Extended classes of pomsets:
Some views on logic, regularity and automata

Jean Fanchon
LAAS–CNRS

Extending language theory to pomsets

Logics for pomsets

Notions of regularity

Relating regularity and MSO definability

Pomsets with autoconcurrency

Finitely generated pomsets without autoconcurrency
Traces, MSC, pomsets without autoconcurrency

\[
\begin{align*}
\text{a} &\rightarrow \text{c} \\
\text{b} &\rightarrow \text{d}
\end{align*}
\]

\[
\begin{align*}
\text{a} &\rightarrow \text{b} \\
\text{c} &\rightarrow \text{d}
\end{align*}
\]

\[
\begin{align*}
\{\text{a,b}\} &\rightarrow \text{f} \\
\{\text{f,g}\} &\rightarrow \text{g}
\end{align*}
\]

\[
\begin{align*}
\{\text{c}\} &\rightarrow \text{c} \\
\{\text{d,e}\} &\rightarrow \text{d}
\end{align*}
\]
Extended classes of pomsets

Words
Mazurkiewicz Traces

Message Sequence Charts
Local traces
Petri–Nets languages

Finitely generated sets

Without autoconcurrency

Prime bounded sets
Width bounded sets

Linear closed sets
Step closed sets

Pomsets with autoconcurrency
Co-rational expressions

Monoid recognisability

Logic MSO

Rational expressions

Automata

Series-rational expressions

Fork-join Automata

Series-parallel pomsets

Series-parallel pomsets

SP recognisability

Logic MSO

Words

Traces

Logic MSO

Logic MSO
Complementary Notions

Expressions (rational) $\rightarrow$ Syntax, process algebras

Automata, Petri Nets $\rightarrow$ Devices, recognition or implementation

Logical formulae $\rightarrow$ Behavioural properties, specification

Morphisms $\rightarrow$ Algebraic properties, closures

$p \models \varphi \iff p \notin L(A)$

Logical specification $\iff$ Automaton acceptance
From words to pomsets.

Usual Linear and branching temporal logics do not apply directly

LtrTL, TLC, ISTL, LCSA ...
Most applied to trace pomsets

Recognisability must be adapted:
monoidal or algebraic structure
specific automata

Monadic second order logic
Subsume Temporal Logics, many useful results on graphs

Pomset regularity
Extends words, traces, local traces and MSC regularity

Pomset automata
Tiling systems and Asynchronous cellular automata
Extended classes of pomsets:
Some views on logic, regularity and automata

Extending language theory to pomsets

Logics for pomsets

Notions of regularity

Relating regularity and MSO definability
Pomsets, events, configurations, cuts.

\[ p = (E, <, l) \quad \text{and} \quad l : E \rightarrow A \]
Logics for LPOs

$FO(\rightarrow, \Sigma)$

$\varphi ::= P_a(x) \mid x \rightarrow y \mid \neg \varphi \mid \varphi \land \varphi \mid \exists x. \varphi$

$EMSO(\rightarrow, \Sigma)$

$\varphi ::= P_a(x) \mid x \in X \mid x \rightarrow y \mid \neg \varphi \mid \varphi \land \varphi \mid \exists x. \varphi$

$\exists X_1 \ldots X_n. \varphi$

$MSO(\rightarrow, \Sigma)$

$\varphi ::= P_a(x) \mid x \in X \mid x \rightarrow y \mid \neg \varphi \mid \varphi \land \varphi \mid \exists x. \varphi \mid \exists X. \varphi$

let $s = (E_s, \leq_s, \lambda_s)$ then $s \models \varphi$:

$(s, e) \models P_a(x)$ iff $\lambda_s(e) = a$

$(s, e, e') \models x \rightarrow y$ iff $e \rightarrow_s e'$

$(s, e, E) \models x \in X$ iff $e \in E$
Logics and LPOs

\[ FO(\rightarrow, \Sigma) \]
Causality \( \leq \) or concurrency not expressible in general.

Bounded distance causality.

\[ FO(\leq, \Sigma) \]
\( \rightarrow \) definable from \( \leq \).

Concurrency \( \neg(x \leq y \lor y \leq x) \), cuts for width bounded sets.

\[ EMZO(\rightarrow, \Sigma) \]
\( \leq \) definable from \( \rightarrow \), concurrency, cuts, configurations.

