SPECIFYING PROPERTIES USING
TEMPORAL LOGIC

1. CTL vs. LTL
2. Examples of temporal properties
3. Requirement specifications
4. Fairness properties and CTL

E. M. Clarke et al.: Model Checking, Chapter 3 (pp. 27–33).

CTL Computation Tree

- For a model $M = (S, R, v)$ and a state $s_0 \in S$, the computation tree $\hat{M} = (\hat{S}, \hat{R}, \hat{v})$, starting from $s_0$ is constructed as follows:
  (i) start with the node $\langle s_0, 0 \rangle$.
  (ii) unfold the model using the rule:
  
  if $\langle s, n \rangle \in \hat{S}$ and $sRt$, then $\langle t, m \rangle \in \hat{S}$ and $\langle \langle s, n \rangle, \langle t, m \rangle \rangle \in \hat{R}$
  where $m$ is a new number not used before.
  (iii) The valuation $\hat{v}$ is given by $\hat{v}(\langle s, n \rangle, P) = v(s, P)$ for all $\langle s, n \rangle$.

- Now temporal formulas are evaluated on $\hat{M}$ as follows:
  - $\hat{M}, s_0 \models A(\hat{P}U\hat{Q})$ iff for all branches of the computation tree $\hat{M}$ ($s_0, s_1, \ldots$) there is some $i \geq 0$ such that $\hat{M}, s_i \models Q$ and $\hat{M}, s_j \models P$ for all $0 \leq j < i$.
  - $\hat{M}, s_0 \models E(\hat{P}U\hat{Q})$ iff there is some branch of $\hat{M}$ ($s_0, s_1, \ldots$) and some $i \geq 0$ such that $\hat{M}, s_i \models Q$ and $\hat{M}, s_j \models P$ for all $0 \leq j < i$.

Comparing CTL and LTL formulas

- For model checking consider the correspondence

  CTL: $\hat{M}, s_0 \models P$  \hspace{1cm} LTL: $\hat{M}, x \models P$ for all full paths $x = (s_0, \ldots)$.

- CTL and LTL operators are similar but differ in some respects.

- For instance, “temporal possibilities” can be expressed in CTL but not in LTL.

Example. For a CTL formula $AGEFp$, there is no corresponding LTL formula.

Consider LTL formula $GFp$ and the model $\hat{M} = (S, R, v)$ where $S = \{s_0, s_1\}$, $R = \{\langle s_0, s_0 \rangle, \langle s_0, s_1 \rangle, \langle s_1, s_0 \rangle\}$, $v(s_0, P) = \text{false}$ and $v(s_1, P) = \text{true}$.

The formula $GFp$ is not valid in $\hat{M}$ because it is false in a full path $(s_0, s_0, s_0, \ldots)$ although CTL formula $AGEFp$ is valid in $\hat{M}$. 
Differences between CTL and LTL—cont’d

- Thus, a CTL formula of the type “there is a path …” is not expressible as an LTL formula.

**Example.** For a CTL formula $\text{EF}P$ there is no corresponding LTL formula.

For instance, the LTL formula $\text{FP}$ is not valid in the previous model $\mathcal{M}$ because it is false in a full path $(s_0, s_0, s_0, \ldots)$ but the CTL formula $\text{EFP}$ is valid.

- Fairness properties are not expressible in CTL.

**Example.** For an LTL formula $\text{FGQ}$ there is no corresponding CTL formula.

Consider the previous model $\mathcal{M}$ where we set $v(s_0, Q) = \text{true}$ and $v(s_1, Q) = \text{false}$.

Now $\text{FGQ}$ is true in a full path $(s_0, s_0, \ldots)$ but the CTL formula $\text{AFAFGQ}$ is not satisfiable in $\mathcal{M}$ and neither is $\text{EFAGQ}$.

---

2. Examples of Temporal Properties

- $\text{EF}(\text{started} \land \neg \text{ready})$:
  
  It is possible to reach a state where $\text{started}$ is true but $\text{ready}$ is not.

- $\text{AG}(%\text{req} \rightarrow \text{AFack})$:
  
  If a request is received then it will be acknowledged.

- $\text{AGAFenabled}$:
  
  $\text{enabled}$ is true infinitely often on every computation path.

- $\text{AGEFrestart}$:
  
  From every state is it possible to reach a state where $\text{restart}$ is true.

---

3. Requirement Specifications

- Temporal logic can be used to state requirement specifications for reactive systems.

