Lecture 10

K-satisfiability and Linear Time Logic

February 10, 1998
Lecturer: Richard Ladner
Notes: Jason Hartline

10.1. K-Satisfiability is in PSPACE

Theorem 10.1: K-satisfiability is in PSPACE.

Proof: We will be using a Tableau method which is often used Modal and Temporal Logic. We will be using $(\mathcal{T}, \mathcal{F}, \mathcal{T}', \mathcal{F}')$. Intuitively these sets represent:

$$\begin{aligned} \alpha \in \mathcal{T} &\Rightarrow \alpha \text{ is true.} \\ \alpha \in \mathcal{F} &\Rightarrow \alpha \text{ is false.} \\ \alpha \in \mathcal{T}' &\Rightarrow \alpha \text{ is true in every next world.} \\ \alpha \in \mathcal{F}' &\Rightarrow \alpha \text{ is false in some next world.} \end{aligned}$$

$K\text{-}SAT(\mathcal{T}, \mathcal{F}, \mathcal{T}', \mathcal{F})$ returns true if and only if there is a K-model, M, and a state, s, such that

$$M, s \models \bigwedge_{\alpha \in \mathcal{T}} \alpha \wedge \bigwedge_{\alpha \in \mathcal{F}} \neg\alpha \wedge \bigwedge_{\alpha \in \mathcal{T}'} \Box\alpha \wedge \bigwedge_{\alpha \in \mathcal{F}'} \neg\Box\alpha$$

We now give the following alternating polynomial time algorithm for $K\text{-}SAT$. VAR is the set of variables.

$K\text{-}SAT(\mathcal{T}, \mathcal{F}, \mathcal{T}', \mathcal{F}')$
  if $\mathcal{T} \cup \mathcal{F} \not\subseteq VAR$
    choose $\alpha \in (\mathcal{T} \cup \mathcal{F}) - VAR$
    case
      $\alpha = \neg\beta$ and $\alpha \in \mathcal{T}$: return $K\text{-}SAT(\mathcal{T} - \{\alpha\}, \mathcal{F} \cup \{\beta\}, \mathcal{T}', \mathcal{F}')$
      $\alpha = \neg\beta$ and $\alpha \in \mathcal{F}$: return $K\text{-}SAT(\mathcal{T} \cup \{\beta\}, \mathcal{F} - \{\alpha\}, \mathcal{T}', \mathcal{F}')$
      $\alpha = \beta \vee \gamma$ and $\alpha \in \mathcal{T}$:
        return $K\text{-}SAT(\mathcal{T} - \{\alpha\} \cup \{\beta\}, \mathcal{F}, \mathcal{T}', \mathcal{F}') \vee K\text{-}SAT(\mathcal{T} - \{\alpha\} \cup \{\gamma\}, \mathcal{F}, \mathcal{T}', \mathcal{F}')$
      $\alpha = \beta \vee \gamma$ and $\alpha \in \mathcal{F}$: return $K\text{-}SAT(\mathcal{T}, \mathcal{F} - \{\alpha\} \cup \{\beta, \gamma\}, \mathcal{T}', \mathcal{F}')$
      $\alpha = \Box\beta$ and $\alpha \in \mathcal{T}$: return $K\text{-}SAT(\mathcal{T} - \{\alpha\}, \mathcal{F}, \mathcal{T}' \cup \{\beta\}, \mathcal{F}')$

$\alpha = \Box\beta$ and $\alpha \in \mathcal{F}$: return $K\text{-}SAT(\mathcal{T}, \mathcal{F} - \{\alpha\}, \mathcal{T}', \mathcal{F}' \cup \{\beta\})$
endcase
else if $\mathcal{T} \cap \mathcal{F} \neq \emptyset$
return *false*
else
return $\bigwedge_{\beta \in \mathcal{F}'} K\text{-}SAT(\mathcal{T}', \{\beta\}, \emptyset, \emptyset)$ (true if $\mathcal{F}' = \emptyset$)
fi

To see if statement α is satisfiable, we then call $K\text{-}SAT(\{\alpha\}, \emptyset, \emptyset.\emptyset)$.

This algorithm is polynomial time because it reduces the size of $|\mathcal{T}| + |\mathcal{F}| + |\mathcal{T}'| + |\mathcal{F}'|$ by 1 at least once every two recursive steps. $|\mathcal{X}|$ denotes the sum of the lengths of the formulas in $\mathcal{X}$. $\Box$

10.1.1. Example

$\neg(\Box x \vee \Box\neg x)$ is satisfiable:

$$
\begin{array}{cccc}
\mathcal{T} & \mathcal{F} & \mathcal{T}' & \mathcal{F}' \\
\{\neg(\Box x \vee \Box\neg x)\} & \emptyset & \emptyset & \emptyset \\
 & \downarrow & & \\
\emptyset & \{\Box x \vee \Box\neg x\} & \emptyset & \emptyset \\
 & \downarrow & & \\
\emptyset & \{\Box x, \Box\neg x\} & \emptyset & \emptyset \\
 & \downarrow & & \\
\emptyset & \{\Box\neg x\} & \emptyset & \{x\} \\
 & \downarrow & & \\
\emptyset & \emptyset & \emptyset & \{x, \neg x\} \\
 & \swarrow & \searrow &
\end{array}
$$

$$
\begin{array}{cccc}
\emptyset & \{x\} & \emptyset & \emptyset \\
 & \downarrow & & \\
 & \mathit{true} & &
\end{array}
\qquad
\begin{array}{cccc}
\emptyset & \{\neg x\} & \emptyset & \emptyset \\
 & \downarrow & & \\
\{x\} & \emptyset & \emptyset & \emptyset \\
 & \downarrow & & \\
 & \mathit{true} & &
\end{array}
$$

10.1.2. K-model checking is in P (in the size of the model and formula).

Make a tableau for each state in the model and fill in whether a particular statement is true or false at that state starting inductively with small length statements.

