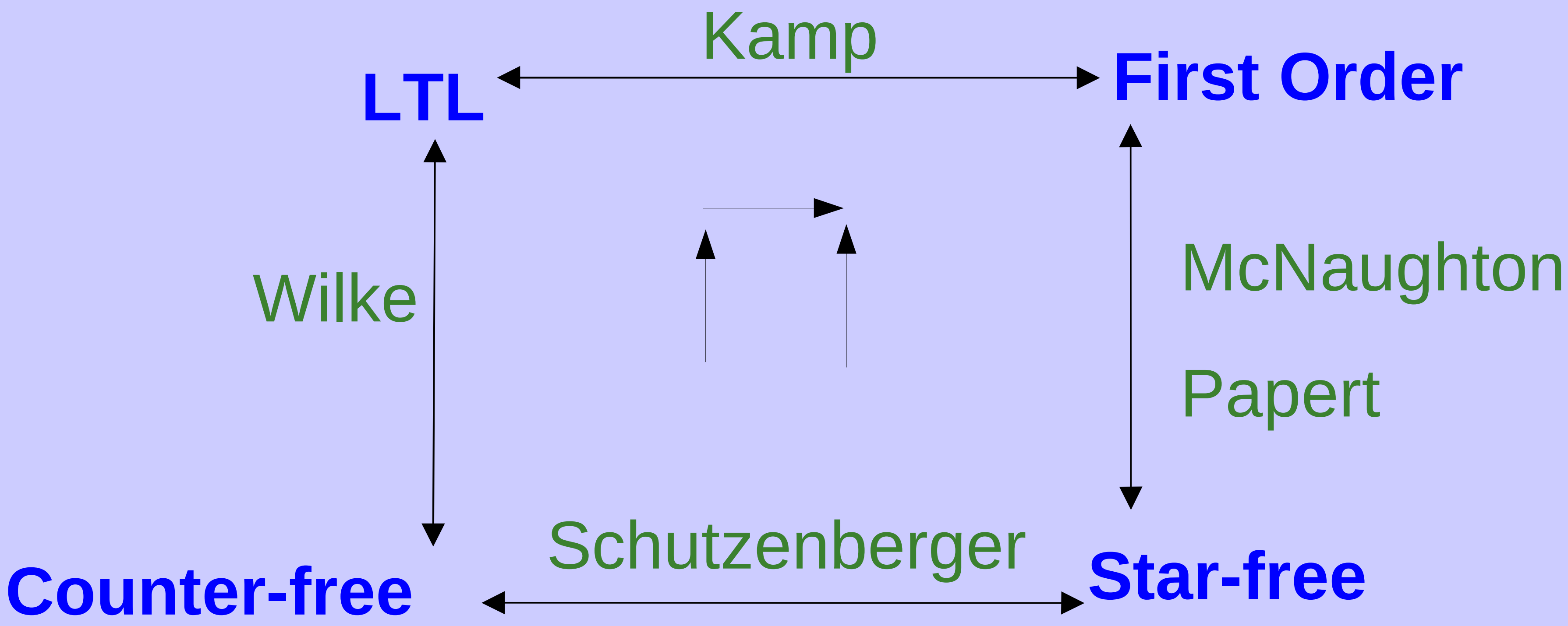
Expressive Completeness of LTL

K Vamsi Krishna & V Deepak
CSA, IISc

Outline

- The big picture
- TL to FO
- Star-free to FO
- Counter-free to TL
- Temporal logic
- Finite Automata
- Examples
- Theorem
- Proof
- References

The big picture



TL to FO

- For every TL formula φ, we give a FO formula $\varphi^+(x)$ such that $L(\varphi) = L(\forall x\,(first(x) \supset \varphi^+(x)))$

φ	$\varphi^+(x)$
a	$Q_a(x)$
$\neg\varphi_1$	$\neg\varphi_1^+(x)$
$\varphi_1 \vee \varphi_2$	$\varphi_1^+(x) \vee \varphi_2^+(x)$
$X\varphi_1$	$\exists y(succ(x,y) \wedge \varphi_1^+(y))$
$F\varphi_1$	$\exists y(x<y \wedge \varphi_1^+(y))$
$\varphi_1 U \varphi_2$	$\exists y(x<y \wedge \varphi_2^+(y) \wedge \forall z((x<z<y) \supset \varphi_1^+(z))$

Star-free to FO

- A language is called star-free if it can be constructed from finite languages by applications of boolean operations and concatenation
  - eg. A*.a.b.~(A*.a.A*) where A = {a,b,c}
  - is same as $\exists x \exists y (S(x,y) \wedge Q_a(x) \wedge Q_b(y) \wedge \neg \exists z (y<z \wedge Q_a(z)))$

Star-free to FO (cont)

- For every star-free expression r, we give a FO formula $\varphi(x,y)$ such that
- $L(r) = L(\forall x\ \forall y\ (first(x) \wedge last(y) \supset \varphi(x,y))$

r	$\boldsymbol{\varphi}$ **(x,y)**
◆ a	$Q_a(x) \wedge x=y$
◆ $\neg\ r_1$	$\neg\ \varphi_1(x,y)$
◆ r_1+r_2	$\varphi_1(x,y) \vee \varphi_2(x,y)$
◆ $r_1.r_2$	$\exists y_1\, \exists x_2(\ (x \leq y_1 \leq y) \wedge \varphi_1(x,y_1) \wedge\ succ(y_1, x_2) \wedge \varphi_2(x_2,y)\ )$

Counter-free to TL

- Every language expressible in temporal logic is a regular language.
- Now, what class of regular languages are expressible in temporal logic?
- Is there any structural property of minimal DFAs of languages that can be expressed in temporal logic?

