Formulating Natural Language in LTL (2.3)

Natural Language:

Whenever a process makes a request, it starts waiting. As soon as no other process is in the critical region, the process is granted access to the critical region.

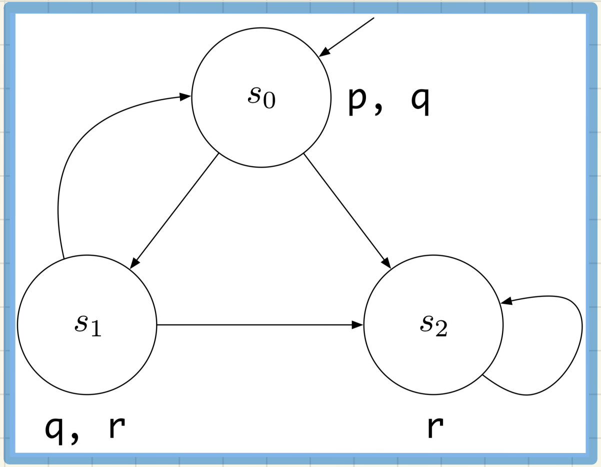
Assumed atoms:

- requested
- waiting
- granted
- noOneInCs

LTL Formulation

Q. Is starvation freedom guaranteed?

Model Satisfaction: Exercises (7.1)



$s \models \phi \Leftrightarrow$ all π starting at s, $\pi \models \phi$

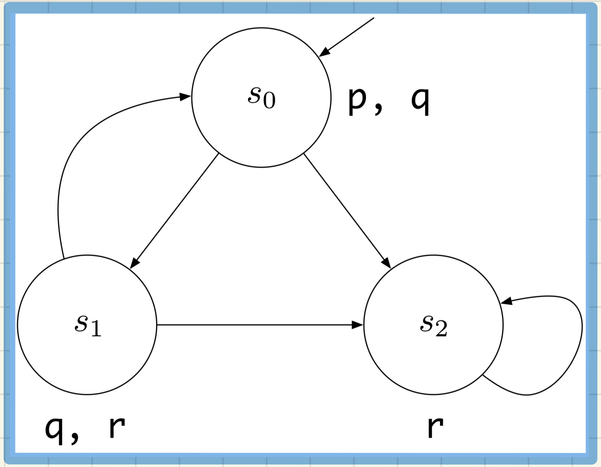
$s_0 \models p \ \mathbf{U} \ r$

$s_0 \models p \ \mathbf{W} \ r$

$s_0 \models r \ \mathbf{R} \ p$

Exercise: What if we change the LHS to s_2?

Model Satisfaction: Exercises (7.2)



$s \models \phi \Leftrightarrow$ all π starting at s, $\pi \models \phi$

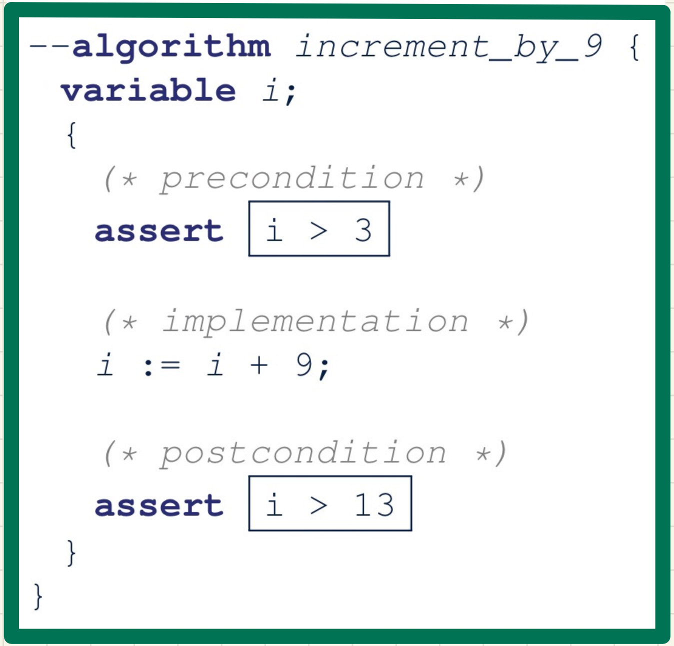
$s_0 \models (p \lor r) \; \mathbf{U} \; (p \land r)$

$s_0 \models (p \lor r) \; \mathbf{W} \; (p \land r)$

$s_0 \models (p \land r) \; \mathbf{R} \; (p \lor r)$

Exercise: What if we change the LHS to s_2?

Program Correctness: Example (1)

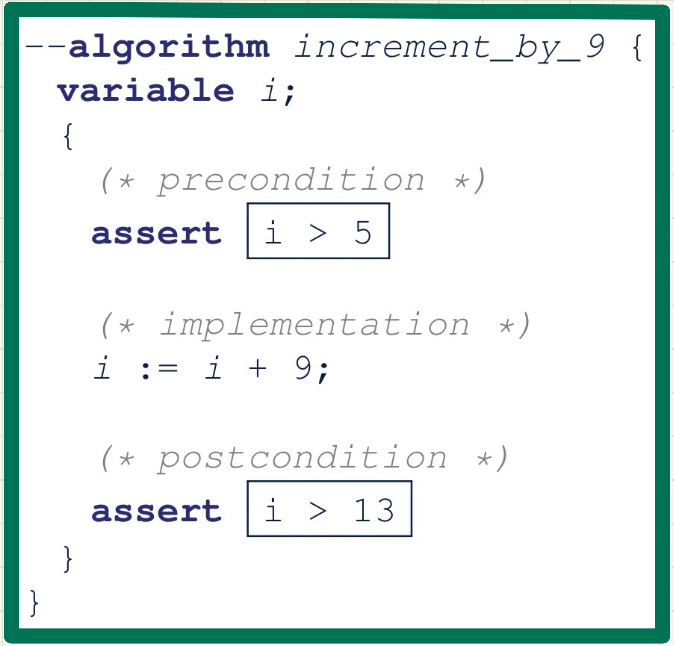


```
--algorithm increment_by_9 {
  variable i;
  {
    (* precondition *)
    assert i > 3

    (* implementation *)
    i := i + 9;

    (* postcondition *)
    assert i > 13
  }
}
```

Program Correctness: Example (2)



```
--algorithm increment_by_9 {
  variable i;
  {
    (* precondition *)
    assert i > 5

    (* implementation *)
    i := i + 9;

    (* postcondition *)
    assert i > 13
  }
}
```

Hoare Triple as a Predicate

$$\boxed{\{Q\}\ \text{S}\ \{R\} \equiv Q \Rightarrow wp(S, R)}$$