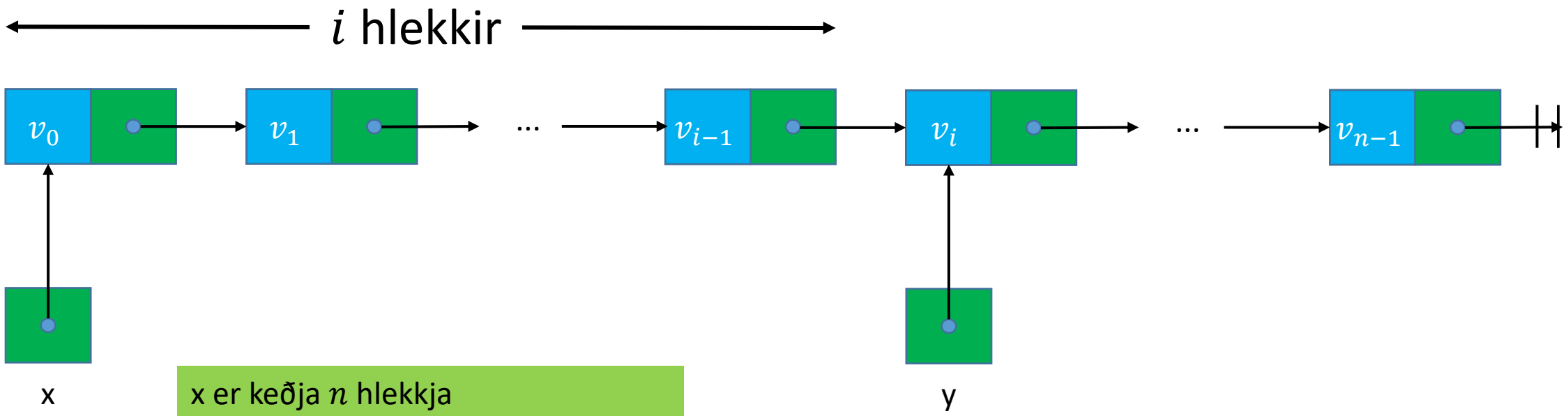
Eintengdir hlekkir og listavinnsla

Ýmsar fastayrðingar

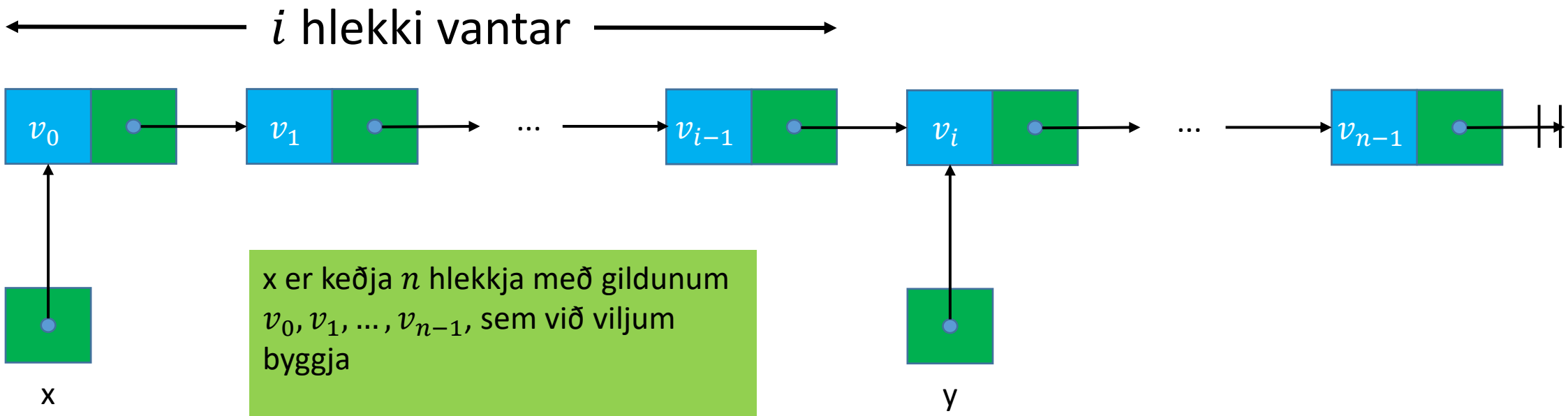
Talning hlekkja



x er keðja n hlekkja

y er einhver aftari hluti keðjunnar x og fjöldi þeirra hlekkja sem eru í x en ekki í y er i

