

Lista zagadnień nr 7

Przed zajęciami

W celu aktywnego udziału w zajęciach, należy wiedzieć, co to jest **składniowe wiązanie zmiennych** i czym różni się od **dynamicznego wiązania zmiennych**. Trzeba też wiedzieć, co to jest **środowisko** i **domknięcie**.

Ćwiczenie 1.

Co jest wynikiem obliczenia następującego programu?

```
(define (f)
  (let ([y 10])
    (lambda (x) (+ x y))))

(define y 0)

((f) 5)
```

Jaki byłby wynik, gdyby Racket używał dynamicznego wiązania zmiennych?

Ćwiczenie 2.

Jakie domknięcie będzie wartością następujących wyrażeń w języku z wykładu?

- `(let (x 5) (lambda (z) (let (y 5) (+ x y z))))`
- `(let (x 5) (lambda (x) (let (y 5) (+ x y))))`
- `((lambda (x) (lambda (y) (+ x y))) 10)`

Na zajęciach

UWAGA: Zadania do rozwiązania podczas zajęć wymagają napisania większej ilości kodu niż zadania z poprzednich list. Być może dobrym pomysłem jest rozwiązywanie ich (w grupach) przy komputerach?

Ćwiczenie 3.

Dodaj do języka z wykładu listy, wprowadzając formy specjalne `null`, `null?` i `pair?` z semantyką taką samą jak w Rackecie. Następnie dodaj do języka formę specjalną `list`. Można dodać ją jako lukier syntaktyczny, pisząc procedurę, która przetwarza wyrażenie używające `list` na wyrażenie używające `cons` i `null`.

Ćwiczenie 4.

Dodaj do języka z wykładu wartości boolowskie i formę specjalną `if` i `cond`. Rozbuduj język o operatory porównania (`=`, `>`, `>=`, itd.). Dodaj formy specjalne `and` i `or`, pamiętając o ich leniwej semantyce.

Ćwiczenie 5.

Pomijając fakt, że `let`-wyrażenia wiążą jedną zmienną, język z wykładu można traktować jako podzbiór Racketa. Ale czy na pewno? Jest pewna drobna różnica: znajdź wyrażenie, które pozwoli odróżnić interpreter Racketa od naszego interpretera (oczywiście rzeczy takie jak treść komunikatu o błędzie się nie liczą).

Ćwiczenie 6.

W kodzie przedstawionym na wykładzie domknięcie przechowuje ciało procedury i środowisko aktualne w momencie utworzenia procedury. Nie zawsze jednak trzeba przechowywać całe środowisko – wystarczy znać wartości zmiennych używanych przez procedurę. Zaproponuj procedurę, która potrafi stworzyć takie okrojone środowisko i użyj jej w ewaluatorze (być może przyda się coś w rodzaju procedury `free-vars` zdefiniowana na wykładzie).

Ćwiczenie 7.

Zdefiniuj procedurę `subst`, która wykonuje podstawienie wyrażenia za zmienną.

Wskazówka: Podobną procedurę widzieliśmy już w przypadku wyrażen `let`-ami na wykładzie 6.