(Imperative) Program Logic Part 2

Robert Keller
February 2011

Recall the **while** rule

- In order to use the while rule in JAPE, it is necessary to supply an **invariant**, I.

$$\{I \wedge P\}\ S\ \{I\}$$

$$\{I\}\ \textbf{while(P) S}\ \{I \wedge \neg P\}$$

Inferring invariants

- There is no fully general automation for inferring invariants (as there is for the weakest pre-condition/assumption for assignment statements).
- This is one of the things that makes totally automated verification difficult.
- Finding the right invariant is still a human intellectual activity.

Using the **while** rule

- In JAPE, an assertion **implying** the invariant (by using the consequent(L) rule) is included as an assertion before the while, and also doubly serves as a post-condition for the preceding statement.

(What does this code do?)

...

$\{i=Ki \wedge j=Kj \wedge i \geq 0\}(k:=0)$

1: $\{i \geq 0 \wedge k+i \times j=Ki \times Kj\}$

while $i \neq 0$ do $k:=k+j;i:=i-1$ od

$\{k=Ki \times Kj\}$

Should imply the invariant

Invariant and negation of test should imply this.

Using the **while** rule

- Before using Jape's while rule, this setup is decomposed using Jape's **Ntuple** rule.

1: $\{i=Ki \wedge j=Kj \wedge i \geq 0\}(k:=0)$
$\{i \geq 0 \wedge k+i \times j=Ki \times Kj\}$
$\ldots$

Prove using assignment rule

$\{i \geq 0 \wedge k+i \times j=Ki \times Kj\}$
2: while $i \neq 0$ do $k:=k+j;i:=i-1$ od
$\{k=Ki \times Kj\}$

Prove using while rule

$\{i=Ki \wedge j=Kj \wedge i \geq 0\}(k:=0)$
3: $\{i \geq 0 \wedge k+i \times j=Ki \times Kj\}$
while $i \neq 0$ do $k:=k+j;i:=i-1$ od
$\{k=Ki \times Kj\}$

Ntuple 1,2

Ntuple rule used

A proof of a simple program

1:	$i=10 \wedge i>0$	assumption
2:	$i-1=10$	obviously
3:	$i=10 \wedge i>0 \rightarrow i-1=10$	$\rightarrow$ intro 1–2
4:	$\{i-1=10\}(i:=i-1)\{i=10\}$	variable-assignment
5:	$\{i=10 \wedge i>0\}(i:=i-1)\{i=10\}$	consequence(L) 3,4
6:	$i=10 \wedge i>0$	assumption
7:	$i>0$	$\wedge$ elim 6
8:	$i=10 \wedge i>0 \rightarrow i>0$	$\rightarrow$ intro 6–7
9:	integer Km	assumption
10:	$i=10 \wedge i>0 \wedge i=Km$	assumption
11:	$i-1<Km$	obviously
12:	$i=10 \wedge i>0 \wedge i=Km \rightarrow i-1<Km$	$\rightarrow$ intro 10–11
13:	$\{i-1<Km\}(i:=i-1)\{i<Km\}$	variable-assignment
14:	$\{i=10 \wedge i>0 \wedge i=Km\}(i:=i-1)\{i<Km\}$	consequence(L) 12,13
15:	$\{i=10\}\text{while } i>0 \text{ do } i:=i-1 \text{ od}\{i=10 \wedge \neg(i>0)\}$	while 5,8,9–14
16:	$i=10 \wedge \neg(i>0) \rightarrow i=0$	obviously
17:	$\{i=10\}(\text{while } i>0 \text{ do } i:=i-1 \text{ od})\{i=0\}$	consequence(R) 15,16

Subtleties about loop invariants

- Can the following be proved?

...

1: $\{y=0 \wedge i=0 \wedge n \geq 0\}(j:=1)\{y=i \times i \wedge i \leq n \wedge i \geq 0\}$

while $i<n$ do $y:=y+j; j:=j+2; i:=i+1$ od$\{y=n \times n\}$

Provided:

DISTINCT i, j, n, y

The loop invariant is not strong enough to enable induction

- This is more likely provable.

...

1: $\{y=0 \wedge i=0 \wedge n \geq 0\}(j:=1)\{y=i \times i \wedge i \leq n \wedge i \geq 0 \wedge j=2 \times i+1\}$

while $i<n$ do $y:=y+j; j:=j+2; i:=i+1$ od$\{y=n \times n\}$

Provided:

DISTINCT i, j, n, y

- What is _M for termination?

...

$$\{y=0 \wedge i=0 \wedge n \geq 0\}(j:=1)\{y=i \times i \wedge i \leq n \wedge i \geq 0 \wedge j=2 \times i+1\}$$

1 :

$$\text{while } i<n \text{ do } y:=y+j; j:=j+2; i:=i+1 \text{ od}\{y=n \times n\}$$

Provided:

DISTINCT i, j, n, y

A Completed Proof (lines 1-24 of 51)

1:	$y=0 \wedge i=0 \wedge n \geq 0$	assumption
2:	$y=0$	$\wedge$ elim 1
3:	$i=0$	$\wedge$ elim 1
4:	$n \geq 0$	$\wedge$ elim 1
5:	$y=i \times i$	obviously, from 3,2
6:	$i \leq n$	obviously, from 4,3
7:	$i \geq 0$	obviously, from 3
8:	$1=2 \times i+1$	obviously, from 3
9:	$y=i \times i \wedge i \leq n \wedge i \geq 0 \wedge 1=2 \times i+1$	$\wedge$ intro 5,6,7,8
10:	$y=0 \wedge i=0 \wedge n \geq 0 \rightarrow y=i \times i \wedge i \leq n \wedge i \geq 0 \wedge 1=2 \times i+1$	$\rightarrow$ intro 1-9
11:	$\{y=i \times i \wedge i \leq n \wedge i \geq 0 \wedge 1=2 \times i+1\}(j:=1)\{y=i \times i \wedge i \leq n \wedge i \geq 0 \wedge j=2 \times i+1\}$	variable-assignment

Initialization

12:	$\{y=0 \wedge i=0 \wedge n \geq 0\}(j:=1)\{y=i \times i \wedge i \leq n \wedge i \geq 0 \wedge j=2 \times i+1\}$	consequence(L) 10,11
13:	$y=i \times i \wedge i \leq n \wedge i \geq 0 \wedge j=2 \times i+1 \wedge i<n$	assumption
14:	$y=i \times i$	$\wedge$ elim 13
15:	$i \geq 0$	$\wedge$ elim 13
16:	$j=2 \times i+1$	$\wedge$ elim 13
17:	$i<n$	$\wedge$ elim 13
18:	$y+j=(i+1) \times (i+1)$	obviously, from 16,14
19:	$i+1 \leq n$	obviously, from 17
20:	$i+1 \geq 0$	obviously, from 15
21:	$j+2=2 \times (i+1)+1$	obviously, from 16
22:	$y+j=(i+1) \times (i+1) \wedge i+1 \leq n \wedge i+1 \geq 0 \wedge j+2=2 \times (i+1)+1$	$\wedge$ intro 18,19,20,21
23:	$y=i \times i \wedge i \leq n \wedge i \geq 0 \wedge j=2 \times i+1 \wedge i<n \rightarrow y+j=(i+1) \times (i+1) \wedge i+1 \leq n \wedge i+1 \geq 0 \wedge j+2=2 \times (i+1)+1$	$\rightarrow$ intro 13-22
24:	$\{y+j=(i+1) \times (i+1) \wedge i+1 \leq n \wedge i+1 \geq 0 \wedge j+2=2 \times (i+1)+1\}(y:=y+j)\{y=(i+1) \times (i+1) \wedge i+1 \leq n \wedge i+1 \geq 0 \wedge j+2=2 \times (i+1)+1\}$	variable-assignment

First assignment in loop body

The Completed Proof (lines 25-51 of 51)

25:	$\{y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\wedge i<n\}(y:=y+j)\{y=(i+1)\times(i+1)\wedge i+1\le n\wedge i+1\ge 0\wedge j+2=2\times(i+1)+1\}$	consequence(L) 23,24	Other assignments in loop body
26:	$\{y=(i+1)\times(i+1)\wedge i+1\le n\wedge i+1\ge 0\wedge j+2=2\times(i+1)+1\}(j:=j+2)\{y=(i+1)\times(i+1)\wedge i+1\le n\wedge i+1\ge 0\wedge j=2\times(i+1)+1\}$	variable-assignment	
27:	$\{y=(i+1)\times(i+1)\wedge i+1\le n\wedge i+1\ge 0\wedge j=2\times(i+1)+1\}(i:=i+1)\{y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\}$	variable-assignment	
28:	$\{y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\wedge i<n\}(y:=y+j;j:=j+2;i:=i+1)\{y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\}$	sequence 25,26,27	
29:	$y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\wedge i<n$	assumption	Condition on _M
30:	$i<n$	$\wedge$ elim 29	
31:	$n-i>0$	obviously, from 30	
32:	$y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\wedge i<n\rightarrow n-i>0$	$\rightarrow$ intro 29-31	
33:	integer Km	assumption	Termination of loop body
34:	$y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\wedge i<n\wedge n-i=Km$	assumption	
35:	$n-i=Km$	$\wedge$ elim 34	
36:	$n-(i+1)<Km$	obviously, from 35	
37:	$y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\wedge i<n\wedge n-i=Km\rightarrow n-(i+1)<Km$	$\rightarrow$ intro 34-36	
38:	$\{n-(i+1)<Km\}(y:=y+j)\{n-(i+1)<Km\}$	variable-assignment	
39:	$\{y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\wedge i<n\wedge n-i=Km\}(y:=y+j)\{n-(i+1)<Km\}$	consequence(L) 37,38	
40:	$\{n-(i+1)<Km\}(j:=j+2)\{n-(i+1)<Km\}$	variable-assignment	
41:	$\{n-(i+1)<Km\}(i:=i+1)\{n-i<Km\}$	variable-assignment	
42:	$\{y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\wedge i<n\wedge n-i=Km\}(y:=y+j;j:=j+2;i:=i+1)\{n-i<Km\}$	sequence 39,40,41	
43:	$\{y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\}$while $i<n$ do $y:=y+j;j:=j+2;i:=i+1$ od$\{y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\wedge\neg(i<n)\}$	while 28,32,33-42	
44:	$y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\wedge\neg(i<n)$	assumption	Exit consequence
45:	$y=i\times i$	$\wedge$ elim 44	
46:	$i\le n$	$\wedge$ elim 44	
47:	$\neg(i<n)$	$\wedge$ elim 44	
48:	$y=n\times n$	obviously, from 47,46,45	
49:	$y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\wedge\neg(i<n)\rightarrow y=n\times n$	$\rightarrow$ intro 44-48	
50:	$\{y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\}$while $i<n$ do $y:=y+j;j:=j+2;i:=i+1$ od$\{y=n\times n\}$	consequence(R) 43,49	
51:	$\{y=0\wedge i=0\wedge n\ge 0\}(j:=1)\{y=i\times i\wedge i\le n\wedge i\ge 0\wedge j=2\times i+1\}$while $i<n$ do $y:=y+j;j:=j+2;i:=i+1$ od$\{y=n\times n\}$	Ntuple 12,50	

Verifying Array Programs

- Arrays present extra challenges and interesting issues.
- A useful dichotomy:
  - Programs with read-only arrays
  - Programs with modifiable arrays

Array Mathematics

- An array can be treated as a **function**:
  - It maps indices into values.
  - e.g. a 1-dimensional array with dimension 10 maps {0, …, 9} into values of the type stored in the array.
  - a[i] is the value of this function with argument I
- Because several indices can have the same value, arrays are more susceptible to variable **aliasing**, e.g.

  i := 5; j = 6-1; a[j] = a[i]+1

Read-Only Array Example

This program sets j to the last index i such that a[i] = 0.
The array is assumed to be indexed 0..n-1.
If there is no such value, it leaves j at its initial value n.