Uppbygging keðju

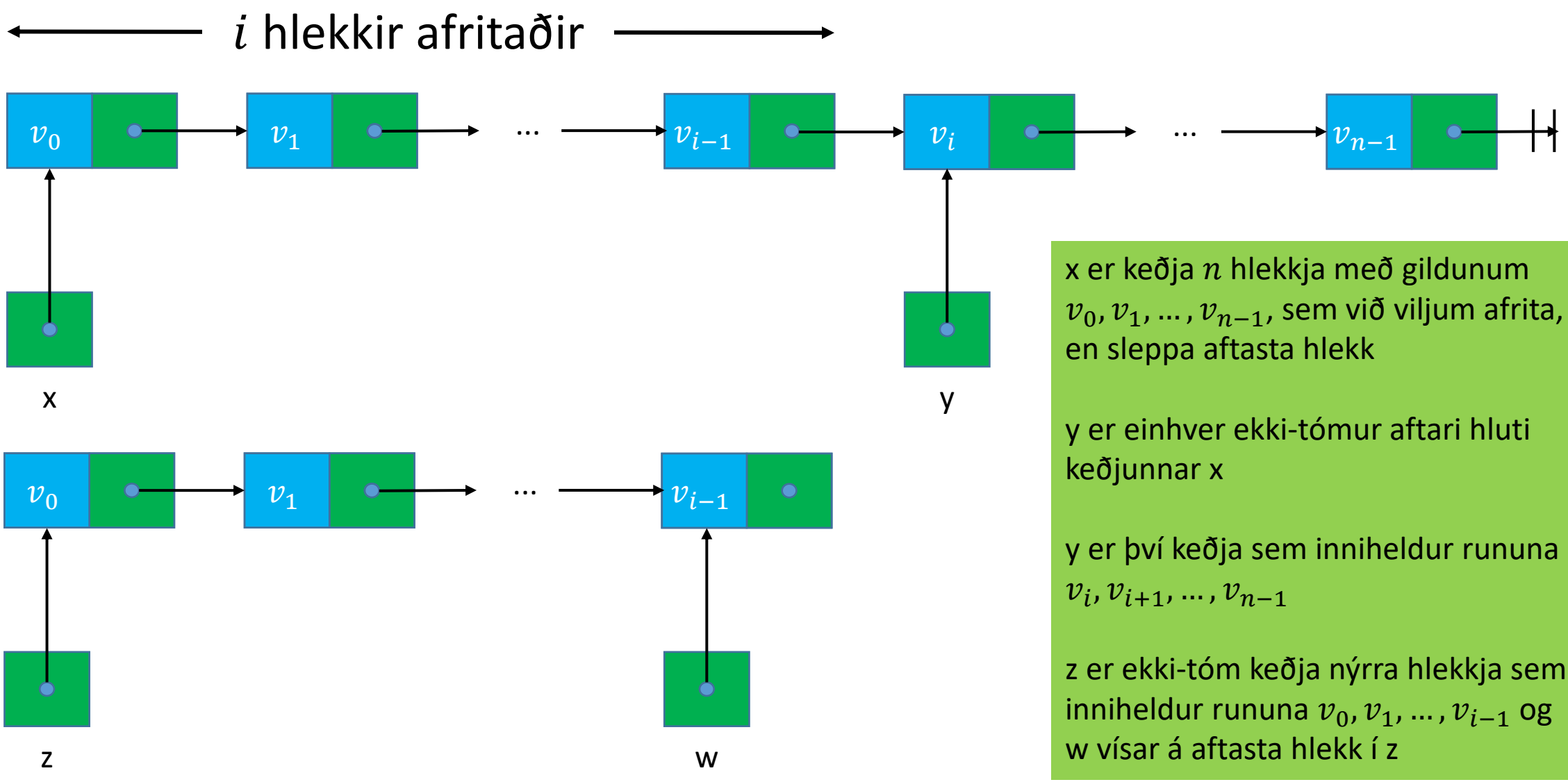


x er keðja n hlekkja með gildunum $v_0, v_1, \ldots, v_{n-1}$, sem við viljum byggja

y er einhver aftari hluti keðjunnar x og fjöldi þeirra hlekkja sem eiga að vera í x en eru ekki ennþá í y er i

