CS 228 : Logic in Computer Science

Krishna. S

Union of NBA/DBA

Normal Form for ω-regular languages

An ω-regular language $L \subseteq \Sigma^\omega$ can be written as $L = \bigcup_{i=1}^{n} U_i V_i^\omega$, where U_i, V_i are regular languages.

One direction : Assume L is accepted by an NBA/DBA.

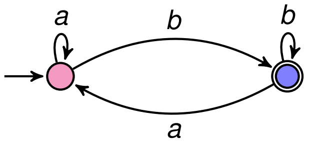
- Define $U_g = \{w \in \Sigma^* \mid q_0 \xrightarrow{w} g\}$
- Define $V_g = \{w \in \Sigma^* \mid g \xrightarrow{w} g\}$
- Then $L = \bigcup_{g \in G} U_g V_g^\omega$, where U_g, V_g are regular
- Show that U_g, V_g are regular.

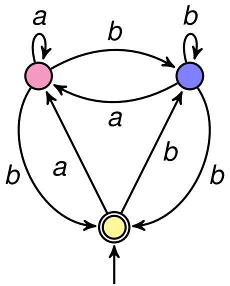
Normal Form for ω-regular languages

An ω-regular language $L \subseteq \Sigma^\omega$ can be written as $L = \bigcup_{i=1}^{n} U_i V_i^\omega$, where U_i, V_i are regular languages.

Other direction : Assume $L = \bigcup_{i=1}^{n} U_i V_i^\omega$. Show that L is accepted by an NBA/DBA.

1. If V is regular, V^ω is ω-regular





Normal Form for ω-regular languages

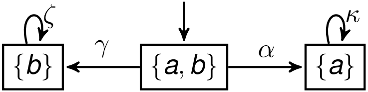
1. If V is regular, V^ω is ω-regular
   - Let $D = (Q, \Sigma, q_0, \delta, F)$ be a DFA accepting V
   - Construct NBA $E = (Q \cup \{p_0\}, \Sigma, p_0, \Delta, G)$ such that $G = \{p_0\}$,
   - $\Delta = \delta \cup \{p_0 \in \Delta(q, a) \mid \delta(q, a) \in F\} \cup \{\Delta(p_0, a) = s \mid \delta(q_0, a) = s\}$
2. Show that if U is regular and V^ω is ω-regular, then UV^ω is ω-regular
   - $D = (Q_1, \Sigma, q_0, \delta_1, F)$ be a DFA, $L(D) = U$ and $E = (Q_2, \Sigma, q_0', \delta_2, G)$ be an NBA, $L(E) = V^\omega$.
   - $A = (Q_1 \cup Q_2, \Sigma, q_0, \delta', G)$ NBA such that $\delta' = \delta_1 \cup \delta_2 \cup \{(q, a, q_0') \mid \delta_1(q, a) \in F\}$

LTL ModelChecking

- Given transition system TS, and LTL formula φ, does $TS \models \varphi$?
- $Tr(TS) \subseteq L(\varphi)$ iff $Tr(TS) \cap \overline{L(\varphi)} = \emptyset$
- First construct NBA $A_{\neg\varphi}$ for $\neg\varphi$.
- Construct product of TS and $A_{\neg\varphi}$, obtaining a new TS, say TS'.
- Check some very simple property on TS', to check $TS \models \varphi$.

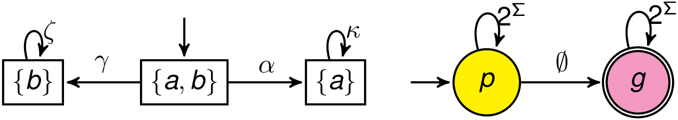
An Example $TS \models \varphi$

- Let $\varphi = \Box(a \vee b), \neg\varphi = \Diamond(\neg a \wedge \neg b)$
- Take TS and $A_{\neg\varphi}$, and check the intersection.



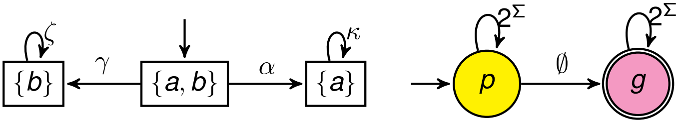
An Example $TS \models \varphi$

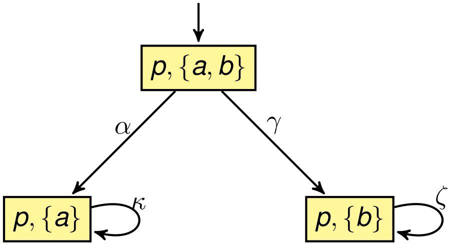
- Let $\varphi = \Box(a \vee b), \neg\varphi = \Diamond(\neg a \wedge \neg b)$
- Take TS and $A_{\neg\varphi}$, and check the intersection.



An Example $TS \models \varphi$

- Let $\varphi = \Box(a \vee b)$, $\neg\varphi = \Diamond(\neg a \wedge \neg b)$
- Take TS and $A_{\neg\varphi}$, and check the intersection.