$\{n \geq 0 \wedge \text{length}(a)=n\}$

```
i := 0;
j := n;
while i < n
do
if a[i] = 0
  then j := i
  else skip
fi
i := i+1
od
```

$\{j < n \rightarrow a[j] = 0\}$

What invariant do we need?

Read-Only Array Example

This program sets j to the last index i such that a[i] = 0.
The array is assumed to be indexed 0..n-1.
If there is no such value, it leaves j at its initial value n.

...

assumption program invariant

1: $\{n \geq 0 \wedge length(a)=n\}\boxed{(i:=0;j:=n)}\{i \leq n \wedge i \geq 0 \wedge length(a)=n \wedge (j<n \rightarrow a[j]=0)\}$

$\boxed{\text{while } i<n \text{ do if } a[i]=0 \text{ then } j:=i \text{ else skip fi}; i:=i+1 \text{ od}}\{j<n \rightarrow a[j]=0\}$

expectation

Provided:

DISTINCT a, i, j, n

Read-Only Example (lines 1-15)

1:	$n \geq 0 \wedge length(a)=n$	assumption
2:	$n \geq 0$	$\wedge$ elim 1
3:	$length(a)=n$	$\wedge$ elim 1
4:	$0 \leq n$	$A \leq B \triangleq B \geq A$ 2
5:	$0 \geq 0$	obviously
6:	$n<n$	assumption
7:	$\perp$	obviously, from 6
8:	$a[n]=0$	contra (constructive) 7
9:	$n<n \rightarrow a[n]=0$	$\rightarrow$ intro 6-8
10:	$0 \leq n \wedge 0 \geq 0 \wedge length(a)=n \wedge (n<n \rightarrow a[n]=0)$	$\wedge$ intro 4,5,3,9
11:	$n \geq 0 \wedge length(a)=n \rightarrow 0 \leq n \wedge 0 \geq 0 \wedge length(a)=n \wedge (n<n \rightarrow a[n]=0)$	$\rightarrow$ intro 1-10
12:	$\{0 \leq n \wedge 0 \geq 0 \wedge length(a)=n \wedge (n<n \rightarrow a[n]=0)\}(i:=0)\{i \leq n \wedge i \geq 0 \wedge length(a)=n \wedge (n<n \rightarrow a[n]=0)\}$	variable-assignment
13:	$\{n \geq 0 \wedge length(a)=n\}(i:=0)\{i \leq n \wedge i \geq 0 \wedge length(a)=n \wedge (n<n \rightarrow a[n]=0)\}$	consequence(L) 11,12
14:	$\{i \leq n \wedge i \geq 0 \wedge length(a)=n \wedge (n<n \rightarrow a[n]=0)\}(j:=n)\{i \leq n \wedge i \geq 0 \wedge length(a)=n \wedge (j<n \rightarrow a[j]=0)\}$	variable-assignment
15:	$\{n \geq 0 \wedge length(a)=n\}(i:=0;j:=n)\{i \leq n \wedge i \geq 0 \wedge length(a)=n \wedge (j<n \rightarrow a[j]=0)\}$	sequence 13,14

Read-Only Example (lines 16-37)

16:	$i \le n \wedge i \ge 0 \wedge \text{length}(a)=n \wedge (j<n \rightarrow a[j]=0) \wedge i<n$	assumption
17:	$i \ge 0$	$\wedge$ elim 16
18:	$\text{length}(a)=n$	$\wedge$ elim 16
19:	$j<n \rightarrow a[j]=0$	$\wedge$ elim 16
20:	$i<n$	$\wedge$ elim 16
21:	$a[i]=0$	assumption
22:	$i+1 \le n$	obviously, from 20
23:	$i+1 \ge 0$	obviously, from 17
24:	$i<n$	assumption
25:	$a[i]=0$	hyp 21
26:	$i<n \rightarrow a[i]=0$	$\rightarrow$ intro 24-25
27:	$i+1 \le n \wedge i+1 \ge 0 \wedge \text{length}(a)=n \wedge (i<n \rightarrow a[i]=0)$	$\wedge$ intro 22,23,18,26
28:	$a[i]=0 \rightarrow i+1 \le n \wedge i+1 \ge 0 \wedge \text{length}(a)=n \wedge (i<n \rightarrow a[i]=0)$	$\rightarrow$ intro 21-27
29:	$\neg(a[i]=0)$	assumption
30:	$i+1 \le n$	obviously, from 20
31:	$i+1 \ge 0$	obviously, from 17
32:	$i+1 \le n \wedge i+1 \ge 0 \wedge \text{length}(a)=n \wedge (j<n \rightarrow a[j]=0)$	$\wedge$ intro 30,31,18,19
33:	$\neg(a[i]=0) \rightarrow i+1 \le n \wedge i+1 \ge 0 \wedge \text{length}(a)=n \wedge (j<n \rightarrow a[j]=0)$	$\rightarrow$ intro 29-32
34:	$0 \le i$	obviously, from 17
35:	$i<\text{length}(a)$	obviously, from 20,18
36:	$(a[i]=0 \rightarrow i+1 \le n \wedge i+1 \ge 0 \wedge \text{length}(a)=n \wedge (i<n \rightarrow a[i]=0)) \wedge (\neg(a[i]=0) \rightarrow i+1 \le n \wedge i+1 \ge 0 \wedge \text{length}(a)=n \wedge (j<n \rightarrow a[j]=0)) \wedge 0 \le i \wedge i<\text{length}(a)$	$\wedge$ intro 28,33,34,35
37:	$i \le n \wedge i \ge 0 \wedge \text{length}(a)=n \wedge (j<n \rightarrow a[j]=0) \wedge i<n$ $\rightarrow (a[i]=0 \rightarrow i+1 \le n \wedge i+1 \ge 0 \wedge \text{length}(a)=n \wedge (i<n \rightarrow a[i]=0)) \wedge (\neg(a[i]=0) \rightarrow i+1 \le n \wedge i+1 \ge 0 \wedge \text{length}(a)=n \wedge (j<n \rightarrow a[j]=0)) \wedge 0 \le i \wedge i<\text{length}(a)$	$\rightarrow$ intro 16-36

Read-Only Example (lines 38-47)

38:	$\{i+1\leq n\wedge i+1\geq 0\wedge \mathrm{length}(a)=n\wedge(i<n\rightarrow a[i]=0)\}(j:=i)\{i+1\leq n\wedge i+1\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0)\}$	variable-assignment
39:	$\{i+1\leq n\wedge i+1\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0)\}\mathrm{skip}\{i+1\leq n\wedge i+1\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0)\}$	skip
40:	$\{(a[i]=0\rightarrow i+1\leq n\wedge i+1\geq 0\wedge \mathrm{length}(a)=n\wedge(i<n\rightarrow a[i]=0))\wedge(\neg(a[i]=0)\rightarrow i+1\leq n\wedge i+1\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0))\wedge 0\leq i\wedge i<\mathrm{length}(a)\}$ $\text{if } a[i]=0 \text{ then } j:=i \text{ else skip fi}\{i+1\leq n\wedge i+1\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0)\}$	choice 38,39
41:	$\{i\leq n\wedge i\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0)\wedge i<n\}\text{if } a[i]=0 \text{ then } j:=i \text{ else skip fi}\{i+1\leq n\wedge i+1\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0)\}$	consequence(L) 37,40
42:	$\{i+1\leq n\wedge i+1\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0)\}(i:=i+1)\{i\leq n\wedge i\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0)\}$	variable-assignment
43:	$\{i\leq n\wedge i\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0)\wedge i<n\}(\text{if } a[i]=0 \text{ then } j:=i \text{ else skip fi}; i:=i+1)\{i\leq n\wedge i\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0)\}$	sequence 41,42
44:	$i\leq n\wedge i\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0)\wedge i<n$	assumption
45:	$i<n$	$\wedge$ elim 44
46:	$n-i>0$	obviously, from 45
47:	$i\leq n\wedge i\geq 0\wedge \mathrm{length}(a)=n\wedge(j<n\rightarrow a[j]=0)\wedge i<n\rightarrow n-i>0$	$\rightarrow$ intro 44-46

Read-Only Example (lines 48-74)

48:	integer Km	assumption
49:	$i\leq n\land i\geq 0\land length(a)=n\land(j<n\to a[j]=0)\land i<n\land n-i=Km$	assumption
50:	$i\geq 0$	$\land$ elim 49
51:	$i<n$	$\land$ elim 49
52:	$n-i=Km$	$\land$ elim 49
53:	$a[i]=0$	assumption
54:	$n-(i+1)<Km$	obviously, from 52
55:	$a[i]=0\to n-(i+1)<Km$	$\to$ intro 53-54
56:	$\neg(a[i]=0)$	assumption
57:	$n-(i+1)<Km$	obviously, from 52
58:	$\neg(a[i]=0)\to n-(i+1)<Km$	$\to$ intro 56-57
59:	$0\leq i$	$A\leq B\triangleq B\geq A$ 50
60:	$i<length(a)$	obviously, from 51
61:	$(a[i]=0\to n-(i+1)<Km)\land(\neg(a[i]=0)\to n-(i+1)<Km)\land 0\leq i\land i<length(a)$	$\land$ intro 55,58,59,60
62:	$i\leq n\land i\geq 0\land length(a)=n\land(j<n\to a[j]=0)\land i<n\land n-i=Km\to(a[i]=0\to n-(i+1)<Km)\land(\neg(a[i]=0)\to n-(i+1)<Km)\land 0\leq i\land i<length(a)$	$\to$ intro 49-61
63:	$\{n-(i+1)<Km\}(j:=i)\{n-(i+1)<Km\}$	variable-assignment
64:	$\{n-(i+1)<Km\}\text{skip}\{n-(i+1)<Km\}$	skip
65:	$\{(a[i]=0\to n-(i+1)<Km)\land(\neg(a[i]=0)\to n-(i+1)<Km)\land 0\leq i\land i<length(a)\}\text{if } a[i]=0 \text{ then } j:=i \text{ else skip fi}\{n-(i+1)<Km\}$	choice 63,64
66:	$\{i\leq n\land i\geq 0\land length(a)=n\land(j<n\to a[j]=0)\land i<n\land n-i=Km\}\text{if } a[i]=0 \text{ then } j:=i \text{ else skip fi}\{n-(i+1)<Km\}$	consequence(L) 62,65
67:	$\{n-(i+1)<Km\}(i:=i+1)\{n-i<Km\}$	variable-assignment
68:	$\{i\leq n\land i\geq 0\land length(a)=n\land(j<n\to a[j]=0)\land i<n\land n-i=Km\}(\text{if } a[i]=0 \text{ then } j:=i \text{ else skip fi};i:=i+1)\{n-i<Km\}$	sequence 66,67
69:	$\{i\leq n\land i\geq 0\land length(a)=n\land(j<n\to a[j]=0)\}\text{while } i<n \text{ do if } a[i]=0 \text{ then } j:=i \text{ else skip fi};i:=i+1 \text{ od}\{i\leq n\land i\geq 0\land length(a)=n\land(j<n\to a[j]=0)\land\neg(i<n)\}$	while 43,47,48-68
70:	$i\leq n\land i\geq 0\land length(a)=n\land(j<n\to a[j]=0)\land\neg(i<n)$	assumption
71:	$j<n\to a[j]=0$	$\land$ elim 70
72:	$i\leq n\land i\geq 0\land length(a)=n\land(j<n\to a[j]=0)\land\neg(i<n)\to j<n\to a[j]=0$	$\to$ intro 70-71
73:	$\{i\leq n\land i\geq 0\land length(a)=n\land(j<n\to a[j]=0)\}\text{while } i<n \text{ do if } a[i]=0 \text{ then } j:=i \text{ else skip fi};i:=i+1 \text{ od}\{j<n\to a[j]=0\}$	consequence(R) 69,72
74:	$\{n\geq 0\land length(a)=n\}(i:=0;j:=n)\{i\leq n\land i\geq 0\land length(a)=n\land(j<n\to a[j]=0)\}\text{while } i<n \text{ do if } a[i]=0 \text{ then } j:=i \text{ else skip fi};i:=i+1 \text{ od}\{j<n\to a[j]=0\}$	Ntuple 15,73

Quantifiers

- Quantifiers are handy representing information about arrays, e.g.

