MP23 @ II UWr 4 maja 2023 r.

Lista zadań nr 9

Zadanie 1. (2 pkt)

Zdefiniuj gramatykę bezkontekstową (infiksowych) wyrażeń zbudowanych z liczb, operatorów dodawania, odejmowania, mnożenia, dzielenia, potęgowania, silni i liczby przeciwnej (unarny minus). Zadbaj o to, by gramatyka była jednoznaczna, a operatory miały odpowiednią łączność i priorytet zgodnie z ogólnoprzyjętymi konwencjami, np. silnia wiąże najsilniej, potęgowanie wiąże w prawo i silniej niż mnożenie.

Zadanie 2. (2 pkt)

Zdefiniuj gramatykę opisującą S-wyrażenia znane nam z języka Racket. Twoja gramatyka powinna uwzględniać specjalne znaczenie kropki, przecinka oraz apostrofów obu rodzajów.

Zadanie 3. (2 pkt)

Podziel moduł parser. rkt z wykładu na dwa moduły:

- parsing.rkt zawierający ogólne definicje dotyczące parsowania,
- expparser.rkt zawierający konkretne zastosowanie modułu parsing.rkt do parsowania wyrażeń arytmetycznych.

Zwróć uwagę na to, które definicje powinny zostać wyeksportowane, a które powinny zostać prywatne dla modułu. W module parsing.rkt zadbaj o odpowiednie kontrakty dla eksportowanych procedur.

MP23 @ II UWr Lista 9

Zadanie 4. (2 pkt)

Napisz funkcję tłumaczącą napis złożony z liter, cyfr, białych znaków i wybranych znaków interpunkcyjnych na kod Morse'a. Kropkę koduj jako znak kropki (#\.), kreskę jako znak podkreślenia (#_), odstęp między znakami jako pojedynczą spację, zaś dowolny ciąg białych znaków jako dwie spacje. Oto przykładowe wywołanie takiej procedury:

Przydatne mogą okazać się procedury string->list, list->string oraz char-whitespace?. Postaraj się zredukować złożoność kodu poprzez zakodowanie części algorytmu za pomocą danych.

Zadanie 5. (2 pkt)

Napisz funkcję tłumaczącą kod Morse'a na zwykły napis, np.

```
> (morse-decode "__ .__. ..__")
"MP 2022"
```

Również postaraj się zakodować część algorytmu za pomocą odpowiednich danych.

Zadanie 6.

Rozbuduj parser infiksowych wyrażeń arytmetycznych z wykładu o operatory mnożenia i dzielenia (o wyższym priorytecie). Nie przejmuj się, jeśli operatory łączą w prawo.

Zadanie 7.

Napraw parser infiksowych wyrażeń tak, by operatory łączyły w lewo.