Expressivity within second-order transitive-closure logic

Jonni Virtema

Hasselt University, Belgium
jonni.virtema@gmail.com

Joint work with Jan Van den Bussche and Flavio Ferrarotti
To be presented in CSL 2018

Horizons of Logic, Computation and Definability
Symposium in Honour of Lauri Hella's 60th birthday
July 6, 2018

Transitive closure

The transitive closure $\mathrm{TC}(R)$ of a binary relation $R \subseteq A \times A$ is defined as follows

$$\mathrm{TC}(R) := \{(a, b) \in A \times A \mid \exists n > 0 \text{ and } e_0, \ldots, e_n \in A \\ \text{such that } a = e_0,\ b = e_n, \text{ and } (e_i, e_{i+1}) \in R \text{ for all } i < n\}.$$

In our setting A is set of tuples $(a_1, \ldots a_n)$, where each a_i is either an *element* or a *relation* over some domain D.

Transitive closure

The transitive closure $\mathrm{TC}(R)$ of a binary relation $R \subseteq A \times A$ is defined as follows

$$\mathrm{TC}(R) := \{(a, b) \in A \times A \mid \exists n > 0 \text{ and } e_0, \ldots, e_n \in A \text{ such that } a = e_0,\ b = e_n, \text{ and } (e_i, e_{i+1}) \in R \text{ for all } i < n\}.$$

In our setting A is set of tuples $(a_1, \ldots a_n)$, where each a_i is either an *element* or a *relation* over some domain D.

Transitive closure

Example

Let $G = (V, E)$ be an undirected graph. Then $(a, b) \in \mathrm{TC}(E)$ if a and b are in the same component of G, or equivalently, if there is a path from a to b in G.

Example

A graph $G = (V, E)$ has a Hamiltonian cycle if the following holds:

1. There is a relation $\mathcal{R}$ such that

$$(Z, z, Z', z') \in \mathcal{R} \quad \text{iff} \quad Z' = Z \cup \{z'\}, z' \notin Z \text{ and } (z, z') \in E.$$

2. The tuple $(\{x\}, x, V, y)$ is in the transitive closure of $\mathcal{R}$, for some $x, y \in V$ such that $(y, x) \in E$.

Transitive closure

Example

Let $G = (V, E)$ be an undirected graph. Then $(a, b) \in \mathrm{TC}(E)$ if a and b are in the same component of G, or equivalently, if there is a path from a to b in G.

Example

A graph $G = (V, E)$ has a Hamiltonian cycle if the following holds:

1. There is a relation $\mathcal{R}$ such that
$$(Z, z, Z', z') \in \mathcal{R} \quad \text{iff} \quad Z' = Z \cup \{z'\}, z' \notin Z \text{ and } (z, z') \in E.$$
2. The tuple $(\{x\}, x, V, y)$ is in the transitive closure of $\mathcal{R}$, for some $x, y \in V$ such that $(y, x) \in E$.

Logics with transitive closure operator

First-order transitive closure logic $\mathrm{FO(TC)}$:

$$\varphi ::= x = y \mid X(x_1, \ldots, x_k) \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \exists x \varphi \mid [\mathrm{TC}_{\vec{x},\vec{x'}} \varphi](\vec{y}, \vec{y'}),$$

where $\vec{x}$, $\vec{x'}$,$\vec{y}$, and $\vec{y'}$ are tuples of first-order variables of the same length.

Semantics for the TC operator:

$$\mathfrak{A} \models_s [\mathrm{TC}_{\vec{x},\vec{x'}} \varphi](\vec{y}, \vec{y'}) \text{ iff } \big(s(\vec{y}), s(\vec{y'})\big) \in \mathrm{TC}(\{(\vec{a}, \vec{a'}) \mid \mathfrak{A} \models_{s(\vec{x} \mapsto \vec{a}, \vec{x'} \mapsto \vec{a'})} \varphi\})$$

Example

The sentence

$$\forall x \forall y [\mathrm{TC}_{z,z'} E(z, z')](x, y)$$

expresses connectivity of graphs (V, E).