- $\forall i\ ((0 < i) \wedge (i < n)) \rightarrow a[i-1] \leq a[i]$

- $\exists i\ ((0 \leq i) \wedge (i < n) \wedge a[i] = 0)$

Quantifier Example

program

$\ldots$ assumption program

1: $\{\exists x.(0\leq x\wedge x<\text{length}(a)\wedge a[x]=0)\}\boxed{(i:=0)}$

$\{0\leq i\wedge i<\text{length}(a)\wedge\exists x.(i\leq x\wedge x<\text{length}(a)\wedge a[x]=0)\}\boxed{\text{while } a[i]\neq 0 \text{ do } i:=i+1 \text{ od}}\{a[i]=0\}$

invariant expectation

This program assumes there is an array element having value 0.
It returns an index of such a value.

This is from the JAPE "factory samples".
Will the invariant do the job?

Subtleties with Array Programs

$\{\exists x.(0 \leq x \wedge x < length(a) \wedge a[x]=0)\}(i:=0)$

$\{0 \leq i \wedge i < length(a) \wedge \boxed{\exists x.(i \leq x \wedge x < length(a) \wedge a[x]=0)}\}$while $a[i] \neq 0$ do $i:=i+1$ od$\{a[i]=0\}$

- Look at part of the invariant here.
- Note that the lower bound on x is a function of the index i.
- This is important, because it says that the element such that **a[x] = 0 is yet to be found**.
- We need this invariant to prove termination.
- The loop test will stop when a[i] = 0.

- The expansion order is tricky.

Quantifier Example Proved

- Things go pretty routinely, until this …

3: $0 \le i \wedge i < length(a) \wedge \exists x.(i \le x \wedge x < length(a) \wedge a[x] = 0) \wedge a[i] \ne 0$

4: $0 \le i$

5: $i < length(a)$

6: $\exists x.(i \le x \wedge x < length(a) \wedge a[x] = 0)$

7: $a[i] \ne 0$

...

8: $0 \le i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \le x \wedge x < length(a) \wedge a[x] = 0)$

How do we get i+1<length(a)?

∃-elimination to the rescue

3: $0 \le i \wedge i < length(a) \wedge \exists x.(i \le x \wedge x < length(a) \wedge a[x] = 0) \wedge a[i] \ne 0$

4: $0 \le i$

5: $i < length(a)$

6: $\boxed{\exists x.(i \le x \wedge x < length(a) \wedge a[x] = 0)}$

7: $a[i] \ne 0$

8: integer $i1$, $i \le i1 \wedge i1 < length(a) \wedge a[i1] = 0$

...

9: $0 \le i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \le x \wedge x < length(a) \wedge a[x] = 0)$

10: $0 \le i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \le x \wedge x < length(a) \wedge a[x] = 0)$

Now case analysis

3: $0 \le i \wedge i < length(a) \wedge \exists x.(i \le x \wedge x < length(a) \wedge a[x]=0) \wedge a[i] \ne 0$

4: $0 \le i$

5: $i < length(a)$

6: $\exists x.(i \le x \wedge x < length(a) \wedge a[x]=0)$

7: $a[i] \ne 0$

8: integer i1, $i \le i1 \wedge i1 < length(a) \wedge a[i1]=0$

9: $i \le i1$

10: $i1 < length(a)$

11: $a[i1]=0$

...

12: $0 \le i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \le x \wedge x < length(a) \wedge a[x]=0)$

13: $0 \le i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \le x \wedge x < length(a) \wedge a[x]=0)$

Setting up for case analysis

3: $0 \le i \wedge i < \mathrm{length}(a) \wedge \exists x.(i \le x \wedge x < \mathrm{length}(a) \wedge a[x] = 0) \wedge a[i] \neq 0$

4: $0 \le i$

5: $i < \mathrm{length}(a)$

6: $\exists x.(i \le x \wedge x < \mathrm{length}(a) \wedge a[x] = 0)$

7: $a[i] \neq 0$

8: integer $i1$, $i \le i1 \wedge i1 < \mathrm{length}(a) \wedge a[i1] = 0$

9: $i \le i1$

10: $i1 < \mathrm{length}(a)$

11: $a[i1] = 0$

12: $\boxed{i < i1 \vee i = i1}$ from 9

...

13: $0 \le i+1 \wedge i+1 < \mathrm{length}(a) \wedge \exists x.(i+1 \le x \wedge x < \mathrm{length}(a) \wedge a[x] = 0)$

14: $0 \le i+1 \wedge i+1 < \mathrm{length}(a) \wedge \exists x.(i+1 \le x \wedge x < \mathrm{length}(a) \wedge a[x] = 0)$

Strategy: ∨-Elimination

3: $0 \le i \land i < \mathrm{length}(a) \land \exists x.(i \le x \land x < \mathrm{length}(a) \land a[x]=0) \land a[i] \ne 0$

4: $0 \le i$

5: $i < \mathrm{length}(a)$

6: $\exists x.(i \le x \land x < \mathrm{length}(a) \land a[x]=0)$

7: $a[i] \ne 0$

8: integer i1, $i \le i1 \land i1 < \mathrm{length}(a) \land a[i1]=0$

9: $i \le i1$

10: $i1 < \mathrm{length}(a)$

11: $a[i1]=0$

12: $i < i1 \lor i=i1$

13: $i < i1$

...

14: $0 \le i+1 \land i+1 < \mathrm{length}(a) \land \exists x.(i+1 \le x \land x < \mathrm{length}(a) \land a[x]=0)$

15: $i=i1$

...

16: $0 \le i+1 \land i+1 < \mathrm{length}(a) \land \exists x.(i+1 \le x \land x < \mathrm{length}(a) \land a[x]=0)$

17: $0 \le i+1 \land i+1 < \mathrm{length}(a) \land \exists x.(i+1 \le x \land x < \mathrm{length}(a) \land a[x]=0)$

18: $0 \le i+1 \land i+1 < \mathrm{length}(a) \land \exists x.(i+1 \le x \land x < \mathrm{length}(a) \land a[x]=0)$

Upper branch: ∃-introduction

12: $i < i1 \lor i = i1$

13: $i < i1$

. . .

14: $0 \leq i+1$

. . .

15: $i+1 < \text{length}(a)$

. . .

16: $\boxed{\exists x.(i+1 \leq x \land x < \text{length}(a) \land a[x] = 0)}$

17: $0 \leq i+1 \land i+1 < \text{length}(a) \land \exists x.(i+1 \leq x \land x < \text{length}(a) \land a[x] = 0)$

Upper branch closure

3: $0 \le i \wedge i < \mathrm{length}(a) \wedge \exists x.(i \le x \wedge x < \mathrm{length}(a) \wedge a[x]=0) \wedge a[i] \ne 0$

4: $0 \le i$

5: $i < \mathrm{length}(a)$

6: $\exists x.(i \le x \wedge x < \mathrm{length}(a) \wedge a[x]=0)$

7: $a[i] \ne 0$

8: integer $i1$, $i \le i1 \wedge i1 < \mathrm{length}(a) \wedge a[i1]=0$

9: $i \le i1$

10: $i1 < \mathrm{length}(a)$

11: $a[i1]=0$

12: $i < i1 \vee i = i1$

13: $i < i1$

14: $0 \le i+1$

15: $i+1 < \mathrm{length}(a)$

16: $i+1 \le i1$

17: $i+1 \le i1 \wedge i1 < \mathrm{length}(a) \wedge a[i1]=0$

18: $\exists x.(i+1 \le x \wedge x < \mathrm{length}(a) \wedge a[x]=0)$

19: $0 \le i+1 \wedge i+1 < \mathrm{length}(a) \wedge \exists x.(i+1 \le x \wedge x < \mathrm{length}(a) \wedge a[x]=0)$

Lower branch

3: $0 \le i \wedge i < length(a) \wedge \exists x.(i \le x \wedge x < length(a) \wedge a[x]=0) \wedge a[i] \ne 0$

4: $0 \le i$

5: $i < length(a)$

6: $\exists x.(i \le x \wedge x < length(a) \wedge a[x]=0)$

7: $a[i] \ne 0$

8: integer i1, $i \le i1 \wedge i1 < length(a) \wedge a[i1]=0$

9: $i \le i1$

10: $i1 < length(a)$

11: $a[i1]=0$

12: $i < i1 \vee i = i1$

13: $i < i1$

14: $0 \le i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \le x \wedge x < length(a) \wedge a[x]=0)$

15: $i = i1$

...

16: $0 \le i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \le x \wedge x < length(a) \wedge a[x]=0)$

Collapsed detail

Closure of lower branch

5:	$a[i] \neq 0$	$\wedge$ elim 3
6:	integer i1, $i \leq i1 \wedge i1 < length(a) \wedge a[i1]=0$	assumptions
7:	$i \leq i1$	$\wedge$ elim 6.2
8:	$a[i1]=0$	$\wedge$ elim 6.2
9:	$i<i1 \vee i=i1$	$A \leq B \triangleq A<B \vee A=B$ 7
10:	$i<i1$	assumption
11:	$0 \leq i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \leq x \wedge x < length(a) \wedge a[x]=0)$	{cut}
12:	$i=i1$	assumption
13:	$a[i]=0$	equality-substitution 12,8
14:	$\neg(a[i]=0)$	$A \neq B \triangleq \neg(A=B)$ 5
15:	$\perp$	$\neg$ elim 13,14
16:	$0 \leq i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \leq x \wedge x < length(a) \wedge a[x]=0)$	contra (constructive) 15

14: $0 \leq i \wedge i < length(a) \wedge \exists x.(i \leq x \wedge x < length(a) \wedge a[x]=0) \wedge a[i] \neq 0$

15: $0 \leq i$

16: $\exists x.(i \leq x \wedge x < length(a) \wedge a[x]=0)$

17: $a[i] \neq 0$

18: integer i1, $i \leq i1 \wedge i1 < length(a) \wedge a[i1]=0$

19: $i \leq i1$

20: $i1 < length(a)$

21: $a[i1]=0$

22: $i < i1 \vee i=i1$

23: $i < i1$

24: $0 \leq i+1$

25: $i+1 < length(a)$

26: $i+1 \leq i1$

27: $i+1 \leq i1 \wedge i1 < length(a) \wedge a[i1]=0$

28: $\exists x.(i+1 \leq x \wedge x < length(a) \wedge a[x]=0)$

29: $0 \leq i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \leq x \wedge x < length(a) \wedge a[x]=0)$

30: $i=i1$

31: $a[i]=0$

32: $\neg(a[i]=0)$

33: $\perp$

34: $0 \leq i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \leq x \wedge x < length(a) \wedge a[x]=0)$

35: $0 \leq i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \leq x \wedge x < length(a) \wedge a[x]=0)$

36: $0 \leq i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \leq x \wedge x < length(a) \wedge a[x]=0)$

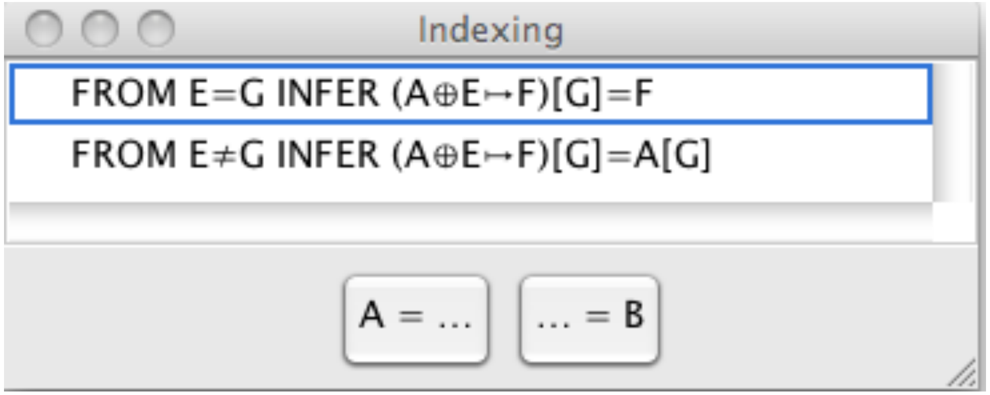
Modifiable Arrays

- Arrays are like functions
- Assigning to an array element is like creating a new function.
- The new function differs from the old in that one element may be different from before.

