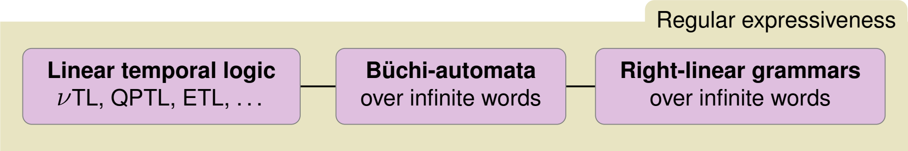
Linear Temporal Logics and Grammars

Joachim Baran

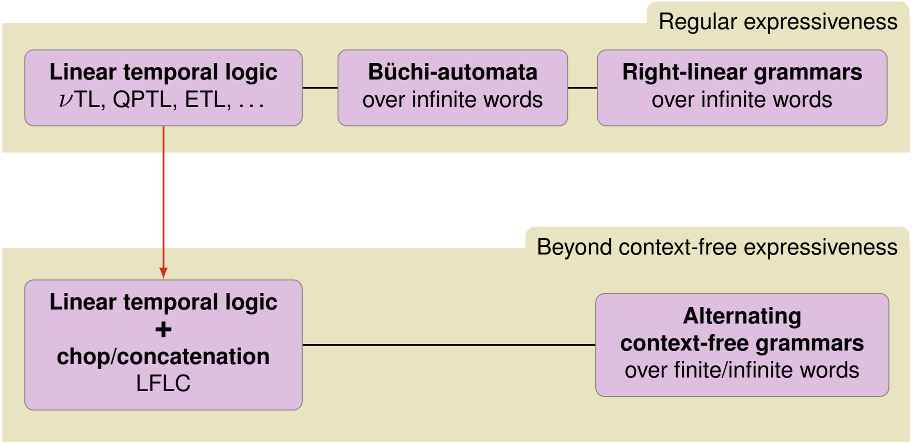
The University of Manchester

April 2006

Overview – Word Systems



Overview – Word Systems



Temporal Logic

Linear-time temporal logic with chop (LFLC):

- propositional constants $p, q, \ldots$
- special “empty” proposition ε
- connectives $\vee, \wedge$
- concatenation $;$
- fixed-point variables $X, Y, \ldots$
- fixed-point operators μ, ν

Temporal Logic

Linear-time temporal logic with chop (LFLC):

- propositional constants $p, q, \ldots$
- special "empty" proposition ε
- connectives $\vee, \wedge$
- concatenation $;$
- fixed-point variables $X, Y, \ldots$
- fixed-point operators μ, ν

Temporal Logic

Linear-time temporal logic with chop (LFLC):

- propositiona $p, q, \ldots$
- special "em ε
- connectives $\vee, \wedge$
- concatenati ;
- fixed-point v $X, Y, \ldots$
- fixed-point operators μ, ν

$$
\begin{aligned}
p &\equiv \{\neg a, \neg b\} \\
q &\equiv \{\neg a, b\} \\
r &\equiv \{a, \neg b\} \\
s &\equiv \{a, b\}
\end{aligned}
$$

Temporal Logic

Linear-time temporal logic with chop (LFLC):

- propositional constants $p, q, \ldots$
- special "empty" proposition ε
- connectives $\vee, \wedge$
- concatenation $;$
- fixed-point variables $X, Y, \ldots$
- fixed-point operators μ, ν

Models: $M \models p$

p

Temporal Logic

Linear-time temporal logic with chop (LFLC):

- propositional constants $p, q, \ldots$
- special "empty" proposition ε
- connectives $\vee, \wedge$
- concatenation $;$
- fixed-point variables $X, Y, \ldots$
- fixed-point operators μ, ν

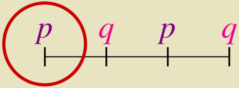
Models: $M \models p; q; p; q$

Temporal Logic

Linear-time temporal logic with chop (LFLC):

- propositional constants $p, q, \ldots$
- special "empty" proposition ε
- connectives $\vee, \wedge$
- concatenation $;$
- fixed-point variables $X, Y, \ldots$
- fixed-point operators μ, ν