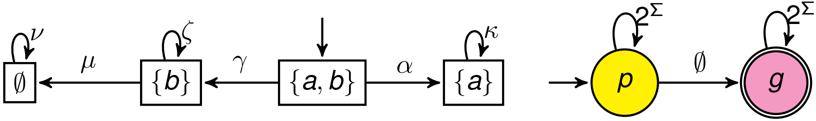
An Example : $TS \not\models \varphi$

- Let $\varphi = \Box(a \vee b), \neg\varphi = \Diamond(\neg a \wedge \neg b)$
- Take TS and $A_{\neg\varphi}$, and check the intersection.

$$
\begin{tikzcd}
 & & {} \arrow[d] & \\
\boxed{\emptyset} \arrow[loop above, "\nu"] & \boxed{\{b\}} \arrow[l, "\mu"'] \arrow[loop above, "\zeta"] & \boxed{\{a, b\}} \arrow[l, "\gamma"'] \arrow[r, "\alpha"] & \boxed{\{a\}} \arrow[loop above, "\kappa"]
\end{tikzcd}
$$

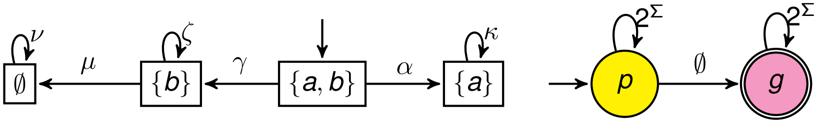
An Example : $TS \not\models \varphi$

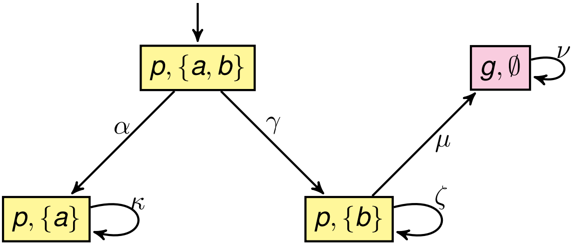
- Let $\varphi = \Box(a \vee b), \neg\varphi = \Diamond(\neg a \wedge \neg b)$
- Take TS and $A_{\neg\varphi}$, and check the intersection.



An Example : $TS \not\models \varphi$

- Let $\varphi = \Box(a \vee b), \neg\varphi = \Diamond(\neg a \wedge \neg b)$
- Take TS and $A_{\neg\varphi}$, and check the intersection.





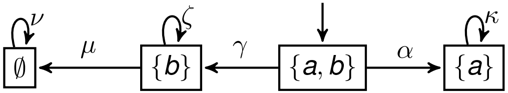
Product of TS and NBA

Given $TS = (S, Act, I, AP, L)$ and $\mathcal{A} = (Q, 2^{AP}, \delta, Q_0, G)$ NBA.
Define $TS \otimes \mathcal{A} = (S \times Q, Act, I', AP', L')$ such that

- $I' = \{(s_0, q) \mid s_0 \in I \text{ and } \exists q_0 \in Q_0, q_0 \stackrel{L(s_0)}{\rightarrow} q\}$
- $AP' = Q$, $L' : S \times Q \rightarrow 2^Q$ such that $L'((s, q)) = \{q\}$
- If $s \stackrel{\alpha}{\rightarrow} t$ and $q \stackrel{L(t)}{\rightarrow} p$, then $(s, q) \stackrel{\alpha}{\rightarrow} (t, p)$

Persistence Properties

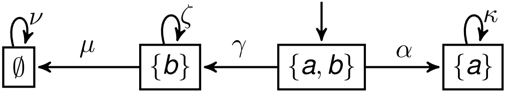
Let η be a propositional logic formula over AP. A persistence property P_{pers} has the form $\Diamond\Box\eta$. How will you check a persistence property on a TS?



- For example, $TS \not\models \Diamond\Box(a \vee b)$
- For example, $TS \models \Diamond\Box(a \vee (a \rightarrow b))$

Persistence Properties

Let η be a propositional logic formula over *AP*. A persistence property P_{pers} has the form $\Diamond\Box\eta$. How will you check a persistence property on a TS?



- For example, $TS \not\models \Diamond\Box(a \vee b)$
- For example, $TS \models \Diamond\Box(a \vee (a \rightarrow b))$
- $TS \not\models P_{pers}$ iff there is a reachable cycle in the TS containing a state with a label which satisfies $\neg\eta$.

LTL ModelChecking

- Given TS and LTL formula φ. Does $TS \models \varphi$?
- Construct $A_{\neg\varphi}$, and let $g_1, \ldots, g_n$ be the good states in $A_{\neg\varphi}$.
- Build $TS' = TS \otimes A_{\neg\varphi}$.
- The labels of TS' are the state names of $A_{\neg\varphi}$.
- Check if $TS' \models \Diamond\Box(\neg g_1 \wedge \ldots \neg g_n)$.

LTL ModelChecking

- Given TS and LTL formula φ. Does $TS \models \varphi$?
- Construct $A_{\neg\varphi}$, and let $g_1, \ldots, g_n$ be the good states in $A_{\neg\varphi}$.
- Build $TS' = TS \otimes A_{\neg\varphi}$.
- The labels of TS' are the state names of $A_{\neg\varphi}$.
- Check if $TS' \models \Diamond\Box(\neg g_1 \wedge \ldots \neg g_n)$.

ModelChecking LTL in TS = Check Persistence in TS'

The following are equivalent.

- $TS \models \varphi$
- $Tr(TS) \cap L(A_{\neg\varphi}) = \emptyset$
- $TS' \models \Diamond\Box(\neg g_1 \wedge \ldots \neg g_n)$.