- Jape Notation: $a \oplus i \rightarrow v$ is the array that is like a except that the value of a[i] is v.
- So $(a \oplus i \rightarrow v)[i] = v$, and
  $(a \oplus i \rightarrow v)[j] = a[j]$ if $j \neq i$.

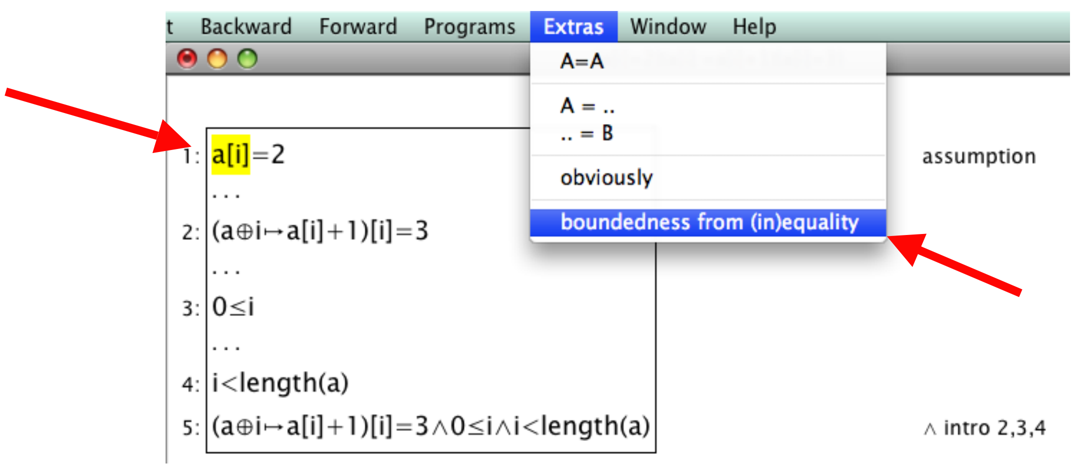
Jape’s Indexing rules

- $(a \oplus i \rightarrow v)[i] = v$, and
  $(a \oplus i \rightarrow v)[j] = a[j]$ if $j \neq i$.
- Two rules below capture the two cases preceding.
  - The first rule simplifies an array modification expression when the index of the new array is provably **the same** as the index to which assignment was done.
  - The second rule simplifies in the case of a **different** index.
- The buttons indicate the direction of substitution.

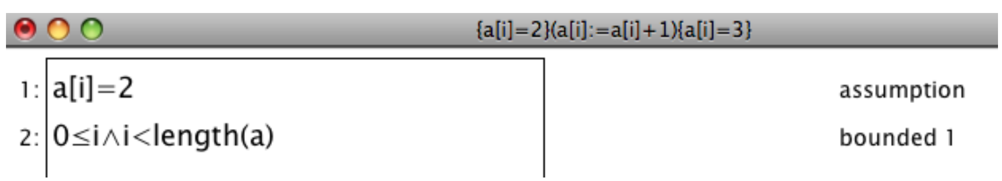


Array Bounds Guarantees

- If an array index value is used in an assumption, the same index value can be used later on without requiring a bounds check.
- Sub-formula select a hypothesis using the desired index.



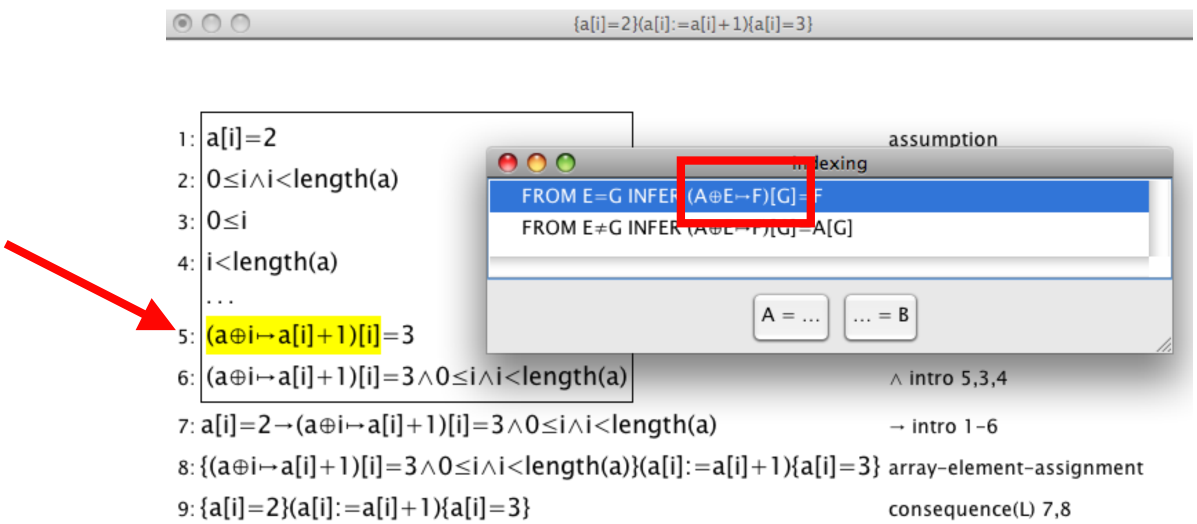
Result:



Using the Array Rule

- Make sure the entire array sub-expression is sub-formula selected.

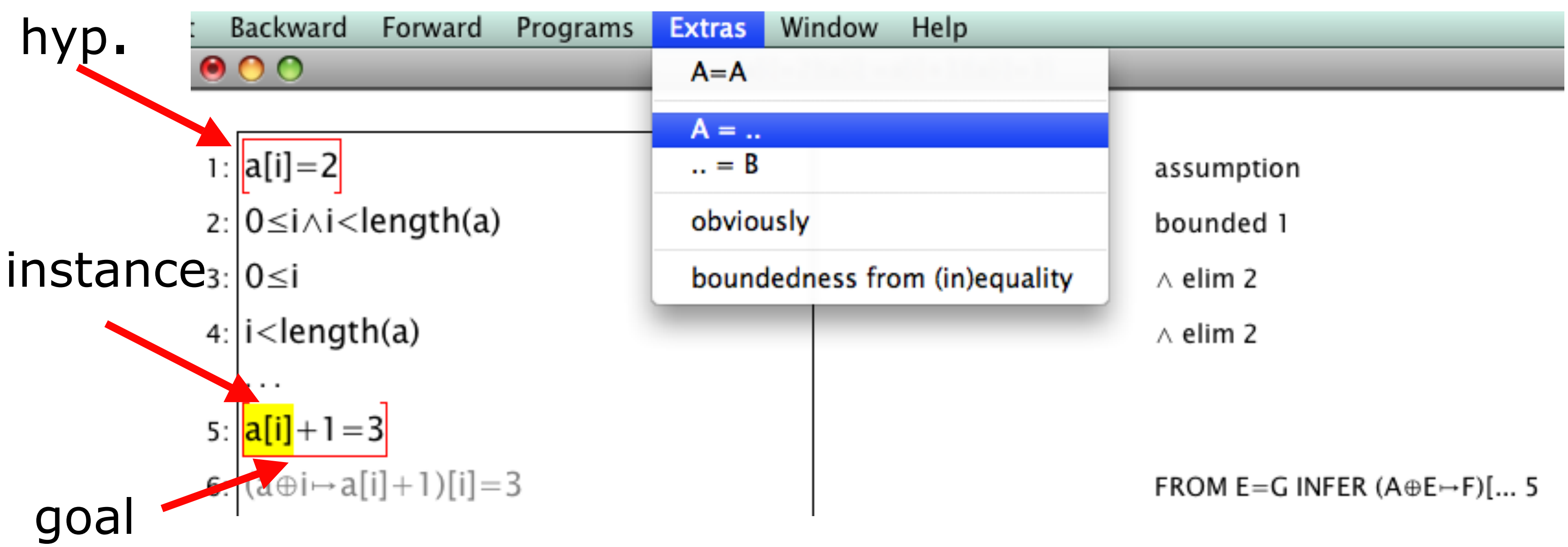
- It should match the form in the rule in the menu:



Here we identify:
A with a
E with i
F with a[i]+1
[G] with [i]
(so E = G).

Using the Equality Rule

- Two selections and a sub-formula selection are needed:
  - Selection an equality hypothesis and a goal.
  - Sub-formula select an instance of the LHS of the equality.



Result of the Equality Rule

	$\ldots$	
5:	$2+1=3$	
6:	$a[i]+1=3$	equality-substitution 1,5
7:	$(a \oplus i \mapsto a[i]+1)[i]=3$	FROM E=G INFER (A⊕E↦F)[G... 6
8:	$(a \oplus i \mapsto a[i]+1)[i]=3 \wedge 0 \leq i \wedge i < \text{length}(a)$	∧ intro 7,3,4
9:	$a[i]=2 \rightarrow (a \oplus i \mapsto a[i]+1)[i]=3 \wedge 0 \leq i \wedge i < \text{length}(a)$	→ intro 1-8
10:	$\{(a \oplus i \mapsto a[i]+1)[i]=3 \wedge 0 \leq i \wedge i < \text{length}(a)\}(a[i]:=a[i]+1)\{a[i]=3\}$	array-element-assignment
11:	$\{a[i]=2\}(a[i]:=a[i]+1)\{a[i]=3\}$	consequence(L) 9,10

The top simple equation
can be justified by “obviously”.

Summary: Jape proof with array modification

1:	$a[i]=2$	assumption
2:	$0\le i \wedge i<\text{length}(a)$	bounded 1
3:	$0\le i$	$\wedge$ elim 2
4:	$i<\text{length}(a)$	$\wedge$ elim 2
5:	$2+1=3$	obviously
6:	$a[i]+1=3$	equality-substitution 1,5
7:	$(a\oplus i\mapsto a[i]+1)[i]=3$	FROM E=G INFER $(A\oplus E\mapsto F)[G]=F$ 6
8:	$(a\oplus i\mapsto a[i]+1)[i]=3\wedge 0\le i\wedge i<\text{length}(a)$	$\wedge$ intro 7,3,4
9:	$a[i]=2\rightarrow(a\oplus i\mapsto a[i]+1)[i]=3\wedge 0\le i\wedge i<\text{length}(a)$	$\rightarrow$ intro 1-8
10:	$\{(a\oplus i\mapsto a[i]+1)[i]=3\wedge 0\le i\wedge i<\text{length}(a)\}(\boxed{a[i]:=a[i]+1})\{a[i]=3\}$	array-element-assignment
11:	$\{a[i]=2\}(a[i]:=a[i]+1)\{a[i]=3\}$	consequence(L) 9,10

infers in-bounds from usage in 1.

A=… rule

index rule

statement

How to get these rules to work in the GUI (It isn't so obvious.)

- Looking at the 2nd provided array program example, we use sequence, then array-assignment twice (from the bottom up) to get to this point:

$\{a[i]=0\}(a[i]:=a[i]+1;a[i]:=a[i]+1)\{a[i]=2\}$ [1]

...