Models: $M \models p; q; p; q$

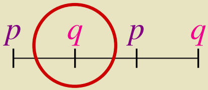


Temporal Logic

Linear-time temporal logic with chop (LFLC):

- propositional constants $p, q, \ldots$
- special "empty" proposition ε
- connectives $\vee, \wedge$
- concatenation $;$
- fixed-point variables $X, Y, \ldots$
- fixed-point operators μ, ν

Models: $M \models p; q; p; q$

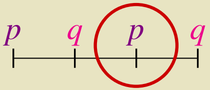


Temporal Logic

Linear-time temporal logic with chop (LFLC):

- propositional constants $p, q, \ldots$
- special "empty" proposition ε
- connectives $\vee, \wedge$
- concatenation $;$
- fixed-point variables $X, Y, \ldots$
- fixed-point operators μ, ν

Models: $M \models p; q; p; q$

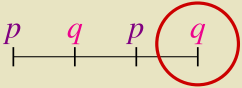


Temporal Logic

Linear-time temporal logic with chop (LFLC):

- propositional constants $p, q, \ldots$
- special "empty" proposition ε
- connectives $\vee, \wedge$
- concatenation $;$
- fixed-point variables $X, Y, \ldots$
- fixed-point operators μ, ν

Models: $M \models p; q; p; q$

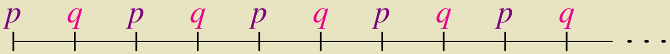


Temporal Logic

Linear-time temporal logic with chop (LFLC):

- propositional constants $p, q, \ldots$
- special "empty" proposition ε
- connectives $\vee, \wedge$
- concatenation $;$
- fixed-point variables $X, Y, \ldots$
- fixed-point operators μ, ν

Models: $M \models \nu X.(p; q; X)$

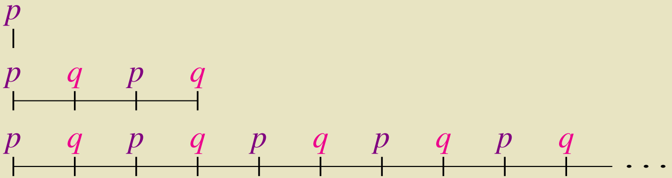


Temporal Logic

Linear-time temporal logic with chop (LFLC):

- propositional constants $p, q, \ldots$
- special "empty" proposition ε
- connectives $\vee, \wedge$
- concatenation $;$
- fixed-point variables $X, Y, \ldots$
- fixed-point operators μ, ν

Models: $M \models p \vee p; q; p; q \vee \nu X.(p; q; X)$



Grammars

Alternating Context-Free Grammar (ACFG):

- terminals $p, q, \ldots \in \Sigma$
- non-terminals $X, Y, \ldots \in N$
- production rules $N \to (N \cup \Sigma)^*$
- designated initial symbol $S \in N$
- alternation function $\lambda : N \to \{\forall, \exists\}$
- parity function $\Omega : N \to \mathbb{N}$

Grammars

Alternating Context-Free Grammar (ACFG):

- terminals $p, q, \ldots \in \Sigma$
- non-terminals $X, Y, \ldots \in N$
- production rules $N \to (N \cup \Sigma)^*$
- designated initial symbol $S \in N$
- alternation function $\lambda : N \to \{\forall, \exists\}$
- parity function $\Omega : N \to \mathbb{N}$

Grammars

Alternating Context-Free Grammar (ACFG):

- terminals $p, q, \ldots \in \Sigma$
- non-termi $X, Y, \ldots \in N$
- productio $N \to (N \cup \Sigma)^*$
- designate $S \in N$
- alternatio $\lambda : N \to \{\forall, \exists\}$
- parity function $\Omega : N \to \mathbb{N}$

$$\begin{aligned} X &\longrightarrow pq \\ X &\longrightarrow pYq \\ X &\longrightarrow \varepsilon \end{aligned}$$

