### Formal Syntax of CTL*

**State formulas**
- Every atomic proposition is a state formula.
- If φ and ψ are state formulas then so are ¬φ, φ ∧ ψ, φ ∨ ψ, φ → ψ, φ ↔ ψ.
- If φ is a path formula, then Aφ and Eφ are state formulas.

**Path formulas**
- Every state formula is a path formula.
- If φ and ψ are path formulas, then so are Xφ, Fφ, Gφ, φ U ψ.

### PLTL and CTL in NuSMV

```
-- CTL
SPEC
  AG (request -> AF grant)

-- PLTL
LTLSPEC
  G (request -> F grant)
```

### Three Temporal Logics

**CTL**
- Every state formula is a CTL*-formula.

**CTL**
- Every state formula in which temporal combinators are applied to state formulas only is a CTL-formula.

**PLTL**
- Every path formula without any path quantifier is a PLTL-formula.

### Semantics of CTL*

**Given**
- Kripke structure $K = (Q, Prop, T, q_0, I)$,
- CTL*-formula $\varphi$,
- we want to define under which conditions the formula $\varphi$ is true in a state $q \in Q$. 

**Examples**

<table>
<thead>
<tr>
<th>CTL*</th>
<th>CTL</th>
<th>PLTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>AG (warm ⇒ AF ¬warm)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AG (warm ⇒ F ¬warm)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G (warm ⇒ X ¬warm)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A (F warm ⇒ F ok)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A (ok U error)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G (warm ⇒ EF ¬warm)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Defining the Semantics

- The truth of a state formula depends on the state
  - \neg \text{train1 on A} \lor \neg \text{train2 on A}
  - AG \neg \text{collision}
- The truth of a path formula depends on the execution path
  - GF \text{req}

Path Formulas

Given:
- Path formula \( \phi \)
  - \( G R_A \land F P_A \land G \neg (P_A \land P_B) \)
- Execution path \( \sigma = \langle \sigma_0, \sigma_1, \ldots \rangle \)

To determine whether \( \phi \) is true in \( \sigma \):
- Determine truth of subformulas.
- Evaluate the temporal combinator for \( \phi \).

Example Kripke Structure

```
1 \ R_A R_B
2 \ W_A R_B
3 \ R_A W_B
4 \ W_A P_B
5 \ P_A W_B
6 \ W_A P_B
7 \ P_A W_B
8 \ R_A P_B
```

Evaluating Temporal Combinators

- X \( \phi \) is true on a path \( \sigma \), if \( \sigma \) has more than one state, and \( \phi \) is true in the second state of \( \sigma \).
- G \( \phi \) is true on a path \( \sigma \), if \( \phi \) is true in every state of \( \sigma \).
- F \( \phi \) is true on a path \( \sigma \), if \( \sigma \) contains a state where \( \phi \) is true.
- \( \phi \lor \psi \) is true on a path \( \sigma \), if \( \sigma \) contains a state \( q \) where \( \psi \) is true, and \( \phi \) is true in every state in \( \sigma \) up to but not necessarily including \( q \).

Propositional Formulas

Given:
- Formula \( \phi \) without temporal combinators
  - \( R_A \land P_A \land P_B \land W_A \Rightarrow P_A \)
- State \( q \in Q \)

To determine whether \( \phi \) is true in \( q \):
- Inspect the labels of state \( q \).

Path Quantifiers

Given:
- State formula \( \phi \) with a path quantifier
  - \( EG R_A \land EF P_A \land AG \neg (P_A \land P_B) \)
- State \( q \in Q \)

To determine whether \( \phi \) is true in \( q \):
- Determine truth of subformulas.
- Evaluate the path quantifier.
Evaluating Path Quantifiers

- $A\varphi$ is true in state $q$, if $\varphi$ is true on every maximal path starting from $q$.
- $E\varphi$ is true in state $q$, if $\varphi$ is true on some maximal path starting from $q$.

**Note:**
- Execution paths must be maximal.
- A finite execution path can only be considered if it ends in a state without outgoing transitions.

More Rules

- $A \varphi \equiv \neg E \neg \varphi$
- $AG \varphi \equiv \neg EF \neg \varphi$
- $AF \varphi \equiv \neg EG \neg \varphi$
- $EG \varphi \equiv \neg AF \neg \varphi$
- $EF \varphi \equiv \neg AG \neg \varphi$

At Last ...

**Given:**
- Path formula $\varphi$
- Kripke-structure $K$

**We say:**
- $\varphi$ is **satisfied in $K$**, if $\varphi$ is true for every maximal execution of $K$.

Reading

Bérard et. al.:
Chapter 2 – Temporal Logic
Chapter 12 – NuSMV

Some Rules

- $G \varphi \equiv \neg F \neg \varphi$
- $F \varphi \equiv \text{true} U \varphi$
- $\varphi W \psi \equiv (\varphi U \psi) \lor G \varphi$

All path formulas can be written using the combinators $U$ and $X$ only.