Temporal logic (TL)

- A temporal formula φ over Σ is defined as
  $\varphi := a \mid \neg\varphi \mid \varphi\vee\varphi \mid X\varphi \mid F\varphi \mid \varphi U\varphi,\ a \in \Sigma$
- The temporal formulas are interpreted over strings (finite words).
- The positions of a string of length n are indexed 0 , . . . , n-1.
- u(i,j) denotes the string u(i)u(i+1)…u(j-1) for $0 \leq i \leq j \leq n$.
- u(i,*) denotes the suffix u(i,n).

TL (cont.)

- u |= a ⇔ u(0) = a
- u |= Xφ ⇔ |u| > 1 and u(1,*) |= φ
- u |= Fφ ⇔ ∃i , 0 < i < |u| ∍ u(i,*) |= φ
- u |= φUψ⇔ ∃i , 0 < i < |u| ∍ u(j,*) |= φ for every j ∈{1,…,i-1} and u(i,*) |= ψ

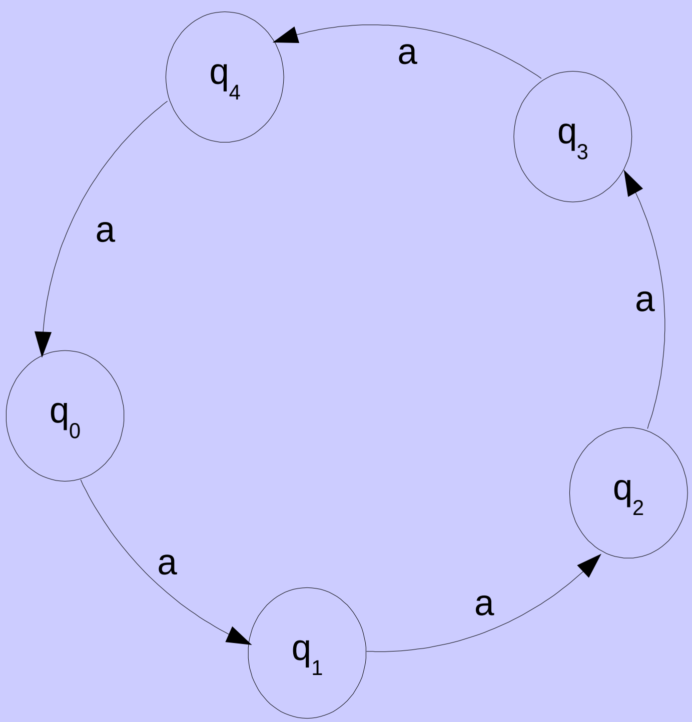
** F and U are strict modalities **

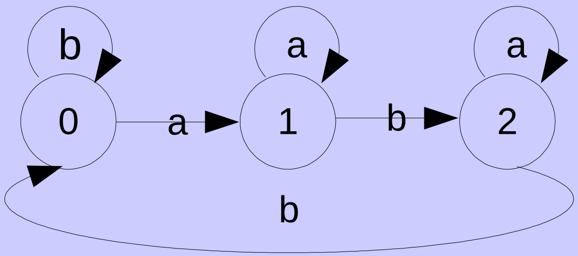
Finite Automata (DFA)

- A DFA is a tuple $A = (\Sigma, Q, q_I, \delta, F)$.
- $L(A) = \{ u \in \Sigma^+ \mid \delta^*(q_I,u) \in F \}$.
- A minimal DFA for a regular language L is denoted by A_L.
- *Counter* : Given a DFA A, a sequence $q_0, q_1, ..., q_{m-1}$ of distinct states is a counter for a string *u* if m>1 and $\delta^*(q_i,u) = q_{i+1}$ for i<m, where by convention, $q_m = q_0$.
- A DFA is *counter-free* if it does not have a counter.

