

Computation Tree

Let $\mathbb{S} = (\mathcal{S}, In, T, \mathcal{X}, dom, L)$ be a transition system and $s \in \mathcal{S}$ be a state. The **computation tree for \mathbb{S} starting at s** is the following (possibly infinite) tree.

1. The **nodes** of the tree are labeled by states in \mathcal{S} .
2. The **root** of the tree is labeled by s .
3. For every node s' in the tree, its **children** are exactly such nodes $s'' \in \mathcal{S}$ that $(s', s'') \in T$.

A **computation path** for \mathbb{S} : any branch s_0, s_1, \dots in the tree.

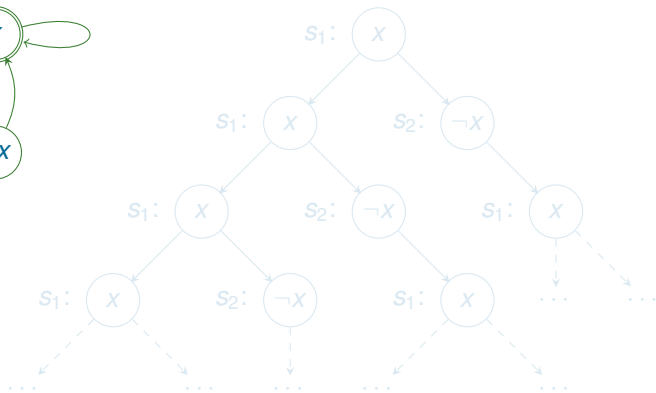
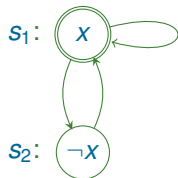
Computation Tree

Let $\mathbb{S} = (\mathcal{S}, In, T, \mathcal{X}, dom, L)$ be a transition system and $s \in \mathcal{S}$ be a state. The **computation tree for \mathbb{S} starting at s** is the following (possibly infinite) tree.

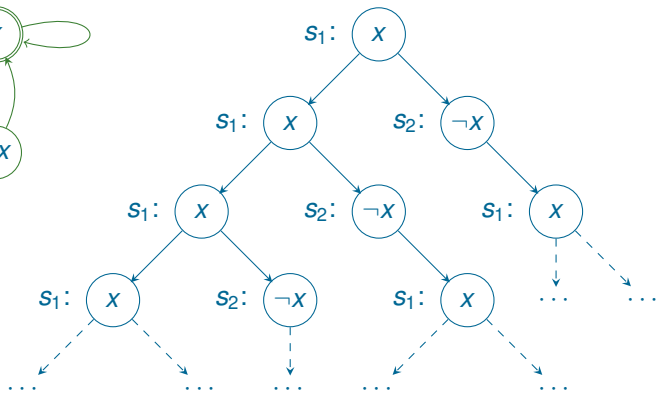
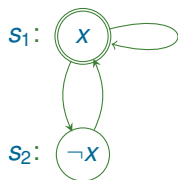
1. The **nodes** of the tree are labeled by states in \mathcal{S} .
2. The **root** of the tree is labeled by s .
3. For every node s' in the tree, its **children** are exactly such nodes $s'' \in \mathcal{S}$ that $(s', s'') \in T$.

A **computation path** for \mathbb{S} : any branch s_0, s_1, \dots in the tree.

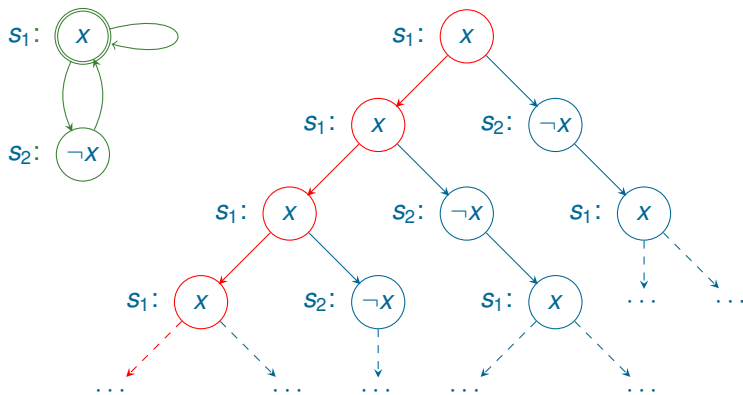
Computation Trees and Paths



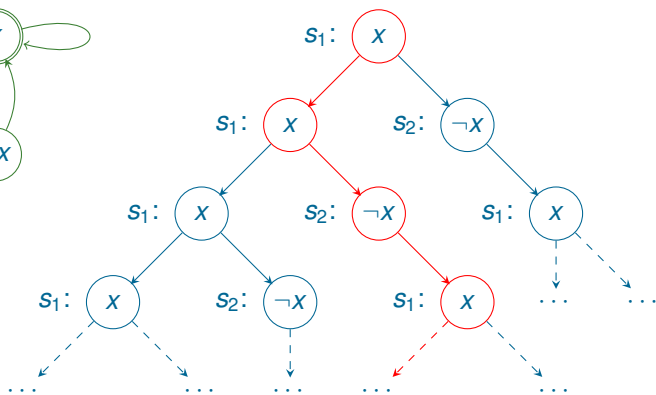
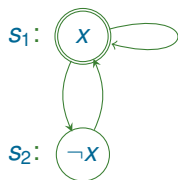
Computation Trees and Paths



Computation Trees and Paths



Computation Trees and Paths



Properties

- ▶ **Computation paths** for a transition system are **exactly all branches** in the computation trees for this transition system.
- ▶ Let n be a node in a computation tree C for \mathbb{S} labeled by s' . Then the **subtree of C rooted at s' is the computation tree for \mathbb{S} starting at s'** . In other words, every subtree of a computation tree rooted at some node is itself a computation tree.
- ▶ For every transition system \mathbb{S} and state s there exists a **unique computation tree** for \mathbb{S} starting at s , up to the order of children.

Properties

- ▶ **Computation paths** for a transition system are **exactly all branches** in the computation trees for this transition system.
- ▶ Let n be a node in a computation tree C for \mathbb{S} labeled by s' . Then the **subtree of C rooted at s' is the computation tree for \mathbb{S} starting at s'** . In other words, every subtree of a computation tree rooted at some node is itself a computation tree.
- ▶ For every transition system \mathbb{S} and state s there exists a **unique computation tree** for \mathbb{S} starting at s , up to the order of children.

Properties

- ▶ **Computation paths** for a transition system are **exactly all branches** in the computation trees for this transition system.
- ▶ Let n be a node in a computation tree C for \mathbb{S} labeled by s' . Then the **subtree of C rooted at s' is the computation tree for \mathbb{S} starting at s'** . In other words, every subtree of a computation tree rooted at some node is itself a computation tree.
- ▶ For every transition system \mathbb{S} and state s there exists a **unique computation tree** for \mathbb{S} starting at s , up to the order of children.

Linear Temporal Logic is a logic for reasoning about properties of computation paths.

Formulas are built in the same way as in propositional logic, with the following additions:

1. If F is a formula, then $\bigcirc F$, $\square F$, and $\diamond F$ are formulas;
2. If F and G are formulas, then $F \cup G$ and $F R G$ are formulas.

Linear Temporal Logic is a logic for reasoning about properties of computation paths.

Formulas are built in the same way as in propositional logic, with the following additions:

1. If F is a formula, then $\bigcirc F$, $\square F$, and $\diamond F$ are formulas;
2. If F and G are formulas, then $F \cup G$ and $F \mathcal{R} G$ are formulas.

Semantics (intuitive)

