The Expressive Power of Linear-time Temporal Logic

K Narayan Kumar

Chennai Mathematical Institute
email:kumar@cmi.ac.in

IIT Guwahati, July 2006

Linear-time Temporal Logic

LTL — convenient specification language

- Atomic propositions, boolean connectives, temporal modalities.

Linear-time Temporal Logic

LTL — convenient specification language

- Atomic propositions, boolean connectives, temporal modalities.
- Models are words.

Linear-time Temporal Logic

LTL — convenient specification language

- Atomic propositions, boolean connectives, temporal modalities.
- Models are words.

Formulas are interpreted at positions of a word.

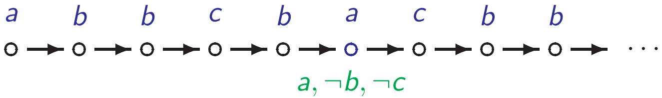
$$w = w_1 w_2 w_3 \ldots \qquad \text{with } w_i \in \Sigma$$

$$w, i \models \varphi \, ?$$

Syntax and Semantics

Atomic propositions: elements of Σ.

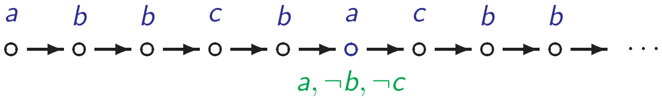
$$w, i \models a \iff w_i = a$$



Syntax and Semantics

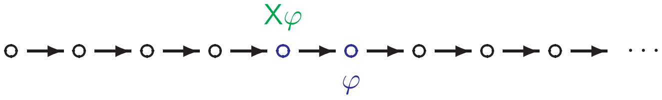
Atomic propositions: elements of Σ.

$$w, i \models a \iff w_i = a$$



The Next state operator:

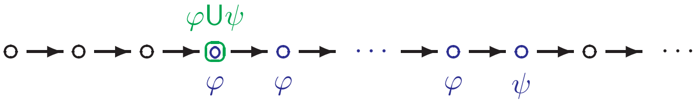
$$w, i \models X\varphi \iff w, i+1 \models \varphi$$



Syntax and Semantics

The Until operator:

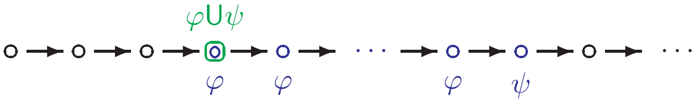
$$w, i \models \varphi \mathsf{U} \psi \quad \Longleftrightarrow \quad \exists j \geq i.\ w, j \models \psi \text{ and } \forall i \leq k < j.\ w, k \models \varphi$$



Syntax and Semantics

The Until operator:

$$w, i \models \varphi \mathsf{U} \psi \quad \iff \quad \exists j \geq i.\ w, j \models \psi \text{ and } \forall i \leq k < j.\ w, k \models \varphi$$



Boolean Connectives:

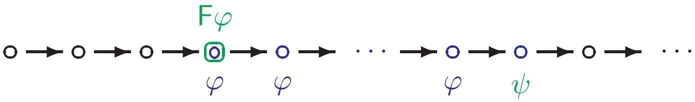
$$\varphi \wedge \psi, \quad \neg\varphi, \quad \ldots$$

with the usual interpretation.

Other Modalities

The Future modality

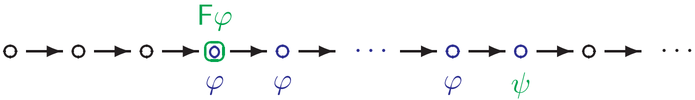
$$w, i \models \mathsf{F}\varphi \quad \Longleftrightarrow \quad \exists j \geq i.\ w, j \models \varphi$$



Other Modalities

The Future modality

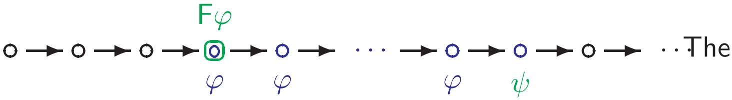
$$F\varphi \quad = \quad \top U \varphi$$



Other Modalities

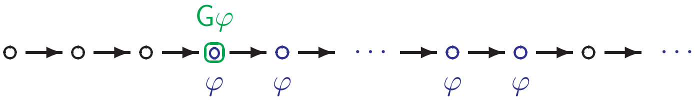
The Future modality

$$F\varphi \quad = \quad \top U \varphi$$



Henceforth modality:

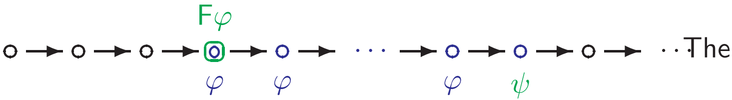
$$w, i \models G\varphi \quad \iff \quad \forall j \geq i.\ w, j \models \varphi$$



Other Modalities

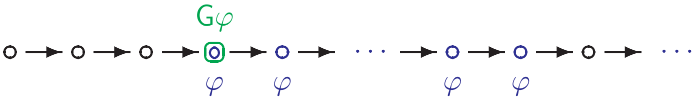
The Future modality

$$\mathsf{F}\varphi \quad = \quad \top \mathsf{U} \varphi$$



Henceforth modality:

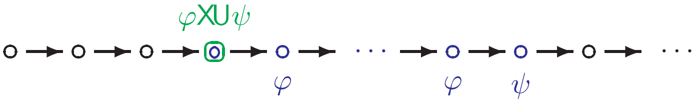
$$\mathsf{G}\varphi \quad = \quad \neg \mathsf{F} \neg \varphi$$



