{\varphi\}\, P\, \{\psi\}, \quad \text{wtedy i tylko wtedy, gdy} \quad \mathcal{A} \models \varphi \Rightarrow wp(P, \psi).$$

Twierdzenie 0.3 (Cook) *Logika Hoare'a dla języka while-programów jest relatywnie pełna dla każdej struktury wyrażalnej $\mathcal{A}$: jeśli $\mathcal{A} \models \{\varphi\}\, P\, \{\psi\}$, to $\mathcal{A} \vdash \{\varphi\}\, P\, \{\psi\}$.*

Dowód: Wystarczy pokazać, że $\mathcal{A} \vdash \{wp(P, \alpha)\}P\{\alpha\}$, dla każdego P i α. Jeśli bowiem $\mathcal{A} \models \{\varphi\}P\{\psi\}$ to $\mathcal{A} \vdash \{\varphi\}P\{\psi\}$ wynika z reguły konsekwencji z pomocą $\varphi \Rightarrow wp(P, \psi)$.

Warunek $\mathcal{A} \vdash \{wp(P, \alpha)\}P\{\alpha\}$ udowodnimy przez indukcję ze względu na P.

Niech najpierw $P = \textit{skip}$. Wtedy $\models wp(\textit{skip}, \alpha) \Leftrightarrow \alpha$ oraz $\{\alpha\}\textit{skip}\{\alpha\}$, więc z reguły konsekwencji wynika $\{wp(\textit{skip}, \alpha)\}\textit{skip}\{\alpha\}$.

Jeśli $P = \textit{loop}$, to $\models wp(\textit{loop}, \alpha) \Leftrightarrow \textit{true}$ i sprawa jest oczywista.

Dla $P = (x := E)$ mamy $\models wp(x := E, \alpha) \Leftrightarrow \alpha[x := E]$, więc to też łatwy przypadek.

Dla $P = P_1; P_2$ korzystamy z równoważności $wp(\texttt{begin } P_1; P_2 \texttt{ end}, \alpha) \Leftrightarrow wp(P_1, wp(P_2, \alpha))$ i stosujemy indukcję. W przypadku `if` należy zauważyć, że $wp(\texttt{if } B \texttt{ then } P_1 \texttt{ else } P_2, \alpha)$ ma postać $(B \wedge wp(P_1, \alpha)) \vee (\neg B \wedge wp(P_2, \alpha))$. Zostaje `while`.

Niech więc $P = \texttt{while}\ B\ \texttt{do}\ Q$ i niech $w = wp(P, \alpha)$. Udowodnimy najpierw, że:

$$(1)\ \ \mathcal{A} \models \{w \wedge B\}Q\{w\} \qquad \text{oraz} \qquad (2)\ \ \mathcal{A} \models w \wedge \neg B \Rightarrow \alpha.$$

(1) Załóżmy, że $\mathcal{A}, s \models w \wedge B$ i niech $\langle Q, s\rangle \Downarrow s'$. Chcemy pokazać $\mathcal{A}, s' \models w$, przypuśćmy więc, że $\langle P, s'\rangle \Downarrow s''$. Z warunków $\mathcal{A}, s \models B$, $\langle Q, s\rangle \Downarrow s'$ i $\langle P, s'\rangle \Downarrow s''$ wynika, że $\langle P, s\rangle \Downarrow s''$, a skoro $\mathcal{A}, s \models w$ to $\mathcal{A}, s'' \models \alpha$.

(2) Jeśli $\mathcal{A}, s \models w \wedge \neg B$, to $\langle P, s\rangle \Downarrow s$, a ponieważ $\mathcal{A}, s \models w$, więc $\mathcal{A}, s \models \alpha$.

Z założenia indukcyjnego $\mathcal{A} \vdash \{wp(Q, w)\}Q\{w\}$. Ponieważ (1), więc $\mathcal{A} \models w \wedge B \Rightarrow wp(Q, w)$, i na mocy reguły konsekwencji $\mathcal{A} \vdash \{w \wedge B\}Q\{w\}$, skąd od razu $\mathcal{A} \vdash \{w\}P\{w \wedge \neg B\}$. Teraz stosujemy regułę konsekwencji, korzystając z (2), i mamy $\mathcal{A} \vdash \{w\}P\{\alpha\}$. ∎

Twierdzenie Gödla o reprezentacji stanowi m.in. że wszystkie zbiory rekurencyjnie przeliczalne są definiowalne formułami arytmetyki pierwszego rzędu. A więc nie na próżno dowodziliśmy twierdzenie Cooka.

Fakt 0.4 *Struktura* $\mathcal{N} = \langle \mathbb{N}, +, \cdot, s, 0\rangle$ *jest wyrażalna.*

Nietrudno pokazać, że wyrażalna jest też każda struktura skończona. Ponieważ teoria struktury skończonej jest zawsze rozstrzygalna, więc z twierdzenia Cooka wynika, że zbiór trójek prawdziwych w takiej strukturze jest rekurencyjnie przeliczalny. Jeśli przypomnimy, że $\{true\}P\{false\}$ wyraża zapętlenie się programu P, to łatwo wywnioskujemy, że pytanie "czy dany program zatrzymuje się dla określonych danych wejściowych w danej strukturze skończonej?" jest rozstrzygalne. Nie jest to może zaskakujące, ale zauważmy, że można tę obserwację uogólnić tak:

> *Warunkiem koniecznym na to, aby dla języka programowania L istniał poprawny i relatywnie pełny system Hoare'a jest aby problem stopu dla języka L był rozstrzygalny na skończonych interpretacjach.*

Problem stopu na skończonych interpretacjach nie jest rozstrzygalny np. dla języka z procedurami wyższych typów, w którym można dokonywać podstawień na zmienne nielokalne. Jeśli tego zabronimy, problem staje się rozstrzygalny i można skonstruować system Hoare'a.

