# Introduction to Software Verification

Vincent Penelle

`<vpenelle@u-bordeaux.fr>`

LaBRI, Université de Bordeaux

September 16, 2019

*— Models and Properties —*

# Example of abstract models

Give examples:

- Finite Automata.
- Turing Machine.
- Pushdown Automata.
- Counter Systems.
- Vector Addition Systems

# Interesting Properties

Explain and put examples:

- Reachability.
- Model-checking.
- Liveness.
- Fairness.
- etc.

# Reachability

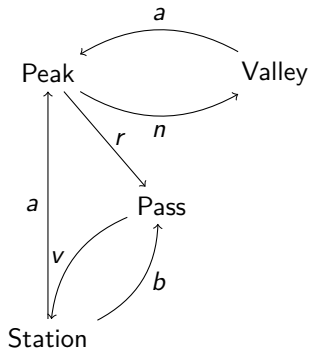**Input:** A graph $\mathcal{G}$ and two vertices $x, y$.

**Output:** If there exists a path between $x$ and $y$.

# Reachability

**Input:** A graph $\mathcal{G}$ and two vertices $x, y$.
**Output:** If there exists a path between $x$ and $y$.

Example: Is it possible to go from the valley to the station?

# Reachability

**Input:** A graph $\mathcal{G}$ and two vertices $x, y$.

**Output:** If there exists a path between $x$ and $y$.

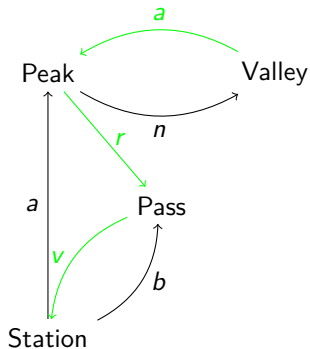Example: Is it possible to go from the valley to the station? Yes.

# Model-checking

**Input:** A graph $\mathcal{G}$ and a logic formula $\phi$.
**Output:** If $\mathcal{G} \models \phi$.

- First Order: $x \xrightarrow{a} y$, $\exists x$, $\forall x$
- First Order with reachability: $x \xrightarrow{*} y$
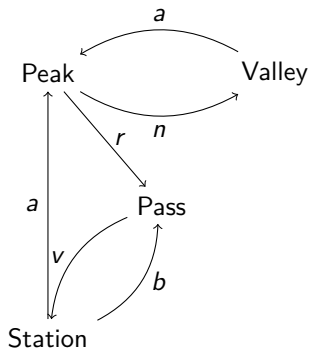- Monadic Second Order: $x \in X$, $\exists X$, $\forall X$

# Model-checking

**Input:** A graph $\mathcal{G}$ and a logic formula $\phi$.
**Output:** If $\mathcal{G} \models \phi$.

Example: Is it possible to leave the valley without taking any ski lift?

$$\phi = \exists x, \left(\text{Valley} \xrightarrow{v} x \vee \text{Valley} \xrightarrow{n} x \vee \text{Valley} \xrightarrow{r} x\right) \wedge x \neq \text{Valley}$$

**Input:** A graph $\mathcal{G}$ and a logic formula $\phi$.

**Output:** If $\mathcal{G} \models \phi$.

Example: Is it possible to leave the valley without taking any ski lift? No.

$$\phi = \exists x, \left(\text{Valley} \xrightarrow{v} x \vee \text{Valley} \xrightarrow{n} x \vee \text{Valley} \xrightarrow{r} x\right) \wedge x \neq \text{Valley}$$
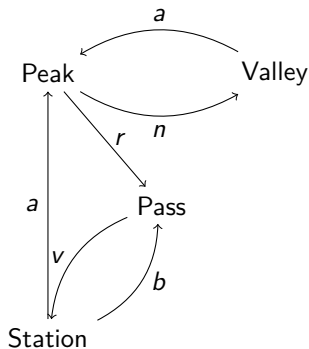
## Safety

Nothing bad happens (bad states not accessible).

# Liveness (concurrent programs)

Something good eventually happens. What to show ?