Characterized by tilings.

\[ MSO(\rightarrow, \Sigma) = MSO(\leq, \Sigma) \]

Most Temporal logics (Not ALL “until” operators).

\[ MSO(\leq, \#, \Sigma) \]
Partial order + branching structure.
Extended classes of pomsets:
Some views on logic, regularity and automata

Extending language theory to pomsets

Logics for pomsets

Notions of regularity

Relating regularity and MSO definability
A general pomset regularity notion

weak concatenation \( u \diamond v \)

\[
\begin{align*}
u \diamond v &= \{ w \in P(A) : 	ext{u prefix of w} \\
v \text{ residue of w after u} \}
\end{align*}
\]

\( u \cdot v \) and \( u \| v \) are in \( u \diamond v \)

\( L/u = \{ v \in P(A) : u \diamond v \text{ intersects } L \} \)

Equivalence \( u =_L v \iff L/u = L/v \) Extends words syntactic equivalence

\( L \text{ regular} \iff \overline{\_L} \text{ has a finite number of classes} \)

Extends words regularity
Order, step, linear extentions

\[ a \leftarrow b \rightarrow a \rightarrow d \]

\[ a \rightarrow b \rightarrow c \rightarrow a \rightarrow a \rightarrow d \]

\[ r \in SE(p) \]

\[ q \in OE(p) \]

\[ s \in LE(p) \]

\[ \in M(A)^* \]

\[ \in A^* \]

\[ \in P(A) \]
\textbf{L step closed} (maximal w.r.t. \textit{SE})

\[ SE(u) \subseteq SE(L) \Rightarrow u \in L \]

\[ SE(L) \text{ reg} \Leftrightarrow L \text{ reg} \]

\textbf{L linear closed} (maximal w.r.t. \textit{LE})

\[ (LE(u) \subseteq LE(L) \Rightarrow u \in L) \]

\[ LE(L) \text{ reg} \Leftrightarrow L \text{ reg} \]
Extended classes of pomsets:
Some views on logic, regularity and automata

Extending language theory to pomsets

Logics for pomsets

Notions of regularity

Relating regularity and MSO definability

Pomsets with autoconcurrency

Finitely generated pomsets without autoconcurrency
Regularity vs definability

Regularity of \( L \)                     \( \rightarrow \)                             Definability of \( L \)
                                  \( \downarrow \)
Regularity of \( OE(L) \)                     \( \rightarrow \)                             Definability of \( OE(L) \)
                                  \( \downarrow \)
Regularity of \( SE(L) \) in \( P(A) \)   \( \leftrightarrow \)       Definability of \( SE(L) \) in \( P(A) \)
                                  \( \downarrow \)
Regularity of \( SE(L) \) in \( M(A)^* \) \( \leftrightarrow \)       Definability of \( SE(L) \) in \( M(A)^* \)
                                  \( \downarrow \)
Regularity of \( LE(L) \) in \( A^* \)   \( \leftrightarrow \)       Definability of \( LE(L) \) in \( A^* \)
Regularity vs MSO Definability

If $L$ (prime) bounded

$L$ definable $\Rightarrow$ $L$ regular

If $L$ width bounded ($\Rightarrow$ $SE(L)$ bounded)

$OE(L)$ definable $\Rightarrow$ $SE(L)$ regular in $P(\Sigma)$

$\Rightarrow$ $SE(L)$ regular in $M(\Sigma)^*$

$\Rightarrow$ $B(SE(L))$ definable in $M(\Sigma)^*$

All sets

$OE(L)$ definable $\Rightarrow$ $LE(L)$ regular in $\Sigma^*$

If $L$ step or linear closed (weak)

$L$ definable $\Rightarrow$ $L$ regular
Words

Rational expressions \rightarrow Automata

Monoid recognisability \rightarrow Logic MSO

Traces

Co–rational expressions \rightarrow Trace Automata

Monoid recognisability \rightarrow Logic MSO

Series–parallel pomsets

Series–rational expressions \rightarrow Fork–join Automata

SP recognisability \rightarrow Logic MSO
Not bounded

Primes
Finitely generated sets of pomsets without autoconcurrency

**Co-rationality** → **Regularity**

**Monoid recognisability** ← **MSO definability**

**Bounded sets** ← **Freely generated**

**ACA recognisability** ← **All sets**