Linear Time Properties

- A linear (time) property talks about the behavior of a system during an execution.
- What is an execution? It is a (finite or infinite) sequence of states. In this context we call it a behavior.
- Recall: a computation of a transition system \((\langle S, \rightarrow \rangle, S^0)\) is a finite or infinite sequence of states,
  \[
  s^0 \rightarrow s^1 \rightarrow \ldots \rightarrow s^n \rightarrow \ldots
  \]
such that
  - \(s^0 \in S\), i.e., the computation starts in an initial state.
  - For each \(n \geq 1\), such that \(n + 1\) is not more than the number of states in the computation, \(s^{n-1} \rightarrow s^n\), i.e., each pair of consecutive states is a transition.
- From now on, it will be convenient to consider only infinite computations.
- We can include finite computations
  \[
  s^0 \rightarrow s^1 \rightarrow \ldots \rightarrow s^n
  \]
e.g., by repeating the last state for ever.
  \[
  s^0 \rightarrow s^1 \rightarrow \ldots \rightarrow s^n \rightarrow s^n \rightarrow \ldots
  \]

Example of Temporal Properties

- Let \(p\) be a set of states. The property
  \(p\) is invariant
  (when it is a linear-time property) denotes the set of behaviors
  \[
  s^0 \rightarrow s^1 \rightarrow \ldots \rightarrow s^n \rightarrow \ldots
  \]
such that \(s^n \in p\) for all \(n\).
- The property
  \(p\) is stable
  denotes the set of behaviors
  \[
  s^0 \rightarrow s^1 \rightarrow \ldots \rightarrow s^n \rightarrow \ldots
  \]
such that if \(s^n \in p\) for some \(n\), then \(s^m \in p\) for all \(m \geq n\).
- Let \(p\) and \(q\) be sets of states. The property
  \(p\) precedes \(q\)
  denotes the set of behaviors
  \[
  s^0 \rightarrow s^1 \rightarrow \ldots \rightarrow s^n \rightarrow \ldots
  \]
such that if \(s^n \in q\) for some \(n\), then there is an \(m < n\) such that \(s^m \in p\).
- The property
  \(p\) leads to \(q\)
  denotes the set of behaviors
  \[
  s^0 \rightarrow s^1 \rightarrow \ldots \rightarrow s^n \rightarrow \ldots
  \]
such that if \(s^n \in p\) for some \(n\), then there is an \(m \geq n\) such that \(s^m \in q\).

Summary of Definitions

- Assume a set \(S\) of states. A behavior \(\overline{s}\) over \(S\) is an infinite sequence of elements in \(S\)
  \[
  s^0 \rightarrow s^1 \rightarrow \ldots \rightarrow s^n \rightarrow \ldots
  \]
- A finite computation can be made into a behavior by repeating the last state an infinite number of times.
- Assume a set \(S\) of states. A behavior \(\overline{s}\) over \(S\) is an infinite sequence of elements in \(S\)
- A (linear time) temporal property \(\phi\) over \(S\) is a set of behaviors over \(S\).
- A behavior \(\beta\) satisfies a temporal property \(\phi\), written \(\beta \models \phi\) iff \(\beta \in \phi\).
- A transition system \(T\) satisfies a correctness property \(\phi\), written \(T \models \phi\), iff all its computations satisfy \(\phi\).

Examples of Temporal Properties

Example: Mutual Exclusion Algorithm

\[\text{Action System}\]

\[\text{variables} \ y_1, y_2 : \text{boolean}\]
\[\text{initially} \ y_1 = y_2 = \text{false} \land t = 1\]
\[\text{end}\]

Properties:

a) The two processes are not both in their critical sections at the same time.
\[\neg (at_{l_1} \land at_{m_1})\] is invariant.
b) whenever the left process is in location \(l_0\), then eventually it will get to \(l_4\).
\[at_{l_1}\] leads to \(at_{l_4}\).
Safety vs. Liveness Properties

• A Safety Property is of the form
   “Nothing bad will ever happen”
   Examples
   p is invariant    p is stable    p precedes q

• A Liveness Property is of the form
   “Something good will eventually happen”
   Examples
   eventually p    p leads to q    infinitely often p

We can define a topology on $S^\omega$:
– the distance between $\beta_1$ and $\beta_2$ is
  \[d(\beta_1, \beta_2) = \frac{1}{lcp(\beta_1, \beta_2) + 1}\]
where $lcp(\beta_1, \beta_2)$ is the length of the longest common prefix of $\beta_1$ and $\beta_2$.
– A property $\phi$ is open if $\beta \in \phi$ implies that
  \[\exists \varepsilon > 0. \forall \beta'. \ d(\beta, \beta') < \varepsilon \implies \beta' \in \phi\]
– A safety property is a property whose complement is open.
  I.e., if $\beta \notin \phi$ then there is a finite prefix of $\beta$ such that all
  extensions of that prefix of $\beta$ are not in $\phi$.

Testers for Safety Properties

• The complement of a safety property can be characterized by
  a set of finite sequences (error traces, counterexamples, bad prefixes, ...)