1:	$a[i]=0 \rightarrow (a \oplus i \mapsto a[i]+1 \oplus i \mapsto (a \oplus i \mapsto a[i]+1)[i]+1)[i]=2 \land 0 \leq i \land i < length(a)$	
2:	$\{(a \oplus i \mapsto a[i]+1 \oplus i \mapsto (a \oplus i \mapsto a[i]+1)[i]+1)[i]=2 \land 0 \leq i \land i < length(a)\}$ $(a[i]:=a[i]+1)\{(a \oplus i \mapsto a[i]+1)[i]=2 \land 0 \leq i \land i < length(a)\}$	array-element-assignment
3:	$\{a[i]=0\}(a[i]:=a[i]+1)\{(a \oplus i \mapsto a[i]+1)[i]=2 \land 0 \leq i \land i < length(a)\}$	consequence(L) 1,2
4:	$\{(a \oplus i \mapsto a[i]+1)[i]=2 \land 0 \leq i \land i < length(a)\}(a[i]:=a[i]+1)\{a[i]=2\}$	array-element-assignment
5:	$\{a[i]=0\}(a[i]:=a[i]+1;a[i]:=a[i]+1)\{a[i]=2\}$	sequence 3,4

Provided:

DISTINCT a, i

How to use GUI (continued)

- The top line is pure logic, so we expand using $\rightarrow$Introduction and $\wedge$Introduction:

1:	$a[i]=0$	assumption
	...	
2:	$(a \oplus i \mapsto a[i]+1 \oplus i \mapsto (a \oplus i \mapsto a[i]+1)[i]+1)[i]=2$	
	...	
3:	$0 \leq i$	
	...	
4:	$i<length(a)$	
5:	$(a \oplus i \mapsto a[i]+1 \oplus i \mapsto (a \oplus i \mapsto a[i]+1)[i]+1)[i]=2 \wedge 0 \leq i \wedge i<length(a)$	$\wedge$ intro 2,3,4
6:	$a[i]=0 \rightarrow (a \oplus i \mapsto a[i]+1 \oplus i \mapsto (a \oplus i \mapsto a[i]+1)[i]+1)[i]=2 \wedge 0 \leq i \wedge i<length(a)$	$\rightarrow$ intro 1-5

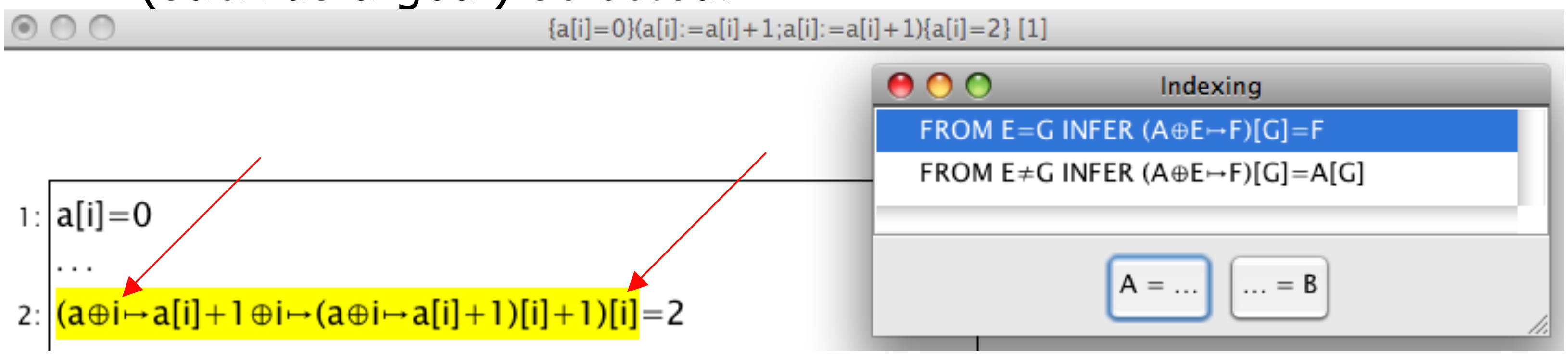
How to use GUI (continued)

- We then conclude the two array index bounds (lines 4, 5) by ∧ Elimination, giving:

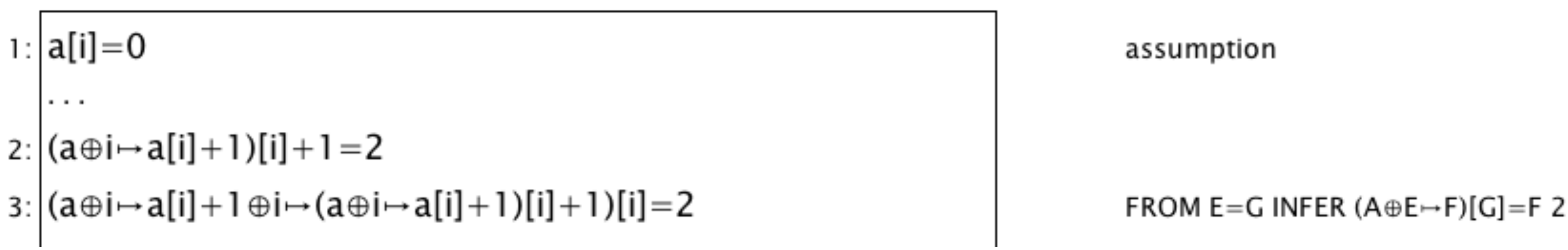
1:	$a[i]=0$	assumption
	...	
2:	$\boxed{(a\oplus i\mapsto a[i]+1\oplus i\mapsto \boxed{(a\oplus i\mapsto a[i]+1)[i]}+1)[i]}=2$	
3:	$0\leq i\wedge i<length(a)$	bounded 1
4:	$0\leq i$	∧ elim 3
5:	$i<length(a)$	∧ elim 3
6:	$(a\oplus i\mapsto a[i]+1\oplus i\mapsto (a\oplus i\mapsto a[i]+1)[i]+1)[i]=2\wedge 0\leq i\wedge i<length(a)$	∧ intro 2,4,5

How to use GUI (continued)

- We are left with a nested array-modification expression. *Carefully* sub-formula select the outer array-modification and apply the rule shown (since we have a⊕i→ …[i]). Do not have anything else (such as a goal) selected.

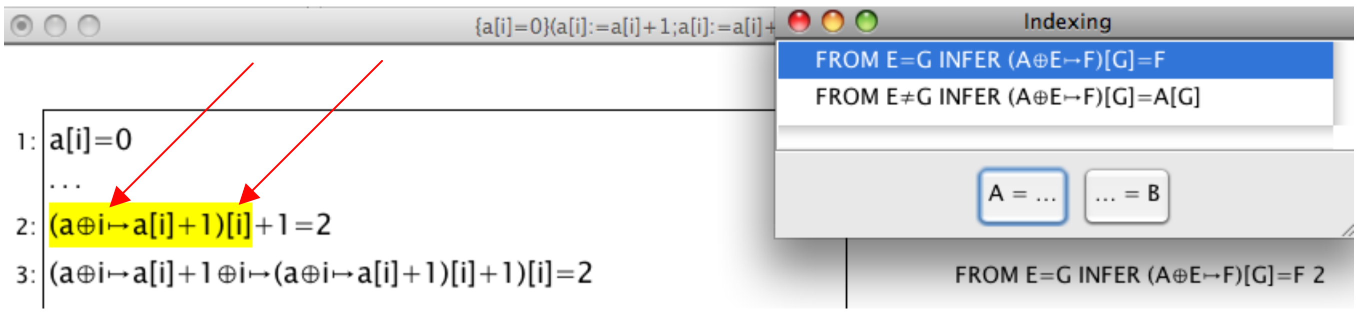


giving:



How to use GUI (continued)

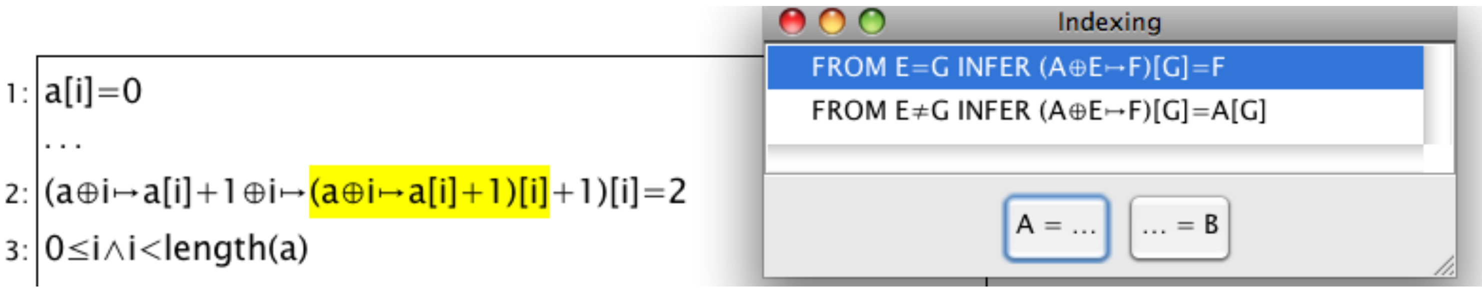
- Repeat the preceding process on the new formula:



giving:

How to use GUI (continued)

- Alternatively we could have selected the *inner* modification expression first:



giving:

1: a[i]=0 — assumption

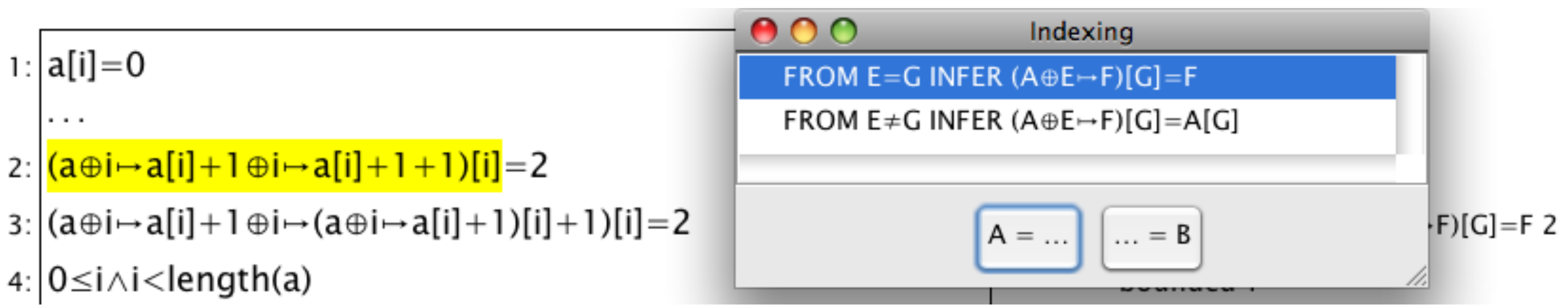
. . .

