

List Comprehensions

Example: $[x * x \mid x \leftarrow [1..5], \text{odd } x] = [1, 9, 25]$

$$[\underline{\text{exp}} \mid \underline{\text{var}} \leftarrow \underline{\text{exp}'}, Q] = \text{concat} (\text{map } f \underline{\text{exp}'})$$

where $f \underline{\text{var}} = [\underline{\text{exp}} \mid Q]$

$$\begin{aligned} & [\underline{\text{exp}} \mid \underline{\text{var}} \leftarrow [a_1, \dots, a_n], Q] \\ &= \text{concat} (\text{map } f [a_1, \dots, a_n]) \text{ where } f \underline{\text{var}} = [\underline{\text{exp}} \mid Q] \\ &= f a_1 ++ \dots ++ f a_n \text{ where } f \underline{\text{var}} = [\underline{\text{exp}} \mid Q] \\ &= [\underline{\text{exp}} \mid Q] [x/a_1] ++ \dots ++ [\underline{\text{exp}} \mid Q] [x/a_n]. \end{aligned}$$
$$[\underline{\text{exp}} \mid \underline{\text{exp}'}, Q] = \text{if } \underline{\text{exp}'} \text{ then } [\underline{\text{exp}} \mid Q] \text{ else } []$$

Quicksort in an Imperative Language (JAVA)

```
static void qsort(int[] a, int lo, int hi) {
    int h, l, p, t;

    if (lo <= hi) { l = lo;
                    h = hi;
                    p = a[hi];

                    do { while ((l < h) && (a[l] <= p))
                            l = l+1;
                        while ((h > l) && (a[h] >= p))
                            h = h-1;
                        if (l < h) { t = a[l];
                                    a[l] = a[h];
                                    a[h] = t;          }
                    } while (l < h);

                    t = a[l];
                    a[l] = a[hi];
                    a[hi] = t;

                    qsort( a, lo, l-1);
                    qsort( a, l+1, hi); } }
```