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`

```latex
= ...
```

```latex
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)
```

```latex
and rhs

= ...
```

```latex
E\_TyApply of (var * ty list)
E\_DCon of (data_con * ty list * ty list)
```

```latex
and lambda = FB of

| Unknown macro: { f } |
```

```latex
and pat

= P\_DCon of (data_con * ty list * var list)
```

```latex
P\_DConst of (data_con * ty list)
```

```latex
and the changes to the BOMTy module:

datatype `ty`

```latex
= T\_Var of ty\_var
```

```latex
T\_Forall of (ty\_var list * ty)
```

```latex
T\_TyCon of (tyc * ty list)
```

```latex
...`
```

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