Finite Automata (FA) and Monadic Second Order logic (MSO).

- **FA**: executable model with good (decidable) properties.
- **MSO (over words)**: very expressive and yet simple logic.
- Both **equally expressive** over words and trees (Büchi).

---

**Qualitative** properties over words.

**Quantitative** properties are also important (today).

**Example**

- number of \(\triangleleft\)-symbols.
- length of the largest sequence of \(\triangleleft\)-symbols.

---

How can we extend finite automata or MSO to define these properties (or functions)?
Weighted automata

General automata framework to define quantitative properties over words.

- (Boolean) automata,
- Probabilistic automata,
- Distance automata,
- Multiplicity automata, etc...

Extension of finite automata with weights from a fix semiring.
Definition

A (commutative) semiring is an algebraic structure $S = (S, \oplus, \otimes, 0, 1)$ where:

- $(S, \oplus, 0)$ and $(S, \otimes, 1)$ are commutative monoids,
- multiplication distributes over addition, and
- $0 \otimes s = s \otimes 0 = 0$ for each $s \in S$.

Example

- Natural numbers: $(\mathbb{N}, +, \cdot, 0, 1)$.
- Boolean: $(\{\text{true, false}\}, \lor, \land, \text{false, true})$.
- Min-plus: $(\mathbb{N}_{\infty}, \min, +, \infty, 0)$.
- Max-plus: $(\mathbb{N}_{-\infty}, \max, +, -\infty, 0)$.
Weighted automata (definition)

Fix a semiring $S$ and a finite alphabet $\Gamma$.

**Definition**

A **weighted automata** over $S$ and $\Gamma$ is a tuple $A = (\Gamma, S, Q, E, I, F)$:

- $E : Q \times \Gamma \times Q \to S$ is the transition relation ($p \xrightarrow{a/s} q$), and
- $I, F : Q \to S$ is the initial and final function.

**Semantics**

- A run $\rho$ of $A$ over $a_1 \ldots a_n \in \Gamma^*$ is:
  \[ \rho = q_0 \xrightarrow{a_1/s_1} q_1 \xrightarrow{a_2/s_2} \ldots \xrightarrow{a_n/s_n} q_n \]

- The weight of run $\rho$ of $A$:
  \[ \text{weight}(\rho) = I(q_0) \odot \bigodot_{i=1}^{n} s_i \odot F(q_n) \]

- $A$ defines the function $[A] : \Gamma^* \to S$:
  \[ [A](w) = \bigoplus_{\rho \in \text{Run}_A(w)} \text{weight}(\rho) \]
Weighted automata (examples)

**Over \((\mathbb{N}, +, \cdot, 0, 1)\)**
- \(f(w) = 3 \cdot |w|_a + 4 \cdot |w|_b\)

**Over \((\mathbb{N}_\infty, \min, +, \infty, 0)\)**
- \(f(w) = \min\{|w|_a, |w|_b\}\)

**Over \((\mathbb{N}_{-\infty}, \max, +, -\infty, 0)\)**
- \(f(w) = \text{maximum length of all infix sequences of } b's\)
What is a good logic to define quantitative properties?

Weighted MSO (Droste & Gastin 2005)

Disadvantages:

- Semantical definition of valid formulas.
- Inherits the undecidability results of weighted automata.

We want a quantitative logic that:

1. has a simple and purely syntactical definition,
2. as expressive as weighted automata, and
3. with good decidability properties.
We propose:

**Quantitative Monadic Second Order Logic (QMSO)**

1. General framework for adding quantitative properties to any boolean logic.

2. Subfragments of QMSO capture different subclasses of WA.

3. Subfragments of QMSO with good decidability properties.

More results in the paper:

- Evaluation of QMSO with respect to counting complexity classes.
Quantitative Monadic Second-Order Logic

Cristian Riveros
University of Oxford

Stephan Kreutzer
Technische Universität Berlin

LICS 2013
Outline

QMSO and WA

QMSO and subclasses of WA

Beyond WA

Conclusions
Quantitative Monadic Second Order Logic (QMSO)

For each \( w \in \Gamma^* \), we represent \( w := (\{1, \ldots, |w|\}, \leq, \{P_a\}_{a \in \Gamma}) \).

