MP23 @ II UWr 20 kwietnia 2023 r.

Lista zadań nr 8

Programowanie ze stanem

Poniższe trzy zadania rozwiąż w języku Racket.

Zadanie 1.

Zdefiniuj procedurę `cycle!`, która zapętla listę mutowalną, czyli przepina wskaźnik ogona ostatniego elementu na początek listy.

Zadanie 2.

Zdefiniuj procedurę `mreverse!`, która odwraca listę mutowalną „w miejscu”, czyli nie tworzy nowych bloczków `mcons`-em, a odpowiednio przepina wskaźniki.

Zadanie 3. (3 pkt)

Wzorując się na implementacji kolejek z wykładu zaimplementuj kolejki dwukierunkowe, czyli takie w których można wstawiać i usuwać element zarówno z jednej jak i z drugiej strony kolejki. Do implementacji kolejek dwukierunkowych użyj list dwukierunkowych, czyli takich w których każdy węzeł ma wskaźnik na następny i poprzedni węzeł listy.

Twoja implementacja powinna znajdować się w osobnym module, a eksportowane procedury powinny mieć odpowiednie kontrakty.

Składnia abstrakcyjna

Poniższe zadania rozwiąż w języku Plait.

Zadanie 4.

Zmodyfikuj parser wyrażeń arytmetycznych z wykładu tak, by nie konstruował *abstrakcyjnego drzewa rozbioru* (drzewa typu `Exp`), ale od razu obliczał podane wyrażenie do liczby.

Zadanie 5. (2 pkt)

Rozszerz kalkulator z wykładu o operacje potęgowania, silni i liczby przeciwnej (unarny minus). W tym celu najpierw uzupełnij składnię abstrakcyjną i interpreter, a następnie rozbuduj parser, tak aby obsługiwał nowe konstrukcje.

Zadanie 6. (2 pkt)

Zaproponuj składnię abstrakcyjną fragmentu języka Racket i zdefiniuj ją jako odpowiedni typ danych w języku Plait. Na fragment języka Racket rozważany w tym zadaniu zawiera tylko wyrażenia (nie zawiera definicji), na które mogą składać się zmienne, liczby, lambda-wyrażenia, aplikacje, formy specjalne **`let`**, **`if`** i **`cond`**. Do reprezentacji zmiennych użyj typu `Symbol`.

Zadanie 7. (2 pkt)

Napisz parser dla języka z poprzedniego zadania. Podobnie jak w parserze z wykładu, załóż, że składnia konkretna jest reprezentowana jako S-wyrażenie.