

Musterlösung Aufgabe 3 Serie 4 Maximum im Array

```

max_el_1 = -inf; odd = 1;
max_el_2 = -inf; even = 2;
n = length(a);
{I1 ∧ I2}
co
  {I1}
  while (odd ≤ n) {
    {I1 ∧ odd ≤ n}
    max_el_1 = max{max_el_1, a[odd]};
    {max_el_1 == maxi ≤ odd, i is odd(a[i]) ∧ odd ≤ n ∧ odd is odd}
    odd = odd + 2;
    {I1 ∧ odd - 2 ≤ n}
  }
  {I1 ∧ odd - 2 ≤ n ∧ odd > n}
//
  {I2}
  while (even ≤ n) {
    {I2 ∧ even ≤ n}
    max_el_2 = max{max_el_2, a[even]};
    {max_el_2 == maxi ≤ even, i is even(a[i]) ∧ even ≤ n ∧ even is even}
    even = even + 2;
    {I2 ∧ even - 2 ≤ n}
  }
  {I1 ∧ even - 2 ≤ n ∧ even > n}
oc
{I1 ∧ I2 ∧ odd ∈ {n + 1, n + 2} ∧ even ∈ {n + 1, n + 2}}
return max{max_el_1, max_el_2}

I1 := {max_el_1 == maxi < odd, i is odd(a[i]) ∧ odd is odd}
I2 := {max_el_2 == maxi < even, i is even(a[i]) ∧ even is even}

I := {max{max_el_1, max_el_2} == maxi < min{even, odd}}

```

Wie leicht zu sehen ist, folgt I direkt aus $I_1 \wedge I_2$.
 Aus den Eigenschaften *even is even* und *odd is odd*, die zu jeder Zeit gelten,
 folgt, dass bis auf das Array a alle Variablen bezüglich der einzelnen Arme
 des *co*-Statements disjunkt sind.

Da das Array nur gelesen wird, gilt die Aussage über die Unabhängigkeit von parallelen Prozessen, da die Mengen der "Schreib"-Variablen der einzelnen Arme disjunkt ist, von den "Lese-" und "Schreib"-Variablen der jeweils anderen Arme.

Natürlich gilt dann auch die *At-Most-Once-Property*.

Das das Programm auch korrekt arbeitet folgt direkt aus I , da gilt:

$$(odd \in \{n + 1, n + 2\} \wedge even \in \{n + 1, n + 2\}) \Rightarrow n + 1 \in \{odd, even\}$$