y er því keðja sem inniheldur rununa $v_i, v_{i+1}, \ldots, v_{n-1}$

Fjarlæging aftasta hlekks með afritun í lykkju



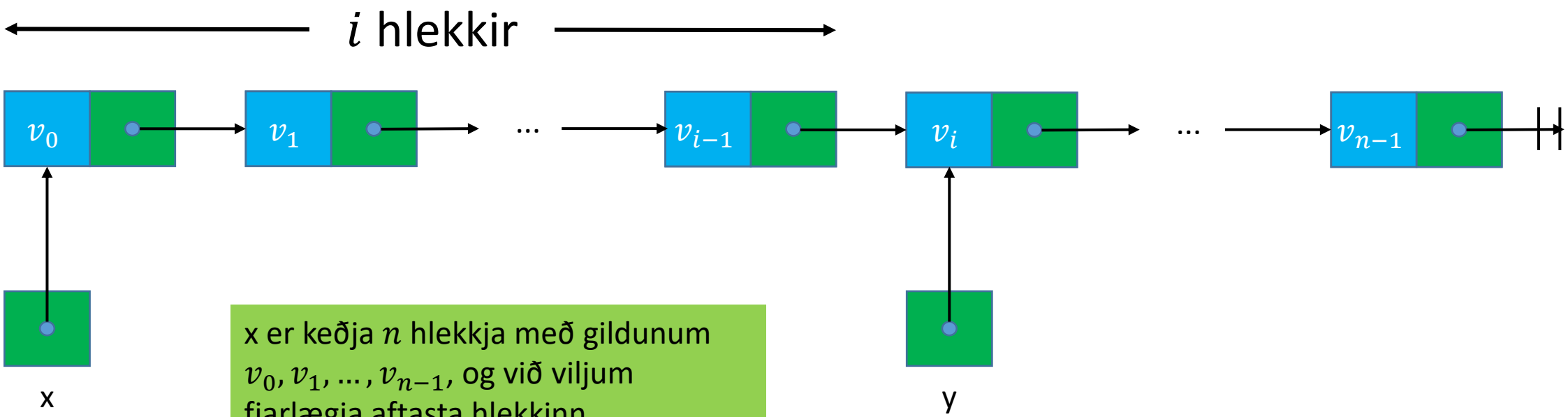
x er keðja n hlekkja með gildunum $v_0, v_1, \ldots, v_{n-1}$, sem við viljum afrita, en sleppa aftasta hlekk

y er einhver ekki-tómur aftari hluti keðjunnar x

y er því keðja sem inniheldur rununa $v_i, v_{i+1}, \ldots, v_{n-1}$

z er ekki-tóm keðja nýrra hlekkja sem inniheldur rununa $v_0, v_1, \ldots, v_{i-1}$ og w vísar á aftasta hlekk í z