The Universal Modality

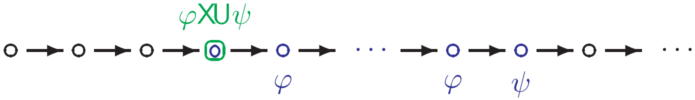
The Next-Until modality:

$$w, i \models \varphi \mathsf{XU} \psi \quad \equiv \quad \exists \mathrm{j} > \mathrm{i}.\ \mathrm{w}, \mathrm{j} \models \psi \text{ and } \forall \mathrm{i} < \mathrm{k} \leq \mathrm{j}.\ \mathrm{w}, \mathrm{k} \models \varphi$$



The Universal Modality

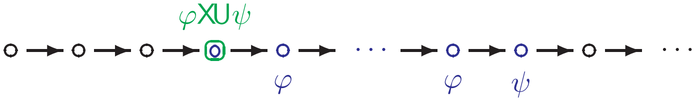
The Next-Until modality:



$$\varphi XU \psi \quad = \quad X(\varphi U \psi)$$

The Universal Modality

The Next-Until modality:



$$\varphi \mathsf{XU} \psi \quad = \quad \mathsf{X}(\varphi \mathsf{U} \psi)$$

Next-Until can express everthing else

$$\begin{array}{rcl} \mathsf{X}\varphi & = & \bot \mathsf{XU} \varphi \\ \varphi \mathsf{U} \psi & = & \psi \vee (\varphi \wedge \varphi \mathsf{XU} \psi) \end{array}$$

LTL definable languages

A word satisfies φ if the initial position satisfies φ

$$w \models \varphi \iff w, 1 \models \varphi$$

LTL definable languages

A word satisfies φ if the initial position satisfies φ

$$w \models \varphi \iff w, 1 \models \varphi$$

Formulas define languages. For example,

$$\mathsf{G}(\mathsf{a} \implies \mathsf{Fb})$$

describes words in which there is a b somewhere to the right of every a.

$$b^*(aa^*bb^*)^*$$

Finite/Infinite Words

- LTL formulas are interpreted over both finite and infinite words.

Finite/Infinite Words

- LTL formulas are interpreted over both finite and infinite words.
- Satisfiability of a formula may depend on the class of models.

Finite/Infinite Words

- LTL formulas are interpreted over both finite and infinite words.
- Satisfiability of a formula may depend on the class of models.

$$\mathsf{GX}\top$$

  is satisfied only over infinite words.

$$\mathsf{F}\neg\mathsf{X}\top$$

  is satisfied only by finite words.
- The empty word is not a model.

First-Order Logic of Words

Consider the First-Order formula

$$\varphi \quad = \quad \forall x.\ (a(x) \implies \exists y.\ ((y > x) \wedge b(x))$$

interpreted over words.

First-Order Logic of Words

Consider the First-Order formula

$$\varphi \quad = \quad \forall x.\ (a(x) \implies \exists y.\ ((y > x) \wedge b(x))$$

interpreted over words.

- The variables x, y etc. refer to positions in the word.

First-Order Logic of Words

Consider the First-Order formula

$$\varphi \quad = \quad \forall x.\ (a(x) \implies \exists y.\ ((y > x) \wedge b(x))$$

interpreted over words.

- The variables x, y etc. refer to positions in the word.
- The formula $a(x)$ asserts that the letter at position x is a.

First-Order Logic of Words

Consider the First-Order formula

$$\varphi \quad = \quad \forall x.\ (a(x) \implies \exists y.\ ((y > x) \wedge b(x))$$

interpreted over words.

- The variables x, y etc. refer to positions in the word.
- The formula $a(x)$ asserts that the letter at position x is a.
- The quantifiers have the usual meaning.

First-Order Logic of Words

Consider the First-Order formula

$$\varphi \quad = \quad \forall x.\ (a(x) \implies \exists y.\ ((y > x) \wedge b(x))$$

interpreted over words.

- The variables x, y etc. refer to positions in the word.
- The formula $a(x)$ asserts that the letter at position x is a.
- The quantifiers have the usual meaning.
- The formula $y > x$ is true if the position y appears somewhere to the right of the position x.

First-Order Logic of Words

Consider the First-Order formula

$$\varphi \quad = \quad \forall x.\ (a(x) \implies \exists y.\ ((y > x) \wedge b(x))$$

interpreted over words.

- The variables x, y etc. refer to positions in the word.
- The formula $a(x)$ asserts that the letter at position x is a.
- The quantifiers have the usual meaning.
- The formula $y > x$ is true if the position y appears somewhere to the right of the position x.

A word w satisfies φ only if for any position (x) with the letter a, there is some position to its right (y) with the letter b.

$$L(\varphi) \quad = \quad b^*(aa^*bb^*)^*$$

First-Order Logic over words

The formula

$$\forall x.\ \forall y.\ (a(x) \wedge a(y)) \implies x = y$$

is true of all words that have at most one a.

First-Order Logic over words

The formula

$$\forall x.\ \forall y.\ (a(x) \wedge a(y)) \implies x = y$$

is true of all words that have at most one a.
The formula

$$\mathrm{First}(x) \triangleq \forall y.(x = y) \vee (x < y)$$

evaluates to true at a position x if and only if it is the first position in the word.

First-Order Logic over words

The formula

$$\forall x.\ \forall y.\ (a(x) \wedge a(y)) \implies x = y$$

is true of all words that have at most one a.
The formula

$$\mathrm{First}(x) \stackrel{\triangle}{=} \forall y.(x = y) \vee (x < y)$$

evaluates to true at a position x if and only if it is the first position in the word. Thus

$$\forall x.(\mathrm{First}(x) \implies a(x))$$

identifies all the words that begin with an a.

LTL to FO over Words

- LTL formulas are interpreted at a pair w, i.

LTL to FO over Words

- LTL formulas are interpreted at a pair w, i.
- Translated to FO formulas with a single free variable.