**Syntax of QMSO\( [\mathcal{S}, \Gamma] \)**

\[
\begin{align*}
\varphi & := P_a(x) \mid x \leq y \mid x \in X \mid \varphi \lor \varphi \mid \neg \varphi \mid \exists x. \varphi \mid \exists X. \varphi \\
\theta & := \varphi \mid s \in S \mid \theta \oplus \theta \mid \theta \odot \theta \mid \Sigma x. \theta \mid \Pi x. \theta \mid \Sigma X. \theta
\end{align*}
\]

**Semantic of QMSO\( [\mathcal{S}, \Gamma] \)**

\[
\begin{align*}
[\varphi](w, \sigma) & := \begin{cases} 
1 & \text{if } (w, \sigma) \models \varphi \\
0 & \text{otherwise}
\end{cases} \\
[S](w, \sigma) & := s \\
[\theta_1 \oplus \theta_2](w, \sigma) & := [\theta_1](w, \sigma) \oplus [\theta_2](w, \sigma) \\
[\Pi x. \theta(x)](w, \sigma) & := \bigodot_{i \in \text{dom}(w)} [\theta(x)](w, \sigma[x \to i]) \\
[\Sigma X. \theta(X)](w, \sigma) & := \bigoplus_{I \subseteq \text{dom}(w)} [\theta(X)](w, \sigma[X \to I])
\end{align*}
\]
Quantitative Monadic Second Order Logic (QMSO)

The syntax of QMSO depends on the semiring.

**Syntax of QMSO**

\[
\begin{align*}
\varphi & ::= \text{P}_a(x) \mid x \leq y \mid x \in X \mid \varphi \lor \varphi \mid \neg \varphi \mid \exists x. \varphi \mid \exists X. \varphi \\
\theta & ::= \varphi \mid s \in \mathbb{N} \mid \theta + \theta \mid \theta \cdot \theta \mid \sum x. \theta \mid \Pi x. \theta \mid \Sigma X. \theta
\end{align*}
\]

**Syntax of QMSO**

\[
\begin{align*}
\varphi & ::= \text{P}_a(x) \mid x \leq y \mid x \in X \mid \varphi \lor \varphi \mid \neg \varphi \mid \exists x. \varphi \mid \exists X. \varphi \\
\theta & ::= \varphi \mid s \in \mathbb{N}_\infty \mid \min \{\theta, \theta\} \mid \theta + \theta \mid \text{Min} x. \theta \mid \sum x. \theta \mid \text{Min} X. \theta
\end{align*}
\]

**Syntax of QMSO**

\[
\begin{align*}
\varphi & ::= \text{P}_a(x) \mid x \leq y \mid x \in X \mid \varphi \lor \varphi \mid \neg \varphi \mid \exists x. \varphi \mid \exists X. \varphi \\
\theta & ::= \varphi \mid s \in \mathbb{N}_{-\infty} \mid \max \{\theta, \theta\} \mid \theta + \theta \mid \text{Max} x. \theta \mid \sum x. \theta \mid \text{Max} X. \theta
\end{align*}
\]
Examples of QMSO formulas

Over \((\mathbb{N}, +, \cdot, 0, 1)\)

- \(f(w) = 3 \cdot |w|_a + 4 \cdot |w|_b\)

\[ \sum x. \left( 3 \cdot P_a(x) + 4 \cdot P_b(x) \right) \]

Over \((\mathbb{N}_\infty, \min, +, \infty, 0)\)

- \(f(w) = \min \{|w|_a, |w|_b\}\)

\[ \min \left\{ \sum x. P_a(x) \mapsto 1 , \sum x. P_b(x) \mapsto 1 \right\} \]

where \(P_a(x) \mapsto 1 := \min \{ P_a(x) + 1, \neg P_a(x) \} \).

Over \((\mathbb{N}_{-\infty}, \max, +, -\infty, 0)\)

- \(f(w) = \text{maximum length of all infix sequences of } b\text{'s}\)

\[ \text{Max } x. \left( \sum y. \text{interval}_b(x, y) \mapsto 1 \right) \]

where \(\text{interval}_b(x, y) := x \leq y \land \forall z. (x \leq z \land z \leq y) \rightarrow P_b(z)\).
Subfragments of QMSO

1. QMSO(Op) restricted to operators \( \text{Op} \subseteq \{ \oplus, \odot, \Sigma_x, \Pi_x, \Sigma_X \} \).

\[
\begin{align*}
\oplus &= \text{semiring addition} \\
\odot &= \text{semiring multiplication} \\
\Sigma_x &= \text{first-order addition} \\
\Pi_x &= \text{first-order multiplication} \\
\Sigma_X &= \text{second-order addition}
\end{align*}
\]

Example

\[
\text{Full QMSO} := \text{QMSO}(\Sigma_x, \Pi_x, \Sigma_x, \oplus, \odot)
\]

Subfragments of QMSO

1. QMSO(Op) restricted to operators $\text{Op} \subseteq \{\oplus, \odot, \Sigma_x, \Pi_x, \Sigma_X\}$.

2. Alternation and Nesting of semiring quantifiers.

Example

- QMSO($\Sigma_X \Sigma_x \Pi_x, \oplus, \odot$):
  $$\Sigma X. \left( \Sigma y. \Pi z. \varphi(X, z) \right) \oplus \left( \Pi z_1. \Pi z_2. \theta(X, z_1, z_2) \right)$$

