

Języki formalne i metody kompilacji

Laboratorium nr 0

Stefan Sokołowski

W dalszym ciągu wykładu zostaną przedstawione systematyczne sposoby rozwiązywania takich zadań jak poniższe. Na razie spróbujmy je rozwiązać, jak kto potrafi. Może docenimy wtedy wartość sposobów przedstawionych w przyszłości...

Zadanie 1:

Napisać program w C¹, który przegląda tekst na pliku i oblicza, ile w nim jest słów `for`. Jednak proszę nie liczyć takich przypadków jak

`formalina` `termofor` `informacja` `for1`

w których `for` jest tylko częścią właściwą słowa. Liter i cyfr nie można uważać za granice słów; granicą słowa może być

- początek lub koniec całego tekstu,
- początek lub koniec wiersza,
- spacja,
- dowolny znak nie będący literą ani cyfrą.

Na przykład dla poniższego tekstu:

Większość języków programowania zawiera komendę `for` w jakiejś postaci; niektóre języki dodatkowo zawierają komendę `forall` albo `foreach`. W języku C składnia pętli `for` jest następująca:
`for(komenda; komenda; komenda) komenda;`
choć to `sformułowanie` może być mylące.

program powinien policzyć zielone `fory` (jest ich 3), a czerwonych (również 3) nie liczyć.

¹Proszę spróbować zrobić to zadanie oraz następne z tego zestawu pod **Unixem (Linuxem)** np. na [unixlabie](http://unixlabie.pl). Pod iis.ans-elblag.pl/~stefan/Dydaktyka/JezForm/Slajdy/naUnixlabie.html znajduje się ściągą z niezbędnymi komendami.

Zadanie 2:

Napisać program, który przegląda tekst na pliku i sprawdza, czy stojące w nim nawiasy okrągłe są prawidłowo postawione — czyli czy każdy lewy nawias ma dalej w tekście odpowiadający mu nawias prawy, oraz czy żaden prawy nawias nie występuje przed odpowiednim lewym.

Na przykład nawiasy w tekście

```
(2*(2+2))^2 = 64 -- prawda :)  
((2*2)+2)^2 = 64 -- nieprawda :(
```

są postawione nieprawidłowo (prawy nawias na końcu pierwszego wiersza nie jest poprzedzony odpowiednim lewym), a w tekście

```
((2*2)+2)^2 = 64 -- nieprawda :(  
(2*(2+2))^2 = 64 -- prawda :)
```

są postawione prawidłowo (lewy nawias na końcu pierwszego wiersza jest zamknięty prawym na końcu drugiego wiersza). Oczywiście to, czy równości są prawdziwe, nie ma znaczenia dla tego zadania.

Wskazówka:

Program, który to robi, może być naprawdę prosty, nawet bez użycia tablic, a tym bardziej struktur bardziej skomplikowanych.

Zadanie 3 (domowe specjalne):

Wszystko tak, jak w zad. 2; ale teraz mamy do czynienia z dwoma rodzajami nawiasów: okrągłymi i prostokątnymi. Można „zagnieżdżać” nawiasy okrągłe w prostokątnych i na odwrót; tak więc na przykład tekst

```
2*(2+2*[(2+2)*2+2])
```

jest OK. Ale nie można dopuścić, żeby nawiasy się „krzyżowały”; więc np. tekst

```
( [ ] )
```

jest niepoprawny.

Uwaga:

Pierwszej osobie, która przyśle mi (na s.sokolowski@ans-elblag.pl) poprawne rozwiązanie tego zadania, przysługują punkty do zaliczenia laboratorium/ćwiczeń. Liczba tych punktów zależy od oryginalności i elegancji rozwiązania. W mailu proszę koniecznie podać swoje **nazwisko, numer zajęć i numer zadania**.