Logics with transitive closure operator

First-order transitive closure logic $\mathrm{FO}(\mathrm{TC})$:

$$\varphi ::= x = y \mid X(x_1, \ldots, x_k) \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \exists x \varphi \mid [\mathrm{TC}_{\vec{x},\vec{x'}} \varphi](\vec{y}, \vec{y'}),$$

where $\vec{x}$, $\vec{x'}$,$\vec{y}$, and $\vec{y'}$ are tuples of first-order variables of the same length.

Semantics for the TC operator:

$$\mathfrak{A} \models_s [\mathrm{TC}_{\vec{x},\vec{x'}} \varphi](\vec{y}, \vec{y'}) \text{ iff } \big(s(\vec{y}), s(\vec{y'})\big) \in \mathrm{TC}(\{(\vec{a}, \vec{a'}) \mid \mathfrak{A} \models_{s(\vec{x} \mapsto \vec{a}, \vec{x'} \mapsto \vec{a'})} \varphi\})$$

Example

The sentence

$$\forall x \forall y [\mathrm{TC}_{z,z'} E(z, z')](x, y)$$

expresses connectivity of graphs (V, E).

Logics with transitive closure operator

Second-order transitive closure logic $\mathrm{SO(TC)}$:

$$\varphi ::= x = y \mid X(x_1, \ldots, x_k) \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \exists x \varphi \mid \exists Y \varphi \mid [\mathrm{TC}_{\vec{X},\vec{X}'}\varphi](\vec{Y}, \vec{Y}'),$$

where $\vec{X}$, $\vec{X}'$, $\vec{Y}$, and $\vec{Y}'$ are tuples of first-order and second-order variables of the same length and sort.

Semantics for the TC operator:

$$\mathfrak{A} \models_s [\mathrm{TC}_{\vec{X},\vec{X}'}\varphi](\vec{Y}, \vec{Y}') \text{ iff } \big(s(\vec{Y}), s(\vec{Y}')\big) \in \mathrm{TC}(\{(\vec{A}, \vec{B}') \mid \mathfrak{A} \models_{s(\vec{X}\mapsto\vec{A}, \vec{X}'\mapsto\vec{A}')} \varphi\})$$

$\mathrm{MSO(TC)}$ is the fragment of $\mathrm{SO(TC)}$ in which all second-order variables have arity 1.

Logics with transitive closure operator

Second-order transitive closure logic $\mathrm{SO(TC)}$:

$$\varphi ::= x = y \mid X(x_1, \dots, x_k) \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \exists x \varphi \mid \exists Y \varphi \mid [\mathrm{TC}_{\vec{X},\vec{X}'}\varphi](\vec{Y}, \vec{Y}'),$$

where $\vec{X}$, $\vec{X}'$, $\vec{Y}$, and $\vec{Y}'$ are tuples of first-order and second-order variables of the same length and sort.

Semantics for the TC operator:

$$\mathfrak{A} \models_s [\mathrm{TC}_{\vec{X},\vec{X}'}\varphi](\vec{Y}, \vec{Y}') \text{ iff } \big(s(\vec{Y}), s(\vec{Y}')\big) \in \mathrm{TC}(\{(\vec{A}, \vec{B}') \mid \mathfrak{A} \models_{s(\vec{X}\mapsto\vec{A},\vec{X}'\mapsto\vec{A}')} \varphi\})$$

$\mathrm{MSO(TC)}$ is the fragment of $\mathrm{SO(TC)}$ in which all second-order variables have arity 1.

The Härtig quantifier

$\mathfrak{A} \models_s \mathrm{H}xy(\varphi(x), \psi(y)) \Leftrightarrow$ the sets $\{a \in A \mid \mathfrak{A} \models_{s(x \mapsto a)} \varphi(x)\}$ and $\{b \in A \mid \mathfrak{A} \models_{s(y \mapsto b)} \psi(y)\}$ have the same cardinality

Example (The Härtig quantifier can be expressed in MSO(TC).)

