Termination

Total Correctness

```
[A] stmt [B]
```

Hoare triple

- If A holds before stmt, stmt terminates and B will hold afterward.

Total Correctness

Definition: a well-ordered set is a set S with a total order $>$ such that every non-empty subset of S has a least element.

E.g., $(\mathbb{N}, >)$ is a w.o. set, $(\mathbb{Z}, >)$ is not

$(\mathbb{N}^2, >)$ where $(a, b) > (a', b')$ if $a > a'$, or $a = a'$ and $b > b'$

Total Correctness

Definition: a well-ordered set is a set S with a total order $>$ such that every non-empty subset of S has a least element.

E.g., $(\mathbb{N}, >)$ is a w.o. set, $(\mathbb{Z}, >)$ is not

$(\mathbb{N}^2, >)$ where $(a, b) > (a', b')$ if $a > a'$, or $a = a'$ and $b > b'$

Theorem: There is no infinite descending chain w.r.t. a well-ordered set.

Total Correctness

Termination:

1. find a ranking function $rank: ProgStates \rightarrow (S, >)$

2. find a set of cutpoints (program points) to cut the program

3. prove for any cutpoint pc, and any two program states P_1, P_2, if (P_1, pc) reaches (P_2, pc) in an execution sequence, then $rank(P_1) > rank(P_2)$

Example

```
int i, j;
i := 1;
j := 1;
while (j != n) {
    i := i + 2*j + 1;
    j := j+1;
}
return i;
```

Example

```
[n>0]
int i, j;
i := 1;
j := 1;
while (j != n) {
    i := i + 2*j + 1;
    j := j+1;
}
return i;
[true]
```

Example

```
[n>0]
int i, j;
i := 1;
j := 1;
while (j != n) {
    decreases (n-j)
    i := i + 2*j + 1;
    j := j+1;
}
return i;
[true]
```