Model Checking VI
Linear-Time Temporal Logic

Edmund M. Clarke, Jr.
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
Model Checking for LTL

- Reduction of LTL model checking to CTL model checking with fairness constraints
- Symbolic LTL model checking algorithm
- Extension of SMV to permit LTL specifications


Motivation

- Succinct and intuitive descriptions
  - No path quantifiers
  - Paths rather than trees

- Expressiveness
  - Some properties such as $FG_p$ cannot be expressed in CTL.
Review of LTL Syntax

Syntax

- **LTL formula:** $\mathbf{A} f$ ($f$ is a path formula).
- **Path formulas:**
  - Propositional operators: $\neg p$ and $p \lor q$
  - Temporal operators: $\mathbf{X} p$ and $p \mathbf{U} q$
Basic Idea for Reduction

- LTL formula
- Kripke structure
- Tableau (Kripke structure)
- Fairness constraints
- Product (Kripke structure)
- CTL formula
- CTL model checking
- CTL model checking
Major Steps in Reduction

1. Translate the given LTL formula $\mathbf{A} f$ to:
   - Tableau (Kripke structure) $T = (S_T, R_T, L_T)$: Includes every path that satisfies $\neg f$.
   - Fairness constraints $\mathcal{F}$: Guarantee that every eventuality $g \mathbf{U} h$ is ultimately filled.
   - CTL formula $\psi$: Guarantees that no state is the start of a path that satisfies $\neg f$.

2. Generate the Product $P$ of $M$ and $T$

3. Perform CTL model checking of $\psi$ in $P$ under $\mathcal{F}$. 
States of Tableau

$S_T$ is $\mathcal{P}(el(p))$, i.e, the power set of elementary formulas of $f$.

- $el(p) = \{p\}$ if $p \in AP$.
- $el(\neg g) = el(g)$.
- $el(g \lor h) = el(g) \cup el(h)$.
- $el(\text{X}g) = \{\text{X}g\} \cup el(g)$.
- $el(g \text{U} h) = \{\text{X}(g \text{U} h)\} \cup el(g) \cup el(h)$.

Example.

$$el(a \text{U} b) = \{a, b, \text{X}(a \text{U} b)\}$$
$$el(a \text{U} (\text{X}b)) = \{a, b, \text{X}b, \text{X}(a \text{U} (\text{X}b))\}$$
States in tableau $T$ for $g = a \cup b$:
Additional function $\text{sat}$:

$\text{sat}(g)$ will be the set of states that satisfy $g$.

- $\text{sat}(g) = \{ \sigma \mid g \in \sigma \}$ where $g \in \text{el}(f)$.
- $\text{sat}(\neg g) = \{ \sigma \mid \sigma \not\in \text{sat}(g) \}$.
- $\text{sat}(g \lor h) = \text{sat}(g) \cup \text{sat}(h)$.
- $\text{sat}(g \cup h) = \text{sat}(h) \cup (\text{sat}(g) \cap \text{sat}(X(g \cup h)))$.

Transition relation $R_T$:

$$R_T(\sigma, \sigma') = \bigwedge_{Xg \in \text{el}(f)} \sigma \in \text{sat}(Xg) \Leftrightarrow \sigma' \in \text{sat}(g).$$
Tableau $T$ for $g = a \cup b$: 

![Diagram of Tableau T](image-url)
We must guarantee that every eventuality is actually fulfilled. For this purpose we use the following fairness constraints.

Fairness Constraints $\mathcal{F}$:

$$\{ \text{sat}(\neg(g \mathcal{U} h) \lor h) \mid g \mathcal{U} h \text{ occurs in } f \}.$$
**Theorem:** If $M, \pi' \models \neg f$ for some $M$ and $\pi'$, then there exists a path $\pi$ in $T$ such that:

- $\pi$ is a fair path.
- The initial state of $\pi$ is in $sat(\neg f)$.

**Theorem:** $M, \sigma' \models A f$ if and only if there is NO state $(\sigma, \sigma')$ in $P$ such that:

- $P, (\sigma, \sigma') \models EG\ true$ under the fairness constraints.
- $(\sigma, \sigma') \in sat(\neg f)$.

