Zadanie domowe na pracownię nr 12 i 13

W pliku `prac12.rkt` zamieszczonym na SKOS-ie znajduje się definicja składni pewnego języka programowania. Jest to język wyrażeń, który znamy z wykładu bez rekurencyjnych let-wyrażeń, ale zawierający inną składnię definicji funkcji i aplikacji. Tym razem chcemy, by funkcje były **wieloargumentowe** (ale nie ze zmienną liczbą argumentów). Nowa składnia abstrakcyjna języka dana jest następującymi strukturami:

- `(struct lambda-expr (xs b))` to definicja lambdy, gdzie `xs` to lista zmiennych, które są argumentami formalnymi definiowanej funkcji. Za to `b` to, tak jak wcześniej, ciało tej procedury. Np. wyrażenie w składni konkretnej Racketa dane jako `(lambda (x y z) k)` reprezentujemy w naszej składni abstrakcyjnej jako `(lambda-expr '(x y z) k)`.
- `(struct app-expr (f es))` to aplikacja funkcji będącej wynikiem wyrażenia `f` do wartości wyrażeń, które w składni abstrakcyjnej są na liście argumentów właściwych `es`. Odpowiednik Racketowego `(foo e1 e2 e3)`.
- `(struct apply-expr (f e))` to aplikacja funkcji do argumentów znajdujących się na liście, odpowiednik Racketowego `(apply foo (list e1 e2 e3))`.

Plik `prac12.rkt` zawiera także niepełną definicję interpretera – brakuje przypadków dla funkcji i aplikacji, zarówno w interpreterze jak i w definicji wartości. W tym zadaniu:

- Uzupełnij definicje `wartości` i `interpretera` tak, by implementowały powyższy nieformalny opis semantyki konstrukcji `lambda-expr`, `app-expr` i `apply-expr`.
- Przetestuj dobrze swoje rozwiązania

Semantyka w przypadku, gdy liczba argumentów w aplikacji nie zgadza się z liczbą argumentów oczekiwanych przez funkcję to błąd (interpreter powinien zakończyć się czytelnym komunikatem o błędzie, tak jak dzieje się to w przypadku próby odczytania wartości niezadeklarowanej zmiennej lub – w pełnym interpreterze z wykładu – próby aplikacji wyrażenia, którego wartością nie jest funkcja).

Uwaga: Rozwiązanie zadania dla `lambda-expr` i `app-expr` to zadanie nr 12, a dodatkowo dla `apply-expr` to zadanie nr 13. Proszę przysłać tylko jeden plik (**bezwzględnie** nazywający się `imie-nazwisko.rkt`, a **nie** `prac12.rkt`). Tak więc jeśli ktoś rozwiązuje zadanie dla wszystkich trzech nowych elementów składni, proszę nie wysyłać rozwiązania częściowego jako zadania 12, a jedynie całość jako rozwiązanie zadania 13. W takim wypadku jeśli rozwiązanie będzie prawidłowe, autor dostanie punkty zarówno za zadanie nr 12, jak i za zadanie nr 13.

Uwaga: Proszę pamiętać o testach i poprawnej nazwie pliku.