LTL to FO over Words

- LTL formulas are interpreted at a pair w, i.
- Translated to FO formulas with a single free variable.

$$
\begin{array}{rcl}
\mathcal{T}(a) & = & a(x) \\
\mathcal{T}(X\alpha) & = & \exists y.\ (y = x + 1) \wedge \mathcal{T}(\alpha)[y/x] \\
\mathcal{T}(\varphi \mathsf{U} \psi) & = & \exists y.\ (y \geq x) \wedge \mathcal{T}(\psi)[y/x] \wedge \\
 & & \quad \forall z.(x \leq z < y) \implies \mathcal{T}(\varphi)[z/x]
\end{array}
$$

LTL to FO over Words

- LTL formulas are interpreted at a pair w, i.
- Translated to FO formulas with a single free variable.

$$\begin{array}{rcl}
\mathcal{T}(a) & = & a(x) \\
\mathcal{T}(X\alpha) & = & \exists y.\ (y = x + 1) \wedge \mathcal{T}(\alpha)[y/x] \\
\mathcal{T}(\varphi \mathsf{U} \psi) & = & \exists y.\ (y \geq x) \wedge \mathcal{T}(\psi)[y/x] \wedge \\
 & & \quad \forall z.(x \leq z < y) \implies \mathcal{T}(\varphi)[z/x]
\end{array}$$

- $w, i \models \mathcal{T}(\varphi) \quad \iff \quad w, i \models \varphi$.

LTL to FO over Words

- LTL formulas are interpreted at a pair w, i.
- Translated to FO formulas with a single free variable.

$$
\begin{array}{rcl}
\mathcal{T}(a) & = & a(x) \\
\mathcal{T}(X\alpha) & = & \exists y.\ (y = x+1) \wedge \mathcal{T}(\alpha)[y/x] \\
\mathcal{T}(\varphi \mathsf{U} \psi) & = & \exists y.\ (y \geq x) \wedge \mathcal{T}(\psi)[y/x] \wedge \\
 & & \quad \forall z. (x \leq z < y) \implies \mathcal{T}(\varphi)[z/x]
\end{array}
$$

- $w, i \models \mathcal{T}(\varphi) \iff w, i \models \varphi$.
- $\mathcal{T}(\varphi)$ uses at the most 3 variables. So, LTL is expressible in FO(3).

Complexity of LTL and FO

Satisfiability: Given a formula φ determine whether there is some word w such tha $w \models \varphi$.

Complexity of LTL and FO

Satisfiability: Given a formula φ determine whether there is some word w such tha $w \models \varphi$.

Theorem: (Clarke-Sistla) Satisfiability problem for LTL formulas is PSPACE complete.

Complexity of LTL and FO

Satisfiability: Given a formula φ determine whether there is some word w such tha $w \models \varphi$.

Theorem: (Clarke-Sistla) Satisfiability problem for LTL formulas is PSPACE complete.

In particular, there is a satisfiability checking algorithm that runs in time $2^{|\varphi|}$.

Complexity of LTL and FO

Satisfiability: Given a formula φ determine whether there is some word w such tha $w \models \varphi$.

Theorem: (Clarke-Sistla) Satisfiability problem for LTL formulas is PSPACE complete.

In particular, there is a satisfiability checking algorithm that runs in time $2^{|\varphi|}$.

Not very different from the best known for propositional formulas.

Complexity of LTL and FO

Satisfiability: Given a formula φ determine whether there is some word w such tha $w \models \varphi$.

Theorem: (Clarke-Sistla) Satisfiability problem for LTL formulas is PSPACE complete.

In particular, there is a satisfiability checking algorithm that runs in time $2^{|\varphi|}$.

Not very different from the best known for propositional formulas.

What about FO?

Complexity of LTL and FO

Satisfiability: Given a formula φ determine whether there is some word w such tha $w \models \varphi$.

Theorem: (Clarke-Sistla) Satisfiability problem for LTL formulas is PSPACE complete.

In particular, there is a satisfiability checking algorithm that runs in time $2^{|\varphi|}$.

Not very different from the best known for propositional formulas.

Theorem: (Albert Meyer) Satisfiability checking for FO over words is non-elementary.

Complexity of LTL and FO

Satisfiability: Given a formula φ determine whether there is some word w such tha $w \models \varphi$.

Theorem: (Clarke-Sistla) Satisfiability problem for LTL formulas is PSPACE complete.

In particular, there is a satisfiability checking algorithm that runs in time $2^{|\varphi|}$.

Not very different from the best known for propositional formulas.

Theorem: (Albert Meyer) Satisfiability checking for FO over words is non-elementary.

Conclusion: FO seems to be a stronger logic than LTL.

Expressive Completeness of LTL

Theorem: (Kamp) LTL is as expressive as FO over words.

Expressive Completeness of LTL

Theorem: (Kamp) LTL is as expressive as FO over words.

- Kamp's logic uses "future" and "past" modalities.

Expressive Completeness of LTL

Theorem: (Kamp) LTL is as expressive as FO over words.

- Kamp's logic uses "future" and "past" modalities.
- Gabbay, Pnueli, Shelah and Stavi: Expressive completeness for the future fragment.

Expressive Completeness of LTL

Theorem: (Kamp) LTL is as expressive as FO over words.

- Kamp's logic uses "future" and "past" modalities.
- Gabbay, Pnueli, Shelah and Stavi: Expressive completeness for the future fragment.
- Other proofs: Cohen, Perrin and Pin, Thomas Wilke.

Expressive Completeness of LTL

Theorem: (Kamp) LTL is as expressive as FO over words.

- Kamp's logic uses "future" and "past" modalities.
- Gabbay, Pnueli, Shelah and Stavi: Expressive completeness for the future fragment.
- Other proofs: Cohen, Perrin and Pin, Thomas Wilke.

Wilke's proof uses a simple double induction. Has been generalized to Mazurkiewicz traces.

