

Souvislost počítání a dokazování

Část I

Opakování

$\models \varphi$, φ je *logicky platná*

φ platí ve všech mat strukturách, interpretacích symbolů jazyka

např. $\models (\forall x : x > x) \rightarrow (1 > 1)$

ale $\not\models 1 + 1 = 2$

$\vdash \varphi$, φ je *dokazatelná*

důkaz = sekvence formulí, které jsou axiomy nebo odvozené odvozovacími pravidly z předchozích, končí φ .

V podstatě syntaktická manipulace s formullemi.

Platné je právě to, co je dokazatelné.

PL je korektní $\vdash \varphi \Rightarrow \models \varphi$

PL je úplná $\models \varphi \Rightarrow \vdash \varphi$

PL je efektivní.

Je možné čistě mechanicky ověřit, co je správný důkaz.

$\{w \mid w \text{ je důkazem v PL}\}$ je rekurzivní.

Efektivnost \Rightarrow generování důkazů

Pro efektivní log. systém existuje program **Generátor důkazů**, který vypisuje (nekonečný) seznam důkazů, a každý důkaz časem vypíše.

```
foreach řetězec do
  if řetězec je důkazem then
    vypiš řetězec
```

- * Důkaz je řetězec symbolů z konečné abecedy symbolů.
- * Řetězce je možné generovat v abecedním pořadí, na každý jednou dojde.
- * Efektivnost: existuje program, který rozhodne, zda je řetězec důkazem.

V efektivním systému je $\{\varphi \mid \varphi \text{ je dokazatelná}\}$ částečně rozhodnutelná.

Příklad běhu Generátoru důkazů

<i>a</i>	$xyp \Rightarrow$
<i>b</i>	...
<i>c</i>	$x = y \rightarrow p$
...	...
\wedge	$xy \Rightarrow (((, ((\wedge \neg \neg x, , \exists$
\neg	...
\exists	$p \rightarrow ((p \rightarrow p) \rightarrow p), (p \rightarrow (p \rightarrow p) \rightarrow (p \rightarrow p)), \forall x \forall y x = y$
<i>aa</i>	...
<i>ab</i>	Proletěl mi zubr zdí, myslel, že to ubrdzí. (Plíhal)
<i>ac</i>	...
...	...
<i>aab</i>	$p \rightarrow ((p \rightarrow p) \rightarrow p), (p \rightarrow ((p \rightarrow p) \rightarrow p) \rightarrow ((p \rightarrow (p \rightarrow p)) \rightarrow (p \rightarrow p))), \underline{(p \rightarrow (p \rightarrow p) \rightarrow (p \rightarrow p))}$
...	...
<i>ab</i> \exists	...
...	...
<i>a</i> \vee <i>b</i> \neg (..... důkaz, <u>$(\forall x x > x) \rightarrow (1 > 1)$</u>
...	...

Efektivnost + korektnost + úplnost \Rightarrow rozhodování platnosti

Tedy umíme řešit „Entscheidungsproblem“.

Pro PL tedy existuje program **Rozhodovač platnosti formulí**. Pro danou φ :

Spust' Generátor důkazů.

if Generátor vypsal důkaz φ **then** φ je platná

if Generátor vypsal důkaz $\neg\varphi$ **then** φ není platná

- * Úplnost zaručuje, že program skončí, protože:
 - Pro každou větu máme $\models \varphi$, nebo $\models \neg\varphi$ (z def. sémantiky).
 - Z úplnosti proto $\vdash \varphi$, nebo $\vdash \neg\varphi$.
- * Korektnost zaručuje, že odpověď je správná.

Prvořádkové teorie

- * Rozhodování platnosti v čisté PL ještě není výhra.
Umíme jen dokazovat logickou platnost formulí jako $\forall x \ x > x \rightarrow 1 > 1$.
- * $1 + 1 = 2$ ale není logicky platná formule, neplatí ve všech interpretacích.
- * Chceme dokazovat a rozhodovat, jestli $1 + 1 = 2$ je platná v přirozených číslech (t.j. když 1, 2, a + jsou interpretovány, jak jsme zvyklí).
- * Zajímají nás pravdy o konkrétních matematických strukturách.

