proposed move to System F

At the Manticore meeting on 11/7/07, we discussed moving BOM in the direction of System F.

The following summarize the proposed changes to the <code>BOM</code> module:

New syntactic forms:

- let \( x : \tau = y \) \([\tauVec]\)
- apply \( f \) \([\tauVec] (...) / (...)\)
- let \( x : \tau = \text{CONS} \) \([\tauVec] (...)\)
- let \( x : \tau = \text{CONST} \) \([\tauVec]\)
- throw \( k \) \([\tau] (...)\)

New types:

- alpha (* type variables *)
- forall \( \alphaVec . \tau \)
- \( T \) \([\tauVec]\)
- \( \tau + \tau \)

Here are the changes to the datatypes in the <code>BOM</code> module:

datatype term
= ...

```
E_Apply of (var * ty list * var list * var list)
E_Throw of (var * ty list * var list)
E_HLOp of (hlop * ty list * var list * var list)
```

and rhs
= ...

```
E_TyApply of (var * ty list)
E_DCon of (data_con * ty list * ty list)
```

and lambda = FB of

```
Unknown macro: { f }
```

and pat
= P_DCon of (data_con * ty list * var list)

```
P_DConst of (data_con * ty list)
...
```

and the changes to the BOMTy module:

datatype ty
= T_Var of ty_var

```
T_Var of ty_var
T_Forall of (ty_var list * ty)
T_TyCon of (tyc * ty list)
```

We'll also need to add the <code>ty_var</code> type and make changes to the representation of <code>tyc</code>s.