Set-once memory

Set-once memory is a special kind of synchronous memory with the following constraints.

- Multiple fibers can access the memory.
- We initialize the memory at most once, during the first \texttt{set} operation.
- We spin until the thunk has evaluated.

Our implementation has the following signature. The \texttt{new} operation creates a set-once cell. The \texttt{get} operation obtains the value of the cell. This operation spins if the thunk is already evaluating.

\begin{verbatim}
val new : 'a thunk -> 'a set_once_mem
val get : 'a set_once_mem -> 'a
\end{verbatim}

We can completely specify the behavior of set-once memory in terms of futures.

\begin{verbatim}
type 'a set_once_mem = 'a future
val new = future
fun set sm = ( case poll sm
    of NONE => touch sm
        SOME v => v
    (* end case *))
\end{verbatim}