Teorie, množina speciálních axiomů, definuje vlastnosti matematických struktur.

- * $T \models \varphi$, φ platí ve všech strukturách, které splňují axiomy T .
- * $T \vdash \varphi$, φ je dokazatelná z axiomů T .

Peanova aritmetika T_{PA}

- $\forall x \neg(S(x) = 0)$ (nula je první)
- $\forall xy(S(x) = S(y) \rightarrow x = y)$ (každý má jiného následníka)
- pro formule φ jazyka T_{PA} s jednou volnou proměnnou:
$$[\varphi(0) \wedge (\forall x(\varphi(x) \rightarrow \varphi(S(x))))] \rightarrow \forall x(\varphi(x))$$
 (axiom indukce)
- $\forall x(x + 0 = x)$ (0 je neutrální k +)
- $\forall xy(x + S(y) = S(x + y))$ (def. sčítání)
- $\forall x(x \cdot 0 = 0)$ (0 je nulová k ·)
- $\forall xy(x \cdot S(y) = x \cdot y + x)$ (def. násobení)

Hurá, umíme říct, že $1 + 1 = 2$ platí.

$$T_{PA} \models S^1(0) + S^1(0) = S^2(0)$$

Vlastnosti teorií

Rozumná teorie T je

- * **efektivní**: T (množina axiomů) je rekurzivní.
- * **bezesporná**: nikdy $T \vdash \varphi$ a zároveň $T \vdash \neg\varphi$.

V efektivní teorii je $\{\varphi \mid T \vdash \varphi\}$ částečně rozhodnutelná.

Když T definuje strukturu přesně (jediný model až na izomorfismus), nebo v ní není možné vyjádřit rozdíl mezi různými modely, pak je

- * **úplná**: $T \vdash \varphi$ nebo $T \vdash \neg\varphi$.

Gödelova úplnost pro prvořádové teorie:

$T \models \varphi$ právě když $T \vdash \varphi$

Rozhodnutelná teorie

Teorie je rozhodnutelná, pokud $\{\varphi \mid T \models \varphi\}$ je rozhodnutelná (rekurzivní).

Pro efektivní, bezespornou a úplnou teorii je $\{\varphi \mid T \vdash \varphi\} = \{\varphi \mid T \models \varphi\}$ rozhodnutelná (rekurzivní) množina. (generátor časem vygeneruje důkaz φ nebo $\neg\varphi$)

Efektivní, bezesporná a úplná teorie je rozhodnutelná.
Efektivní a bezesporná teorie je částečně rozhodnutelná.

Sporná teorie je rozhodnutelná triviálně, protože každá formule je jejím důsledkem.

Část II

Souvislost neúplnosti a nerozhodnutelnosti

Gödel: Teorie PL, která zahrnuje Peanovu aritmetiku, nemůže být úplná.
V PL není možné přesně definovat, co jsou přirozená čísla
(ani v žádném jiném korektním a efektivním axiomatickém systému).

Turing: Existují problémy, které nejsou obecně algoritmicky řešitelné.
(jako problém zastavení)

Tyto dva výsledky těsně souvisejí.

Důkaz a výpočet

- * Důkaz a výpočet jsou velmi podobné věci:
 - Je to sekvence *formulí / konfigurací*,
 - které jsou buď *axiomy / iniciální konfigurace*
 - nebo jsou odvozeny z předchozích pomocí jednoduchých mechanických *odvozovacích pravidel / pravidel daných přechodovou funkcí*.
- * Dá se ukázat, že:
 - Dokazatelnost aritm. formule můžeme redukovat na zastavení Turingova stroje (generátor důkazů).
 - Zastavení Turingova stroje můžeme redukovat na platnost aritm. formule (ukážeme na dalším slajdu).

Redukce problému zastavení na problém platnosti aritm. formule

Mějme DTS M který, pro jednoduchost, nikdy neskončí abnormálně.

Sestrojíme aritm. formuli φ_w^M platnou právě tehdy, když M zastaví na slově $w = a_1 \cdots a_n$.

