Documentation for the `compound.rkt` teachpack

The teachpack `compound.rkt` contains structures used for manipulating chemical compounds. Information on how to install teachpacks can be found on the course website.

1 Structure and data definitions

(*define-struct compound* (*name lop*))
;; A Compound is a (make-compound Sym Part-List)

;; A Part-List is a (listof Part)

(*define-struct part* (*size eoc*))
;; A Part is a (make-part Nat (anyof Element Compound))
;; This indicates that the part consists of size elements (or compounds) eoc.

(*define-struct element* (*name mmass*))
;; An Element is a (make-element Sym Num)
;; where mmass denotes the molar mass, i.e., the mass of
;; one mole of the substance, 6.02 x 10^23 atoms.

2 Built-in data

To save you time, the teachpack includes the definition of many constants. You are welcome to create more definitions along with the functions that you create for labs and assignments.

(**define** *hydrogen* (*make-element* 'H 1.01))
(**define** *carbon* (*make-element* 'C 12.01))
(**define** *oxygen* (*make-element* 'O 16.00))
(**define** *sodium* (*make-element* 'Na 22.99))
(**define** *sulfur* (*make-element* 'S 32.07))
(**define** *argon* (*make-element* 'Ar 39.95))
(**define** *calcium* (*make-element* 'Ca 40.08))
(**define** *iron* (*make-element* 'Fe 55.85))
(**define** *phosphorus* (*make-element* 'P 30.97))

(**define** *po-four* (*make-compound* 'PO4 (*list* (*make-part* 1 *phosphorus*) (*make-part* 4 *oxygen*))))
(**define** *so-four* (*make-compound* 'SO4 (*list* (*make-part* 1 *sulfur*) (*make-part* 4 *oxygen*))))
(**define** *calcium-phosphate*
  (*make-compound* 'calcium-phosphate (*list* (*make-part* 3 *calcium*) (*make-part* 2 *po-four*))))
(**define** *glucose*
  (*make-compound* 'glucose
                   (*list* (*make-part* 6 *carbon*) (*make-part* 12 *hydrogen*) (*make-part* 6 *oxygen*))))

```
(define iron-sulfate
  (make-compound 'iron-sulfate (list (make-part 2 iron) (make-part 3 so-four))))
(define cinnamaldehyde
  (make-compound 'cinnamaldehyde
                 (list (make-part 9 carbon) (make-part 8 hydrogen) (make-part 1 oxygen))))
```

For convenience in testing, we have created "elements" and "compounds" that indicate their "molar mass" in their names. It is also possible to read from the names of the compounds the numbers of elements in each (that is, none has more than nine of each of the elements). It may be easier to untangle where an error occurs than in a compound where the relation between molar mass, elements, and parts is less obvious.

```
(define e1 (make-element 'e1 1))
(define e10 (make-element 'e10 10))
(define e100 (make-element 'e100 100))

(define c123 (make-compound 'c123
                            (list (make-part 1 e100) (make-part 2 e10) (make-part 3 e1))))
(define c45 (make-compound 'c45
                           (list (make-part 4 e10) (make-part 5 e1))))
(define c200 (make-compound 'c200 (list (make-part 2 e100))))
(define c489 (make-compound 'c489
                            (list (make-part 1 c123) (make-part 1 c45) (make-part 3 e100)
                                  (make-part 2 e10) (make-part 1 e1))))
(define c304 (make-compound 'c304
                            (list (make-part 1 c200) (make-part 1 e100) (make-part 4 e1))))
```