2: (a⊕i↦a[i]+1⊕i↦a[i]+1+1)[i]=2

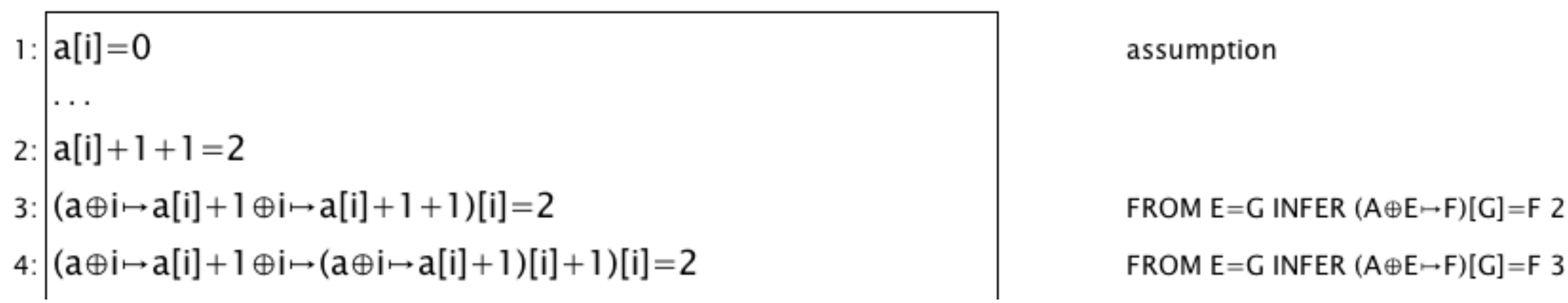
3: (a⊕i↦a[i]+1⊕i↦(a⊕i↦a[i]+1)[i]+1)[i]=2 — FROM E=G INFER (A⊕E↦F)[G]=F 2

How to use GUI (continued)

- (Note that this is different from two slides ago). Then simplify that result:

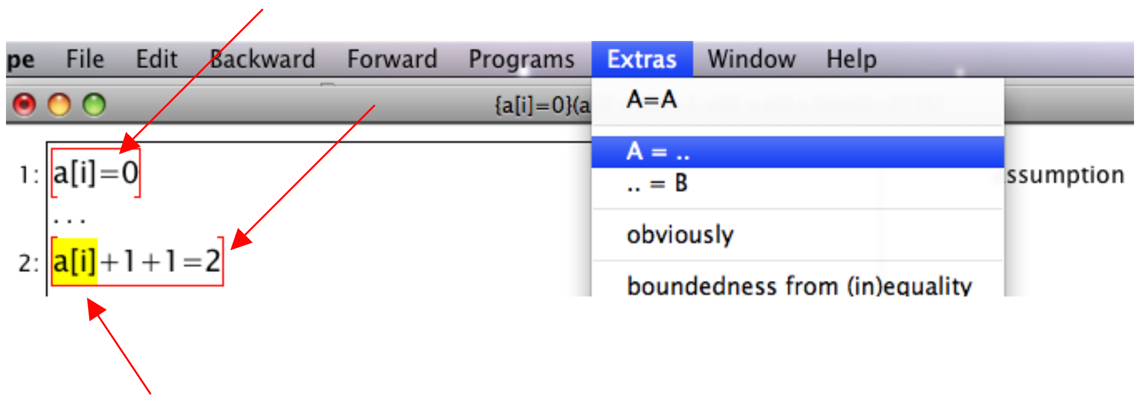


giving (as before):

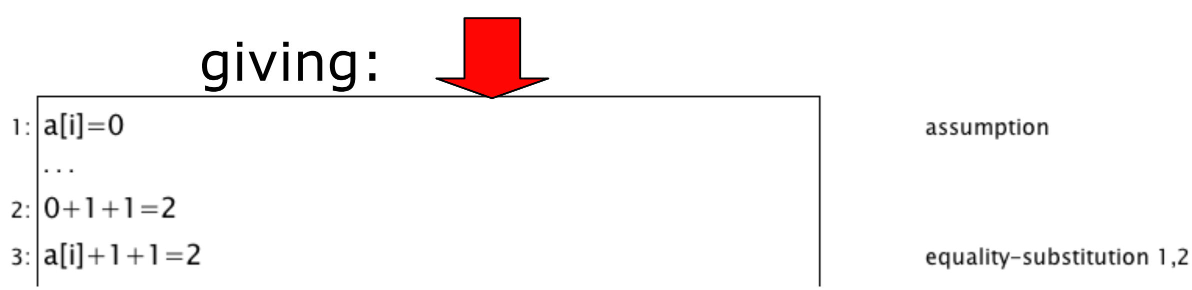


How to use GUI (continued)

- Use plain equality substitution to simplify the new goal. Note that both the goal and the equation defining the substitution are selected, and the sub-formula for which substitution is to occur is sub-formula selected (3 selections).



giving:



Consecutive Array Modification

1:	$a[i]=0$	assumption
2:	$0+1+1=2$	obviously
3:	$a[i]+1+1=2$	equality-substitution 1,2
4:	$(a\oplus i\mapsto a[i]+1)[i]+1=2$	FROM E=G INFER (A⊕E↦F)[G]=F 3
5:	$(a\oplus i\mapsto a[i]+1\oplus i\mapsto(a\oplus i\mapsto a[i]+1)[i]+1)[i]=2$	FROM E=G INFER (A⊕E↦F)[G]=F 4
6:	$0\leq i\wedge i<length(a)$	bounded 1
7:	$0\leq i$	∧ elim 6
8:	$i<length(a)$	∧ elim 6
9:	$(a\oplus i\mapsto a[i]+1\oplus i\mapsto(a\oplus i\mapsto a[i]+1)[i]+1)[i]=2\wedge 0\leq i\wedge i<length(a)$	∧ intro 5,7,8
10:	$a[i]=0\rightarrow(a\oplus i\mapsto a[i]+1\oplus i\mapsto(a\oplus i\mapsto a[i]+1)[i]+1)[i]=2\wedge 0\leq i\wedge i<length(a)$	→ intro 1-9
11:	$\{(a\oplus i\mapsto a[i]+1\oplus i\mapsto(a\oplus i\mapsto a[i]+1)[i]+1)[i]=2\wedge 0\leq i\wedge i<length(a)\}$ $(a[i]:=a[i]+1)\{(a\oplus i\mapsto a[i]+1)[i]=2\wedge 0\leq i\wedge i<length(a)\}$	array-element-assignment
12:	$\{a[i]=0\}(a[i]:=a[i]+1)\{(a\oplus i\mapsto a[i]+1)[i]=2\wedge 0\leq i\wedge i<length(a)\}$	consequence(L) 10,11
13:	$\{(a\oplus i\mapsto a[i]+1)[i]=2\wedge 0\leq i\wedge i<length(a)\}(a[i]:=a[i]+1)\{a[i]=2\}$	array-element-assignment
14:	$\{a[i]=0\}(\boxed{a[i]:=a[i]+1;a[i]:=a[i]+1})\{a[i]=2\}$	sequence 12,13

original program

Provided:

DISTINCT a, i

The following are more detail on an earlier example.

$\{\exists x.(0 \leq x \wedge x < \text{length}(a) \wedge a[x]=0)\}(i:=0)$

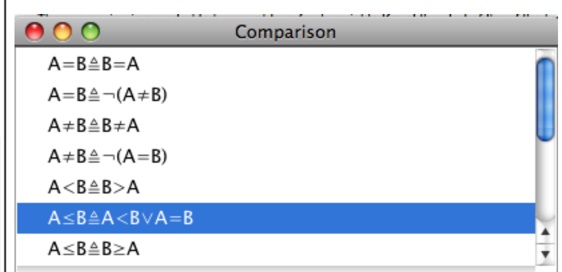
$\{0 \leq i \wedge i < \text{length}(a) \wedge \boxed{\exists x.(i \leq x \wedge x < \text{length}(a) \wedge a[x]=0)}\}\text{while } a[i] \neq 0 \text{ do } i:=i+1 \text{ od}\{a[i]=0\}$

- Look at part of the invariant here.
- Note that the lower bound on x is a function of the index i.
- This is important, because it says that the element such that **a[x] = 0 is yet to be found**.
- We need this invariant to prove termination.
- The loop test will stop when a[i] = 0.

- The expansion order is tricky.

Key Step #1: Split i $\underline{\leq}$ i1

2:	$0\leq i \wedge i<length(a) \wedge \exists x.(i\leq x \wedge x<length(a) \wedge a[x]=0) \rightarrow 0\leq i \wedge i<length(a)$	
3:	$0\leq i \wedge i<length(a) \wedge \exists x.(i\leq x \wedge x<length(a) \wedge a[x]=0) \wedge a[i]\neq 0$	assumption
4:	$0\leq i$	∧ elim 3
5:	$i<length(a)$	∧ elim 3
6:	$\exists x.(i\leq x \wedge x<length(a) \wedge a[x]=0)$	∧ elim 3
7:	$a[i]\neq 0$	∧ elim 3
8:	integer i1	assumption
9:	$i\leq i1 \wedge i1<length(a) \wedge a[i1]=0$	assumption
10:	$i\leq i1$	∧ elim 9
11:	$i<i1 \vee i=i1$	$A\leq B \triangleq A<B \vee A=B$ 10
12:	$i1<length(a)$	∧ elim 9
13:	$a[i1]=0$	∧ elim 9
	...	
14:	$0\leq i+1 \wedge i+1<length(a) \wedge \exists x.(i+1\leq x \wedge x<length(a) \wedge a[x]=0)$	
15:	$0\leq i+1 \wedge i+1<length(a) \wedge \exists x.(i+1\leq x \wedge x<length(a) \wedge a[x]=0)$	∃ elim 6,8-14
16:	$0\leq i \wedge i<length(a) \wedge \exists x.(i\leq x \wedge x<length(a) \wedge a[x]=0) \wedge a[i]\neq 0$ $\rightarrow 0\leq i+1 \wedge i+1<length(a) \wedge \exists x.(i+1\leq x \wedge x<length(a) \wedge a[x]=0)$	→ intro 3-15



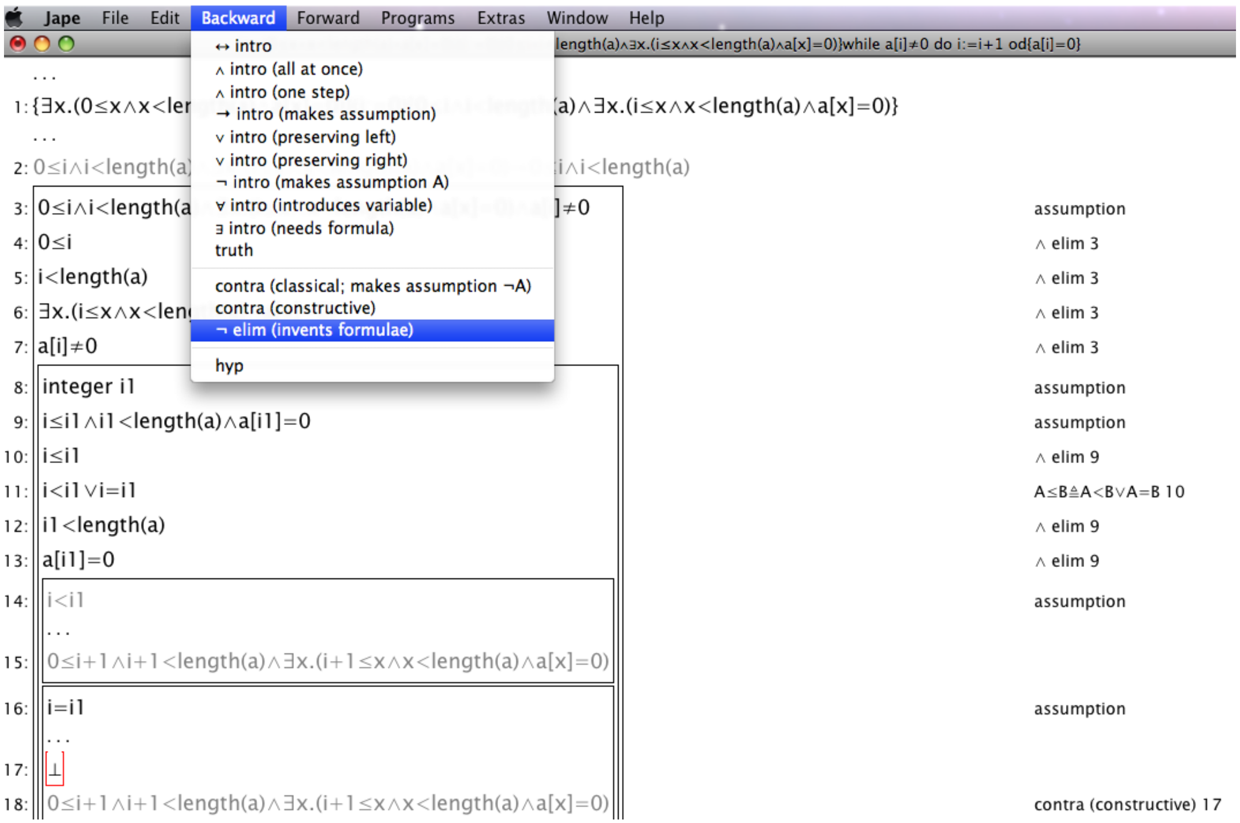
Then use ∨Elimination.