Fjarlæging aftasta hlekks með uppskurði

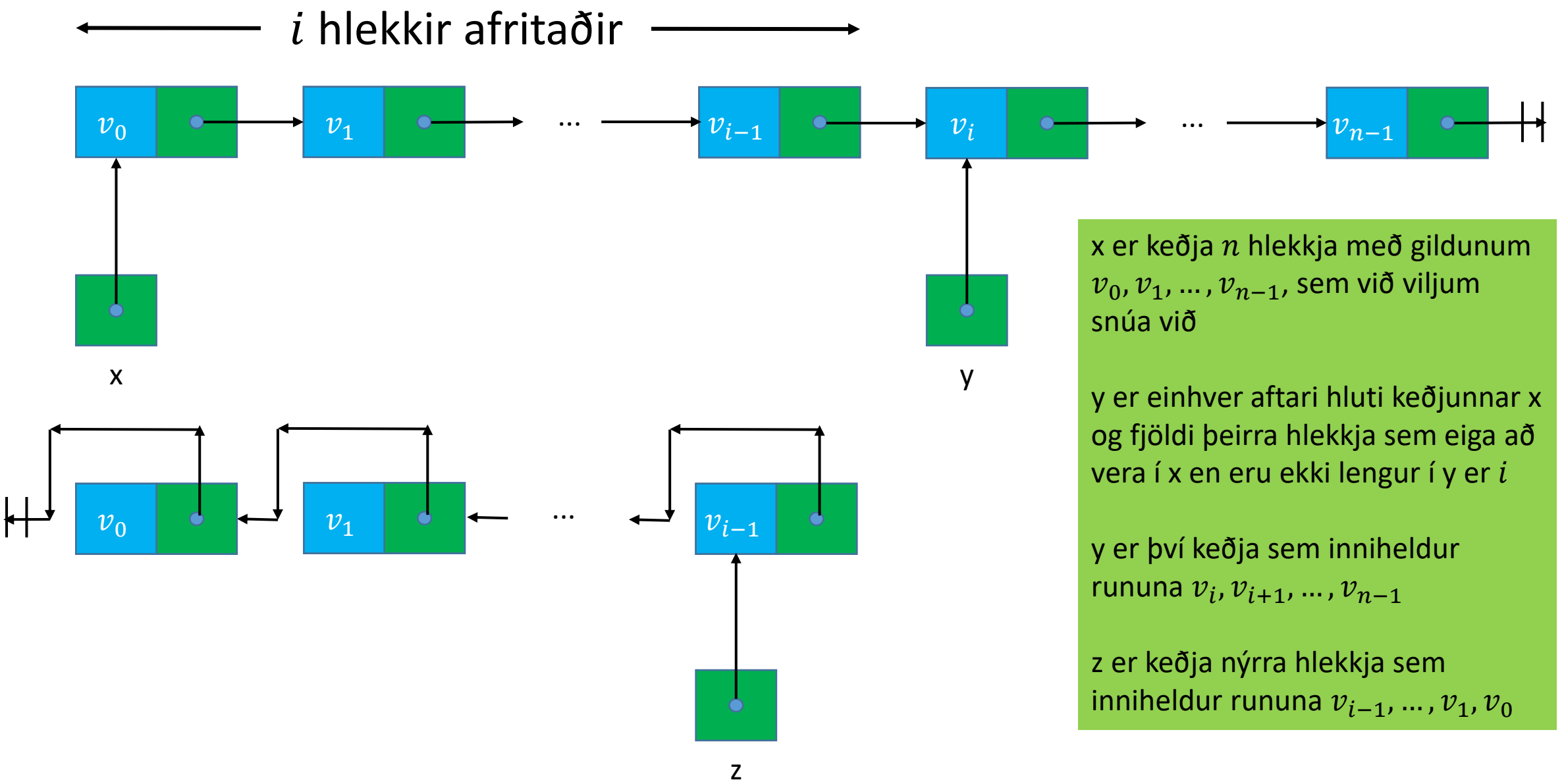


x er keðja n hlekkja með gildunum $v_0, v_1, \ldots, v_{n-1}$, og við viljum fjarlægja aftasta hlekkinn

y er einhver aftari hluti keðjunnar x með a.m.k. tveimur hlekkjum

Tilvikið þegar x inniheldur aðeins einn hlekk þarf að meðhöndla sérstaklega

Viðsnúningur keðju án eyðingar



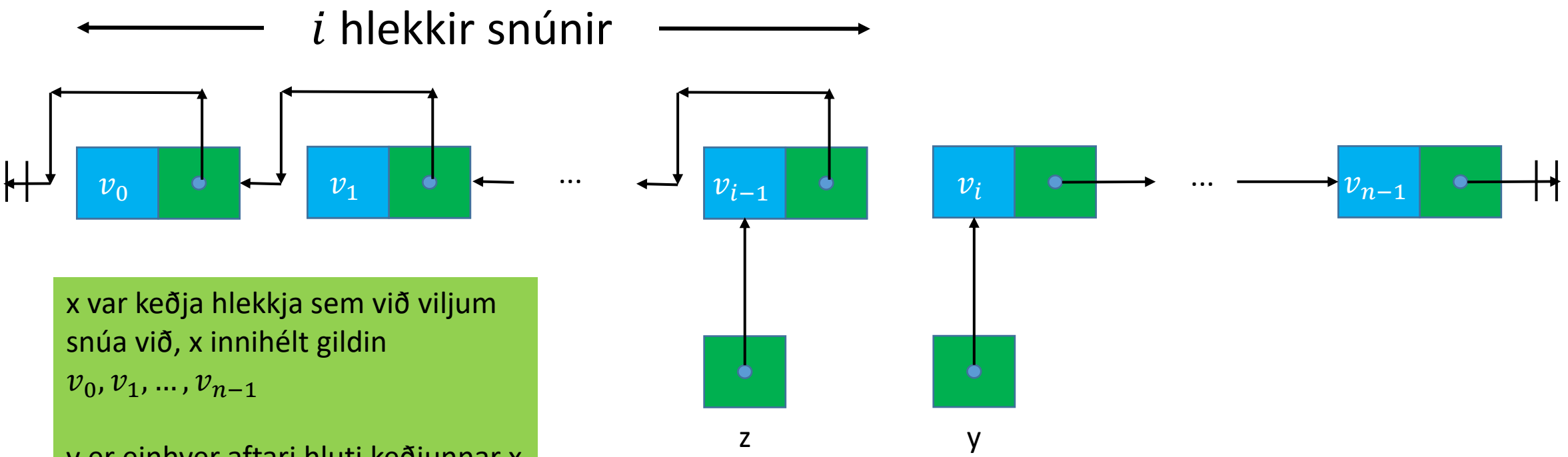
x er keðja n hlekkja með gildunum $v_0, v_1, \ldots, v_{n-1}$, sem við viljum snúa við

y er einhver aftari hluti keðjunnar x og fjöldi þeirra hlekkja sem eiga að vera í x en eru ekki lengur í y er i

y er því keðja sem inniheldur rununa $v_i, v_{i+1}, \ldots, v_{n-1}$

z er keðja nýrra hlekkja sem inniheldur rununa $v_{i-1}, \ldots, v_1, v_0$

Viðsnúningur keðju með eyðingu



x var keðja hlekkja sem við viljum snúa við, x innihélt gildin
$v_0, v_1, \ldots, v_{n-1}$