Expressive Completeness of LTL

Theorem: (Kamp) LTL is as expressive as FO over words.

- Kamp's logic uses "future" and "past" modalities.
- Gabbay, Pnueli, Shelah and Stavi: Expressive completeness for the future fragment.
- Other proofs: Cohen, Perrin and Pin, Thomas Wilke.

Wilke's proof uses a simple double induction. Has been generalized to Mazurkiewicz traces.

Our presentation shall follow a variation of Wilke's proof due to Volker Diekert and Paul Gastin.

Expressive Completeness of LTL

Theorem: (Kamp) LTL is as expressive as FO over words.

- Kamp's logic uses "future" and "past" modalities.
- Gabbay, Pnueli, Shelah and Stavi: Expressive completeness for the future fragment.
- Other proofs: Cohen, Perrin and Pin, Thomas Wilke.

Wilke's proof uses a simple double induction. Has been generalized to Mazurkiewicz traces.

Our presentation shall follow a variation of Wilke's proof due to Volker Diekert and Paul Gastin.

The rest of this talk and the next would be devoted to proving this result.

An Outline of the proof

1. Characterize the languages defined by FO.

An Outline of the proof

1. Characterize the languages defined by FO.
    1. Every FO formula defines a regular language.

An Outline of the proof

1. Characterize the languages defined by FO.
    1. Every FO formula defines a regular language.
    2. Every regular language is recognized by a finite monoid.

An Outline of the proof

1. Characterize the languages defined by FO.
    1. Every FO formula defines a regular language.
    2. Every regular language is recognized by a finite monoid.
    3. Every FO formula defines a regular language recognized by an aperiodic monoid.

An Outline of the proof

1. Characterize the languages defined by FO.
    1. Every FO formula defines a regular language.
    2. Every regular language is recognized by a finite monoid.
    3. Every FO formula defines a regular language recognized by an aperiodic monoid.

    Ehrenfeucht-Fraisse Games

An Outline of the proof

1. Characterize the languages defined by FO.
    1. Every FO formula defines a regular language.
    2. Every regular language is recognized by a finite monoid.
    3. Every FO formula defines a regular language recognized by an aperiodic monoid.

    Ehrenfeucht-Fraisse Games

2. Transform aperiodic monoids into equivalent LTL formulas.

    Wilke's technique.

Formulas with free variables

Let φ be a FO formula with free variables $x_1, \ldots x_k$.

Formulas with free variables

Let φ be a FO formula with free variables $x_1, \ldots x_k$. A model of φ: A word w along with an assignment of positions to $x_1, x_2 \ldots x_k$.

Formulas with free variables

Let φ be a FO formula with free variables $x_1, \ldots x_k$. A model of φ: A word w along with an assignment of positions to $x_1, x_2 \ldots x_k$.

Example: $\phi = (x < y) \wedge a(x) \wedge b(y)$.

The *bacabc* with x assigned position 2 and y assigned position 5 satisfies ϕ.

Formulas with free variables

Let φ be a FO formula with free variables $x_1, \ldots x_k$. A model of φ: A word w along with an assignment of positions to $x_1, x_2 \ldots x_k$.

Example: $\phi = (x < y) \wedge a(x) \wedge b(y)$.

The *bacabc* with x assigned position 2 and y assigned position 5 satisfies ϕ.

Model as a word decorated with the variables x and y.

$$\begin{array}{cccccc} b & a & c & a & b & c \\ & x & & & y & \end{array}$$

Formulas with free variables

Let φ be a FO formula with free variables $x_1, \ldots x_k$. A model of φ: A word w along with an assignment of positions to $x_1, x_2 \ldots x_k$.

Example: $\phi = (x < y) \wedge a(x) \wedge b(y)$.

Another decorated word:

$$\begin{array}{cccccc} b & a & c & a & b & c \\ & & & & x & \\ & & & & y & \end{array}$$

ϕ is not satisifed by this word.

Formulas with free variables

Let φ be a FO formula with free variables $x_1, \ldots x_k$. A model of φ: A word w along with an assignment of positions to $x_1, x_2 \ldots x_k$.

Example: $\phi = (x < y) \wedge a(x) \wedge b(y)$.

Any formula defines a language of decorated words

Decorated word models

A decorated word is a word over the alphabet $\Sigma \times 2^V$, where V is a set of free variables.

Words corresponding to the decorated words:

$$\begin{array}{cccccc} b & a & c & a & b & c \\ & x & & & y & \end{array}$$

is $(b, \emptyset)(a, \{x\})(c, \emptyset)(a, \emptyset)(b, \{y\})(c, \emptyset)$.

Decorated word models

A decorated word is a word over the alphabet $\Sigma \times 2^V$, where V is a set of free variables.

Words corresponding to the decorated words:

$$\begin{array}{cccccc} b & a & c & a & b & c \\ & & & & x & \\ & & & & y & \end{array}$$

is $(b, \emptyset)(a, \emptyset)(c, \emptyset)(a, \emptyset)(b, \{x, y\})(c, \emptyset)$.

Decorated word models

A decorated word is a word over the alphabet $\Sigma \times 2^V$, where V is a set of free variables.

Words corresponding to the decorated words:

$$\begin{array}{cccccc} b & a & c & a & b & c \\ & & & & x & \\ & & & & y & \end{array}$$

is $(b, \emptyset)(a, \emptyset)(c, \emptyset)(a, \emptyset)(b, \{x, y\})(c, \emptyset)$.

A V-word is a word $(a_1, U_1)(a_2, U_2)\ldots(a_k, U_k)$ with

1. $U_i \cap U_j = \emptyset$ for $i \neq j$.
2. $\bigcup_{1 \leq i \leq k} U_i = V$.

