

Skolemizator - programátorská dokumentace

V tomto textu vysvětlím, jak program funguje a popíši ideu jeho jednotlivých částí. Detailnější komentáře jsou napsané přímo v kódu programu.

Program Skolemizator je naprogramován v **SWI-Prolog**. Nyní popíši jednotlivé soubory, ze kterých se jeho kód skládá.

main.pl

Jedná se o zaváděcí skript, který konzultuje ostatní skripty. Tento skript pak konzultuje uživatel.

prefix.pl

Při převodu formule do prenexního tvaru je někdy třeba zavést nové identifikátory. To, že tyto identifikátory budou unikátní (tedy nebudou použité v zadané formuli) se řeší tím, že všechny nově vytvořené identifikátory začínají na prefix, který není prefixem žádného identifikátoru v zadané formuli. Takovému prefixu dále budeme říkat *volný prefix*.

Soubor [prefix.pl](#) definuje proceduru **find_prefix(+Formule, +IdealniPrefix, -VolnyPrefix)**, která vrácí volný prefix. Funguje tak, že vrátí IdealniPrefix, pokud je to volný prefix ve Formule. Pokud IdealniPrefix není volný ve Formule, začne za něj přidávat opakovaně znak `_`, dokud tak volný prefix nezíská. Ten pak vrátí.

utils.pl

Tento skript obsahuje pomocné jednoduché procedury, které jsou pak použité ve více jiných skriptech. Například proceduru k substituci identifikátorů (*substitute*), která je použita jak ve skolemizaci, tak v převodu do prenexního tvaru.

unique.pl

Při převodu do prenexního tvaru je třeba dávat pozor na to, aby dvě různé proměnné se stejným

identifikátorem v zadané formuli neměly stejný identifikátor ve formuli v prenexním tvaru, protože tam budou použité ve stejném kontextu (budou znamenat to samé).

Soubor [unique.pl](#) definuje proceduru **unique(+Formule, -Formule)**, která nahradí názvy proměnných ve formuli takovými názvy, aby se každý název ve formuli vyskytoval jen v daném kontextu (nepoužíval se na více nezávislých místech ve formuli).

Využívá k tomu již zmíněné procedury *substitute* a *find_prefix*.

[prenex.pl](#)

Definuje proceduru **prenex(+Formule, -FormuleVPrenexnimTvaru)**, která převede formuli do prenexního tvaru.

Převod funguje tak, že se nejdříve formule převede do formule s unikátními názvy pomocí procedury *unique*, aby se později zabránilo problémům s kolizí identifikátorů. Poté se pomocí obvyklých pravidel pro převod do prenexního tvaru formule převede do prenexního tvaru.

[skolemization.pl](#)

Definuje proceduru **skolemization(+Formule, -SkolemizovanaFormule)**, která skolemizuje formuli (převede formuli do prenexního tvaru jen s univerzálními kvantifikátory).

Procedura funguje tak, že formuli nejdříve převede do prenexního tvaru a pak postupně odstraňuje existenční kvantifikátory.

[converter.pl](#)

Všechny zmiňované formule výše byly v programu reprezentovány pomocí termů. Tedy například formule popsaná slovy jako

Pro všechna x platí $p(x)$ a zároveň $q(x)$.

byla v programu reprezentována jako

```
forall(x, and(p(x), q(x)))
```

Protože taková reprezentace složitějších formulí může být pro uživatele nepřehledná, skript [converter.pl](#) definuje procedury **tonice** a **fromnice**, které slouží k převodu formule do resp. z “hezkého” tvaru. “Hezký” tvar formule výše by pak byl

```
x* (p(x)/\q(x))
```

Dále skript definuje procedury pro převod formulí v “hezkém” tvaru do prenexního tvaru a do skolemizované formy **pren** a **skol**.

Pro testování formulí v “nehezkém” tvaru je možné použít soubor **examples.txt**, pro testování formulí v “hezkém” tvaru je možné využít [tests.pl](#).