

Костянтин Любченко

ЗАДАЧА ПОШУКУ ПІДРЯДКА В РЯДКУ У МОВІ PROLOG

Однією з найпоширеніших структур даних, яка використовується в мовах програмування для подання фактичної інформації, є рядок.

Як відомо, для комп'ютера рядок являє собою скінчений набір символів. В мовах програмування символами є букви латинського (українського, російського) алфавіту, спеціальні символи (+, @, \$ та ін.), а також управляючі символи, які відповідають деяким клавішам, наприклад, Enter і Esc.

Над рядками виконуються такі операції [1]:

- об'єднання рядків для утворення нового рядка;
- поділ рядка на два нових рядка, кожний з яких містить деякі з вихідних символів;
- пошук символу або підрядка всередині даного рядка.

Ці операції над рядками є одними з основних при розв'язуванні практичних задач за допомогою мови Prolog.

Для зручності програмування мова Turbo-Prolog має декілька вбудованих предикатів:

- `str_len` – предикат для знаходження довжини рядка;
- `concat` – предикат для об'єднання двох рядків;
- `frontstr` – предикат для поділу рядка на два підрядка;
- `frontchar` – предикат для поділу рядка на перший символ і залишок та створення символічних префіксів;
- `fronttoken` – предикат для поділу рядка на лексему і залишок;
- `isname` – предикат, який перевіряє, чи буде рядок спеціальним рядком мови Turbo-Prolog.

Але при створенні програм на мові Prolog виникає потреба розширити цей список вбудованих предикатів обробки рядків.

Створимо один з важливих і зручних у використанні таких предикатів – предикат пошуку позиції першого входження підрядка у заданий рядок:

```
pos(string, string, integer).
```

Перші два параметри цього предикату є вхідними – відповідно підрядок і рядок, в якому шукається підрядок, третій параметр – вихідний – повинен містити позицію входження підрядка у рядок.

Подібні функції реалізовані у деяких мовах програмування (наприклад Pascal, Delphi), але у мові Turbo-Prolog – ні.

Також при реалізації предикату `pos` слід врахувати випадок, коли заданий підрядок відсутній у заданому рядку.

Отже, програма, в якій створено предикат `pos`, має наступний вигляд:

```
predicates
  pos(string,string, integer)
clauses
  pos(S,S, 1): -!.
  pos(SubS,S, 0): -
    str_len(SubS, L),
    str_len(S, L),
    SubS <> S, !.
  pos(SubS,S, 1): -
    str_len(SubS, SubSLen),
    frontstr(SubSLen,S, Sub,_),
    Sub = SubS, !.
  pos(SubS,S, P): -
    frontchar(S,_, S1),
    pos(SubS,S1, P1),
    P1 > 0, !,
    P = P1 + 1.
  pos(_,_, 0).
```

Розглянемо кожне з речень предикату `pos` даної програми:

1) `pos(S,S, 1)` призначено для обробки ситуації, коли підрядок співпадає з рядком, в якому він шукається; тому повертається число 1;

2) `pos(SubS,S, 0)` обробляє випадок, коли підрядок `SubS` і рядок `S` мають однакову кількість символів, але є різними; тому повертається число 0;

3) `pos(SubS,S, 1)` повертає результат 1, якщо підрядок `SubS` є початком рядку `S`;

4) `pos(SubS,S, P)` призначено для випадку, коли підрядок `SubS` міститься у рядку `S`, починаючи з позиції `P`, але не є його початком;

5) `pos(_,_, 0)` повертає 0 у ситуації, якщо не виник жоден з попередніх випадків і, отже, підрядок не знайдено у рядку.

Наведемо декілька прикладів роботи створеного предикату:

– `pos("Prolog", "Мова Prolog", P)` повертає результат 0;

– `pos("Prolog", "Prolog", P)` повертає результат 1;

– `pos("Мова", "Prolog", P)` повертає результат 0.

Таким чином, створений предикат працює правильно і може використовуватись при розробці програм, які потребують обробки рядків.

Джерела та література

1. Ин Ц., Соломон Д. Использование Турбо-Пролога: Пер. с англ. – М.: Мир, 1993. – 608 с., ил.