6??

Split heap into From-Space and To-Space.

Allocate blocks from the From-Space.

When From-Space is exhausted, copy live blocks to To-Space, and switch the roles.

Advantages: fast (no pointer manipulations), copying automatically compacts the heap.

Disadvantages: wastes half of memory, will copy blocks that are long-lived.

* 1. We proceed by induction over the structure of the grammar.

To show that there exists an n such that num = 3n.

Base case: num = 11

112 = 310 so set n = 1.

Base case: num = 1001

10012 = 910 so set n = 3.

Inductive case: num = m 0

Assume I.H P(m)

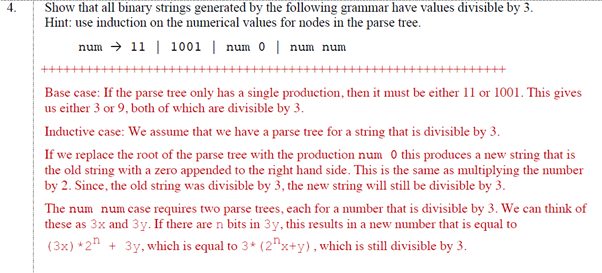
This operation is equivalent to m2 = m\*m. If m = 3n, then m2 = 9n2 which is also clearly divisible by 3.

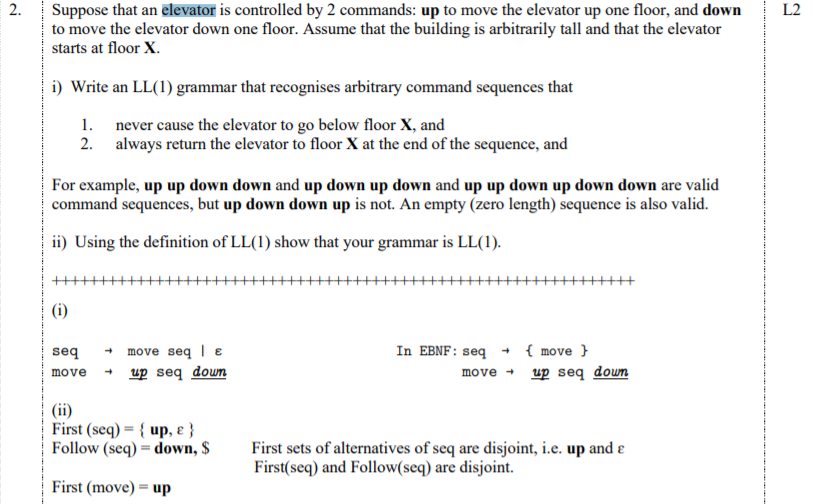
Inductive case: num = m1 m2

Assume I.H P(m1) and P(m2)

This is equivalent to shifting m1 by the length of m2 and then adding them.

If m1 = 3n1 and m2 = 3n2, then m1 m2 = (3n1)l + 3n2 = 3 ((3)l-1(n1)l + n2) which is clearly divisible by 3.





1. 1. Recall that a variable is live if its current value will be used in the future.

|  |  |
| --- | --- |
| Instruction | Variables live after instruction |
| int i = 0; | i |
| int j = i + 1; | j |
| i = j; | i |
| int j = i + 1; | j |



while (true) {

if (a) {

// ...

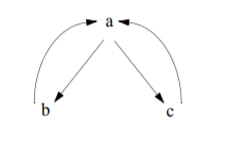
} else {

// ...

}

}

Which should look like





int a = ...;

int b = ...;

int t = 0;

while (false) {

t = a + b;

}

int x = t;

Since t = a + b does not dominate all exits of the loop.

* 1. For very short functions, where the call overhead is larger than the body of the function, or single-use functions where no duplication occurs.
  2. “If it is a leaf procedure, just use the caller-save registers as often as possible. If it is a non-leaf procedure, if the variable is live across the call, callee-save register is preferred. Otherwise, just use the caller-save registers.”

<https://zhongshugu.wordpress.com/2011/02/23/caller-save-registers-and-callee-save-registers/>