Sufficient to check the CTL formula $\psi$:

$$\neg (EG\ true \ & \ Sat_{\neg f})$$
The Kripke structure $M$: 
Product $P$ of the structure $M$ and the tableau $T$:
Fairness Constraint $\mathcal{F}$: $a \mathbf{U} b$ is eventually fulfilled.

CTL formula $\psi$: 

$$\neg (\mathbf{EG} \text{true} \& \text{Sat}_{\neg(a \mathbf{U} b)})$$
We have developed a translator that extends SMV to permit LTL specifications.

The translator replaces a given LTL formula with SMV code for

- a tableau,
- fairness constraints and
- a CTL formula.

The tableau description is implicit!!
MODULE main -- simple program

VAR
a: boolean;
b: boolean;

TRANS  ( a & !b) -> next(!(a & !b))
TRANS  ( a & b) -> next(a & !b)
TRANS  (!a & b) -> next(!a & b)
TRANS  (!a & !b) -> next(!a & b)

SPEC A[a U b]
Translation

-- Kripke structure
MODULE

    ...
    :

MODULE main
    :
    :

-- LTL formula
SPEC A f

An SMV program
Translation

--- Tableau for \( f \)

\textbf{VAR} -- new variables

\begin{align*}
\text{EL} \, \mathbf{X}_{g_1} & : \text{boolean}; \\
\vdots \\
\text{EL} \, \mathbf{X}_{g_N} & : \text{boolean};
\end{align*}

\textbf{DEFINE} -- characteristic function

\begin{align*}
S_{h_1} & := \cdots; \\
\vdots \\
S_{h_M} & := \cdots;
\end{align*}

\textbf{TRANS} -- transition relation

\begin{align*}
( \, \text{EL} \, \mathbf{X}_{g_1} = \text{next} \, (S_{g_1}) \, ) & \land \\
\vdots \\
( \, \text{EL} \, \mathbf{X}_{g_N} = \text{next} \, (S_{g_N}) \, )
\end{align*}

--- fairness constraints

\textbf{FAIRNESS} \quad !\, S'_{g_1} \, \mathbf{U} \, h_1 \mid S'_{h_1}

\vdots

\textbf{FAIRNESS} \quad !\, S'_{g_3} \, \mathbf{U} \, h_3 \mid S'_{h_3}

--- new specification

\textbf{SPEC} \quad !(\neg f \, \& \, \text{EG} \, \text{true})

Translator output for SMV program
Result of translation for simple example:

```plaintext
MODULE main  -- simple program
VAR
  a: boolean;
  b: boolean;
TRANS
  ( a & !b) -> next(!(a & !b))
TRANS
  ( a & b) -> next(a & !b)
TRANS
  (!a & b) -> next(!a & b)
TRANS
  (!a & !b) -> next(!a & b)
VAR
  EL_X_a_U_b : boolean;
DEFINE
  S_a := a;
  S_b := b;
  S_X_a_U_b := EL_X_a_U_b;
  S_a_U_b := S_b | (S_a & S_X_a_U_b);
  S_NOT_a_U_b := !S_a_U_b;
TRANS
  S_X_a_U_b = next(S_a_U_b)
FAIRNESS
  !S_a_U_b | b
SPEC
  !(S_NOT_a_U_b & EG true)
```
Experimental Results

- Distributed mutual exclusion (DME) circuit
  - Speed-independent token ring composed of identical cells
  - Each gate is modeled as a non-deterministic finite-state machine.

- Specifications
  1. (Safety) No two users are acknowledged simultaneously.
  2. (Liveness) All requests are eventually acknowledged.

- Results (time and space) : Comparison with CTL model checking
  - (Safety) Within 10% increase
  - (Liveness) Within 1.5-3 times increase (2 times for large circuits)
### Experimental Results (Cont.)