Let $\psi_{\mathrm{decrement}}$ denote an FO-formula expressing that $s(X') = s(X) \setminus \{a\}$ and $s(Y') = s(Y) \setminus \{b\}$ for some a and b. Define

$$\psi_{\mathrm{ec}} := \exists X_\emptyset \Big((\forall x \neg X_\emptyset(x)) \wedge [\mathrm{TC}_{X,Y,X',Y'} \psi_{\mathrm{decrement}}](Z, Z', X_\emptyset, X_\emptyset) \Big).$$

Now ψ_{ec} holds under s if and only if the cardinalities of $s(Z)$ and $s(Z')$ are the same. Therefore $\mathrm{H}xy(\varphi(x), \psi(y))$ is equivalent with the formula

$$\exists Z \exists Z' (\forall x (\varphi(x) \leftrightarrow Z(x)) \wedge \forall y (\psi(y) \leftrightarrow Z'(y)) \wedge \psi_{\mathrm{ec}}).$$

The Härtig quantifier

$\mathfrak{A} \models_s \mathrm{H}xy(\varphi(x), \psi(y)) \Leftrightarrow$ the sets $\{a \in A \mid \mathfrak{A} \models_{s(x \mapsto a)} \varphi(x)\}$ and $\{b \in A \mid \mathfrak{A} \models_{s(y \mapsto b)} \psi(y)\}$ have the same cardinality

Example (The Härtig quantifier can be expressed in MSO(TC).)

Let $\psi_{\mathrm{decrement}}$ denote an FO-formula expressing that $s(X') = s(X) \setminus \{a\}$ and $s(Y') = s(Y) \setminus \{b\}$ for some a and b. Define

$$\psi_{\mathrm{ec}} := \exists X_\emptyset \Big((\forall x \neg X_\emptyset(x)) \wedge [\mathrm{TC}_{X,Y,X',Y'} \psi_{\mathrm{decrement}}](Z, Z', X_\emptyset, X_\emptyset) \Big).$$

Now ψ_{ec} holds under s if and only if the cardinalities of $s(Z)$ and $s(Z')$ are the same. Therefore $\mathrm{H}xy(\varphi(x), \psi(y))$ is equivalent with the formula

$$\exists Z \exists Z' (\forall x(\varphi(x) \leftrightarrow Z(x)) \wedge \forall y(\psi(y) \leftrightarrow Z'(y)) \wedge \psi_{\mathrm{ec}}).$$

The Härtig quantifier

$\mathfrak{A} \models_s \mathrm{H}xy(\varphi(x), \psi(y)) \Leftrightarrow$ the sets $\{a \in A \mid \mathfrak{A} \models_{s(x \mapsto a)} \varphi(x)\}$ and $\{b \in A \mid \mathfrak{A} \models_{s(y \mapsto b)} \psi(y)\}$ have the same cardinality

Example (The Härtig quantifier can be expressed in MSO(TC).)

Let $\psi_{\mathrm{decrement}}$ denote an FO-formula expressing that $s(X') = s(X) \setminus \{a\}$ and $s(Y') = s(Y) \setminus \{b\}$ for some a and b. Define

$$\psi_{\mathrm{ec}} := \exists X_\emptyset \Big((\forall x \neg X_\emptyset(x)) \wedge [\mathrm{TC}_{X,Y,X',Y'} \psi_{\mathrm{decrement}}](Z, Z', X_\emptyset, X_\emptyset) \Big).$$

Now ψ_{ec} holds under s if and only if the cardinalities of $s(Z)$ and $s(Z')$ are the same. Therefore $\mathrm{H}xy(\varphi(x), \psi(y))$ is equivalent with the formula

$$\exists Z \exists Z' (\forall x(\varphi(x) \leftrightarrow Z(x)) \wedge \forall y(\psi(y) \leftrightarrow Z'(y)) \wedge \psi_{\mathrm{ec}}).$$

Hamiltonian cycle

Example

A graph $G = (V, E)$ has a Hamiltonian cycle if the following holds:

1. There is a relation $\mathcal{R}$ such that

$$(Z, z, Z', z') \in \mathcal{R} \quad \text{iff} \quad Z' = Z \cup \{z'\}, z' \notin Z \text{ and } (z, z') \in E.$$

2. The tuple $(\{x\}, x, V, y)$ is in the transitive closure of $\mathcal{R}$, for some $x, y \in V$ such that $(y, x) \in E$.

In the language of $\mathrm{MSO(TC)}$ this can be written as follows:

$$\exists XYxy(X(x) \wedge \forall z(z \neq x \to \neg X(x)) \wedge \forall z(Y(z)) \wedge E(y,x) \wedge [\mathrm{TC}_{Z,z,Z',z'}\varphi](X,x,Y,y))$$

where $\varphi := \neg Z(z') \wedge \forall x(Z'(x) \leftrightarrow (Z(x) \vee z' = x)) \wedge E(z, z')$.