Key Step #2:
Aim for a contradiction in the i = i1 case.

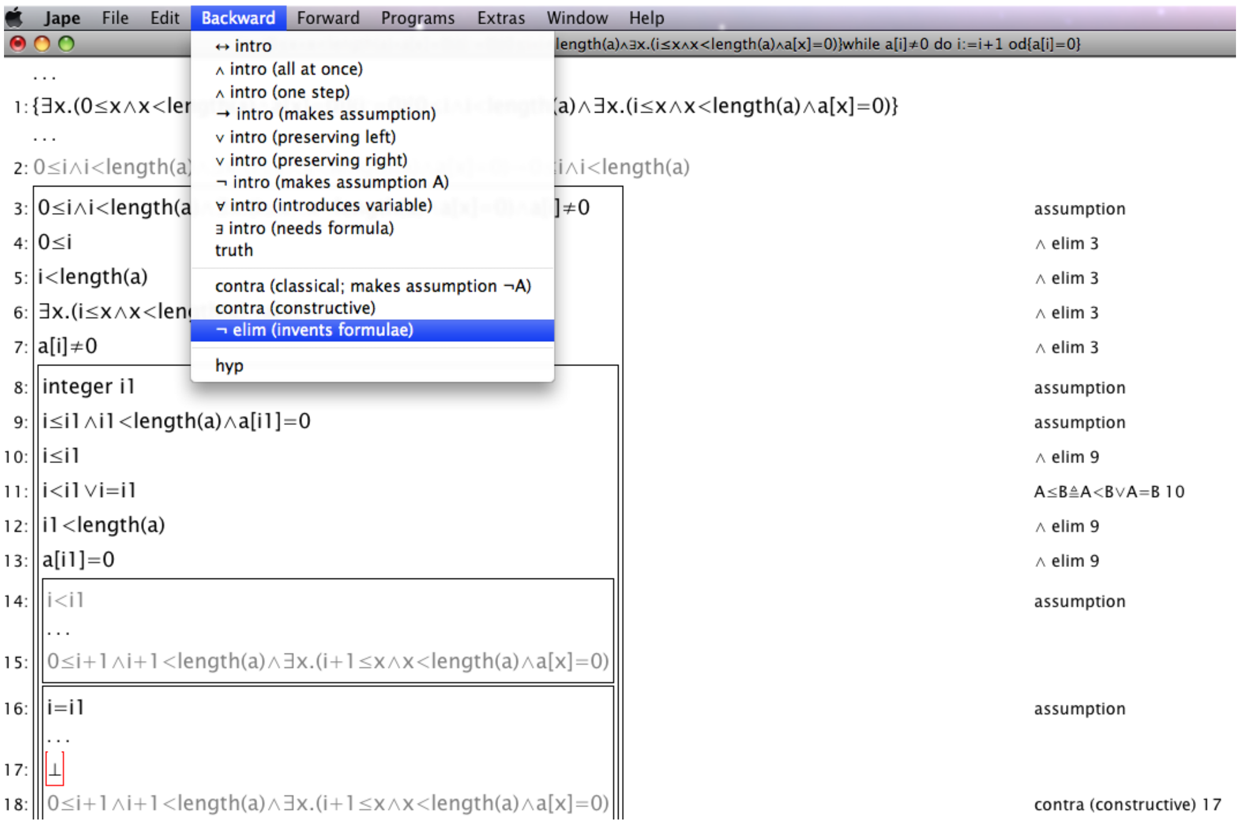
7:	$a[i] \neq 0$	$\wedge$ elim 3
8:	integer i1	assumption
9:	$i \leq i1 \wedge i1 < length(a) \wedge a[i1] = 0$	assumption
10:	$i \leq i1$	$\wedge$ elim 9
11:	$i < i1 \vee i = i1$	$A \leq B \triangleq A < B \vee A = B$ 10
12:	$i1 < length(a)$	$\wedge$ elim 9
13:	$a[i1] = 0$	$\wedge$ elim 9
14:	$i < i1$	assumption
	...	
15:	$0 \leq i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \leq x \wedge x < length(a) \wedge a[x] = 0)$	
16:	$i = i1$	assumption
	...	
17:	$\perp$	
18:	$0 \leq i+1 \wedge i+1 < length(a) \wedge \exists x.(i+1 \leq x \wedge x < length(a) \wedge a[x] = 0)$	contra (constructive) 17

Now introduce a backward $\neg$Elimination.

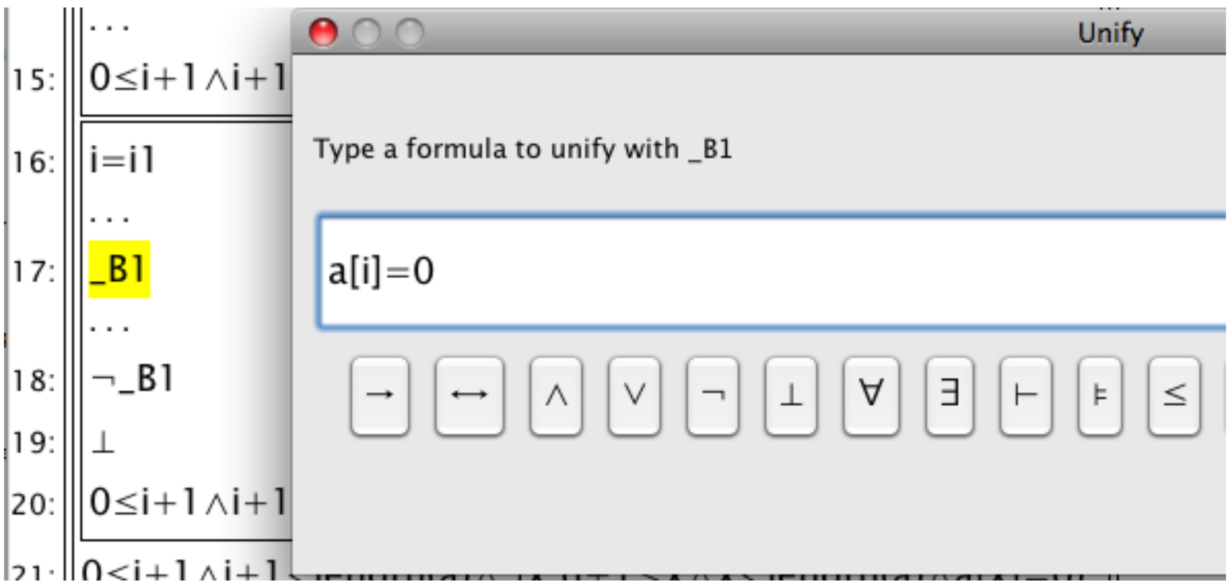
Key Step #2, continued:



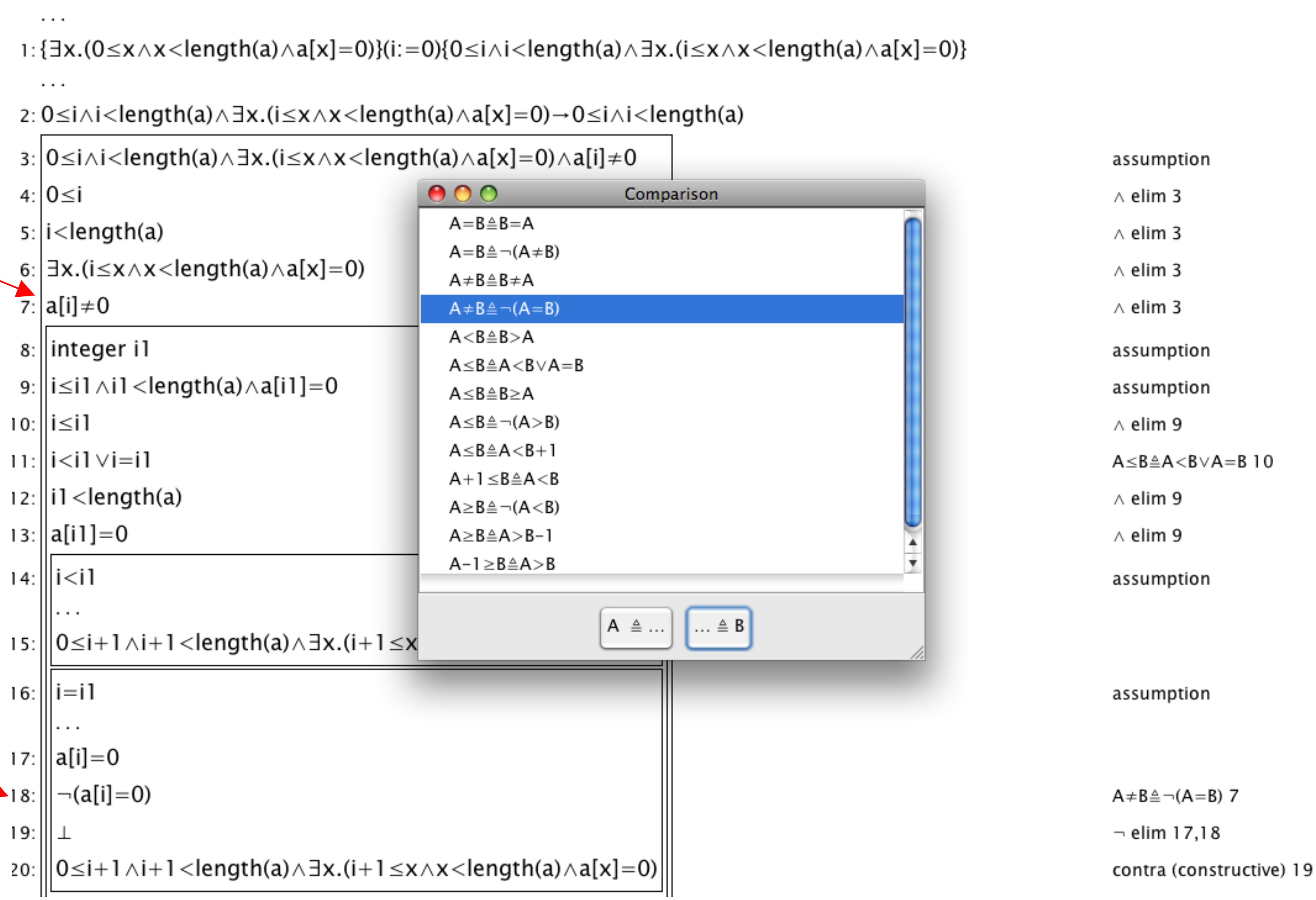
Key Step #2, continued:



