Hoare's Logic: Axioms for arrays

- Assignment:

(an special case of simple assignment $\{\varphi[t/y]y:=t\{\varphi\}\}$)

$$\{P[A\{i \leftarrow t\}/A]\}\ A(i):=t\ \{P\}$$

- $A\{i \leftarrow t\}$ denotes the array identical to A, except that A(i) is t.

$$A\{i \leftarrow t\}(i) = t$$

$$i \neq j \Rightarrow A\{i \leftarrow t\}(j) = A(j)$$

Ex.1

- Proof the partial correctness of the following algorithm which reverses an array.

```
{n ≥ 0} = pre
i := 0
while (i < n)
      b[i] := a [n-i-1];
      i++;
{∀x ( 0 ≤ x ≤ n-1) => b[x] := a [n-x-1]} = post
```

Ex.1

- Proof the partial correctness of the following algorithm which reverses an array.

```
{n ≥ 0} = pre
i := 0
while (i < n)
      b[i] := a [n-i-1];
      i++;
{∀x ( 0 ≤ x ≤ n-1) => b[x] := a [n-x-1]} = post
```

Proof obligations:

```
1. {n ≥ 0} i :=0 {Inv}
2. {Inv} while … {Inv ∧ i ≥ n }
3. {Inv ∧ i ≥ n } => {post}
```

Ex.1

- Proof the partial correctness of the following algorithm which reverses an array.

```
{n ≥ 0} = pre
i := 0
while (i < n)
      b[i] := a [n-i-1];
      i++;
{∀x ( 0 ≤ x ≤ n-1) => b[x] := a [n-x-1]} = post
```

Proof obligations:

```
1. {n ≥ 0} i :=0 {Inv}
2. {Inv} while … {Inv ∧ i ≥ n }
3. {Inv ∧ i ≥ n } => {post}
```

Ex.1

1. `{n ≥ 0} i :=0 {Inv}`

Ex.1

1. {n ≥ 0} i :=0 {Inv}

Inv = {∀x (0 ≤ x ≤ i-1) => b[x] = a [n-x-1] ∧ i ≤ n }

Ex.1

1. $\{n \geq 0\}$ i :=0 {Inv}

Inv = $\{\forall x\ (0 \leq x \leq i-1) \Rightarrow b[x] = a[n-x-1] \wedge i \leq n\}$

$\{n \geq 0\}$ i :=0 $\{\forall x\ (0 \leq x \leq i-1) \Rightarrow b[x] = a[n-x-1] \wedge i \leq n\}$

Ex.1

1. $\{n \geq 0\}$ `i :=0` $\{Inv\}$

$Inv = \{\forall x\ (0 \leq x \leq i-1) \Rightarrow b[x] = a[n-x-1] \wedge i \leq n\}$

$\{n \geq 0\}$ `i :=0` $\{\forall x\ (0 \leq x \leq i-1) \Rightarrow b[x] = a[n-x-1] \wedge i \leq n\}$

Ex.1

1. $\{n \geq 0\}$ i :=0 $\{Inv\}$

$Inv = \{\forall x \ (\ 0 \leq x \leq i-1) \Rightarrow b[x] = a\ [n-x-1] \land i \leq n \}$

$\{n \geq 0\}$ i :=0 $\{\forall x \ (\ 0 \leq x \leq i-1) \Rightarrow b[x] = a\ [n-x-1] \land i \leq n \}$

$\{n \geq 0\} \Rightarrow \{\forall x \ (\ 0 \leq x \leq -1) \Rightarrow b[x] = a\ [n-x-1] \land 0 \leq n \}$

Ex.1

1. $\{n \geq 0\}\ i := 0\ \{Inv\}$

$Inv = \{\forall x\ (0 \leq x \leq i-1) \Rightarrow b[x] = a[n-x-1] \wedge i \leq n\}$

$\{n \geq 0\}\ i := 0\ \{\forall x\ (0 \leq x \leq i-1) \Rightarrow b[x] = a[n-x-1] \wedge i \leq n\}$