Hamiltonian cycle

Example

A graph $G = (V, E)$ has a Hamiltonian cycle if the following holds:

1. There is a relation $\mathcal{R}$ such that

$$(Z, z, Z', z') \in \mathcal{R} \quad \text{iff} \quad Z' = Z \cup \{z'\}, z' \notin Z \text{ and } (z, z') \in E.$$

2. The tuple $(\{x\}, x, V, y)$ is in the transitive closure of $\mathcal{R}$, for some $x, y \in V$ such that $(y, x) \in E$.

In the language of $\mathrm{MSO(TC)}$ this can be written as follows:

$$\exists XYxy\big(X(x) \wedge \forall z(z \neq x \rightarrow \neg X(x)) \wedge \forall z(Y(z)) \wedge E(y, x) \wedge [\mathrm{TC}_{Z,z,Z',z'}\varphi](X, x, Y, y)\big)$$

where $\varphi := \neg Z(z') \wedge \forall x\big(Z'(x) \leftrightarrow (Z(x) \vee z' = x)\big) \wedge E(z, z')$.

Descriptive complexity

Theorem (Harel and Peleg 84)

$\mathrm{SO(TC)}$ *captures polynomial space* PSPACE.

Theorem (Immerman 87)

- *On finite ordered structures, first-order transitive-closure logic* $\mathrm{FO(TC)}$ *captures nondeterministic logarithmic space* $\mathrm{NLOGSPACE}$.
- *On strings (word structures),* $\mathrm{SO(arity}\ k\mathrm{)(TC)}$ *captures the complexity class* $\mathrm{NSPACE}(n^k)$.

In particular, on strings $\mathrm{MSO(TC)}$ captures nondeterministic linear space NLIN.

Existential positive $\mathrm{SO(2TC)}$

$\exists\mathrm{SO(2TC)}$ is the syntactic fragment of $\mathrm{SO(TC)}$ in which

1. the existential quantifiers and the TC-operators occur only positively.
2. TC-operators bound only second-order variables.

Rosen noted (99) that $\exists\mathrm{SO}$ collapses to existential first-order logic $\exists\mathrm{FO}$.

Theorem

The expressive powers of $\exists\mathrm{SO(2TC)}$ and $\exists\mathrm{FO}$ coincide.

Proof.

$$[\mathrm{TC}_{\vec{X},\vec{X}'}\exists x_1 \dots \exists x_n \theta](\vec{Y},\vec{Y}') \text{ and } \mathfrak{A} \models [\mathrm{TC}^k_{\vec{X},\vec{X}'}\exists x_1 \dots \exists x_n \theta](\vec{Y},\vec{Y}'),$$

where θ is quantifier free FO-formula, are equivalent for large enough k. (Note that k independent of the model in question and depends only on the formula.) □

Existential positive $\mathrm{SO}(2\mathrm{TC})$

$\exists\mathrm{SO}(2\mathrm{TC})$ is the syntactic fragment of $\mathrm{SO}(\mathrm{TC})$ in which

1. the existential quantifiers and the TC-operators occur only positively.
2. TC-operators bound only second-order variables.

Rosen noted (99) that $\exists\mathrm{SO}$ collapses to existential first-order logic $\exists\mathrm{FO}$.

Theorem

The expressive powers of $\exists\mathrm{SO}(2\mathrm{TC})$ and $\exists\mathrm{FO}$ coincide.

Proof.

$$[\mathrm{TC}_{\vec{X},\vec{X}'}\exists x_1\dots\exists x_n\theta](\vec{Y},\vec{Y}') \text{ and } \mathfrak{A}\models[\mathrm{TC}^k_{\vec{X},\vec{X}'}\exists x_1\dots\exists x_n\theta](\vec{Y},\vec{Y}'),$$

where θ is quantifier free FO-formula, are equivalent for large enough k. (Note that k independent of the model in question and depends only on the formula.) □

Corridor tiling problem

The *corridor tiling problem* is the following PSPACE-complete decision problem (Chlebus 86):

Input: An instance $P = (T, H, V, \vec{b}, \vec{t}, n)$ of the corridor tiling problem.

