1 Pretty Printing

1.1 Additional Preliminaries

Save your solutions for this problem to a file named pretty.rkt.

Run the following expression (you must have required [extras.rkt](http://www.ccs.neu.edu/course/cs5010sp15/files/extras.rkt)) to check that your file is properly named and is in the proper directory:

(check-location "07" "pretty.rkt")

Add these additional provides at the top of your file (below the requires), so that we can test your solution:

([provide](http://docs.racket-lang.org/reference/require.html#%28form._%28%28lib._racket%2Fprivate%2Fbase..rkt%29._provide%29%29) expr->strings)

1.2 Problem Description

Here is a data definition for numeric expressions.

|  |
| --- |
| ; An Expr is one of: |
| ; - Integer |
| ;  - (make-add (cons Expr NEListOf<Expr>) Boolean) |
| ;  - (make-mul (cons Expr NEListOf<Expr>) Boolean) |
| ; Represents a numeric expression that is either an integer, or an addition |
| ; or multiplication of two or more expressions. |
| ; The Boolean flag indicates whether the expression uses infix or prefix notation. |
| ([define-struct](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28form._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._define-struct%29%29) add (exprs infix?)) |
| ([define-struct](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28form._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._define-struct%29%29) mul (exprs infix?)) |

Complete the data definition with appropriate data examples and templates.

Note that there are two pre-defined alternative data definitions of [NEListOf<X>](http://www.ccs.neu.edu/course/cs5010sp15/predatadefs.html" \l "%28tech._nelistof~3cx~3e%29), each with a different template. You may use either representation (or both).

Then implement the expr->strings function.

|  |
| --- |
| ; expr->strings : Expr Natural -> ListOf<String> |
| ; Returns a rendering of exp as a sequence of lines, |
| ; where each line is a string not longer than width characters. |
| ; EFFECT: errors if expr cannot fit width |
| ([define](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28form._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._define%29%29) (expr->strings expr width) [...](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28form._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._......%29%29)) |

Here are the rendering rules:

* Numbers are rendered with [number->string](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._number-~3estring%29%29).
* Compound expressions are rendered with surrounding parentheses.
* There are no spaces immediately inside open and closing parentheses.
* An expression is rendered as a single line if it fits within the specified width.
* Infix expressions are rendered with an operator between each expression and a space between an operator and each subexpression.

|  |
| --- |
| ([check-equal?](http://docs.racket-lang.org/rackunit/api.html#%28def._%28%28lib._rackunit%2Fmain..rkt%29._check-equal~3f%29%29) (expr->strings (make-add ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) 1 2 3) [true](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._true%29%29)) 11) |
| ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) "(1 + 2 + 3)") |
| "infix, single line") |

* Prefix expressions are rendered with only one operator at the beginning and a space separating each subexpression, like in Racket.

|  |
| --- |
| ([check-equal?](http://docs.racket-lang.org/rackunit/api.html#%28def._%28%28lib._rackunit%2Fmain..rkt%29._check-equal~3f%29%29) (expr->strings (make-add ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) 1 2 3) [false](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._false%29%29)) 100) |
| ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) "(+ 1 2 3)") |
| "prefix, single line") |

* If an expression cannot be rendered on one line, it should be stacked across multiple lines.
  + There are no spaces before a new line.
  + Parentheses should never go on their own line.
  + Infix expressions are stacked with each operator on its own line and subexpressions and operators of that expression at the same left alignment. This may require indenting subexpressions with spaces, depending on how many surrounding parentheses there are.

|  |
| --- |
| ([check-equal?](http://docs.racket-lang.org/rackunit/api.html#%28def._%28%28lib._rackunit%2Fmain..rkt%29._check-equal~3f%29%29) (expr->strings (make-add ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) 10 20 30) [true](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._true%29%29)) 7) |
| ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) "(10" |
| " +" |
| " 20" |
| " +" |
| " 30)") |
| "infix stacking") |

* + Prefix expressions are stacked such that the first line has the operator and the first subexpression. All other subexpressions should be on their own line, indented with spaces to align with the first subexpression.

|  |
| --- |
| ([check-equal?](http://docs.racket-lang.org/rackunit/api.html#%28def._%28%28lib._rackunit%2Fmain..rkt%29._check-equal~3f%29%29) (expr->strings (make-add ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) 10 20 30) [false](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._false%29%29)) 7) |
| ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) "(+ 10" |
| "   20" |
| "   30)") |
| "prefix stacking") |

* + An expression may have mixed infix and prefix notation expressions.

|  |
| --- |
| ([define](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28form._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._define%29%29) EXAMPLE |
| (make-add ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) (make-mul ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) 1 2) [true](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._true%29%29)) |
| (make-add ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) 3 4) [false](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._false%29%29))) |
| [true](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._true%29%29))) |
| (begin-for-test |
| ([check-equal?](http://docs.racket-lang.org/rackunit/api.html#%28def._%28%28lib._rackunit%2Fmain..rkt%29._check-equal~3f%29%29) (expr->strings EXAMPLE 100) |
| ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) "((1 \* 2) + (+ 3 4))") |
| "unstacked") |
| ([check-equal?](http://docs.racket-lang.org/rackunit/api.html#%28def._%28%28lib._rackunit%2Fmain..rkt%29._check-equal~3f%29%29) (expr->strings EXAMPLE 9) |
| ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) "((1 \* 2)" |
| " +" |
| " (+ 3 4))") |
| "one level stacking") |
| ([check-equal?](http://docs.racket-lang.org/rackunit/api.html#%28def._%28%28lib._rackunit%2Fmain..rkt%29._check-equal~3f%29%29) (expr->strings EXAMPLE 8) |
| ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) "((1 \* 2)" |
| " +" |
| " (+ 3" |
| "    4))") |
| "two level stacking (1 subexpression)") |
| ([check-equal?](http://docs.racket-lang.org/rackunit/api.html#%28def._%28%28lib._rackunit%2Fmain..rkt%29._check-equal~3f%29%29) (expr->strings EXAMPLE 7) |
| ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) "((1" |
| "  \*" |
| "  2)" |
| " +" |
| " (+ 3" |
| "    4))") |
| "two level stacking (2 subexpressions)")) |

* If an expression cannot be rendered, even with stacking, then the function should report an appropriate error message. Use the [error](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._error%29%29) function to report an error. To test erroring function calls, use check-error.

([check-error](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28form._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._check-error%29%29) (expr->strings EXAMPLE 4) "doesn't fit")

There is a display-strings! function in extras.rkt to help you debug.

|  |
| --- |
| ; display-strings! : ListOf<String> -> Void |
| ; EFFECT: Prints the given list of strings, one per line. |
| Examples: |
| |  | | --- | | > ([require](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28form._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._require%29%29) "extras.rkt") | | extras.rkt Mon Jan 12 13:33:22 2015 | | |  | | --- | |  | | | |  | | --- | | > (display-strings! ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) "(+ 22" | | "   333" | | "   44)")) | | | |  | | --- | | (+ 22 | | 333 | | 44) | | | |  | | --- | |  | | | |  | | --- | | > (display-strings! ([list](http://docs.racket-lang.org/htdp-langs/intermediate-lam.html#%28def._htdp-intermediate-lambda._%28%28lib._lang%2Fhtdp-intermediate-lambda..rkt%29._list%29%29) "(+ (\* 22" | | "      333" | | "      44)" | | "   (66" | | "    +" | | "    67" | | "    +" | | "    68)" | | "   (\* 77 88))")) | | | |  | | --- | | (+ (\* 22 | | 333 | | 44) | | (66 | | + | | 67 | | + | | 68) | | (\* 77 88)) | | |