$L(\varphi)$ is a language of V-words for any V with $\mathrm{free}(\varphi) \subseteq V$.

Stratifying FO formulas

A natural measure of the complexity of a FO formula is its quantifier-depth.

$$
\begin{array}{lcll}
\mathrm{qd}(\varphi) & = & 0 & \text{if } \varphi \text{ is an atomic formula} \\
\mathrm{qd}(\varphi \wedge \psi) & = & \mathsf{Maximum}(\mathrm{qd}(\varphi), \mathrm{qd}(\psi)) & \\
\mathrm{qd}(\neg\varphi) & = & \mathrm{qd}(\varphi) & \\
\mathrm{qd}(\exists \mathrm{x}.\varphi) & = & 1 + \mathrm{qd}(\varphi) &
\end{array}
$$

Stratifying FO formulas

A natural measure of the complexity of a FO formula is its quantifier-depth.

$$
\begin{array}{lcll}
\mathrm{qd}(\varphi) & = & 0 & \text{if } \varphi \text{ is an atomic formula} \\
\mathrm{qd}(\varphi \wedge \psi) & = & \text{Maximum}(\mathrm{qd}(\varphi), \mathrm{qd}(\psi)) & \\
\mathrm{qd}(\neg\varphi) & = & \mathrm{qd}(\varphi) & \\
\mathrm{qd}(\exists \mathrm{x}.\varphi) & = & 1 + \mathrm{qd}(\varphi) &
\end{array}
$$

Theorem: For any i there are only finitely many formulas of quantifier depth i or less (upto logical equivalence).

Stratifying FO formulas

Why are we doing all this?

Stratifying FO formulas

Why are we doing all this?

This allows us to establish properties of FO via induction.

For example, we could show, by induction on quantifier-depth, that any language definable in FO is a regular language.

Stratifying FO formulas

Why are we doing all this?

This allows us to establish properties of FO via induction.

For example, we could show, by induction on quantifier-depth, that any language definable in FO is a regular language.

To do this we need an alternative characterization of quantifier-depth.

FO(k) definability

Question: When is L definable in FO(k)?
or equivalently
Question: When is L not definable in FO(k)?

FO(k) definability

Question: When is L definable in FO(k)?
or equivalently
Question: When is L not definable in FO(k)?

Find a pair of words w, w' such that

1. $w \in L$, $w' \notin L$.
2. $\forall \phi \in FO(k).\ (w \models \phi) \iff (w' \models \phi)$.

FO(k) definability

Question: When is L definable in FO(k)?
or equivalently
Question: When is L not definable in FO(k)?

Find a pair of words w, w' such that

1. $w \in L$, $w' \notin L$.
2. $\forall \phi \in FO(k).\ (w \models \phi) \iff (w' \models \phi)$.

Question: When are two words distinguishable by FO(k) ?

FO(k) definability

Question: When is L definable in FO(k)?
or equivalently
Question: When is L not definable in FO(k)?

Find a pair of words w, w' such that

1. $w \in L$, $w' \notin L$.
2. $\forall \phi \in FO(k). \ (w \models \phi) \iff (w' \models \phi)$.

Question: When are two words distinguishable by FO(k) ?

EF-Games: Set up k-round two player game (between say player 0 and player 1) based on w and w'. Relate winning strategies to distinguishability.

The Game

Let w, w' be two words V-words and let k be an integer. The k round EF-game consists of the two players making k moves. In round i:

The Game

Let w, w' be two words V-words and let k be an integer. The k round EF-game consists of the two players making k moves. In round i:

1. Player 0 (who is trying to show that the two words are distinguishable) picks one of the two words and a position p in that word and labels it with x_i.

The Game

Let w, w' be two words V-words and let k be an integer. The k round EF-game consists of the two players making k moves. In round i:

1. Player 0 (who is trying to show that the two words are distinguishable) picks one of the two words and a position p in that word and labels it with x_i.
2. Player 1 must then pick the other word (i.e. the one not picked by player 0 in round i), pick some position p' and label it with x_i.

The Game

Let w, w' be two words V-words and let k be an integer. The k round EF-game consists of the two players making k moves. In round i:

1. Player 0 (who is trying to show that the two words are distinguishable) picks one of the two words and a position p in that word and labels it with x_i.
2. Player 1 must then pick the other word (i.e. the one not picked by player 0 in round i), pick some position p' and label it with x_i.

Let W and W' be the two $V \cup \{x_1, x_2 \ldots, x_k\}$ words resulting from the k-round game.

The Game

Let w, w' be two words V-words and let k be an integer. The k round EF-game consists of the two players making k moves. In round i:

1. Player 0 (who is trying to show that the two words are distinguishable) picks one of the two words and a position p in that word and labels it with x_i.
2. Player 1 must then pick the other word (i.e. the one not picked by player 0 in round i), pick some position p' and label it with x_i.

Let W and W' be the two $V \cup \{x_1, x_2 \ldots, x_k\}$ words resulting from the k-round game.

1. If W and W' are distinguishable by atomic formulas then Player 0 is the winner.

The Game

Let w, w' be two words V-words and let k be an integer. The k round EF-game consists of the two players making k moves. In round i:

1. Player 0 (who is trying to show that the two words are distinguishable) picks one of the two words and a position p in that word and labels it with x_i.
2. Player 1 must then pick the other word (i.e. the one not picked by player 0 in round i), pick some position p' and label it with x_i.

Let W and W' be the two $V \cup \{x_1, x_2 \ldots, x_k\}$ words resulting from the k-round game.

1. If W and W' are distinguishable by atomic formulas then Player 0 is the winner.
2. Otherwise Player 1 is the winner.

An Example

Consider the words *abba* and *ababa*. Here is a winning strategy for Player 0.

An Example

Consider the words *abba* and *ababa*. Here is a winning strategy for Player 0.