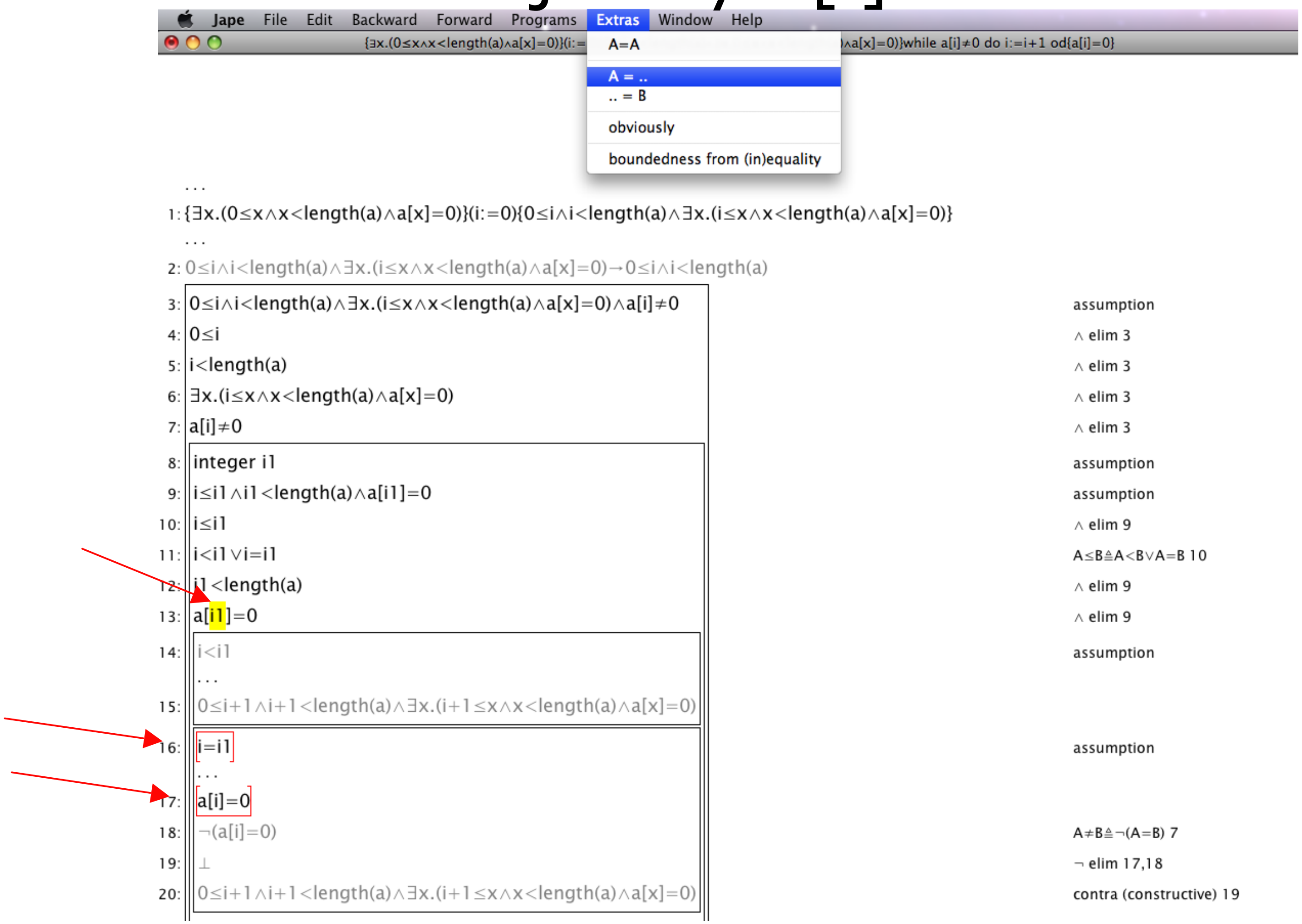
Key Step #2, continued: unify



Key Step #2, continued: use comparison menu to justify ¬(a[i] = 0)



Key Step #2, concluded: substitute to justify a[i] = 0



Status following key step #2

1:	$\{\exists x.(0 \le x \land x < length(a) \land a[x]=0)\}(i:=0)\{0 \le i \land i < length(a) \land \exists x.(i \le x \land x < length(a) \land a[x]=0)\}$	
	...	
2:	$0 \le i \land i < length(a) \land \exists x.(i \le x \land x < length(a) \land a[x]=0) \rightarrow 0 \le i \land i < length(a)$	
3:	$0 \le i \land i < length(a) \land \exists x.(i \le x \land x < length(a) \land a[x]=0) \land a[i] \ne 0$	assumption
4:	$0 \le i$	$\land$ elim 3
5:	$i < length(a)$	$\land$ elim 3
6:	$\exists x.(i \le x \land x < length(a) \land a[x]=0)$	$\land$ elim 3
7:	$a[i] \ne 0$	$\land$ elim 3
8:	integer i1	assumption
9:	$i \le i1 \land i1 < length(a) \land a[i1]=0$	assumption
10:	$i \le i1$	$\land$ elim 9
11:	$i < i1 \lor i = i1$	$A \le B \triangleq A < B \lor A = B$ 10
12:	$i1 < length(a)$	$\land$ elim 9
13:	$a[i1]=0$	$\land$ elim 9
14:	$i < i1$	assumption
	...	
15:	$0 \le i+1 \land i+1 < length(a) \land \exists x.(i+1 \le x \land x < length(a) \land a[x]=0)$	
16:	$i = i1$	assumption
17:	$a[i]=0$	equality-substitution 16,13
18:	$\neg(a[i]=0)$	$A \ne B \triangleq \neg(A=B)$ 7
19:	$\bot$	$\neg$ elim 17,18
20:	$0 \le i+1 \land i+1 < length(a) \land \exists x.(i+1 \le x \land x < length(a) \land a[x]=0)$	contra (constructive) 19
21:	$0 \le i+1 \land i+1 < length(a) \land \exists x.(i+1 \le x \land x < length(a) \land a[x]=0)$	$\lor$ elim 11,14-15,16-20
22:	$0 \le i+1 \land i+1 < length(a) \land \exists x.(i+1 \le x \land x < length(a) \land a[x]=0)$	$\exists$ elim 6,8-21

Completed Proof (lines 1-15)

1:	$\exists x.(0 \le x \wedge x < length(a) \wedge a[x]=0)$	assumption
2:	$0 \le 0$	obviously
3:	integer i2	assumption
4:	$0 \le i2 \wedge i2 < length(a) \wedge a[i2]=0$	assumption
5:	$0 \le i2$	$\wedge$ elim 4
6:	$i2 < length(a)$	$\wedge$ elim 4
7:	$0 < length(a)$	obviously, from 6,5
8:	$0 < length(a)$	$\exists$ elim 1,3-7
9:	$0 \le 0 \wedge 0 < length(a) \wedge \exists x.(0 \le x \wedge x < length(a) \wedge a[x]=0)$	$\wedge$ intro 2,8,1
10:	$\exists x.(0 \le x \wedge x < length(a) \wedge a[x]=0) \rightarrow 0 \le 0 \wedge 0 < length(a) \wedge \exists x.(0 \le x \wedge x < length(a) \wedge a[x]=0)$	$\rightarrow$ intro 1-9
11:	$\{0 \le 0 \wedge 0 < length(a) \wedge \exists x.(0 \le x \wedge x < length(a) \wedge a[x]=0)\}(i:=0)\{0 \le i \wedge i < length(a) \wedge \exists x.(i \le x \wedge x < length(a) \wedge a[x]=0)\}$	variable-assignment
12:	$\{\exists x.(0 \le x \wedge x < length(a) \wedge a[x]=0)\}(i:=0)\{0 \le i \wedge i < length(a) \wedge \exists x.(i \le x \wedge x < length(a) \wedge a[x]=0)\}$	consequence(L) 10,11
13:	$0 \le i \wedge i < length(a) \wedge \exists x.(i \le x \wedge x < length(a) \wedge a[x]=0)$	assumption
14:	$0 \le i \wedge i < length(a)$	$\wedge$ elim(L) 13
15:	$0 \le i \wedge i < length(a) \wedge \exists x.(i \le x \wedge x < length(a) \wedge a[x]=0) \rightarrow 0 \le i \wedge i < length(a)$	$\rightarrow$ intro 13-14

Completed Proof (lines 16-39)

16:	$0 \le i \land i < length(a) \land \exists x.(i \le x \land x < length(a) \land a[x]=0) \land a[i] \ne 0$	assumption
17:	$0 \le i$	$\land$ elim 16
18:	$\exists x.(i \le x \land x < length(a) \land a[x]=0)$	$\land$ elim 16
19:	$a[i] \ne 0$	$\land$ elim 16
20:	integer $i1$	assumption
21:	$i \le i1 \land i1 < length(a) \land a[i1]=0$	assumption
22:	$i \le i1$	$\land$ elim 21
23:	$i < i1 \lor i = i1$	$A \le B \triangleq A < B \lor A = B$ 22
24:	$i1 < length(a)$	$\land$ elim 21
25:	$a[i1]=0$	$\land$ elim 21
26:	$i < i1$	assumption
27:	$0 \le i+1$	obviously, from 17
28:	$i+1 < length(a)$	obviously, from 26,24
29:	$i+1 \le i1$	obviously, from 26
30:	$i+1 \le i1 \land i1 < length(a) \land a[i1]=0$	$\land$ intro 29,24,25
31:	$\exists x.(i+1 \le x \land x < length(a) \land a[x]=0)$	$\exists$ intro 30
32:	$0 \le i+1 \land i+1 < length(a) \land \exists x.(i+1 \le x \land x < length(a) \land a[x]=0)$	$\land$ intro 27,28,31
33:	$i = i1$	assumption
34:	$a[i]=0$	equality-substitution 33,25
35:	$\neg(a[i]=0)$	$A \ne B \triangleq \neg(A=B)$ 19
36:	$\bot$	$\neg$ elim 34,35
37:	$0 \le i+1 \land i+1 < length(a) \land \exists x.(i+1 \le x \land x < length(a) \land a[x]=0)$	contra (constructive) 36
38:	$0 \le i+1 \land i+1 < length(a) \land \exists x.(i+1 \le x \land x < length(a) \land a[x]=0)$	$\lor$ elim 23,26-32,33-37
39:	$0 \le i+1 \land i+1 < length(a) \land \exists x.(i+1 \le x \land x < length(a) \land a[x]=0)$	$\exists$ elim 18,20-38

Completed Proof (lines 40-60)

40:	$0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\land a[i]\neq 0\rightarrow 0\le i+1\land i+1<\text{length}(a)\land\exists x.(i+1\le x\land x<\text{length}(a)\land a[x]=0)$	→ intro 16-39
41:	$\{0\le i+1\land i+1<\text{length}(a)\land\exists x.(i+1\le x\land x<\text{length}(a)\land a[x]=0)\}(i:=i+1)\{0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\}$	variable-assignment
42:	$\{0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\land a[i]\neq 0\}(i:=i+1)\{0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\}$	consequence(L) 40,41
43:	$0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\land a[i]\neq 0$	assumption
44:	$i<\text{length}(a)$	∧ elim 43
45:	$\text{length}(a)-i>0$	obviously, from 44
46:	$0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\land a[i]\neq 0\rightarrow\text{length}(a)-i>0$	→ intro 43-45
47:	integer Km	assumption
48:	$0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\land a[i]\neq 0\land\text{length}(a)-i=Km$	assumption
49:	$\text{length}(a)-i=Km$	∧ elim 48
50:	$\text{length}(a)-(i+1)<Km$	obviously, from 49
51:	$0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\land a[i]\neq 0\land\text{length}(a)-i=Km\rightarrow\text{length}(a)-(i+1)<Km$	→ intro 48-50
52:	$\{\text{length}(a)-(i+1)<Km\}(i:=i+1)\{\text{length}(a)-i<Km\}$	variable-assignment
53:	$\{0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\land a[i]\neq 0\land\text{length}(a)-i=Km\}(i:=i+1)\{\text{length}(a)-i<Km\}$	consequence(L) 51,52
54:	$\{0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\}\text{while } a[i]\neq 0\text{ do } i:=i+1\text{ od}$ $\{0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\land\lnot(a[i]\neq 0)\}$	while 15,42,46,47-53
55:	$0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\land\lnot(a[i]\neq 0)$	assumption
56:	$\lnot(a[i]\neq 0)$	∧ elim 55
57:	$a[i]=0$	$A=B\triangleq\lnot(A\neq B)$ 56
58:	$0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\land\lnot(a[i]\neq 0)\rightarrow a[i]=0$	→ intro 55-57
59:	$\{0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\}\text{while } a[i]\neq 0\text{ do } i:=i+1\text{ od}\{a[i]=0\}$	consequence(R) 54,58
60:	$\{\exists x.(0\le x\land x<\text{length}(a)\land a[x]=0)\}(i:=0)\{0\le i\land i<\text{length}(a)\land\exists x.(i\le x\land x<\text{length}(a)\land a[x]=0)\}\text{while } a[i]\neq 0\text{ do } i:=i+1\text{ od}\{a[i]=0\}$	Ntuple 12,59