$\{n \geq 0\} \Rightarrow \{\forall x\ (\underbrace{0 \leq x \leq -1}_{\text{false}}) \Rightarrow b[x] = a[n-x-1] \wedge 0 \leq n\}$

Ex.1

1. $\{n \ge 0\}$ i :=0 $\{Inv\}$

$Inv = \{\forall x\ (\ 0 \le x \le i-1) \Rightarrow b[x] = a\ [n-x-1] \wedge i \le n\ \}$

$\{n \ge 0\}$ i :=0 $\{\forall x\ (\ 0 \le x \le i-1) \Rightarrow b[x] = a\ [n-x-1] \wedge i \le n\ \}$

$\{n \ge 0\} \Rightarrow \{\forall x\ (\ 0 \le x \le -1) \Rightarrow b[x] = a\ [n-x-1] \wedge 0 \le n\ \}$

true

Ex.1

1. {n ≥ 0} i :=0 {Inv}

Inv = {∀x (0 ≤ x ≤ i-1) => b[x] = a [n-x-1] ∧ i ≤ n }

{n ≥ 0} i :=0 {∀x (0 ≤ x ≤ i-1) => b[x] = a [n-x-1] ∧ i ≤ n }

{n ≥ 0} => {∀x (0 ≤ x ≤ -1) => b[x] = a [n-x-1] ∧ 0 ≤ n }

{n ≥ 0} => {n ≥ 0}

Ex.1

- Proof the partial correctness of the following algorithm which reverses an array.

```
{n ≥ 0} = pre
i := 0
while (i < n)
      b[i] := a [n-i-1];
      i++;
{∀x ( 0 ≤ x ≤ n-1) => b[x] := a [n-x-1]} = post
```

Proof obligations:

```
1. {n ≥ 0} i :=0 {Inv}
2. {Inv} while … {Inv ∧ i ≥ n }
3. {Inv ∧ i ≥ n } => {post}
```

Ex.1

2. `{Inv} while … {Inv ∧ i ≥ n }`

Ex.1

2. {Inv} while … {Inv ∧ i ≥ n }

```
{Inv ∧ i < n} b[i] := a [n-i-1]; i++; {Inv}
```

Ex.1

2. `{Inv} while … {Inv ∧ i ≥ n }`

   `{Inv ∧ i < n} b[i] := a [n-i-1]; i++; {Inv}`

Ex.1

2.
```
{Inv} while … {Inv ∧ i ≥ n }
{Inv ∧ i < n} b[i] := a [n-i-1]; i++; {Inv}
```

```
{Inv ∧ i < n} b[i] := a [n-i-1]; {∀x ( 0 ≤ x ≤ i)
=> b[x] = a [n-x-1] ∧ i+1 ≤ n}
```

Ex.1

2.
```
{Inv} while … {Inv ∧ i ≥ n }
{Inv ∧ i < n} b[i] := a [n-i-1]; i++; {Inv}
```

```
{Inv ∧ i < n} b[i] := a [n-i-1]; {∀x ( 0 ≤ x ≤ i)
=> b[x] = a [n-x-1] ∧ i+1 ≤ n}
```

```
{Inv ∧ i < n} b[i] := a [n-i-1]; {∀x ( 0 ≤ x < i)
=> b[x] = a [n-x-1] ∧ i < n ∧ b[i] = a[n-i-1]}
```

Ex.1

2. {Inv} while … {Inv ∧ i ≥ n }

   {Inv ∧ i < n} b[i] := a [n-i-1]; i++; {Inv}

{Inv ∧ i < n} b[i] := a [n-i-1]; {∀x (0 ≤ x ≤ i) => b[x] = a [n-x-1] ∧ i+1 ≤ n}

{Inv ∧ i < n} b[i] := a [n-i-1]; {∀x (0 ≤ x < i) => b[x] = a [n-x-1] ∧ i < n ∧ b[i] = a[n-i-1]}

Ex.1

2. {Inv} while … {Inv ∧ i ≥ n }

{Inv ∧ i < n} b[i] := a [n-i-1]; i++; {Inv}

{Inv ∧ i < n} b[i] := a [n-i-1]; {∀x (0 ≤ x ≤ i) => b[x] = a [n-x-1] ∧ i+1 ≤ n}

{Inv ∧ i < n} b[i] := a [n-i-1]; {∀x (0 ≤ x < i) => b[x] = a [n-x-1] ∧ i < n ∧ b[i] = a[n-i-1]}