Output: Does there exist a corridor tiling for P? (Does there exists a tiling of width n having $\vec{b}$ as the first row and $\vec{t}$ as the last row?)

Complexity of model checking

Theorem

Combined complexity of model checking for monadic $2\mathrm{TC}[\forall\mathrm{FO}]$ *is* PSPACE*-complete.*

Proof.

Hardness follows from a reduction from corridor tiling. Input: $(T, H, V, \vec{b}, \vec{t})$. Let s be a successor relation on $\{0, 1, \ldots, n\}$ and $X_1, \ldots X_k, Y_1, \ldots, Y_k$ monadic second-order variables that correspond to tile types.

$$\varphi_H := \forall xy\big(s(x,y) \rightarrow \bigvee_{(i,j)\in H} Z_i'(x) \wedge Z_j'(y)\big), \quad \varphi_V := \forall x \bigvee_{(i,j)\in V} Z_i(x) \wedge Z_j'(x)$$

$$\varphi_T := \forall x \bigvee_{i\in T} \big(Z_i'(x) \wedge \bigwedge_{j\in T, i\neq j} \neg Z_j'(x)\big),$$

The formula $\mathrm{TC}_{\vec{Z},\vec{Z}'}[\varphi_T \wedge \varphi_H \wedge \varphi_V](\vec{X}, \vec{Y})$ describes proper tiling. □

MSO(TC) and counting

- Assume a supply of *counter variables* μ and ν (with subscripts). Counters range over $\{0, \ldots, n\}$, where n is the cardinality the model.
- Assume a supply of k-ary *numeric predicates* $p(\mu_1, \ldots, \mu_k)$.
  - Intuitively relations over natural numbers such as the tables of multiplication and addition.
  - Technically similar to generalised quantifiers; a k-ary numeric predicate is a class $Q_p \subseteq \mathbb{N}^{k+1}$ of $k+1$-tuples of natural numbers.
  - When evaluating a k-ary numeric predicate $p(\mu_1, \ldots, \mu_k)$, the numeric predicate Q_p accesses also the cardinality of the structure in question.

MSO(TC) and counting

Definition

The syntax of CMSO(TC) extends the syntax of MSO(TC) as follows:

$$\varphi ::= (\mu = \#\{x : \varphi\}) \mid p(\mu_1, \ldots, \mu_k) \mid \exists \mu \varphi \mid [\mathrm{TC}_{\vec{X}, \vec{X}'} \varphi](\vec{Y}, \vec{Y}'),$$

where $\vec{X}$, $\vec{X}'$, $\vec{Y}$, and $\vec{Y}'$ may also include counter variables.

Semantics:

$$\mathfrak{A} \models_s \mu = \#\{x : \varphi\} \text{ iff } s(\mu) \text{ equals the cardinality of } \{a \in A \mid \mathfrak{A} \models_{s(x \mapsto a)} \varphi\}.$$

$$\mathfrak{A} \models_s p(\mu_1, \ldots, \mu_k) \text{ iff } (|A|, s(\mu_1), \ldots, s(\mu_k)) \in Q_p$$

$$\mathfrak{A} \models_s \exists \mu \varphi \text{ iff there exists } i \in \{0, \ldots, n\} \text{ such that } \mathfrak{A} \models_{s(\mu \mapsto i)} \varphi.$$

$\mathrm{MSO(TC)}$ and counting

Definition

The syntax of $\mathrm{CMSO(TC)}$ extends the syntax of $\mathrm{MSO(TC)}$ as follows:

$$\varphi ::= (\mu = \#\{x : \varphi\}) \mid p(\mu_1, \ldots, \mu_k) \mid \exists \mu \varphi \mid [\mathrm{TC}_{\vec{X},\vec{X}'} \varphi](\vec{Y}, \vec{Y}'),$$

where $\vec{X}$, $\vec{X}'$, $\vec{Y}$, and $\vec{Y}'$ may also include counter variables.

