data node {

int val;

node left;

node right;

};

avl<h, n, b, s, B> ==

self = null & h = 0 & n = 0 & b = 0 & B = {}

or self::node<v, p, q> \* p::avl<h1, n1, b1, s1, B1> \* q::avl<h2, n2, b2, s2, B2>

& v >= 0

& n = 1 + n1 + n2

& -1 <= h1 - h2 <= 1 & h = 1 + max(h1, h2)

& **$:**b = h1 - h2

& s = **$:v** + s1 + s2

& B = union({**v**}, B1, B2) & forall (x : (x notin B1 | x <= v)) & forall (x : (x notin B2 | x > v))

inv n >= 0 & h >= 0 & **$:**n >= h & **#:**-1 <= b <= 1 & s >= 0 & forall (x : (x notin B | x >= 0))

Because

A |- C

B |- C

----------------

A \/ B |- C

and the invariant is satisfied under the case self = null, we only need to consider the case self != null.

**Automatic slicing:** performance problem

- Linking variables (v causes s and B grouped together)

- Linking constraints (n >= h (RHS) and b = h1 - h2 (LHS) cause n, h and B grouped together)

**Forced slicing:**

- Indicate the linking variables and linking constraints by **$**

- Ignore the variables and constraints with $ when doing slicing

~~1. 🡪 Start from RHS to choose the relevant constraints by calculating the closure of the RHS’s variables?~~

2. 🡪 Or slice the LHS and RHS independently?

~~- Use # to indicate the proofs do not need the corresponding inductive hypotheses?~~ (Do not include the hypotheses that are not related to the added constraints in the LHS or RHS)

1.

2. Slicing the LHS and RHS independently

**a. The original LHS with inductive hypotheses**

v >= 0

& n = 1 + n1 + n2

& -1 <= h1 - h2 <= 1 & h = 1 + max(h1, h2)

& **$:**b = h1 - h2

& s = **$:v** + s1 + s2

& B = union({**$:v**}, B1, B2) & forall (x : (x notin B1 | x <= **$:v**)) & forall (x : (x notin B2 | x > **$:v**))

& n1 >= 0 & n2 >= 0

& h1 >= 0 & h2 >= 0

& **$:**n1 >= h1 & **$:**n2 >= h2

& **#:**-1 <= b1 <= 1 & **#:**-1 <= b2 <= 1

& s1 >= 0 & s2 >= 0

& forall (x : (x notin B1 | x >= 0)) & forall (x : (x notin B2 | x >= 0))

**b. Do slicing after removing all linking constraints and ignoring linking variables:**

LHS:

v >= 0

& n = 1 + n1 + n2

& -1 <= h1 - h2 <= 1 & h = 1 + max(h1, h2)

~~&~~ **~~$:~~**~~b = h1 - h2~~

& s = **~~$:v~~** + s1 + s2

& B = union({**~~$:v~~**}, B1, B2) & forall (x : (x notin B1 | x <= **~~$:v~~**)) & forall (x : (x notin B2 | x > **~~$:v~~**))

& n1 >= 0 & n2 >= 0

& h1 >= 0 & h2 >= 0

~~&~~ **~~$:~~**~~n1 >= h1 &~~ **~~$:~~**~~n2 >= h2~~

& **#:**-1 <= b1 <= 1 & **#:**-1 <= b2 <= 1

& s1 >= 0 & s2 >= 0

& forall (x : (x notin B1 | x >= 0)) & forall (x : (x notin B2 | x >= 0))

RHS:

n >= 0 & h >= 0 & **~~$:~~**~~n >= h~~ & **#:**-1 <= b <= 1 & s >= 0 & forall (x : (x notin B | x >= 0))

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Partition | LHS | Propagation (P) | RHS | |
|  | $:v | v >= 0 |  |  | |
| 1 | n, n1, n2 | n = 1 + n1 + n2 | n1 >= 0 (P1),  n2 >= 0 (P1) | n >= 0  (LHS1 + P1) | $:n >= h  (LHS1 + LHS2 + P6) |
| 2 | h1, h2, h | -1 <= h1 - h2 <= 1,  h = 1 + max(h1, h2) | h1 >= 0 (P2),  h2 >= 0 (P2) | h >= 0  (LHS2 + P2) |
| 3 | s, s1, s2 | s = **$:v** + s1 + s2 | s1 >= 0 (P3),  s2 >= 0 (P3) | s >= 0  (LHS3 + P3 + $:v) | |
| 4 | B, B1, B2 | B = union({**$:v**}, B1, B2),  forall (x : (x notin B1 | x <= **$:v**)),  forall (x : (x notin B2 | x > **$:v**)) | forall (x : (x notin B1 | x >= 0)) (P4),  forall (x : (x notin B2 | x >= 0)) (P4) | forall (x : (x notin B | x >= 0))  (LHS4 + P4 + $:v) | |
| 5 | b | $:b = h1 – h2 | **#:**-1 <= b1 <= 1 (P5),  **#:**-1 <= b2 <= 1 (P5) | **#:**-1 <= b <= 1  (LHS5 + {h1, h2}) | |
|  |  |  | $:n1 >= h1 (P6)  $:n2 >= h2 (P6) |  | |