Logika Hoare'a jako semantyka aksjomatyczna

Teraz potraktujemy logikę Hoare'a w sposób bardziej uniwersalny. Napiszemy $\vdash \{\varphi\}\, P\, \{\psi\}$, gdy trójkę $\{\varphi\}\, P\, \{\psi\}$ można wyprowadzić ograniczając regułę konsekwencji do twierdzeń logiki pierwszego rzędu. Zbiór wszystkich takich trójek nazwiemy *teorią częściowej poprawności* programu P. Przyjmijmy oznaczenie $\mathrm{PC}(P) = \{\langle \varphi, \psi\rangle \mid \vdash \{\varphi\}\, P\, \{\psi\}\}$.

Programy P i Q są *równoważne* (piszemy $P \equiv Q$) wtedy i tylko wtedy, gdy w dowolnej strukturze $\mathcal{A}$, warunki $\langle P, s\rangle \Downarrow s'$ i $\langle Q, s\rangle \Downarrow s'$ są równoważne. Równoważne programy mają oczywiście takie same teorie częściowej poprawności. Okazuje się, że zachodzi też twierdzenie odwrotne, a zatem *teoria częściowej poprawności determinuje semantykę programu*. Ten fakt uzasadnia określenie „semantyka aksjomatyczna".

Twierdzenie 0.5 (Meyer) *Jeśli* $P \not\equiv Q$ *to* $\mathrm{PC}(P) \neq \mathrm{PC}(Q)$.

Szkic dowodu: Dla uproszczenia rozpatrzymy przypadek, gdy $\langle P, s\rangle \Downarrow s'$ oraz $\langle Q, s\rangle \not\Downarrow\ s''$, dla dowolnego s'' (program P się zatrzymuje, a Q się zapętla dla pewnego s) w pewnej strukturze $\mathcal{A}$. Załóżmy przy tym, że $\vec{x} = x_1, \ldots, x_n$ to wszystkie zmienne występujące w programach P i Q. Analiza obliczenia P w stanie s pozwala wskazać pewną formułę otwartą $\varphi(\vec{x})$ o następujących własnościach:

- $\mathcal{A}, s \models \varphi(\vec{x})$;
- Dla dowolnych $\mathcal{B}, r$, jeśli $\mathcal{B}, r \models \varphi(\vec{x})$, to $\langle P, r\rangle \Downarrow r'$, dla pewnego r'.

Formuła φ to tak naprawdę koniunkcja wszystkich testów, które wykonane są w czasie obliczenia (uwzględniając wcześniejsze podstawienia). Ponieważ obliczenie programu Q dla wejścia s jest nieskończone, więc zamiast jednej formuły mamy nieskończony zbiór $\Gamma(\vec{x})$:

- $\mathcal{A}, s \models \Gamma(\vec{x})$;
- Dla dowolnych $\mathcal{B}, r$, jeśli $\mathcal{B}, r \models \Gamma(\vec{x})$, to $\langle Q, r\rangle \not\Downarrow r'$, dla każdego r'.

Zbiór $\Gamma(\vec{x}) \cup \{\varphi(\vec{x})\}$ jest niesprzeczny i składa się tylko z formuł otwartych, więc ma model mocy $\aleph_0$. Można uważać, że dziedziną tego modelu jest po prostu $\mathbb{N}$. Dodając do sygnatury zwykłe operacje arytmetyczne, wzbogacamy ten model do wyrażalnej struktury $\mathcal{B}$, w której dla pewnych r, r' zachodzi $\langle P, r\rangle \Downarrow r'$, ale dla każdego r'' mamy $\langle Q, r\rangle \not\Downarrow r''$. Niech $\psi(\vec{x})$ bedzie formułą definiującą w $\mathcal{B}$ dziedzinę programu Q. Ponieważ stan r nie należy do dziedziny programu Q, więc $\mathcal{B} \not\models \{\neg\psi\}P\{\mathit{false}\}$ i tym bardziej $\not\vdash \{\neg\psi\}Q\{\mathit{false}\}$. Wtedy $\mathcal{B} \models \{\neg\psi\}Q\{\mathit{false}\}$ więc z relatywnej pełności wynika $\mathcal{B} \vdash \{\neg\psi\}Q\{\mathit{false}\}$. W takim dowodzie występuje tylko skończenie wiele zdań prawdziwych w $\mathcal{B}$, jeśli ϑ jest koniunkcją tych formuł, to w istocie mamy $\vdash \{\vartheta \wedge \neg\psi\}Q\{\mathit{false}\}$.

Ale ponieważ stan r nie należy do dziedziny programu Q, więc $\mathcal{B} \not\models \{\vartheta \wedge \neg\psi\}P\{\mathit{false}\}$ i tym bardziej $\mathcal{B} \not\vdash \{\vartheta \wedge \neg\psi\}Q\{\mathit{false}\}$. A zatem $\langle \vartheta \wedge \neg\psi, \mathit{false}\rangle \in \mathrm{PC(Q)} - \mathrm{PC(P)}$. ∎

Ćwiczenia:

1. Niech $P =$ `while` $x > 0$ `do begin` $y := x \cdot y; x := x - 1$ `end`. Udowodnić, że
$$\mathcal{N} \vdash \{x = n \wedge n \geq 0 \wedge y = 1\}P\{y = n!\}.$$
2. Uogólnić dowód twierdzenia 0.5 na przypadek gdy $\langle P, s\rangle \Downarrow s'$ oraz $\langle Q, s\rangle \Downarrow\ s''$, gdzie $s' \neq s''$.