

PLT Redex – zadanie

Wojciech Jedynek

Wrocław, 13 listopada 2013

Zadanie polega na tym, aby po pierwsze rozszerzyć kod prezentowany podczas prezentacji, a po drugie zapoznać się z dodatkowymi możliwościami, które zostały naszkicowane w przykładzie piątym (plik przykładowy `p_05...`).

Konkretnie, rozważamy język L składający się z wyrażeń arytmetycznych oraz boolowskich:

$$e ::= n \mid (+ e e) \mid \text{true} \mid \text{false} \mid (\text{or } e e) \mid (\text{and } e e) \mid (\text{eq? } e e)$$

W ramach zadania należy (w PLT Redeksie):

1. Zakodować gramatykę języka L .
2. Zadać i zakodować gramatykę wartości v .
3. Zadać i zakodować gramatykę kontekstów.
4. Zadać i zakodować semantykę redukcyjną dla języka L . Dodawanie (+) i porównanie wartości (eq?) powinny być zdefiniowane tylko dla liczb; koniunkcja (and) i alternatywa (or) powinny być zdefiniowane tylko dla wartości boolowskich.
5. Korzystając z testowania losowego, sprawdzić czy Twoja relacja redukcji dla L jest konfluentna, tzn. czy idąc po dowolnej ścieżce redukcji zawsze dojdziemy do tego samego wyniku (do tej samej postaci normalnej).
6. Korzystając z testowania losowego, sprawdzić czy każde wyrażenie można sprowadzić do wartości.
7. Zmodyfikować gramatykę kontekstów i relację redukcji tak, aby otrzymać deterministyczną semantykę redukcyjną. Sprawdzić (poprzez testowanie losowe), że faktycznie zawsze można wykonać redukcję na co najwyżej jeden sposób. Należy rozważyć zarówno wariant leniwy jak i gorliwy dla wyrażeń `and` oraz `or`.
8. Wzorując się na przykładach z pliku `p05_bonus_arith_expr_many_formats.rkt`, należy zadać i zakodować semantykę naturalną dla języka L oraz sprawdzić (przez testowanie losowe) czy semantyki naturalna oraz redukcyjna są równoważne. Jeśli nie, to należy to poprawić :-)
9. Wzorując się i upraszczając przykład z tutoriala, należy zadać i zakodować prosty system typów dla wyrażeń. Sprawdzić następujące własności (opierając się o semantykę redukcyjną):
 - (a) *Postęp*: wyrażenie typowalne jest albo wartością, albo jest redukowalne (czyli wyrażenia zablokowane (and. *stuck*) nie typują się).
 - (b) *Zachowanie typu*: jeśli e ma typ τ oraz e redukuje się do e' , to e' także ma typ τ .

Po wykonaniu sprawdzenia przez testowanie losowe należy udowodnić te dwie własności na papierze.

Rozwiązania należy przesyłać na adres wjedynek@gmail.com do 12 grudnia 2013 r.