- Typical requirement specifications can be divided into the following classes:
  1. Reachability properties
  2. Safety properties
  3. Liveness properties
  4. Fairness properties

---

Reachability Properties

- This is a simple class of properties stating that some state (where a given condition $P$ is true) can be reached (from the initial state of the system).

- Can be expressed using temporal formulas of the form $\text{EFP}$.

- Conditional reachability can be expressed using temporal formulas of the form $\text{E}(\text{QUP})$ (there is an execution where $Q$ is true reaching a state where $P$ is true).

**Example.** Typical reachability properties:

1. $\text{EF}(\text{started} \land \neg \text{ready})$.
2. $\text{EF}(\text{restart})$.
3. $\text{E}(\neg \text{restartUready})$. 
### Safety Properties

- Safety properties state that nothing "bad" happens during an execution of the system.
- A safety property is a requirement which has a finite counter-execution:
  - if the system does not satisfy a safety property $P$, then it has a finite execution where the property $P$ does not hold.

**Example.** Examples of typical safety properties:

1. Mutual exclusion: $\text{AG} \neg(\text{atCS}_1 \land \text{atCS}_2)$.
2. Partial correctness: $\text{atl}_0 \land P \rightarrow \text{AG}(\text{atl}_h \land Q)$.

### Liveness Properties

- Liveness properties express that something "good" happens.
- Liveness properties do not have finite counter-executions:
  - if a system does not satisfy a liveness property $P$, then this can be demonstrated only using an infinite counter-execution.

**Example.** Typical examples of liveness properties:

1. (nested) reachability: $\text{AGFrestart}$.
2. Temporal implication: $\text{AG}(P \rightarrow \text{AF}Q)$.
3. Starvation freeness: $\text{AG}(\text{atTry}_i \rightarrow \text{AFatCS}_i)$.
4. Total correctness: $\text{atl}_0 \land P \rightarrow \text{AF}(\text{atl}_h \land Q)$.

### Fairness Properties

- Fairness properties are liveness properties which require that states where a given condition is true occur infinitely often.
- Fairness properties are not directly expressible in CTL but they are in LTL.

**Example.** Consider two atomic propositions for a process:
- $\text{en}$ (the process is enabled) and
- $\text{ex}$ (the process is executed).

1. Unconditional fairness: $\text{GFex}$.
2. Strong fairness: $\text{GFen} \rightarrow \text{GFex}$.
3. Weak fairness: $\text{FGen} \rightarrow \text{GFex}$.

### 4. Fairness Properties and CTL

- When using CTL fairness properties are handled by modifying the semantics of the path quantifiers ($\text{A/E}$).
- Quantification is considered over all fair paths (and not over all paths as in the basic case).
- Fairness conditions are given as a set of formulas $F$ and when evaluating the truth of a formula only $F$-fair paths are considered.

**Definition.** A full path $x$ is $F$-fair iff every $P \in F$ is true infinitely often on the path $x$. 

Modified Semantics

Relation $|=F$ is defined as $|=_{F}$ except that path quantification is over $F$-fair paths.

- $M, s |=_F P$ iff there is a $F$-fair full path starting from the state $s$ and $v(s, P) = \text{true}$ when $P$ is an atomic proposition.
- $M, s |=_F A(P \cup Q)$ iff for all $F$-fair full path $(s_0, s_1, \ldots)$ where $s = s_0$ there is some $i \geq 0$ such that $M, s_i |=_F Q$ and $M, s_j |=_F P$ for all $0 \leq j < i$.
- $M, s |=_F E(P \cup Q)$ iff there is some $F$-fair full path $(s_0, s_1, \ldots)$ with $s = s_0$ and there is some $i \geq 0$ such that $M, s_i |=_F Q$ and $M, s_j |=_F P$ for all $0 \leq j < i$.

Example. Unconditional fairness can be express using the set $F = \{ \text{ex} \}$ and a fair channel using a set $F = \{ \text{send} \rightarrow \text{rec} \}$.

Summary

- Although CTL and LTL are based on similar temporal operators, they are different because LTL is a linear time logic where formulas are evaluated on paths whereas CTL is a branching time logic where formulas are evaluated on computation trees.
- Hence, there are CTL formulas (for instance of the form “there is a path . . .”) which cannot be expressed in LTL and LTL formulas (for example fairness formulas) which cannot be expressed in CTL.
- Temporal logics are suitable for requirement specification of reactive systems.
- Typical requirement specifications include reachability properties, safety properties, liveness properties, and fairness properties.