Example

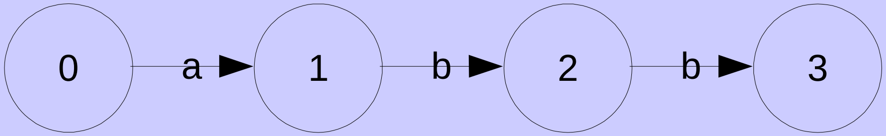
- Counter automata

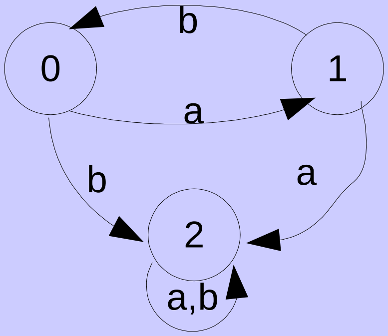




Example

- Counter-free automata





Definitions

- A pre-automaton is a triple (Σ, Q, δ).
- Given a set Q, Q^Q is the set of all functions on Q
- For $\alpha : Q \rightarrow Q$, $Q' \subseteq Q$, $\alpha[Q'] = \{\alpha(q) \mid q \in Q'\}$.
- Given $\alpha, \beta : Q \rightarrow Q$, we write $\alpha\beta$ for the composition of α and β, i.e., for the function given by $q \rightarrow \beta(\alpha(q))$.

Definitions (cont.)

- For a pre-automaton $A=(\Sigma, Q, \delta)$ and for every $u \in \Sigma^*$,we define its *transformation*, u^A, as follows: For every $q \in Q$, we set $u^A(q) = \delta^*(q,u)$.
- $S^A = \{u^A \mid u \in \Sigma^+\}$
- For $\alpha:Q \rightarrow Q$, we set $L_\alpha{}^A = \{ u \in \Sigma^+ \mid u^A = \alpha \}$.
- $L_\alpha{}^{A'} = L_\alpha{}^A \cup \{\in\}$ if $\alpha = id_Q$.
- If A is counter-free and $u^A[Q] = Q$, then $u^A = id_Q$.

Theorem

- A regular language L is expressible in TL if and only if A_L is counter-free

Proof (Outline)

- It suffices to prove that for every pre-automaton A, and every $\alpha \in S^A$, the language L_α^A is expressible in temporal logic.
- Proof goes by induction on $|Q|$ in the first place and then on $|\Sigma|$.
- We distinguish two cases:
  - Case1: For all symbols $a \in \Sigma$, $a^A[Q]=Q$.
  - Case2: There is some symbol $b \in \Sigma$, $b^A[Q] \subset Q$.

Proof

- Let $b^A[Q] = Q' \subset Q$
- $\Gamma = \Sigma$ - {b}
- B is the pre-automaton that results from A by restricting it to the symbols from Γ.
- $U_0 = \Gamma^* b$
- $\Delta = \{u^A \mid u \in U_0\}$
- $C = (\Delta, Q', \delta')$ where $\delta'(q, \alpha) = \alpha(q)$
- $h : U_0^+ \rightarrow \Delta^+$ is the function defined by $h(u_0 \ldots u_{n-1}) = u_0^A \ldots u_{n-1}^A$ for $u_0, \ldots, u_{n-1} \in U_0$

Proof (cont.)

- L_α^A is TL-expressible.
- $L_\alpha^A = L_0 \cup L_1 \cup L_2$
- $L_0 = L_\alpha^B$
- $L_1 = \cup_{\alpha=\beta b^A \beta'} L_\beta^{B'} b L_{\beta'}^{B'}$

  $= \cup_{\alpha=\beta b^A \beta'} (L_\beta^{B'} b\Sigma^* \cap \Gamma^* b L_{\beta'}^{B'})$
- $L_2 = \cup_{\alpha=\beta b^A \gamma\beta'} L_\beta^{B'} b h^{-1}(L_\gamma^C) L_{\beta'}^{B'}$

  $= \cup_{\alpha=\beta b^A \gamma\beta'} (L_\beta^{B'} b\Sigma^* \cap \Gamma^* b h^{-1}(L_\gamma^C) \Gamma^* \cap \Sigma^* b L_{\beta'}^{B'})$

Lemma 1

- Let Σ be an alphabet, $b \in \Sigma$, and $\Gamma = \Sigma - \{b\}$. Assume $L \subseteq \Sigma^+$ and $L' \subseteq \Gamma^+$ are TL-expressible. Then so are Γ^*bL, $\Gamma^*b(L+\epsilon)$, Σ^*bL, $\Sigma^*b(L+\epsilon)$, $L'b\Sigma^*$ and $(L'+\epsilon)b\Sigma^*$.

Lemma 2

- Let Σ, Δ be alphabets, b $\in \Sigma$, $\Gamma = \Sigma$ - {b}, and $U_0 = \Gamma^* b$. Further, let $h_0 : U_0 \rightarrow \Delta$ be an arbitrary function and $h : U_0^+ \rightarrow \Delta^+$ be defined by $h(u_0 \ldots u_{n-1}) = h_0(u_0) \ldots h_0(u_{n-1})$ for $u_0, \ldots, u_{n-1} \in U_0$. For every $d \in \Delta$, let $L_d = \{u \in \Gamma^+ \mid h_0(ub) = d\}$.
- If $L \subseteq \Delta^+$ and L_d for every $d \in \Delta$ are TL-expressible then so is $h^{-1}(L)\, \Gamma^*$.

References

- Thomas Wilke. Classifying discrete temporal properties.
- Wolfgang Thomas. Languages, Automata, and Logic.