CS 686 Weakest Pre-Condition

1. **Statements**

0: $(y \geq 0) \wedge (x \geq 0 \Rightarrow y = x) \wedge (x < 0 \Rightarrow y =- x)$ (ensured post-condition)

1: $(x \geq 0) \wedge (x \geq 0 \Rightarrow x = x) \wedge (x < 0 \Rightarrow x =- x)$ (for $(if\ x \geq 0\ then\ y \leftarrow x)$ statement)

= $(x \geq 0)$

2: $(-x \geq 0) \wedge (x \geq 0 \Rightarrow -x = x) \wedge (x < 0 \Rightarrow -x =- x)$ (for $(if\ x < 0\ then\ y \leftarrow -x)$ statement)

= $(x \leq 0)$

3: $wp(if\ \ x < 0\ \ then\ 2\ \ else\ \ 1,\ Q)$

= $(\ x < 0\ \ \wedge\ \ wp(2,\ Q))\ \vee\ (\ x \geq 0\ \ \wedge\ \ wp(1,\ Q))$

= $(\ x < 0\ \ \wedge\ \ x \leq 0)\ \vee\ (\ x \geq 0\ \ \wedge\ \ x \geq 0)$

= True

Lines/Branches (numbered by code line)

6: if true to line 7 else line 9; statement 3 holds

7: to line 10; statement 2 holds

8:

9: to line 10; statement 1 holds

10: statement 0 holds

Therefore, the weakest precondition is "requires True", which is implied by any other constraints (it is the universally weakest condition).

2. **Statements**

0: $(big\ >\ small)$

1: $(y\ >\ x)$

2: $(x\ >\ y)$

3: $wp(if\ \ x > y\ \ then\ 2\ \ else\ \ 1,\ Q)$

= $(\ x > y\ \ \wedge\ \ wp(2,\ Q))\ \vee\ (\ x \leq y\ \ \wedge\ \ wp(1,\ Q))$

= $(\ x > y\ \ \wedge\ \ x > y)\ \vee\ (\ x \leq y\ \ \wedge\ \ x\ <\ y)$

= $(\ x > y\ )\ \vee\ (\ x\ <\ y)$

= $(\ x \neq y\ )$

Lines/Branches (numbered by code line)

17: i if true to line 18 else line 20; statement 3 holds

18: to line 21; statement 2 holds

19:

20: to line 21; statement 1 holds

21: statement 0 holds

Therefore, the weakest precondition is "requires $(\ x \neq y\ )$", which does verify when entered into the Dafny program.

3. Statements

0: $res == n_0m_0$

1: I (assert true at end of loop if assumed true at the start)

= $res + nm == n_0m_0$

2: $res + (n - 1)m == n_0m_0$

= $res + nm - m == n_0m_0$

3: $res + m + nm - m == n_0m_0$

= $res + nm == n_0m_0$ (the precondition for I)

4: $D > 0$ (assume true at the end of the loop; find precondition)

5: $n > 0$

6: $(old(D) == n) \wedge (D < old(D))$

= $(old(D) == n) \wedge (n < old(D))$

7: $(old(D) == n) \wedge (n - 1 < old(D))$

8: $(n - 1 < n)$ = True

9: $wp(while\ (0 < n)\ (res + nm == n_0m_0)\ S,\ res == n_0m_0)$

= $I \wedge (\forall xs,\ I \wedge E \Rightarrow wp(S, I)) \wedge (\forall xs,\ I \wedge \neg E \Rightarrow Q) \wedge (\forall xs,\ I \wedge E \Rightarrow D > 0) \wedge (\forall xs,\ I \wedge E \Rightarrow wp(S,\ old(D) > D))$

= $(res + nm == n_0m_0)$

$\wedge$ $(\forall xs,\ (res + nm == n_0m_0) \wedge (0 < n) \Rightarrow (res + nm = n_0m_0)$ (vacuously true)

$\wedge$ $(\forall xs,\ (res + nm == n_0m_0) \wedge (0 \geq n) \Rightarrow res == n_0m_0)$ (exit condition)

$\wedge$ $(\forall xs,\ (res + nm == n_0m_0) \wedge (0 < n) \Rightarrow n > 0)$ (finite decreasing; vacuous)

$\wedge$ $(\forall xs,\ (res + nm == n_0m_0) \wedge (0 < n) \Rightarrow True)$ (vacuously true)

= $(res + nm == n_0m_0) \wedge (0 \geq n \Rightarrow res = n_0m_0)$

10: $(res + n_0m_0 == n_0m_0) \wedge (0 \geq - n_0 \Rightarrow res == n_0m_0)$

= $(res == 0) \wedge (n_0 \geq 0 \Rightarrow res = n_0m_0)$

11 $(res + n_0m_0 == n_0m_0) \wedge (0 \geq n_0 \Rightarrow res == n_0m_0)$

=$(res == 0) \wedge (n_0 \leq 0 \Rightarrow res == n_0m_0)$

12: $wp(if\ \ n_0 \geq 0\ \ then\ s_T\ else\ s_F,\ Q)$

= $((n_0 \geq 0) \wedge (res == 0) \wedge (n_0 \leq 0 \Rightarrow res == n_0m_0))$