Grammars

Alternating Context-Free Grammar (ACFG):

- terminals $p, q, \ldots \in \Sigma$
- non-terminals $X, Y, \ldots \in N$
- production rules $N \to (N \cup \Sigma)^*$
- designated initial symbol $S \in N$
- alternation function $\lambda : N \to \{\forall, \exists\}$
- parity function $\Omega : N \to \mathbb{N}$

Languages:

p

$$\mathcal{L}\left(\begin{array}{l} S \to p \\ \Omega(S) = 0 \end{array} \right)$$

Grammars

Alternating Context-Free Grammar (ACFG):

- terminals $p, q, \ldots \in \Sigma$
- non-terminals $X, Y, \ldots \in N$
- production rules $N \to (N \cup \Sigma)^*$
- designated initial symbol $S \in N$
- alternation function $\lambda : N \to \{\forall, \exists\}$
- parity function $\Omega : N \to \mathbb{N}$

Languages:

$$\mathcal{L}\left(\begin{array}{l} S \to pqpq \\ \Omega(S) = 0 \end{array}\right)$$

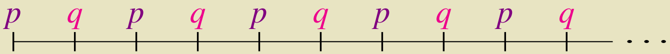
Grammars

Alternating Context-Free Grammar (ACFG):

- terminals $p, q, \ldots \in \Sigma$
- non-terminals $X, Y, \ldots \in N$
- production rules $N \to (N \cup \Sigma)^*$
- designated initial symbol $S \in N$
- alternation function $\lambda : N \to \{\forall, \exists\}$
- parity function $\Omega : N \to \mathbb{N}$

Languages:

$$\mathcal{L}\left(\begin{array}{l} S \to pqS \\ \Omega(S) = 0 \end{array}\right)$$



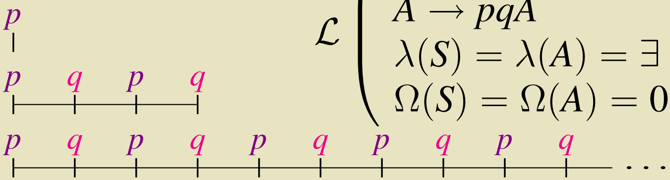
Grammars

Alternating Context-Free Grammar (ACFG):

- terminals $p, q, \ldots \in \Sigma$
- non-terminals $X, Y, \ldots \in N$
- production rules $N \rightarrow (N \cup \Sigma)^*$
- designated initial symbol $S \in N$
- alternation function $\lambda : N \rightarrow \{\forall, \exists\}$
- parity function $\Omega : N \rightarrow \mathbb{N}$

Languages:

$$\mathcal{L}\begin{pmatrix} S \rightarrow p \mid pqpq \mid A \\ A \rightarrow pqA \\ \lambda(S) = \lambda(A) = \exists \\ \Omega(S) = \Omega(A) = 0 \end{pmatrix}$$



Relationship of LFLC and ACFGs

$\mu X.(a; X \vee b)$	$\mathcal{L}\begin{pmatrix} S \to aS \mid b \\ \Omega(S) = 1 \end{pmatrix}$
$\mu X.(a; X \vee b)$	S

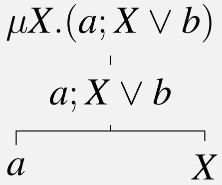
Relationship of LFLC and ACFGs

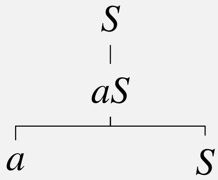
$$\mu X.(a; X \vee b) \qquad \mathcal{L}\left(\begin{array}{l} S \to aS \mid b \\ \Omega(S) = 1 \end{array} \right)$$

$$\begin{array}{c} \mu X.(a; X \vee b) \\ \mid \\ a; X \vee b \end{array} \qquad \begin{array}{c} S \\ \mid \\ aS \end{array}$$