Bude definovat vztah mezi stavem $S(k)$ v kroce k výpočtu, pozicí hlavy $H(k)$ v kroce k , a znakem $Z(k, p)$ v kroce k na poli pásky p :

$$\varphi_w^M \equiv \varphi_{init} \wedge \varphi_{\Delta} \wedge \varphi_{stop}, \text{ kde}$$

$$\varphi_{init} \equiv S(0) = q_0 \wedge H(0) = 1 \wedge Z(0, 1) = \Delta \wedge \left(\bigwedge_{p=2}^{n+1} Z(0, p) = a_p \right) \wedge (\forall p > n+1 : Z(0, p) = \Delta)$$

$$\varphi_{\Delta} \equiv \forall k \forall p \bigwedge_{q \in Q, a \in \Gamma} \varphi_{(q,a)}, \text{ kde pokud } \delta(q, a) = (q', X), X \in \{L, R\} \cup \Sigma, \text{ potom}$$

$$\varphi_{(q,a)} \equiv (S(k) = q \wedge H(k) = p \wedge Z(k, p) = a) \rightarrow$$

$$(s(k+1) = q') \wedge H(k+1) = p' \wedge Z(k+1, p) = a' \wedge$$

$$\forall p' \neq p : Z(k+1, p') = Z(k, p'))$$

$$\text{kde } p' = \begin{cases} p & \text{pokud } X \in \Sigma \\ p+1 & \text{pokud } X = R \\ p-1 & \text{pokud } X = L \end{cases}, \quad a' = \begin{cases} X & \text{pokud } X \in \Sigma \\ a & \text{pokud } X \in \{L, R\} \end{cases}.$$

$$\varphi_{stop} \equiv \exists k : S(k) = q_f$$

Redukce problému zastavení na problém platnosti aritm. formule

φ_w^M je platná právě tehdy, když T zastaví na w .

Tedy, platnost aritmetických formulí nemůže být rozhodnutelná, protože potom by byl rozhodnutelný problém zastavení.

T_{PA} nemůže být úplná, protože pak by platnost aritm. formulí byla rozhodnutelná. Stejně ani žádné rozšíření T_{PA} (efektivní a bezesporné), ani jakýkoliv jiný efektivní a korektní systém charakterizující aritmetiku přirozených čísel přesněji, nemůže být úplný.

Kostra Gödelova důkazu

Gödel ještě neměl Turingovy stroje. Podařilo se mu ale „programovat“ v aritmetice. Hlavní myšlenkou je konstrukce formule, která, velmi neformálně, říká „Nejsem dokazatelná“. Na to je potřeba do sčítání a násobení zakódovat formule a důkazy.

- * Každý symbol c je kódován číslem. Formule φ je tedy zapsána jako slovo $w_\varphi = a_1 \cdots a_n \in \mathbb{N}^*$, a je kódována **Gödelovým číslem**

$$G(\varphi) = 2^{a_1} \cdot 3^{a_2} \cdot 5^{a_3} \cdot p_4^{a_4} \cdots p_n^{a_n}, \quad \text{kde } p_i \text{ je } i\text{-té prvočíslo.}$$

- * Kódování φ do $G(\varphi)$ a dekodování je popsatelné aritmetickými formullemi.
- * Důkaz je sekvence formulí, a má tedy také své Gödelovo číslo.
- * Aplikace odvozovacích pravidel je popsatelná aritmetickými formullemi.
- * V aritmetice je možné definovat predikát D tak, že $D(m, n)$ platí, právě když m je kódem důkazu formule $\varphi(G(\varphi))$, kde n je číslo formule $\varphi(x)$ ($\varphi(x)$ je formule s jednou volnou proměnnou, za kterou je v $\varphi(G(\varphi))$ dosazeno $G(\varphi)$).
- * Necht' $\psi(x)$ je formule $\neg \exists y D(y, x)$.
- * Formule $\psi(G(\psi)) : \neg \exists y D(y, G(\psi))$ komplikovaně říká „Nejsem dokazatelná“.
 - $\psi(G(\psi))$ je dokazatelná. Pak podle definice D neexistuje důkaz $\psi(G(\psi))$.
 - $\neg \psi(G(\psi))$ je dokazatelná. Pak podle definice D existuje důkaz $\psi(G(\psi))$. Spor.

