A statically safe alternative to virtual types

Kim B. Bruce

Williams College

Martin Odersky

University of South Australia

Philip Wadler

Bell Labs, Lucent Technologies

Part I: Parametric vs. Virtual

- Collections: Parametric beats Virtual
  - zip
- Families: Virtual beats Parametric
  - list with length

Zip: Parametric

```
public class List<A> {

  public class Zipper<B> {

    public List<Pair<A,B>> zip(List<B> yl) {...}

  }

}
```

Zip: Virtual

```
public class List {
  typedef A as Object;
  public class Zipper {
    typedef B as Object;
    public class PairAB extends Pair {
      typedef Fst as A;
      typedef Snd as B;
    }
    public class ListB extends List {
      typedef A as B;
    }
    public class ListPairAB extends List {
      typedef A as PairAB;
    }
    public ListPairAB zip (ListB yl) { ... }
  }
}
```

List with length: Virtual

```
public class List {
  deftype This as List;
  protected char h;
  protected This t;
  public List (char h, This t) { setHead(h); setTail(t); }
  public char head () { return h; }
  public This tail () { return t; }
  public void setHead (char h) { this.h = h; }
  public void setTail (This t) { this.t = t; }
}
public class LenList extend List {
  deftype This as LenList;
  protected int l;
  public LenList (char h, This t) { super(h,t); }
  public void setTail (This t) { super.setTail(t); l = ...; }
  public int length () { return l; }
}
```

List with length: Virtual, continued

```
public class Breakit {

  public void breakit () {

    List xl = new LenList('a', null);

    List yl = new List('b', null);

    xl.setTail(yl);  // fails at run-time

  }

}
```

List with length: Parametric

```
public class ListF<This extends ListF<This>> {
  protected char h;
  protected This t;
  protected List (char h, This t) { setHead(h); setTail(t); }
  ...
}
public class List extends ListF<List> {
  public List (char h, List t) { super(h, t); }
}
public class LenListF<This extends LenListF<This>> extend ListF<This> {
  protected int l;
  public LenList (char h, This t) { super(h,t); }
  ...
}
public class LenList extends LenListF<LenList> {
  public LenList (char h, List t) { super(h, t); }
}
```

List with length: Parametric, continued

```
public class Breakit {

  public void breakit () {

    List xl = new LenList('a', null);

    List yl = new List('b', null);

    xl.setTail(yl);  // fails at compile-time

  }

}
```