Before:

Example Code from ASM-2.0 (BFagent.m):

```
// Declare a pointer to space for BF_fcast struct
struct BF_fcast *fcast;
// Allocate memory for forecasts
  fcast = calloc(p->numfcasts,sizeof(struct BF_fcast));
  if(!fcast)
    printf("There was an error allocating space for
           fcast.");

struct BF_fcast *fptr, *topfptr;

// Initialize the forecasts
  topfptr = fcast + p->numfcasts;
  for (fptr = fcast; fptr < topfptr; fptr++)
    {
      fptr->forecast = 0.0;
      fptr->lforecast = global_mean;
      fptr->count = 0;
      fptr->lastactive = 1;
      fptr->specificity = 0;
      fptr->next = fptr->lnext = NULL;
 /* Allocate space for this forecast's conditions out of
           total allocation */
      fptr->conditions = conditions;
      conditions += condwords;
 /* Initialise all conditions to don't care /
      cond = fptr->conditions;
      for (word = 0; word < condwords; word+)
        cond[word] = 0;
  /* Add non-zero bits as specified by probabilities */
      if(fptr!=fcast) /* protect rule 0 */

      /* Compute average specificity */
        specificity = 0;
        for (fptr = fcast; fptr < topfptr; fptr++)
          {
           specificity += fptr->specificity;
          }
        avspecificity = ((double) specificity)/p->numfcasts;
      }
```

After:

Current ASM Code:

```
//Create Array to hold forecast objects
 fcastList=[Array create: [self getZone]
                    setCount: numfcasts];

// Create Forecast objects, add to fcastList
// The "know nothing" forecast is first
  [fcastList atOffset: 0 put: [self
createNewForecast]];
//create rest of forecasts with random conditions
  for ( i = 1; i < numfcasts; i++)
    {
      id aForecast =[self createNewForecast] ;
//initialize
      [self setConditionsRandomly: aForecast];
      [fcastList atOffset: i put: aForecast];
     }

/* Compute average specificity */

 index = [ fcastList begin: [self getZone] ];
 for ( aForecast = [index next]; [index getLoc]
== Member; aForecast = [index next] )
    {
          sumspecificity += [aForecast
getSpecificity];
     }
  avspecificity =
          (double)
sumspecificity/(double)numfcasts;
  [index drop];
}
```