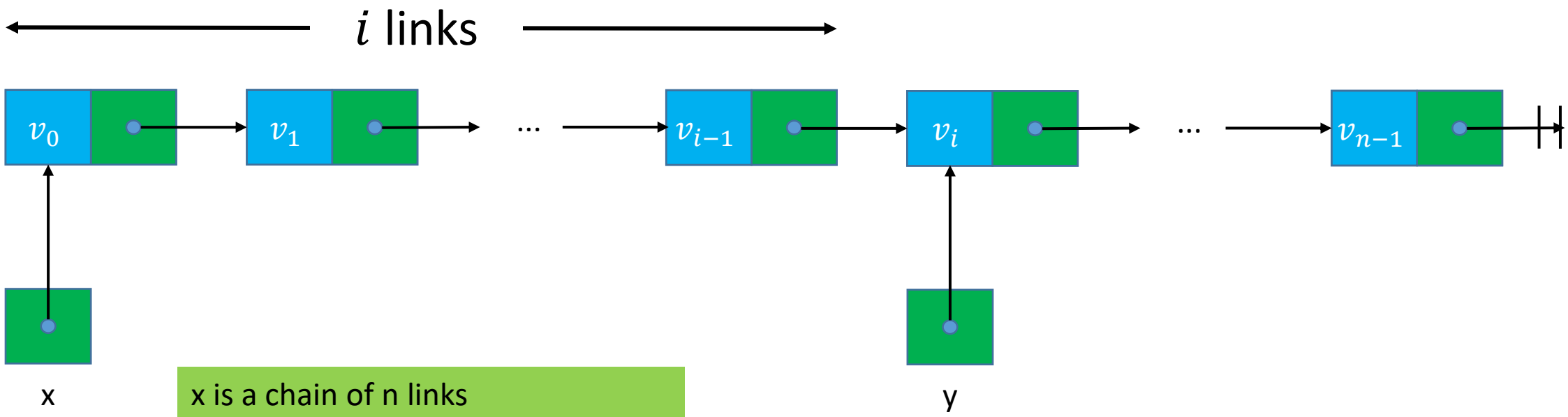
y er einhver aftari hluti keðjunnar x og inniheldur óbreytta og ósnúna hlekki með gildunum
$v_i, v_{i+1}, \ldots, v_{n-1}$

z er keðja sem er viðsnúin keðja þeirra hlekkja úr x sem ekki eru í y með gildunum $v_{i-1}, \ldots, v_1, v_0$

Singly Linked Lists and List Processing

Various Loop Invariants

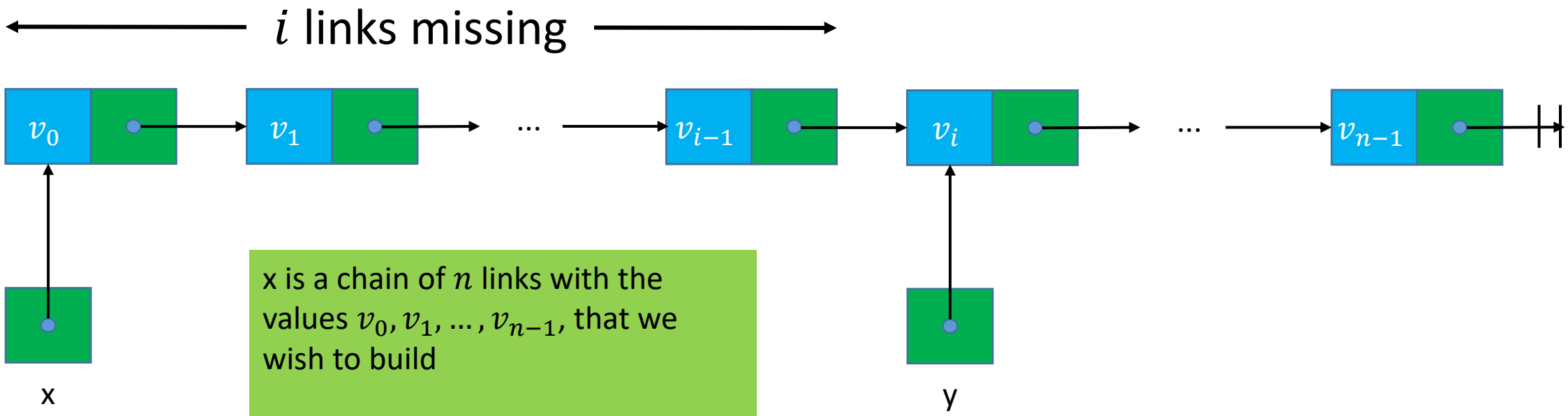
Counting Links



x is a chain of n links

y is some suffix of the chain x and the number of links that are in x but not in y is i

