1 Types and measures

1.1 Unit of Measure

F# allows for assigning *unit of measure* to the following types, · unit of measure

`sbyte`, `int`, `int16`, `int32`, `int64`, `single`, `float32`, `float`, and `decimal`.

by using the syntax,

```
"[<Measure>] type" unit-name [ "=" unit-expr ]
```

and then use `"<" unit-name ">"` as suffix for literals. E.g., defining unit of measure 'm' and 's', then we can make calculations like,

Listing 1.1: fsharpi, floating point and integer numbers may be assigned unit of measures.

```
> [<Measure>] type m
- [<Measure>] type s
- let a = 3<m/s^2>
- let b = a * 10<s>
- let c = 4 * b;;

[<Measure>]
type m
[<Measure>]
type s
val a : int<m/s ^ 2> = 3
val b : int<m/s> = 30
val c : int<m/s> = 120
```

However, if we mixup unit of measures under addition, then we get an error,

Listing 1.2: fsharpi, unit of measures adds an extra layer of types for syntax checking at compile time.

```
> [<Measure>] type m
- [<Measure>] type s
- let a = 1<m>
- let b = 1<s>
- let c = a + b;;

  let c = a + b;;
  ------------^

/Users/sporring/repositories/fsharpNotes/stdin(63,13): error
  FS0001: The unit of measure 's' does not match the unit of
  measure 'm'
```

Unit of measures allow for "`*`", "`/`", and "`^`" for multiplication, division and exponentiation. Values with units can be casted to *unit-less* values by casting, and back again by · unit-less

multiplication as,

Listing 1.3: fsharpi, typecasting unit of measures.

```
> [<Measure>] type m
- let a = 2<m>
- let b = int a
- let c = b * 1<m>;;

[<Measure>]
type m
val a : int<m> = 2
val b : int = 2
val c : int<m> = 2
```

Compound symbols can be declared as,

Listing 1.4: fsharpi, aggregated unit of measures.

```
> [<Measure>] type s
- [<Measure>] type m
- [<Measure>] type kg
- [<Measure>] type N = kg * m / s^2;;

[<Measure>]
type s
[<Measure>]
type m
[<Measure>]
type kg
[<Measure>]
type N = kg m/s ^ 2
```

For fans of the metric system there is the International System of Units, and these are built-in in `Microsoft.FSharp.Data.UnitSystems.SI.UnitSymbols` and give in Table 1.1. Hence, using the predefined unit of seconds, we may write,

Listing 1.5: fsharpi, SI unit of measures are built-in.

```
> let a =
   10.0<Microsoft.FSharp.Data.UnitSystems.SI.UnitSymbols.s>;;

val a : float<Data.UnitSystems.SI.UnitSymbols.s> = 10.0
```

To make the use of these predefined symbols easier, we can import them into the present scope by the *open* keyword,

· open

Listing 1.6: fsharpi, simpler syntax by importing, but beware of namespace pollution.

```
> open Microsoft.FSharp.Data.UnitSystems.SI.UnitSymbols;;
> let a = 10.0<s>;;

val a : float<s> = 10.0
```

The open keyword should be used with care, since now all the bindings in `Microsoft.FSharp.Data.UnitSystems.SI` have been imported into the present scope, and since we most likely do not know, which bindings have been used by the programmers of `Microsoft.FSharp.Data.UnitSystems.SI.UnitSymbols`, we do not know which identifiers to avoid, when using `let` statements. We have obtained, what is known as *namespace pollution*. Read more about namespaces in **??**.

· namespace pollution

Unit	Description
`A`	Ampere, unit of electric current.
`Bq`	Becquerel, unit of radioactivity.
`C`	Coulomb, unit of electric charge, amount of electricity.
`cd`	Candela, unit of luminous intensity.
`F`	Farad, unit of capacitance.
`Gy`	Gray, unit of an absorbed dose of radiation.
`H`	Henry, unit of inductance.
`Hz`	Hertz, unit of frequency.
`J`	Joule, unit of energy, work, amount of heat.
`K`	Kelvin, unit of thermodynamic (absolute) temperature.
`kat`	Katal, unit of catalytic activity.
`kg`	Kilogram, unit of mass.
`lm`	Lumen, unit of luminous flux.
`lx`	Lux, unit of illuminance.
`m`	Metre, unit of length.
`mol`	Mole, unit of an amount of a substance.
`N`	Newton, unit of force.
`ohm`	Unitnames.o SI unit of electric resistance.
`Pa`	Pascal, unit of pressure, stress.
`s`	Second, unit of time.
`S`	Siemens, unit of electric conductance.
`Sv`	Sievert, unit of dose equivalent.
`T`	Tesla, unit of magnetic flux density.
`V`	Volt, unit of electric potential difference, electromotive force.
`W`	Watt, unit of power, radiant flux.
`Wb`	Weber, unit of magnetic flux.

Table 1.1: International System of Units.

Using unit of measures is advisable for calculations involving real-world values, since some semantical errors of arithmetic expressions may be discovered by checking the resulting unit of measure.