Podobnost Gödelova a Turingova důkazu

Gödel: Nemůžeme dokázat nebo vyvrátit každou formuli.

Sporem. Formule $\psi(x) : \neg\exists y D(y, x)$ říká, že formule s G. číslem x není dokazatelná, pokud je x dosazeno za její volnou proměnnou.

- * $\psi(G(\psi))$ je dokazatelná. Pak, podle definice ψ , neexistuje důkaz $\psi(G(\psi))$.
- * $\neg\psi(G(\psi))$ je dokazatelná. Pak, podle definice ψ , existuje důkaz $\psi(G(\psi))$.

Turing: Nemůžeme rozhodovat problém zastavení.

Sporem. TS M , který zastaví, právě když jeho vstup je kódem TS, který nezastaví na vlastním kódu.

- * $M(\langle M \rangle)$ zastaví. Pak, podle definice M , M nezastaví s vlastním kódem na vstupu.
- * $M(\langle M \rangle)$ nezastaví. Pak, podle definice M , M zastaví s vlastním kódem na vstupu.

Za všechno může sebereferece?

- * **Krétan**: Ted' lžu.
- * **Russel**: Je množina množin, které nejsou prvkem sama sebe, prvkem sama sebe?
- * **Gödel**: Je formule „Nejsem dokazatelná.“ dokazatelná?
- * **Turing**: Stroj M zastaví na kódu stroje M' právě tehdy, když M' nezastaví na vlastním kódu. Zastaví M na vlastním kódu?

O čem přemýšlejí vrány na elektrickém vedení



Je možné logicky zdůvodnit (dokázat) všechno, co je pravda?

(např. o přirozených číslech?)

Už víme, že to nejde v žádném jednom rozumném axiomatickém systému.

Můžeme ale dokazovací systémy dál vyvíjet, např. objevovat nové axiomy, které nám umožní dokázat více a více formulí.

Můžeme tak časem dokázat nebo vyvrátit cokoliv?

Dvě možnosti:

1. Proces vymýšlení nových axiomů a systémů je také výpočtem stroje s Turingovskou silou. Pak je to jen komplikovaný TS generující teorémy. Aplikují se tedy věty o neúplnosti a nerozhodnutelnosti: Nemůžeme dokázat každou platnou formuli. Existují nepoznatelné pravdy.
2. Pokud můžeme každou formuli někdy nějak logicky zdůvodnit, dokázat, pak je lidské přemýšlení a vývoj procesem s větší než Turingovskou silou.

Poznámky a cvičení pojmů

- * **Úplnost teorie implikuje rozhodnutelnost** (generátorem teorémů a důkazů).
Ne naopak. Teorie může být rozhodnutelné, tj., existuje algoritmus rozhodující platnost v teorii, a stále neúplná.
- * **Čistá prvořádková logiky s rovností** (axiomy rovnosti: reflexivita, funkční a predikátová kongruence) **je neúplná** (např. jednoprvkový vs. nekonečný model a formule $\forall x \forall y (x = y)$), je možné ji zúplnit přidáním axiomu specifikujícího velikost domény).
Je ale rozhodnutelná (Leopold Löwenheim, 1915).
- * Úplnost a „přesnost definice“ není přesně to stejné. Peanova aritmetika je neúplná, ale Presburgerova aritmetika (Peanova aritmetika bez násobení) je úplná. Protože T_{PA} je neúplná, má více modelů, a protože presburgerova aritmetika je podmnožinou speciálních axiomů T_{PA} , musí mít také více modelů. Presburgerova aritmetika je ale úplná, protože **rozdíl mezi různými modely se v ní nedá vyjádřit**.
- * Úplná teorie je rozhodnutelné generátorem teorémů/důkazů, ale většinou existuje i mnohem efektivnější algoritmus. Například pro Presburgerovu aritmetiku.