Constructing a Chain

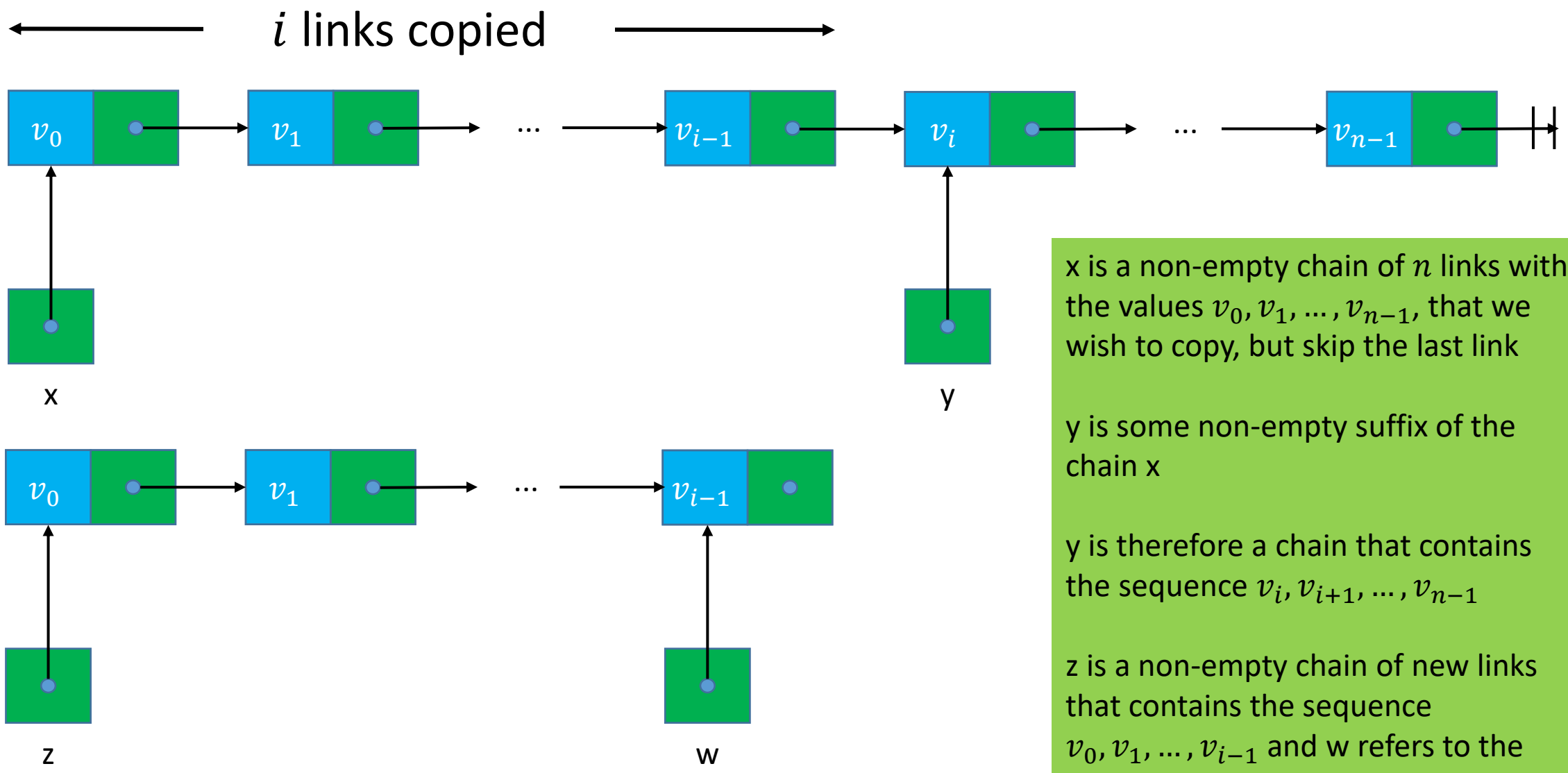


x is a chain of n links with the values $v_0, v_1, \ldots, v_{n-1}$, that we wish to build

y is some suffix of the chain x and the number of links that should be in x but are not yet in y is i

y is therefore a chain that contains the sequence $v_i, v_{i+1}, \ldots, v_{n-1}$

