Spatial complexity

- So far we have measured execution time, which is called temporal complexity
- We can also measure memory use, which is called spatial complexity
  - How much memory is used for input size *n*?
- There are *two* ways to measure space:
  - Active memory $m_{active}(n,t)$ in memory words: total number of words in use by the program at time *t*
  - Memory consumption $m_{consume}(n,t)$ in words per second: number of words allocated per second at time *t*

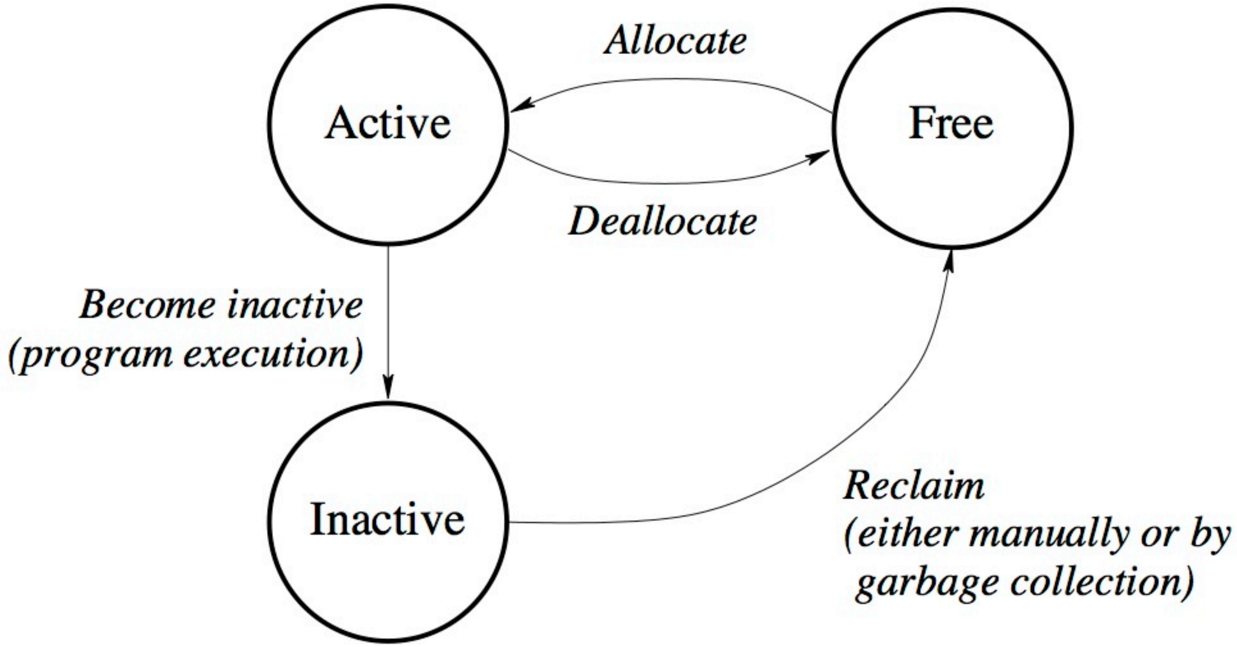
Active memory versus memory consumption

- Active memory is how many words the program needs at any time
  - An in-memory database has a large active memory (= the size of the database) but a small memory consumption (= little memory is needed to calculate the result of a query)
- Memory consumption is how many words the program creates per time unit
  - A simulation of molecules moving in a box has a large memory consumption (= each particle position is recalculated at every time step according to a complex computation that needs much temporary data) but a small active memory (= little memory is needed to store positions and velocities of all particles)

Intuition: Your active size is how much you weigh (in kg); your food consumption is how much you eat (in kg/day)

- The food you eat is used by your metabolism but only a small part (or none) becomes part of your body! Even if you eat 2 kg/day you won't weigh 200 kg after 100 days.

Life cycle of a memory word



We can explain the difference between active memory and memory consumption through the life cycle of a memory word

- Active memory is all words the program needs at a given time; memory consumption is how many words are allocated from the Free set per second
- When during execution the program no longer needs a word (when the word is no longer referenced), then the word automatically becomes Inactive
- Periodically, the system collects all Inactive words and puts them back in the Free set (this is usually done by an algorithm called *garbage collection*)

Computational complexity and kernel language

- We can calculate the temporal complexity using the kernel language
    - Each instruction in the kernel language consists of one or more primitive operations with a constant time
    - We count the primitive operations as the program executes
- We can calculate the spatial complexity using the kernel language
    - We calculate memory consumption by counting the words that the kernel instructions allocate
    - We calculate active memory by starting from the semantic stack and following all references (see lesson 6)
- So we can use the semantics of lesson 6 to calculate both the temporal and spatial complexity
    - We can even do garbage collection using this semantics!