Relationship of LFLC and ACFGs

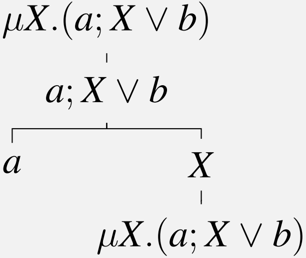
$$\mu X.(a; X \vee b) \qquad \mathcal{L}\left(\begin{array}{l} S \to aS \mid b \\ \Omega(S) = 1 \end{array}\right)$$

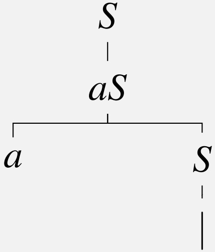




Relationship of LFLC and ACFGs

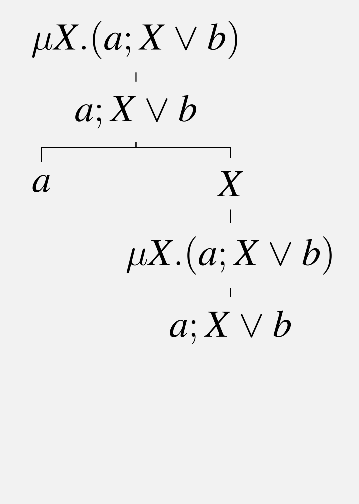
$$\mu X.(a; X \vee b) \qquad \mathcal{L}\begin{pmatrix} S \to aS \mid b \\ \Omega(S) = 1 \end{pmatrix}$$

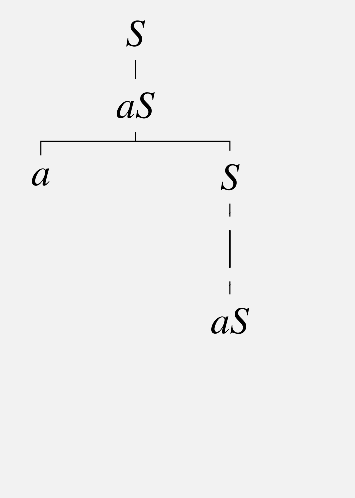




Relationship of LFLC and ACFGs

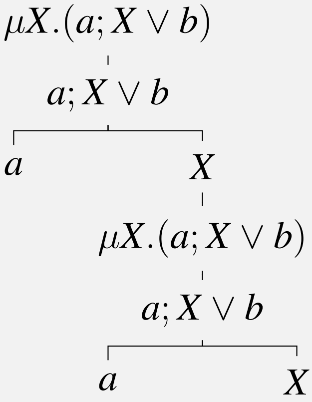
$$\mu X.(a; X \vee b) \qquad \mathcal{L}\begin{pmatrix} S \to aS \mid b \\ \Omega(S) = 1 \end{pmatrix}$$

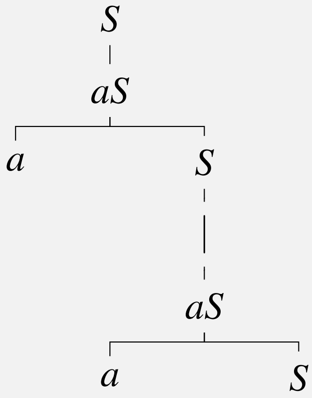




Relationship of LFLC and ACFGs

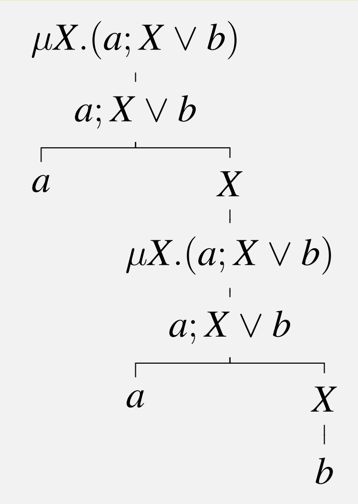
$$\mu X.(a; X \vee b) \qquad \mathcal{L}\left(\begin{array}{l} S \to aS \mid b \\ \Omega(S) = 1 \end{array} \right)$$