|  |  |  |
| --- | --- | --- |
| Optimal | Automatic Slicing  [[n, h, b], [s, B]] | Forced Slicing  [[n, n1, n2], [h, h1, h2], [b], [s, s1, s2], [B, B1, B2]] |
| n = 1 + n1 + n2  & n1 >= 0 & n2 >= 0  |- n >= 0 | n = 1 + n1 + n2  & -1 <= h1 - h2 <= 1  & h = 1 + max(h1, h2)  & b = h1 - h2  & n1 >= 0 & n2 >= 0  & h1 >= 0 & h2 >= 0  & n1 >= h1 & n2 >= h2  & -1 <= b1 <= 1 & -1 <= b2 <= 1  |- n >= 0 | n = 1 + n1 + n2  & n1 >= 0 & n2 >= 0  ~~& n1 >= h1 & n2 >= h2~~  |- n >= 0 |
| h = 1 + max(h1, h2)  & h1 >= 0 & h2 >= 0  |- h >= 0 | n = 1 + n1 + n2  & -1 <= h1 - h2 <= 1  & h = 1 + max(h1, h2)  & b = h1 - h2  & n1 >= 0 & n2 >= 0  & h1 >= 0 & h2 >= 0  & n1 >= h1 & n2 >= h2  & -1 <= b1 <= 1 & -1 <= b2 <= 1  |- h >= 0 | -1 <= h1 - h2 <= 1  & h = 1 + max(h1, h2)  & h1 >= 0 & h2 >= 0  ~~& n1 >= h1 & n2 >= h2~~  |- h >= 0 |
| n = 1 + n1 + n2  & h = 1 + max(h1, h2)  & n1 >= h1 & n2 >= h2  |- n >= h | n = 1 + n1 + n2  & -1 <= h1 - h2 <= 1  & h = 1 + max(h1, h2)  & b = h1 - h2  & n1 >= 0 & n2 >= 0  & h1 >= 0 & h2 >= 0  & n1 >= h1 & n2 >= h2  & -1 <= b1 <= 1 & -1 <= b2 <= 1  |- n >= h | n = 1 + n1 + n2  & -1 <= h1 - h2 <= 1  & h = 1 + max(h1, h2)  ~~& n1 >= 0 & n2 >= 0~~  ~~& h1 >= 0 & h2 >= 0~~  & n1 >= h1 & n2 >= h2  |- n >= h |
| -1 <= h1 - h2 <= 1  & b = h1 - h2  |- -1 <= b <= 1 | n = 1 + n1 + n2  & -1 <= h1 - h2 <= 1  & h = 1 + max(h1, h2)  & b = h1 - h2  & n1 >= 0 & n2 >= 0  & h1 >= 0 & h2 >= 0  & n1 >= h1 & n2 >= h2  & -1 <= b1 <= 1 & -1 <= b2 <= 1  |- -1 <= b <= 1 | -1 <= h1 - h2 <= 1  ~~& h = 1 + max(h1, h2)~~  & b = h1 - h2  ~~& h1 >= 0 & h2 >= 0~~  ~~& n1 >= h1 & n2 >= h2~~  ~~& -1 <= b1 <= 1 & -1 <= b2 <= 1~~  |- -1 <= b <= 1 |
| v >= 0  & s = v + s1 + s2  & s1 >= 0 & s2 >= 0  |- s >= 0 | v >= 0  & s = v + s1 + s2  & B = union({v}, B1, B2)  & forall (x : (x notin B1 | x <= v))  & forall (x : (x notin B2 | x > v))  & s1 >= 0 & s2 >= 0  & forall (x : (x notin B1 | x >= 0))  & forall (x : (x notin B2 | x >= 0))  |- s >= 0 | v >= 0  & s = v + s1 + s2  & s1 >= 0 & s2 >= 0  |- s >= 0 |
| v >= 0  & B = union({v}, B1, B2)  & forall (x : (x notin B1 | x >= 0))  & forall (x : (x notin B2 | x >= 0))  |- forall (x : (x notin B | x >= 0)) | v >= 0  & s = v + s1 + s2  & B = union({v}, B1, B2)  & forall (x : (x notin B1 | x <= v))  & forall (x : (x notin B2 | x > v))  & s1 >= 0 & s2 >= 0  & forall (x : (x notin B1 | x >= 0))  & forall (x : (x notin B2 | x >= 0))  |- forall (x : (x notin B | x >= 0)) | v >= 0  & B = union({v}, B1, B2)  & forall (x : (x notin B1 | x <= v))  & forall (x : (x notin B2 | x > v))  & forall (x : (x notin B1 | x >= 0))  & forall (x : (x notin B2 | x >= 0))  |- forall (x : (x notin B | x >= 0)) |