Removal of last link by copying in a loop



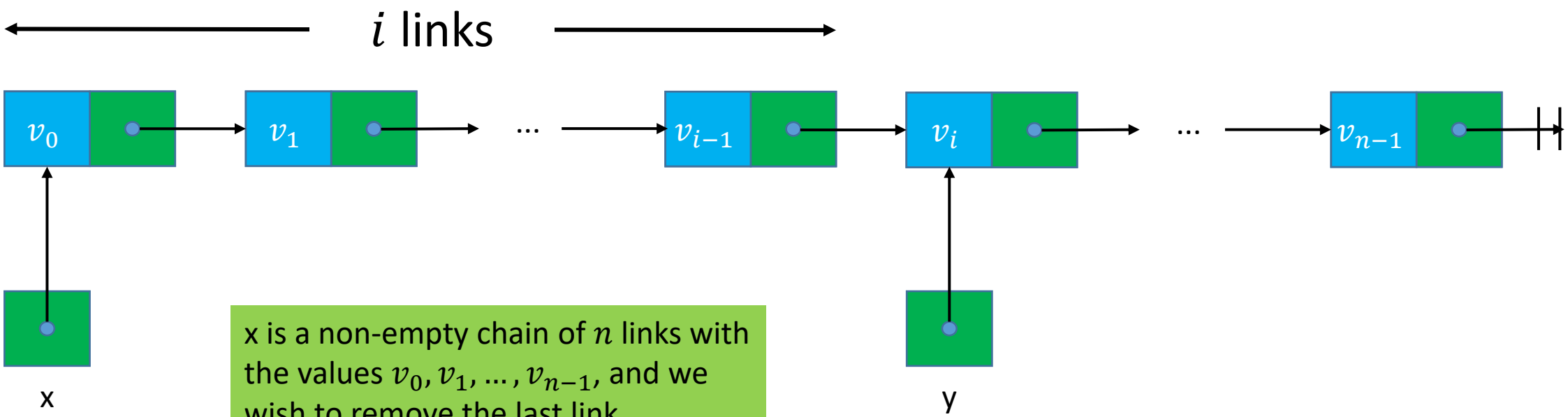
x is a non-empty chain of n links with the values $v_0, v_1, \dots, v_{n-1}$, that we wish to copy, but skip the last link

y is some non-empty suffix of the chain x

y is therefore a chain that contains the sequence $v_i, v_{i+1}, \dots, v_{n-1}$

z is a non-empty chain of new links that contains the sequence $v_0, v_1, \dots, v_{i-1}$ and w refers to the last link of z