- Pick the first word and position 3.

An Example

Consider the words *abba* and *ababa*. Here is a winning strategy for Player 0.

- Pick the first word and position 3.
- No matter how Player 1 responded, pick the first word and position 2.

An Example

Consider the words *abba* and *ababa*. Here is a winning strategy for Player 0.

- Pick the first word and position 3.
- No matter how Player 1 responded, pick the first word and position 2.
- If the positions picked by player 1 are not 2 and 4, Player 0 has already won.

An Example

Consider the words *abba* and *ababa*. Here is a winning strategy for Player 0.

- Pick the first word and position 3.
- No matter how Player 1 responded, pick the first word and position 2.
- If the positions picked by player 1 are not 2 and 4, Player 0 has already won.
- Otherwise, pick the second word and position 3.

EF Games ...

Theorem: (Ehrenfeucht,Fraisse) Player 0 has a winning strategy in the k round game on w, w' if and only if there is a FO(k) formula that distinguishes w and w'.

EF Games ...

Theorem: (Ehrenfeucht,Fraisse) Player 0 has a winning strategy in the k round game on w, w' if and only if there is a FO(k) formula that distinguishes w and w'.

The proof is an easy inductive argument.

Note that any distinuishing formula dictates a winning strategy for player 0.

EF Games ...

Theorem: (Ehrenfeucht,Fraisse) Player 0 has a winning strategy in the k round game on w, w' if and only if there is a FO(k) formula that distinguishes w and w'.

Example: Consider the words

$$a \quad b \quad b \quad a \quad b \quad b \quad a \quad b$$

$$a \quad b \quad a \quad b \quad b \quad a \quad b \quad b$$

Here is a distinguishing formula:

$$\exists x_1.\ (\ b(x_1) \wedge \exists x_2.\ (x_1 < x_2) \wedge \forall x_2 > x_1.\ b(x_2)\ )$$

EF Games ...

Theorem: (Ehrenfeucht,Fraisse) Player 0 has a winning strategy in the k round game on w, w' if and only if there is a FO(k) formula that distinguishes w and w'.

$$a \quad b \quad b \quad a \quad b \quad b \quad a \quad b$$

$$\begin{array}{cccccccc} a & b & a & b & b & a & b & b \\ & & & & & & x_1 & \end{array}$$

Here is a distinguishing formula:

$$b(x_1) \wedge \exists x_2.\ (x_1 < x_2) \wedge \forall x_2 > x_1.\ b(x_2)$$

EF Games ...

Theorem: (Ehrenfeucht,Fraisse) Player 0 has a winning strategy in the k round game on w, w' if and only if there is a FO(k) formula that distinguishes w and w'.

$$\begin{array}{cccccccc} a & b & b & a & b & b & a & b \\ & & & & & & & x_1 \\ a & b & a & b & b & a & b & b \\ & & & & & & x_1 & x_2 \end{array}$$

Distinguishing formula:

$$\exists x_2.\ (x_1 < x_2)$$

EF Games ...

Theorem: (Ehrenfeucht,Fraisse) Player 0 has a winning strategy in the k round game on w, w' if and only if there is a FO(k) formula that distinguishes w and w'.

$$\begin{array}{cccccccc} a & b & b & a & b & b & a & b \\ & & x_1 & & & & & \\ a & b & a & b & b & a & b & b \\ & & & & & & x_1 & \end{array}$$

Here is a distinguishing formula:

$$\forall x_2 > x_1.\ b(x_2)$$

EF Games ...

Theorem: (Ehrenfeucht,Fraisse) Player 0 has a winning strategy in the k round game on w, w' if and only if there is a FO(k) formula that distinguishes w and w'.

$$\begin{array}{cccccccc}
a & b & b & a & b & b & a & b \\
 & & x_1 & & & & x_2 & \\
a & b & a & b & b & a & b & b \\
 & & & & & & x_1 &
\end{array}$$

EF Games ...

Theorem: (Ehrenfeucht,Fraisse) Player 0 has a winning strategy in the k round game on w, w' if and only if there is a FO(k) formula that distinguishes w and w'.

Conversely, winning strategies for Player 0 can be turned into distinguishing formulas.

k-equivalence

Two words w and w' are said to be *k-equivalent* if they are indistinguishable by formulas with quantifier depth k or less.

$$w \equiv_k w'$$

$abbabbab$ and $ababbabb$ are 1-equivalent but not 2-equivalent.

k-equivalence

Two words w and w' are said to be *k-equivalent* if they are indistinguishable by formulas with quantifier depth k or less.

$$w \equiv_k w'$$

abbabbab and *ababbabb* are 1-equivalent but not 2-equivalent.

- $\equiv_k$ is of finite index.

k-equivalence

Two words w and w' are said to be *k-equivalent* if they are indistinguishable by formulas with quantifier depth k or less.

$$w \equiv_k w'$$

abbabbab and *ababbabb* are 1-equivalent but not 2-equivalent.

- $\equiv_k$ is of finite index.
- Let φ be a FO(k) formula. Then $L(\varphi)$ is a (disjoint) union of some of the equivalence classes of $\equiv_k$.

FO definable languages are Regular

Theorem: (Myhill-Nerode) A language L is regular if and only if it is the union of some of the equivalence classes of a right-invariant equivalence relation of finite index.

It suffices to show that $\equiv_k$ is right-invariant.

FO definable languages are Regular

Theorem: (Myhill-Nerode) A language L is regular if and only if it is the union of some of the equivalence classes of a right-invariant equivalence relation of finite index.

It suffices to show that $\equiv_k$ is right-invariant.

- x and y are k-equivalent and z is any word.

FO definable languages are Regular

Theorem: (Myhill-Nerode) A language L is regular if and only if it is the union of some of the equivalence classes of a right-invariant equivalence relation of finite index.