$\vee$ $((n_0 < 0) \wedge (res == 0) \wedge (n_0 \geq 0 \Rightarrow res == n_0m_0))$

= $(res == 0) \wedge [((n_0 \geq 0) \wedge (n_0 \leq 0 \Rightarrow res == n_0m_0)) \vee ((n_0 < 0))]$

= $(res == 0) \wedge [((n_0 < 0) \vee (n_0 \geq 0)) \wedge ((n_0 < 0) \vee (m_0n_0 \neq 0 \Rightarrow n_0 > 0))]$

= $(res == 0) \wedge [(n_0 < 0) \vee (m_0n_0 \neq 0 \Rightarrow n_0 > 0)]$

= $(res == 0) \wedge [(n_0 < 0) \vee (m_0n_0 == 0) \vee (n_0 > 0)]$

= $(res == 0) \wedge [(n_0 \neq 0) \vee (m_0n_0 == 0)]$

= $(res == 0) \wedge [(n_0 == 0) \Rightarrow (m_0n_0 == 0)]$

= $(res == 0)$

13: $(0 == 0)$

= True

Lines/Branches (numbered by code line)

28: to line 29; statement 13 holds
29: if true to line 30 else to line 32; statement 12 holds
30: to line 33; statement 11 holds (same as 10)
32: to line 33; statement 10 holds
33: if true to line 37 else to line 39; statements 3,5,8, and 9 (loop equivalent) hold
37: to line 38; statements 2,5,7 hold
38: to line 33; statements 0 (postcondition), 1 (invarient), 4 (finite sequence), and 6 (strictly decresing) hold
39: statement 0 holds

Therefore, for my postcondition $True$ the weakest precondition is "requires true", which does verify when entered into the Dafny program.

4. **Statements**

0: $Q = (res == fact(n))$
1: $I = (2 \le i \le n + 1) \wedge (res == fact(i - 1))$
2: $(2 \le i + 1 \le n + 1) \wedge (res == fact(i))$
3: $(1 \le i \le n) \wedge (res \times i == fact(i))$
4: $wp(S,\ D \ge 0)$
5: $n - i - 1 \ge 0$ (simplify; no S) eq. To $n \ge i + 1$
6: $(old(D) \equiv n - i) \wedge (D < old(D))$
= $(old(D) \equiv n - i) \wedge (n - i < old(D))$
7: $(old(D) \equiv n - i) \wedge (n - i - 1 < old(D))$
8: $(n - i - 1 < n - i)$ = True
9: $wp(while\ (i \le n)\ I\ S,\ Q)$
= $I \wedge (\forall xs,\ I \wedge E \Rightarrow wp(S, I)) \wedge (\forall\ xs,\ I \wedge \neg E \Rightarrow Q) \wedge (\forall\ xs,\ I \wedge E \Rightarrow wp(S,\ D \ge 0)) \wedge (\forall\ xs,\ I \wedge E \Rightarrow wp(S,\ old(D) > D))$
= $(2 \le i \le n + 1) \wedge (res == fact(i - 1))$
$\wedge\ ((i \le n) \Rightarrow [(1 \le i \le n) \wedge (res \times i == fact(i))])$
$\wedge\ ((i > n) \Rightarrow [res == fact(n)])$
$\wedge\ ((i \le n) \Rightarrow (n > i + 1))$
$\wedge\ ((i \le n) \Rightarrow True)$ vacuous
= $(2 \le i \le n + 1)$
$\wedge\ (res == fact(i - 1))$
$\wedge\ ((i \le n) \Rightarrow (1 \le i \le n)$
$\wedge\ ((i \le n) \Rightarrow (res \times i == fact(i)))$
$\wedge\ ((i > n) \Rightarrow (res == fact(n)))$
$\wedge\ ((i \le n) \Rightarrow (i + 1 < n))$
10: $(2 \le 2 \le n + 1)$ eq. to n > 0 for integers
$\wedge\ (res == fact(1))$
$\wedge\ ((2 \le n) \Rightarrow (1 \le 2 \le n)$
$\wedge\ ((2 \le n) \Rightarrow (res \times 2 == fact(2)))$
$\wedge\ ((2 > n) \Rightarrow (res == fact(n)))$
$\wedge\ ((2 \le n) \Rightarrow (3 < n))$ True
11: $(n > 0)$
$\wedge\ (1 == fact(1))$ True
$\wedge\ ((2 \le n) \Rightarrow (2 \le n)$ True
$\wedge\ ((2 \le n) \Rightarrow (2 == fact(2)))$ Vacuously True
$\wedge\ ((2 > n) \Rightarrow (1 == fact(n)))$ Only non-vacuous case is n=1, where it is true
= $(n > 0)$

Lines/Branches (numbered by code line)

55: to line 56; statement 11 holds
56: to line 57; statement 10 holds
57: if true to line 63 else to line 66; statements 3,5,8, and 9 (loop equivalent) hold
63: to line 64; statements 2,5,7 hold
64: to line 57; statements 0 (post), 1(invariant), 4 (finite), and 6 (strictly dec) hold
66: statement 0 holds

Therefore, for this function to be the factorial function, the weakest precondition is $(n > 0)$.