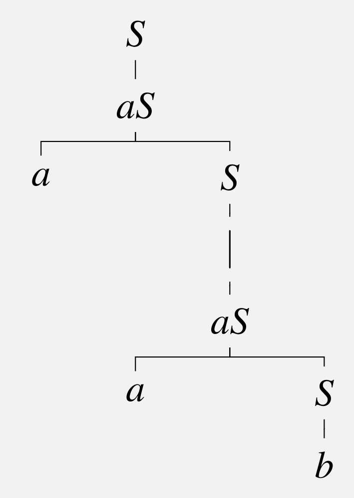




Relationship of LFLC and ACFGs

$$\mu X.(a; X \vee b) \qquad \mathcal{L}\left(\begin{array}{l} S \rightarrow aS \mid b \\ \Omega(S) = 1 \end{array}\right)$$

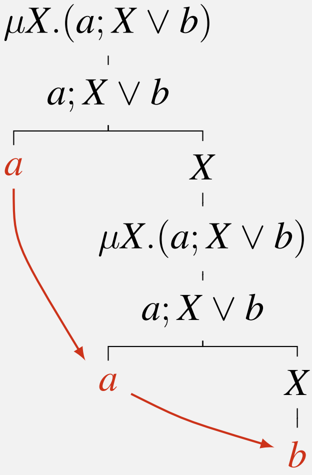


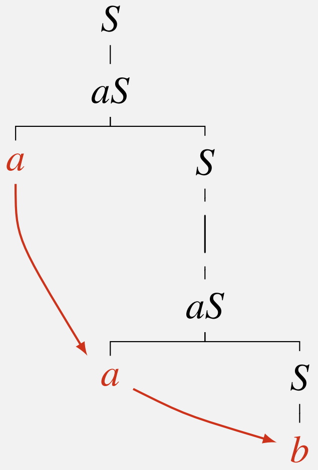


Relationship of LFLC and ACFGs

$\mu X.(a; X \vee b)$

$\mathcal{L}\begin{pmatrix} S \to aS \mid b \\ \Omega(S) = 1 \end{pmatrix}$





Expressiveness of LFLC and ACFGs

- beyond context-free expressiveness

$$\mathcal{L} = \{a^n b^n c^n \mid n \geq 0\}$$

- satisfiability is undecidable

$\mathcal{L} = \mathcal{L}_1 \cap \mathcal{L}_2$, where $\mathcal{L}_1$,$\mathcal{L}_2$ are context-free

due to

$\varphi = \varphi_1 \wedge \varphi_2$, φ_1,φ_2 are context-free

- model-checking of finite words and ultimately periodic infinite words is decidable

$$w \models \varphi \text{ or } w \in \mathcal{L}(G)$$

ε is Obsolete

As long as $\varepsilon \not\models \varphi$:

- φ can be rewritten as an ε-free formula φ'
- since $\varepsilon \overset{?}{\models} \varphi$ is decidable:
  - φ becomes either $\varphi' = \varphi$ or $\varphi' = \varphi \vee \varepsilon$
  - new initial symbol S', such that either $S' \to S$ or $S' \to S \mid \varepsilon$, $\lambda(S) = \exists$

$\Rightarrow$ LFLC's syntax/semantics is not concise

Alternation Hierarchy

- fixed-point alternation:

φ	$\mathcal{L}(\varphi)$	depth(φ)
$\nu X.(\mu Y.(\varepsilon \vee b; Y) \vee b; X)$	$(a^* b)^\omega$	0
$\nu X.\mu Y.(a; Y \vee b; X)$	$(a^* b)^\omega$	1

- private communication with Martin Lange:
  - Logic's fixed-point alternation:
    hierarchy collapses at level 0
  - Grammar's acceptance condition:
    parity acceptance is equiv. to weak parity acc.

c.f. his submitted "Specifying Non-Regular Properties of Runs"

Future Work

- true expressiveness of LFLC/ACFGs
  - proper inclusion in context-sensitive languages
- model-checking decidability reconsidered
  - non-periodic infinite words
- decidable model-checking fragments
  - extending $w \models \varphi$ to regular-$\mathcal{L} \models \varphi$