It suffices to show that $\equiv_k$ is right-invariant.

- x and y are k-equivalent and z is any word.
- Player 1 has winning strategy in the k round game on x and y.

FO definable languages are Regular

Theorem: (Myhill-Nerode) A language L is regular if and only if it is the union of some of the equivalence classes of a right-invariant equivalence relation of finite index.

It suffices to show that $\equiv_k$ is right-invariant.

- x and y are k-equivalent and z is any word.
- Player 1 has winning strategy in the k round game on x and y.
- What about the k-round game on xz and yz ?

  Simulate strategy on x and y, duplicate moves on z.

FO definable languages are Regular

Theorem: (Myhill-Nerode) A language L is regular if and only if it is the union of some of the equivalence classes of a right-invariant equivalence relation of finite index.

It suffices to show that $\equiv_k$ is right-invariant.

- x and y are k-equivalent and z is any word.
- Player 1 has winning strategy in the k round game on x and y.
- What about the k-round game on xz and yz ?

  Simulate strategy on x and y, duplicate moves on z.

Theorem: Every First order definable language of words is regular.

A Non-FO definable Language

Claim: The words a^m and a^{m+1} are k-equivalent whenever $m > 2^k$.

A Non-FO definable Language

Claim: The words a^m and a^{m+1} are k-equivalent whenever $m > 2^k$.

The proof is by induction on k.

A Non-FO definable Language

Claim: The words a^m and a^{m+1} are k-equivalent whenever $m > 2^k$.

The proof is by induction on k.

- Clearly $a \equiv_0 aa$.

A Non-FO definable Language

Claim: The words a^m and a^{m+1} are k-equivalent whenever $m > 2^k$.

The proof is by induction on k.

- Clearly $a \equiv_0 aa$.
- Player 0 will pick one of the two words and pick a position in that word and label it with x to give

$$a^s.(a,x).a^t$$

where $s + t = m$ or $s + t + 1 = m$.

A Non-FO definable Language

Claim: The words a^m and a^{m+1} are k-equivalent whenever $m > 2^k$.

The proof is by induction on k.

- Clearly $a \equiv_0 aa$.
- Player 0 will pick one of the two words and pick a position in that word and label it with x to give

$$a^s.(a,x).a^t$$

  where $s + t = m$ or $s + t + 1 = m$.
- Suppose $s \leq t$. Player 1 breaks up the other word as

$$a^s.(a,x).a^{t'}$$

  with $s + t' = m$ or $s + t' + 1 = m$.

A Non-FO definable Language

$$a^s.(a,x).a^t \qquad\qquad a^s.(a,x).a^{t'}$$

From now on:

A Non-FO definable Language

$$a^s.(a,x).a^t \qquad\qquad a^s.(a,x).a^{t'}$$

From now on:

- On s duplicate moves.

A Non-FO definable Language

$$a^s.(a,x).a^t \qquad\qquad a^s.(a,x).a^{t'}$$

From now on:

- On s duplicate moves.
- $t, t' > 2^{k-1}$ and differ by 1. On $a^t, a^{t'}$ use the winning strategy that exists by the induction hypothesis.

A Non-FO definable Language

$$a^s.(a,x).a^t \qquad\qquad a^s.(a,x).a^{t'}$$

From now on:

- On s duplicate moves.
- $t, t' > 2^{k-1}$ and differ by 1. On $a^t, a^{t'}$ use the winning strategy that exists by the induction hypothesis.
- Player 1 has a winning strategy!

A Non-FO definable Language

$$a^s.(a,x).a^t \qquad\qquad a^s.(a,x).a^{t'}$$

From now on:

- On s duplicate moves.
- $t, t' > 2^{k-1}$ and differ by 1. On $a^t, a^{t'}$ use the winning strategy that exists by the induction hypothesis.
- Player 1 has a winning strategy!

Theorem: $\{a^{2i} \mid i \geq 1\}$ is not a FO definable language.

A Non-FO definable Language

$a^s.(a,x).a^t$ $\qquad\qquad$ $a^s.(a,x).a^{t'}$

From now on:

- On s duplicate moves.
- $t, t' > 2^{k-1}$ and differ by 1. On $a^t, a^{t'}$ use the winning strategy that exists by the induction hypothesis.
- Player 1 has a winning strategy!

Theorem: $\{a^{2i} \mid i \geq 1\}$ is not a FO definable language.

Theorem: For all $m > 2^k$ and $w \in \Sigma^+$, w^m and w^{m+1} are k-equivalent.

A Non-FO definable Language

$$a^s.(a,x).a^t \qquad\qquad a^s.(a,x).a^{t'}$$

From now on:

- On s duplicate moves.
- $t, t' > 2^{k-1}$ and differ by 1. On $a^t, a^{t'}$ use the winning strategy that exists by the induction hypothesis.
- Player 1 has a winning strategy!

Theorem: $\{a^{2i} \mid i \geq 1\}$ is not a FO definable language.

Theorem: For all $m > 2^k$ and $w \in \Sigma^+$, w^m and w^{m+1} are k-equivalent.

The latter asserts that FO definable languages are aperiodic.

Monoids as Language recognizers

Let $(M, ., 1)$ be a finite monoid. Let $h : \Sigma^* \longrightarrow M$ be a morphism.

Theorem: For any $X \subseteq M$, $h^{-1}(X)$ is a regular language.

Monoids as Language recognizers

Let $(M, ., 1)$ be a finite monoid. Let $h : \Sigma^* \longrightarrow M$ be a morphism.

Theorem: For any $X \subseteq M$, $h^{-1}(X)$ is a regular language.

Let $A_M = (M, \Sigma, \delta, 1, X)$ with $\delta(m, a) = m.h(a)$. Then,

$$L(A) = h^{-1}(X)$$

Monoids as Language recognizers

Let $(M, ., 1)$ be a finite monoid. Let $h : \Sigma^* \longrightarrow M$ be a morphism.

