```
⊢ ∀r:ℕ. ∀u:ℤ. ∀v:ℤ List.  (initseg_sum(r;[u / v]) = (u + initseg_sum(r - 1;v)))
|
BY RepeatFor 3 ((D 0 THENA Auto))
|
1. r: ℕ
2. u: ℤ
3. v: ℤ List
⊢ initseg_sum(r;[u / v]) = (u + initseg_sum(r - 1;v))
|
BY Assert ⌈∃x:ℤ. (x = (u + initseg_sum(r - 1;v)))⌉⋅
|\
| ⊢ ∃x:ℤ. (x = (u + initseg_sum(r - 1;v)))
| |
1 BY (InstConcl [⌈u + initseg_sum(r - 1;v)⌉]⋅ THEN Auto)
 \
  4. ∃x:ℤ. (x = (u + initseg_sum(r - 1;v)))
  ⊢ initseg_sum(r;[u / v]) = (u + initseg_sum(r - 1;v))
  |
  BY D 4
  |
  4. x: ℤ
  5. x = (u + initseg_sum(r - 1;v))
  ⊢ initseg_sum(r;[u / v]) = (u + initseg_sum(r - 1;v))
  |
  BY (RevHypSubst 5 0 THENA Auto)
  |
  ⊢ initseg_sum(r;[u / v]) = x
  |
  BY Unfold ‘initseg_sum‘ 0
  |
  ⊢ l_sum(firstn(r + 1;[u / v])) = x
  |
  BY RecUnfold ‘firstn‘ 0
  |
  ⊢ l_sum(case [u / v] of
  |          [] => []
  |          a::as’ =>
  |           if 0 <z r + 1 then [a / firstn((r + 1) - 1;as’)] else [] fi
  |  esac)
  |  = x
  |
  BY (AutoBoolCase ⌈0 <z r + 1⌉⋅ THEN Reduce 0)
  |
  6. 0 < (r + 1)
  ⊢ l_sum([u / firstn((r + 1) - 1;v)]) = x
  |
  BY (RepUR ‘‘l_sum‘‘ 0 THEN Fold ‘l_sum‘ 0)
  |
  ⊢ (u + l_sum(firstn((r + 1) - 1;v))) = x
  |
  BY Assert ⌈((r + 1) - 1) = ((r - 1) + 1)⌉⋅
  |\
  | ⊢ ((r + 1) - 1) = ((r - 1) + 1)
```

```
| |
1 BY Auto
 \
  7. ((r + 1) - 1) = ((r - 1) + 1)
  ⊢ (u + l_sum(firstn((r + 1) - 1;v))) = x
  |
  BY (HypSubst 7 0 THENA Auto)
  |
  ⊢ (u + l_sum(firstn((r - 1) + 1;v))) = x
  |
  BY Fold ‘initseg_sum‘ 0
  |
  ⊢ (u + initseg_sum(r - 1;v)) = x
  |
  BY Auto
```