• Idea: Make a (finite-state or infinite-state) automaton which
  accepts this set. Just as in the theory of finite automata.

• A tester for a safety property $\phi$ (or a safety automaton) is a labeled
  transition system $\langle L, T, T_0, \rightarrow \rangle$ where $L = S$ together with a
  set $T^f \subseteq T$ of accepting states.

• A safety tester $\langle L, T, T_0, \rightarrow \rangle$ accepts a behavior
  \[s^0 \rightarrow s^1 \rightarrow \cdots \rightarrow s^n \rightarrow \cdots\]
  iff for some $n$ there is a computation of the tester
  \[\phi^n : s^0 \rightarrow s^1 \rightarrow \cdots \rightarrow s^n \rightarrow \cdots\]
  such that $s^n \in T^f$

• The property defined by a safety tester is the set of behaviors
  that are not accepted by the tester.

Composition of Transition System and Tester

The superposition of the safety tester $\langle L, T, T_0, \rightarrow \rangle$, $T^f$ onto
the transition system $T = \langle S, S^0, \rightarrow \rangle$ is the transition system
where
• $S \times T$ is the set of states,
• $(s, t) \rightarrow (s', t')$ is a transition if $s \rightarrow s'$ is a transition of the
  transition system and $t \xrightarrow{t'} s'$ is a transition of the tester.

Checking a Safety Property now amounts to:
$T$ satisfies the property $\phi$
iff
No final state (i.e., state of form $(s, t)$ with $t \in T^f$) is
reachable in the superposition.
**Fairness**

An example of a rather uninteresting program with two control components is the following.

```plaintext
Program concurrent
declare
initially
end
```

![Two concurrent state-transition diagrams](image)

**Figure 1:** Two concurrent state-transition diagrams

---

**Fairness Sets**

- An action is a subset of $\rightarrow$.
- An action $A$ is enabled in a state $s$ if there is a $s' \in S$ such that $(s, s') \in A$.
- The action $A$ is disabled in the state $s$ if it is not enabled in $s$.
- Let $A$ be an action. Let $C$ be an infinite computation

$$C = s^0 \rightarrow s^1 \rightarrow \cdots \rightarrow s^n \rightarrow \cdots$$

- $C$ is weakly fair with respect to $A$ if for each $i \geq 0$, the following holds.

  if for all $k \geq i$, the set $A$ is enabled in $s^k$ then there is a $j \geq i$ such that the transition $s^j \rightarrow s^{j+1}$ is in $A$.

  i.e., the action can not be continuously enabled beyond some point in a computation without being executed.

- $C$ is strongly fair with respect to $A$ if for each $i \geq 0$, the following holds.

  if for infinitely many $k \geq i$, the set $A$ is enabled in $s^k$ then there is a $j \geq i$ such that the transition $s^j \rightarrow s^{j+1}$ is in $A$.

  i.e., the action can not be infinitely often enabled beyond some point in a computation without being executed.

---

**How does Fairness Work?**

Consider

**Action System Inc**

variables $x, y : \text{integer}$

initially $x = y = 0$

actions $x := x + 1$

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$y := y + 1$</td>
</tr>
</tbody>
</table>

end

Alternatives

- No fairness constraints.
- Weak fairness for the action $x := x + 1$.
- Weak fairness for the action $x := x + 1$
  Weak fairness for the action $y := y + 1$.

---

**Weak vs. Strong Fairness**

Consider

**Action System Inc**

variables $x, y : \text{integer}$

initially $x = y = 0$

actions $x := x + 1$

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>even($x$) $\rightarrow y := y + 1$</td>
</tr>
<tr>
<td>odd($x$) $\rightarrow y := y + 1$</td>
</tr>
</tbody>
</table>

end

Alternatives

- Weak fairness for all three actions.
- Weak fairness for $x := x + 1$
  weak fairness for even($x$) $\rightarrow y := y + 1$
  strong fairness for odd($x$) $\rightarrow y := y + 1$
- Weak fairness for $x := x + 1$
  strong fairness for even($x$) $\rightarrow y := y + 1$
  strong fairness for odd($x$) $\rightarrow y := y + 1$
Examples (cnt’d)

- Assume that for some action
  - enabled means that the action is enabled
  - taken means that the action was taken in the preceding transition.

then

- $(\Diamond \square \text{enabled}) \Rightarrow (\square \Diamond \text{taken})$ expresses weak fairness.
- $\square (\Diamond \neg \text{enabled} \lor \Diamond \text{taken})$ expresses weak fairness.
- $(\Box \diamond \text{enabled}) \Rightarrow (\Box \diamond \text{taken})$ expresses strong fairness.
- $\square (\Diamond \neg \text{enabled} \lor \Diamond \text{taken})$ expresses strong fairness.
- $\Box (p \Rightarrow \Diamond q)$ expresses $p$ leads to $q$.

Safety Testers in SPIN

- Testers marked by keyword never
- A safety tester accepts a behavior by reaching its end.
- Thus, let the never-claim reach its end when property is violated.
- Example: $p$ is stable $\Box (p \Rightarrow \Box p)$

```
never

do
:: ! p
:: p -> break
od;
do
:: p
:: p -> break
od
```