Chapter 25

Locales

```
module Locale(TimeLocale(..), defaultTimeLocale) where

data TimeLocale = TimeLocale {
        wDays  :: [(String, String)],   -- full and abbreviated week days
        months :: [(String, String)],   -- full and abbreviated months
        amPm   :: (String, String),     -- AM/PM symbols
        dateTimeFmt, dateFmt,           -- formatting strings
          timeFmt, time12Fmt :: String
        } deriving (Eq, Ord, Show)

defaultTimeLocale :: TimeLocale
```

The `Locale` library provides the ability to adapt to local conventions. At present, it supports only time and date information as used by `calendarTimeToString` from the `Time` library.

25.1 Library `Locale`

```
module Locale(TimeLocale(..), defaultTimeLocale) where

data TimeLocale = TimeLocale {
        wDays  :: [(String, String)],   -- full and abbreviated week days
        months :: [(String, String)],   -- full and abbreviated months
        amPm   :: (String, String),     -- AM/PM symbols
        dateTimeFmt, dateFmt,           -- formatting strings
          timeFmt, time12Fmt :: String
        } deriving (Eq, Ord, Show)
```

```haskell
defaultTimeLocale :: TimeLocale
defaultTimeLocale =  TimeLocale {
        wDays  = [("Sunday",    "Sun"),  ("Monday",    "Mon"),
                  ("Tuesday",   "Tue"),  ("Wednesday", "Wed"),
                  ("Thursday", "Thu"),   ("Friday",    "Fri"),
                  ("Saturday", "Sat")],
        months = [("January",    "Jan"), ("February",  "Feb"),
                  ("March",      "Mar"), ("April",     "Apr"),
                  ("May",        "May"), ("June",      "Jun"),
                  ("July",       "Jul"), ("August",    "Aug"),
                  ("September", "Sep"), ("October",    "Oct"),
                  ("November",  "Nov"), ("December",   "Dec")],
        amPm = ("AM", "PM"),
        dateTimeFmt = "%a %b %e %H:%M:%S %Z %Y",
        dateFmt = "%m/%d/%y",
        timeFmt = "%H:%M:%S",
        time12Fmt = "%I:%M:%S %p"
        }
```