10.2. Linear Time Logic: *LTL*

10.2.1. The *LTL* Model

Our model is a sequence of states such that every state looks like every other state. We will introduce two new unary operators and one new binary operator:

$$\circ \to \circ \to \circ \to \circ \to \circ \to \circ \to \circ \to \circ \to \circ \to \cdots$$

$$\begin{array}{rcl} G\alpha & \Rightarrow & \textit{globally } \alpha \text{ is true.} \\ X\alpha & \Rightarrow & \text{in the } \textit{next} \text{ state } \alpha \text{ is true.} \\ \alpha U\beta & \Rightarrow & \alpha \text{ is true } \textit{until } \beta \text{ is true.} \end{array}$$

Relative to our model, these operators represent the following scenarios.

$$\begin{array}{cccccccccc} & & & & G\alpha & & & & & \\ G\alpha & : & \circ & \to & \circ & \to & \circ & \to & \circ & \cdots \\ & & & & \alpha & & \alpha & & \alpha & \end{array}$$

$$\begin{array}{cccccccccc} & & & & X\alpha & & & & & \\ X\alpha & : & \circ & \to & \circ & \to & \circ & \to & \circ & \cdots \\ & & & & & & \alpha & & & \end{array}$$

$$\begin{array}{cccccccccccccc} & & & & \alpha U\beta & & & & & & & & & \\ \alpha U\beta & : & \circ & \to & \circ & \to & \circ & \cdots & \circ & \to & \circ & \to & \circ & \cdots \\ & & & & \alpha & & \alpha & & \alpha & & \beta & & & \end{array}$$

$$\begin{array}{cccccc} & & \alpha U\beta & & & \\ \circ & \to & \circ & \to & \circ & \cdots \\ & & \beta & & & \end{array}$$

10.2.2. Semantics

An *LTL* model is $M = (S, \pi, x)$ where S is a set of states, π is an assignment that takes $s \in S$ to variable assignments, and x is an infinite path through the states $s_0 s_1 s_2 \ldots$.

$$x^i \stackrel{\text{def}}{=} s_i s_{i+1} s_{i+2} \cdots$$

This can then be defined inductively as:

$$\begin{array}{rcl} x^0 & = & x \\ x^i & = & s_i x^{i+1} \end{array}$$

We will use the following semantics. Note that since every state is like any other state we can pick our starting state to be s_0.

$$
\begin{aligned}
M, x &\models z \text{ if } \pi(s_0)(z) = \mathit{true} \\
M, x &\models \alpha \vee \beta \text{ if } M, x \models \alpha \text{ or } M, x \models \beta \\
M, x &\models \neg\alpha \text{ if } M, x \not\models \alpha \\
M, x &\models G\alpha \text{ if } \forall i \geq 0\ M, x^i \models \alpha \\
M, x &\models X\alpha \text{ if } M, x^1 \models \alpha \\
M, x &\models \alpha U \beta \text{ if } \exists j \geq 0\ M, x^j \models \beta \text{ and } \forall i < j\ M, x^i \models \alpha
\end{aligned}
$$

An *LTL* statement α is valid, denoted $\models_{\text{LTL}} \alpha$, if and only if, for all $M - (S, \pi, x)$, $M, x \models \alpha$.

Now we will introduce some new operators that are defined in terms of G, and U.

$$
\begin{aligned}
F\alpha &\stackrel{\text{def}}{=} \neg G \neg \alpha \quad \text{"eventually } \alpha\text{"} \\
\alpha B \beta &\stackrel{\text{def}}{=} \neg(\neg\alpha U \beta) \quad \text{"}\alpha \text{ before } \beta\text{"}
\end{aligned}
$$

Curiously: $\models_{\text{LTL}} F\alpha \equiv \mathit{true}\, U \alpha$

10.2.3. *LTL* validities

The following validities push negations towards variables to transform a formula into a "positive" form where only variables are negated.

$$
\begin{aligned}
&\models_{\text{LTL}} \neg G\alpha \equiv F \neg\alpha \\
&\models_{\text{LTL}} \neg F\alpha \equiv G \neg\alpha \\
&\models_{\text{LTL}} \neg(\alpha U \beta) \equiv \neg\alpha B \beta \\
&\models_{\text{LTL}} \neg X\alpha \equiv X \neg\alpha
\end{aligned}
$$

Here are some more validities of interest:

$$
\begin{aligned}
&\models_{\text{LTL}} FF\alpha \equiv F\alpha \\
&\models_{\text{LTL}} GG\alpha \equiv G\alpha \\
&\models_{\text{LTL}} \alpha U \beta \equiv \beta \vee [X(\alpha U \beta) \wedge \alpha] \\
&\models_{\text{LTL}} \alpha B \beta \equiv \neg\beta \wedge [\alpha \vee X(\alpha B \beta)] \\
&\models_{\text{LTL}} G\alpha \equiv \alpha \wedge XG\alpha \\
&\models_{\text{LTL}} F\alpha \equiv \alpha \vee XF\alpha
\end{aligned}
$$