Theorem: For any $X \subseteq M$, $h^{-1}(X)$ is a regular language.

Let $A_M = (M, \Sigma, \delta, 1, X)$ with $\delta(m, a) = m.h(a)$. Then,

$$L(A) \quad = \quad h^{-1}(X)$$

We say that $L = h^{-1}(X)$ is recognized by the monoid M.

Monoids as Language recognizers

Let $(M, ., 1)$ be a finite monoid. Let $h : \Sigma^* \longrightarrow M$ be a morphism.

Theorem: For any $X \subseteq M$, $h^{-1}(X)$ is a regular language.

Let $A_M = (M, \Sigma, \delta, 1, X)$ with $\delta(m, a) = m.h(a)$. Then,

$$L(A) \quad = \quad h^{-1}(X)$$

We say that $L = h^{-1}(X)$ is recognized by the monoid M.

The Syntactic Monoid of a Regular Language:

Monoids as Language recognizers

Let $(M, ., 1)$ be a finite monoid. Let $h : \Sigma^* \longrightarrow M$ be a morphism.

Theorem: For any $X \subseteq M$, $h^{-1}(X)$ is a regular language.

Let $A_M = (M, \Sigma, \delta, 1, X)$ with $\delta(m, a) = m.h(a)$. Then,

$$L(A) \quad = \quad h^{-1}(X)$$

We say that $L = h^{-1}(X)$ is recognized by the monoid M.

The Syntactic Monoid of a Regular Language:

- Let $x \equiv_L y$ iff $\forall u, v.\ uxv \in L \iff uyv \in L$.

Monoids as Language recognizers

Let $(M, ., 1)$ be a finite monoid. Let $h : \Sigma^* \longrightarrow M$ be a morphism.

Theorem: For any $X \subseteq M$, $h^{-1}(X)$ is a regular language.

Let $A_M = (M, \Sigma, \delta, 1, X)$ with $\delta(m, a) = m.h(a)$. Then,

$$L(A) \quad = \quad h^{-1}(X)$$

We say that $L = h^{-1}(X)$ is recognized by the monoid M.

The Syntactic Monoid of a Regular Language:

- Let $x \equiv_L y$ iff $\forall u, v.\ uxv \in L \iff uyv \in L$.
- $\equiv_L$ is a congruence on Σ^*.

Monoids as Language recognizers

Let $(M, ., 1)$ be a finite monoid. Let $h : \Sigma^* \longrightarrow M$ be a morphism.

Theorem: For any $X \subseteq M$, $h^{-1}(X)$ is a regular language.

Let $A_M = (M, \Sigma, \delta, 1, X)$ with $\delta(m, a) = m.h(a)$. Then,

$$L(A) \quad = \quad h^{-1}(X)$$

We say that $L = h^{-1}(X)$ is recognized by the monoid M.

The Syntactic Monoid of a Regular Language:

- Let $x \equiv_L y$ iff $\forall u, v.\ uxv \in L \iff uyv \in L$.
- $\equiv_L$ is a congruence on Σ^*.
- $SYN(L) = (\Sigma^*/\equiv_L, ., [\epsilon]_{\equiv_L})$ is a finite monoid.

Monoids recognize Regular languages

Let $\eta_L : \Sigma^* \longrightarrow SYN(L)$ be the morphism

$$\eta_L(x) = [x]_{\equiv_L}$$

Then,

$$L = \bigcup_{x \in L} \eta_L^{-1}([x]_{\equiv_L})$$

Theorem: A language is regular if and only if it is recognized by a finite monoid.

Aperiodic Monoids

A Monoid M is said to be aperiodic iff there is an integer N such that

$$a^k = a^{k+1} \quad \text{for all } k \geq N \text{ and } a \in M$$

A language L is aperiodic iff it is recognized by an aperiodic monoid.

Aperiodic Monoids

A Monoid M is said to be aperiodic iff there is an integer N such that

$$a^k \;=\; a^{k+1} \quad \text{for all } k \geq N \text{ and } a \in M$$

A language L is aperiodic iff it is recognized by an aperiodic monoid.

Theorem: $\Sigma^*/\equiv_k$ is an aperiodic monoid. Thus, every FO definable language is aperiodic.

Aperiodic Monoids

A Monoid M is said to be aperiodic iff there is an integer N such that

$$a^k = a^{k+1} \quad \text{for all } k \geq N \text{ and } a \in M$$

A language L is aperiodic iff it is recognized by an aperiodic monoid.

Theorem: $\Sigma^*/\equiv_k$ is an aperiodic monoid. Thus, every FO definable language is aperiodic.

This follows from the fact that $w^m \equiv_k w^{m+1}$ for all $m > 2^k$.

An useful result

If M is an aperiodic monoid and $x, y \in M$ and $x \neq y$ then, $x.y \neq 1$.

An useful result

If M is an aperiodic monoid and $x, y \in M$ and $x \neq y$ then, $x.y \neq 1$.

Suppose $x.y = 1$. Then, $x = x.x^N.y^N = x^N.y^N = 1$.

Similarly, $y = 1$, contradicting $x \neq y$.

Summary

- LTL is expressible in FO.

Summary

- LTL is expressible in FO.
- FO definable languages are regular. (Via EF Games)

Summary

- LTL is expressible in FO.
- FO definable languages are regular. (Via EF Games)
- FO definable languages are aperiodic. (Via EF Games, Syntacic Monoid)

Summary

- LTL is expressible in FO.
- FO definable languages are regular. (Via EF Games)
- FO definable languages are aperiodic. (Via EF Games, Syntactic Monoid)

Schutzenberger's Theorem: A regular language L is aperiodic if and only if it expressible as a star-free regular expression.