Unit Conversion

There are hundreds of units of measurement in common use.[1]

Conversion between different units is an important problem:

- Most programming languages do not support or check units.
- Humans have difficulty converting units.
- Use of the wrong units caused a $327 million spacecraft to crash into Mars.

  `en.wikipedia.org/wiki/Mars_Climate_Orbiter`
- Although unit conversion is actually easy, several complex and costly methods have been published.

[1] G. Novak, "Conversion of Units of Measurement", *IEEE Transactions on Software Engineering*, vol. 21, no. 8 (August 1995), pp. 651-661.

Conversion Using SI Units

The scientific standard for units is the *Systeme Internationale*

`en.wikipedia.org/wiki/International_System_of_Units`

previously known as the meter-kilogram-second (MKS) system.

Conversion of units is simple: Each unit is assigned a number that converts it to the corresponding SI unit. Given two units, *source* and *goal*,

$$source * f_{source} = SI = goal * f_{goal}$$

$$goal = source * (f_{source} / f_{goal})$$

When there are multiple units, their factors are multiplied or divided as above. This process is $O(n)$ for a quotient involving n units.

Unit Checking

It is necessary to check that a unit conversion is correct: a length cannot be converted to a mass. Each unit has a corresponding abstract unit, which is a quotient of two products. For example, a force such as `newton` has an abstract unit:

```
(/ (* mass length) (* time time) )
```

If the abstract units for source and goal are divided, and corresponding units in numerator and denominator are cancelled, the result should be 1.

Symbolically, if the terms are collected into a quotient of two lists and the lists are sorted, the two lists should be equal.

For efficiency, the units can be encoded as 32-bit integers that are added and subtracted, giving a result of 0 for a correct conversion.

Special Conversions

Two special conversions of incompatible units are often seen:

- Mass to weight (force)
- Mass to energy

These can be detected by the pattern of abstract units, allowing the correct conversion factor to be applied.

Unit Simplification

A combination of units can be simplified symbolically as follows:

- Cancel corresponding units in numerator and denominator.
- Search for the known composite unit that covers the most terms in the existing expression.
- Repeat until all terms are covered.

```
>(glsimplifyunit '(/ volt ohm))

AMPERE
```

Problem Solving by Unit Conversion

Some physics problems are just unit conversions:

How many Watts is a person on average?

```
To convert from source unit:
(/ (* 2000 KILO CALORIE) DAY)
to goal unit:
WATT
multiply source quantity by: 96.85185185
```

Fixing Conversion Errors

Experience with an on-line conversion system showed that users would ask for impossible conversions, e.g. convert amps to horsepower.

However, it is possible to lead the user to a correct specification:

- Divide the goal unit by the source unit symbolically
- Simplify the resulting unit expression
- Present the result to the user:

  > The units could be converted if you multiplied by an appropriate quantity of VOLT

Units in Programming Languages

It is possible to incorporate units into the type system, make legal conversions automatically, and detect errors:

```
>(gldefun test ( (x (units real meter))
                 (z (units real inch)) )
            (z = x) )

result type: (UNITS REAL INCH)
(LAMBDA (X Z) (SETQ Z (* 39.37007874015748 X)))
```

```
(gldefun testb ( (x (units real meter))
                 (z (units real kilogram)) )
           (z = x) )

glisp error detected in function TESTB
Cannot convert METER to KILOGRAM
in expression: (Z = X)
```