- QMSO($\Sigma_x \Pi^1_x, \oplus, \odot$):
  $$\Sigma x. \left( \Sigma y. \Pi z. \varphi(x, y, z) \right) \odot \left( \Pi z. \theta(x, z) \right)$$

- QMSO($\Pi^n_x, \oplus, \odot$), $n \in \mathbb{N}$:
  $$\Pi x_1 \ldots \Pi x_n. \theta(x_1, \ldots, x_n)$$
QMSO and weighted automata

QMSO is too expressive to capture weighted automata!

Over \((\mathbb{N}, +, \cdot, 0, 1)\)

- \([\Pi x. \Pi y. 2](w) = 2^{|w|^2}\).
- For every weighted automata \(A\) over \((\mathbb{N}, +, \cdot, 0, 1)\):
  \[[A](w) \in 2^{O(|w|)}\)
QMSO and weighted automata

**QMSO is too expressive to capture weighted automata!**

**Definition**

Quantitative Iteration Logic (QIL) := QMSO(\(\Sigma, x^1, \oplus, \odot\)).

**Theorem**

A function \(f : \Gamma^* \rightarrow S\) is **definable** by a weighted automaton over \(S\) and \(\Gamma\) if, and only if, \(f\) is **definable** by a formula in QIL[\(S, \Gamma\)].

Weighted Automata \(\equiv\) QIL.
Undecidable properties of QIL

Quantitative generalization of classical decision problems:

- **Equivalence:** $[\theta_1](w) = [\theta_2](w)$ for all $w \in \Gamma^*$,

- **Containment:** $[\theta_1](w) \leq [\theta_2](w)$ for all $w \in \Gamma^*$.

**Proposition**

The following problems are undecidable:

1. Containment of formulas in QMSO$(\Sigma_x \Pi^1_x, \oplus, \odot)$ over $(\mathbb{N}, +, \cdot, 0, 1)$.

2. Equivalence and containment of formulas in QMSO$(\Sigma_x \Pi^1_x, \oplus, \odot)$ over $(\mathbb{N}_\infty, \min, +, \infty, 0)$. 
Outline

QMSO and WA

QMSO and subclasses of WA

Beyond WA

Conclusions
Different fragments of QMSO captures different subclasses of WA

Classes of Weighted Automata (WA) depending on the ambiguity:

- Deterministic WA (DWA).
- Unambiguous WA (unamb- WA):
  \[ |\text{Run}_A(w)| \leq 1 \text{ for all } w \in \Sigma^* \]
- Finite Ambiguous WA (fin- WA):
  \[ |\text{Run}_A(w)| < k \text{ for all } w \in \Sigma^* \]
- Polynomially Ambiguous WA (poly- WA):
  \[ |\text{Run}_A(w)| \in O(|w|^k) \]
Unambiguous and finitely ambiguous weighted automata are captured by QMSO

Subfragment QMSO$(\Op, \oplus_b)$: $\oplus$-operator is restricted to a “base” level.

Example

\[
\begin{align*}
(\Pi x. P_a(x) \oplus P_b(x)) \odot (\Pi x. \exists z. x \leq z \land P_a(z)) & \in \text{QMSO}(\Pi^1_x, \oplus_b, \odot) \\
(\Pi x. P_a(x) \oplus P_b(x)) \oplus (\Pi x. \exists z. x \leq z \land P_a(z)) & \notin \text{QMSO}(\Pi^1_x, \oplus_b, \odot)
\end{align*}
\]
Unambiguous and finitely ambiguous weighted automata are captured by QMSO

Subfragment QMSO(\(\text{Op}, \oplus_b\)): \(\oplus\)-operator is restricted to a “base” level.

Theorem

\[
\begin{align*}
\text{unamb-WA} & \equiv \text{QMSO}(\prod^1_x, \oplus_b, \odot) \\
\text{fin-WA} & \equiv \text{QMSO}(\prod^1_x, \oplus, \odot)
\end{align*}
\]

Proof idea.

From QMSO(\(\prod^1_x, \oplus_b, \odot\)) to \text{unamb-WA}:

- Exploit unambiguity to express formulas of the form \(\prod_x \bigoplus_{i \in I} \bigodot_{j \in J} \varphi_{i,j}(x)\).