{Inv ∧ i < n} => {∀x (0 ≤ x < i) => b[x] = a [n-x-1] ∧ i < n ∧ a[n-i-1] = a[n-i-1]}

Ex.1

2. $\{Inv\}$ while … $\{Inv \wedge i \geq n\}$

   $\{Inv \wedge i < n\}$ b[i] := a [n-i-1]; i++; $\{Inv\}$

$\{Inv \wedge i < n\}$ b[i] := a [n-i-1]; $\{\forall x\ (0 \leq x \leq i) \Rightarrow b[x] = a[n-x-1] \wedge i+1 \leq n\}$

$\{Inv \wedge i < n\}$ b[i] := a [n-i-1]; $\{\forall x\ (0 \leq x < i) \Rightarrow b[x] = a[n-x-1] \wedge i < n \wedge b[i] = a[n-i-1]\}$

$\{Inv \wedge i < n\} \Rightarrow \{\forall x\ (0 \leq x < i) \Rightarrow b[x] = a[n-x-1] \wedge i < n\}$

Ex.1

2. {Inv} while … {Inv ∧ i ≥ n }

   {Inv ∧ i < n} b[i] := a [n-i-1]; i++; {Inv}

{Inv ∧ i < n} b[i] := a [n-i-1]; {∀x (0 ≤ x ≤ i) => b[x] = a [n-x-1] ∧ i+1 ≤ n}

{Inv ∧ i < n} b[i] := a [n-i-1]; {∀x (0 ≤ x < i) => b[x] = a [n-x-1] ∧ i < n ∧ b[i] = a[n-i-1]}

{Inv ∧ i < n} => {∀x (0 ≤ x < i) => b[x] = a [n-x-1] ∧ i < n }   equal sides

Ex.1

- Proof the partial correctness of the following algorithm which reverses an array.

```
{n ≥ 0} = pre
i := 0
while (i < n)
      b[i] := a [n-i-1];
      i++;
{∀x ( 0 ≤ x ≤ n-1) => b[x] := a [n-x-1]} = post
```

Proof obligations:

```
1. {n ≥ 0} i :=0 {Inv}
2. {Inv} while … {Inv ∧ i ≥ n }
3. {Inv ∧ i ≥ n } => {post}
```

Ex.1

3. `{Inv ∧ i ≥ n } => {post}`

Ex.1

3. $\{Inv \wedge i \geq n\} \Rightarrow \{post\}$

$Inv \wedge i \geq n = \{\forall x\ (0 \leq x \leq i-1) \Rightarrow b[x] = a[n-x-1] \wedge i \leq n \wedge i \geq n\}$

Ex.1

3. $\{Inv \land i \geq n\} \Rightarrow \{post\}$

$Inv \land i \geq n = \{\forall x\, (0 \leq x \leq i-1) \Rightarrow b[x] = a[n-x-1] \land \underbrace{i \leq n \land i \geq n}_{i = n}\}$

Ex.1

3. $\{Inv \land i \geq n\} \Rightarrow \{post\}$

$Inv \land i \geq n = \{\forall x\ (0 \leq x \leq i-1) \Rightarrow b[x] = a[n-x-1] \land \underbrace{i \leq n \land i \geq n}_{i = n}\}$

$Inv \land i \geq n = \{\forall x\ (0 \leq x \leq n-1) \Rightarrow b[x] = a[n-x-1]\}$

Ex.1

3. $\{Inv \wedge i \geq n\} \Rightarrow \{post\}$

$Inv \wedge i \geq n = \{\forall x\ (0 \leq x \leq i-1) \Rightarrow b[x] = a[n-x-1] \wedge \underbrace{i \leq n \wedge i \geq n}_{i = n}\}$

$Inv \wedge i \geq n = \{\forall x\ (0 \leq x \leq n-1) \Rightarrow b[x] = a[n-x-1]\}$

which is the same as `post`