<table>
<thead>
<tr>
<th>#cell</th>
<th>#nodes</th>
<th>#time(sec)</th>
<th>trans.</th>
<th>#reachable states</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CTL</td>
<td>LTL</td>
<td>CTL</td>
<td>LTL</td>
</tr>
<tr>
<td>3</td>
<td>11326</td>
<td>11362</td>
<td>17.9</td>
<td>20.5</td>
</tr>
<tr>
<td>4</td>
<td>13458</td>
<td>15357</td>
<td>47.5</td>
<td>49.4</td>
</tr>
<tr>
<td>5</td>
<td>22321</td>
<td>22348</td>
<td>100.5</td>
<td>104.4</td>
</tr>
<tr>
<td>6</td>
<td>25869</td>
<td>27318</td>
<td>182.3</td>
<td>193.6</td>
</tr>
<tr>
<td>7</td>
<td>28413</td>
<td>33310</td>
<td>326.4</td>
<td>329.3</td>
</tr>
<tr>
<td>8</td>
<td>44322</td>
<td>44369</td>
<td>509.2</td>
<td>526.3</td>
</tr>
<tr>
<td>9</td>
<td>49702</td>
<td>49755</td>
<td>794.0</td>
<td>794.8</td>
</tr>
<tr>
<td>10</td>
<td>55082</td>
<td>55141</td>
<td>1125.2</td>
<td>1362.7</td>
</tr>
</tbody>
</table>

**Table:** Safety specification for the DME circuit

<table>
<thead>
<tr>
<th>#cell</th>
<th>#nodes</th>
<th>#time(sec)</th>
<th>trans.</th>
<th>#reachable states</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CTL</td>
<td>LTL</td>
<td>CTL</td>
<td>LTL</td>
</tr>
<tr>
<td>3</td>
<td>12721</td>
<td>33940</td>
<td>426.1</td>
<td>1260.5</td>
</tr>
<tr>
<td>4</td>
<td>26541</td>
<td>72029</td>
<td>2553.2</td>
<td>6096.7</td>
</tr>
<tr>
<td>5</td>
<td>47346</td>
<td>120299</td>
<td>9623.1</td>
<td>21950.1</td>
</tr>
<tr>
<td>6</td>
<td>92080</td>
<td>183043</td>
<td>36995.3</td>
<td>66502.5</td>
</tr>
<tr>
<td>7</td>
<td>163867</td>
<td>263380</td>
<td>97807.1</td>
<td>191990.0</td>
</tr>
</tbody>
</table>

**Table:** Liveness specification for the DME circuit
Experimental Results (Cont.)

- Synchronous bus arbiter
  - Daisy chain circuit composed of identical cells
  - Each gate is modeled by a deterministic machine

- Specifications
  1. (Safety) No two users are acknowledged simultaneously.
  2. (Liveness) All requests are eventually acknowledged.

- Results (time and space) : Comparison with CTL model checking
  - (Safety) Within 1.5 times increase
  - (Liveness) Within 1.5-2 times increase
Experimental Results (Cont.)

<table>
<thead>
<tr>
<th>#cell</th>
<th>#nodes</th>
<th>#time(sec)</th>
<th>trans.</th>
<th>#reachable states</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.11</td>
<td>144</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>0.13</td>
<td>176</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>0.16</td>
<td>208</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>0.16</td>
<td>240</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>0.16</td>
<td>272</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.21</td>
<td>304</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11</td>
<td>0.21</td>
<td>336</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>0.31</td>
<td>368</td>
</tr>
</tbody>
</table>

Table: Safety specification for the sync. arbiter

<table>
<thead>
<tr>
<th>#cell</th>
<th>#nodes</th>
<th>#time(sec)</th>
<th>trans.</th>
<th>#reachable states</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.38</td>
<td>144</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>0.43</td>
<td>176</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>0.48</td>
<td>208</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>0.53</td>
<td>240</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>0.71</td>
<td>272</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.83</td>
<td>304</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11</td>
<td>1.00</td>
<td>336</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>1.16</td>
<td>368</td>
</tr>
</tbody>
</table>

Table: Liveness specification for the sync. arbiter