From QMSO(\(\prod^1_x, \oplus, \odot\)) to \text{fin-WA}:

- Use \textit{disambiguation} theorem presented in Klimann et al., 2004.
Polynomial ambiguous weighted automata are also captured by QMSO

Theorem

\[ \text{poly-WA} \equiv \text{QMSO}(\Sigma_x \Pi_x^1, \oplus, \odot) \]

Proof idea.
From \text{poly-WA} to QMSO\((\Sigma_x \Pi_x^1, \oplus, \odot)\):

- Exploit structural properties of the components of a \text{poly-WA}. 
Which fragment captures deterministic weighted automata?

The *forward-iterator* \((\cdot)^\rightarrow\) and the *backward-iterator* \((\cdot)^\leftarrow\)

\[
\begin{align*}
[\theta^\rightarrow](w, \sigma) &= \bigcirc_{i=1}^{n} [\theta](w[1..i], \sigma) \\
[\theta^\leftarrow](w, \sigma) &= \bigcirc_{i=1}^{n} [\theta](w[i..n], \sigma)
\end{align*}
\]

**Over** \((\mathbb{N}_\infty, \min, +, \infty, 0)\)

- \(f(w) = \text{number of prefixes of } w \text{ that satisfy } \varphi.\)

\[(\min\{ \varphi + 1, \neg \varphi \})^\rightarrow.\]
Which fragment captures deterministic weighted automata?

The forward-iterator \((\cdot)\rightarrow\) and the backward-iterator \((\cdot)\leftarrow\) :

\[
\begin{align*}
[\theta^{\rightarrow}](w, \sigma) &= \bigotimes_{i=1}^{n} [\theta](w[1..i], \sigma) \\
[\theta^{\leftarrow}](w, \sigma) &= \bigotimes_{i=1}^{n} [\theta](w[i..n], \sigma)
\end{align*}
\]

Theorem

\[
\begin{align*}
\text{DWA} & \equiv \text{QMSO}(\rightarrow, \oplus_{b}, \odot) \\
\text{co-DWA} & \equiv \text{QMSO}(\leftarrow, \oplus_{b}, \odot)
\end{align*}
\]

* the \((\cdot)\rightarrow\)- and \((\cdot)\leftarrow\)-operator cannot be nested.

Connection of determinization of WA with logic.
Corollary

The following problems are decidable:

1. Equivalence and containment problem of formulas in $\text{QMSO}(\Pi^1_x, \oplus_b, \odot)$ over $(\mathbb{N}, +, \cdot, 0, 1)$.

2. Equivalence and containment problem of formulas in $\text{QMSO}(\Pi^1_x, \oplus, \odot)$ over $(\mathbb{N}_\infty, \min, +, \infty, 0)$.

$\text{QMSO}(\Pi^1_x, \oplus_b, \odot)$ and $\text{QMSO}(\Pi^1_x, \oplus, \odot)$ are good fragments.
Outline

QMSO and WA

QMSO and subclasses of WA

Beyond WA

Conclusions
How to go further from these (good) fragments?

1. **Additive** fragment: $\text{QMSO}(\Sigma^k_x \Pi^1_x, \oplus, \odot_b)$.

**Theorem**

For all $k \in \mathbb{N}$: $\text{poly}^k$-WA $\equiv$ $\text{QMSO}(\Sigma^k_x \Pi^1_x, \oplus, \odot_b)$.

2. **Multiplicative** fragment: $\text{QMSO}(\Pi^k_x, \oplus_b, \odot)$.

Two-way weighted automata with nested pebbles.
Two-way weighted automata with nested pebbles

In the boolean case:

Two-way weighted automata with nested pebbles \( \equiv \) regular languages

Different subclasses of 2WA:

- Two-way WA with \( k \)-nested pebbles (2WA-\( k \)).
- Deterministic 2WA-\( k \) (2DWA-\( k \)).
- Unambiguous 2WA-\( k \) (\textit{unamb}- 2WA-\( k \)).
Multiplicative fragment and two-way WA with nested pebbles

Theorem
The following classes of WA and subfragments of QMSO are equally expressive over $\Gamma$ and $S$:
1. 2DWA-0,
2. $unamb$-2WA-0,
3. $unamb$-WA, and
4. QMSO($\Pi_1^1, \oplus b, \odot$).

Theorem
For every $k \in \mathbb{N}$, there exists an effective translation between the following classes of WA and subfragments of QMSO over $\Gamma$ and $S$:
1. 2DWA-$k$,
2. $unamb$-2WA-$k$, and
3. QMSO($\Pi_x^{k+1}, \oplus b, \odot$).
Outline

QMSO and WA

QMSO and subclasses of WA

Beyond WA

Conclusions
Future work

**Logic-side:**
- Relation between (inner) boolean logic and semiring operators.
- Expressibility of QMSO over more general structures.

**Automata-side:**
- Decidability properties of subclasses of WA motivated by QMSO.
- Determinization of WA.