Removing the last link by surgery

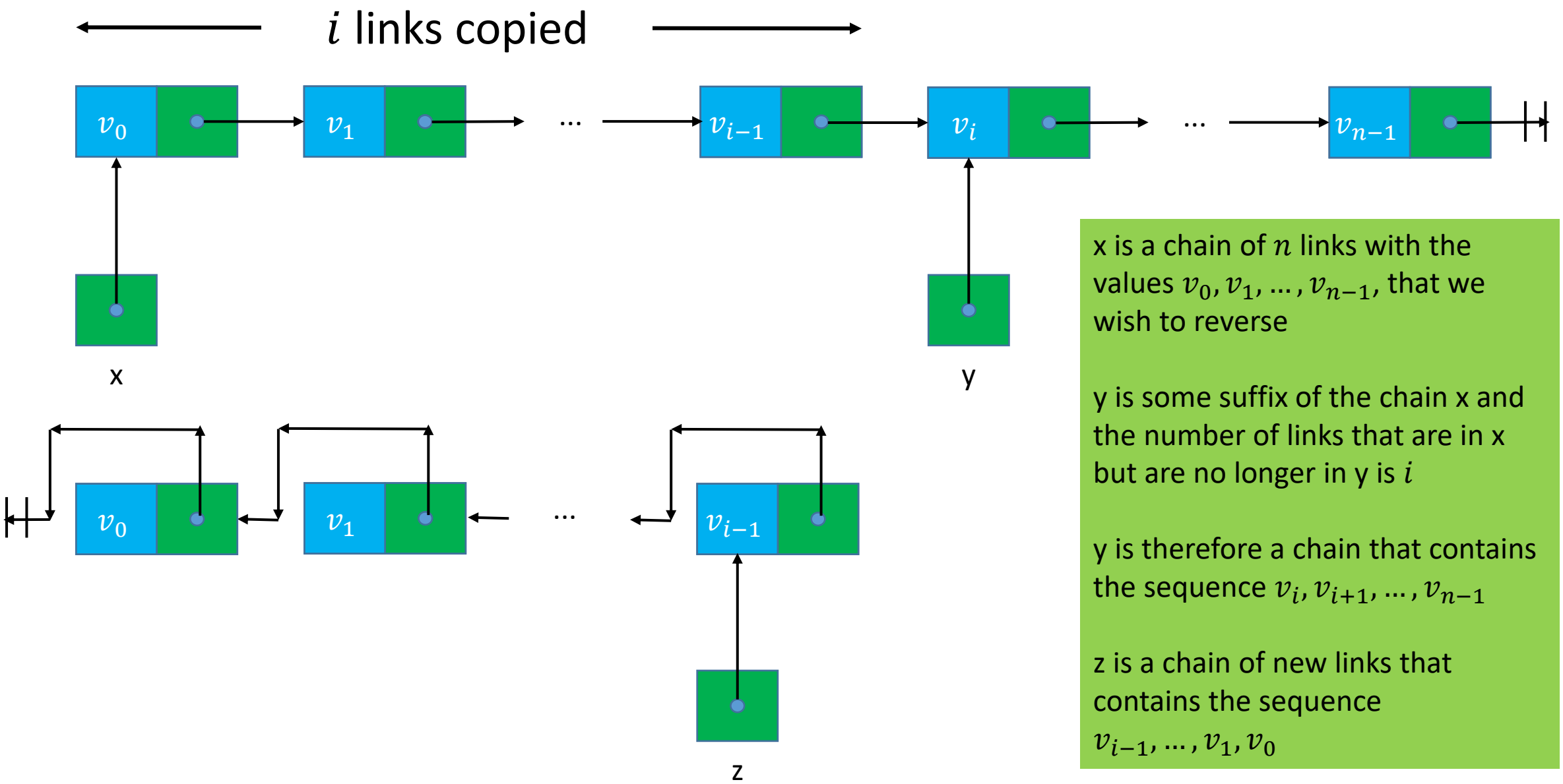


x is a non-empty chain of n links with the values $v_0, v_1, \ldots, v_{n-1}$, and we wish to remove the last link

y is some suffix of x with at least two links

The case when x contains only one link must be handled separately

Reversing a chain non-destructively



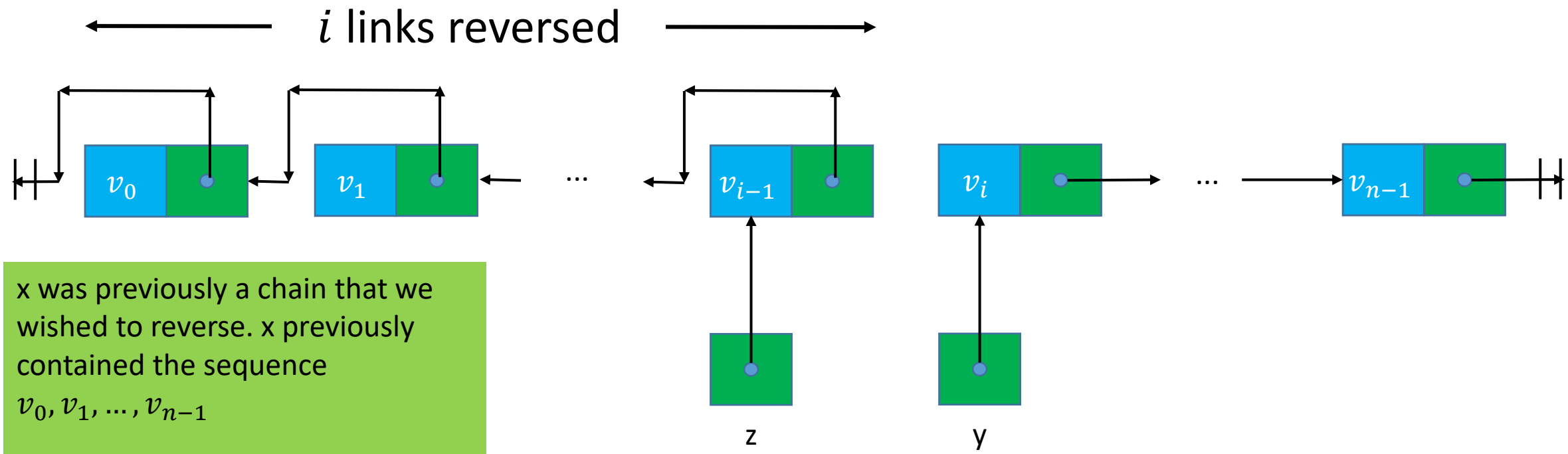
x is a chain of n links with the values $v_0, v_1, \ldots, v_{n-1}$, that we wish to reverse

y is some suffix of the chain x and the number of links that are in x but are no longer in y is i

y is therefore a chain that contains the sequence $v_i, v_{i+1}, \ldots, v_{n-1}$

z is a chain of new links that contains the sequence $v_{i-1}, \ldots, v_1, v_0$

Reversing a chain destructively



x was previously a chain that we wished to reverse. x previously contained the sequence $v_0, v_1, \ldots, v_{n-1}$

y is some suffix of the original x and contains unmodified and unreversed links with the values $v_i, v_{i+1}, \ldots, v_{n-1}$

z is a chain that is a reversed chain of those links from the original x that are not in y, with the value sequence $v_{i-1}, \ldots, v_1, v_0$