Semantics:

$$\mathfrak{A} \models_s \mu = \#\{x : \varphi\} \text{ iff } s(\mu) \text{ equals the cardinality of } \{a \in A \mid \mathfrak{A} \models_{s(x \mapsto a)} \varphi\}.$$
$$\mathfrak{A} \models_s p(\mu_1, \ldots, \mu_k) \text{ iff } \big(|A|, s(\mu_1), \ldots, s(\mu_k)\big) \in Q_p$$
$$\mathfrak{A} \models_s \exists \mu \varphi \text{ iff there exists } i \in \{0, \ldots, n\} \text{ such that } \mathfrak{A} \models_{s(\mu \mapsto i)} \varphi.$$

Counting in NLOGSPACE

Definition

A k-ary numeric predicate Q_p is *decidable in* NLOGSPACE if the membership $(n_0, \ldots, n_k) \in Q_p$ can be decided by a nondeterministic Turing machine that uses logarithmic space when the numbers $n_0, \ldots, n_k$ are given in unary. Note that this is equivalent to linear space when $n_0, \ldots, n_k$ are given in binary.

We restrict to numeric predicates that are decidable in NLOGSPACE.

Example

Let k be a natural number, $X, Y, Z, X_1, \ldots, X_n$ monadic second-order variables. The following numeric predicates are clearly NLOGSPACE-definable:

- $\mathfrak{A} \models_s \mathrm{size}(X, k)$ iff $|s(X)| = k$,
- $\mathfrak{A} \models_s \times(X, Y, Z)$ iff $|s(X)| \times |s(Y)| = |s(Z)|$,
- $\mathfrak{A} \models_s +(X_1, \ldots, X_n, Y)$ iff $|s(X_1)| + \cdots + |s(X_n)| = |s(Y)|$.

Counting in NLOGSPACE

Definition

A k-ary numeric predicate Q_p is *decidable in* NLOGSPACE if the membership $(n_0, \dots, n_k) \in Q_p$ can be decided by a nondeterministic Turing machine that uses logarithmic space when the numbers $n_0, \dots, n_k$ are given in unary. Note that this is equivalent to linear space when $n_0, \dots, n_k$ are given in binary.

We restrict to numeric predicates that are decidable in NLOGSPACE.

Example

Let k be a natural number, $X, Y, Z, X_1, \dots, X_n$ monadic second-order variables. The following numeric predicates are clearly NLOGSPACE-definable:

- $\mathfrak{A} \models_s \mathrm{size}(X, k)$ iff $|s(X)| = k$,
- $\mathfrak{A} \models_s \times(X, Y, Z)$ iff $|s(X)| \times |s(Y)| = |s(Z)|$,
- $\mathfrak{A} \models_s +(X_1, \dots, X_n, Y)$ iff $|s(X_1)| + \cdots + |s(X_n)| = |s(Y)|$.

Counting in NLOGSPACE

Proposition (Immerman 87)

For every k-ary numeric predicate Q_p decidable in $\mathrm{NLOGSPACE}$ there exists a formula φ_p of $\mathrm{FO(TC)}$ over $\{s, x_1, \ldots, x_k\}$,

$$\mathfrak{A} \models_s p(\mu_1, \ldots, \mu_k) \text{ iff } \mathfrak{B} \models_t \varphi_p,$$

where $B = \{0, 1, \ldots, |A|\}$, $t(s)$ is the successor relation of B, and $t(x_i) = s(\mu_i)$, for $1 \leq i \leq k$.

$\mathrm{MSO(TC)}$ (without order) simulates $\mathrm{FO(TC)}$ with order

The idea is that natural numbers i are simulated by sets of cardinality i. Recall that $\mathrm{MSO(TC)}$ can express the Härtig quantifier!

The translation $^+ : \mathrm{FO(TC)} \to \mathrm{MSO(TC)}$ is defined as follows:

- For ψ of the form $x_i = x_j$, define $\psi^+ := \mathrm{H}xy\big(X_i(x), X_j(y)\big)$.
- For ψ of the form $s(x_i, x_j)$, define $\psi^+ := \exists z\Big(\neg X_i(z) \wedge \mathrm{H}xy\big(X_i(x) \vee x = z, X_j(y)\big)\Big)$.
- For ψ of the form $\exists x_i \varphi$, define $\psi^+ := \exists X_i \varphi^+$.
- For ψ of the form $[\mathrm{TC}_{\vec{x},\vec{x}'}\varphi](\vec{y}, \vec{y}')$, define $\psi^+ := [\mathrm{TC}_{\vec{X},\vec{X}'}\varphi^+](\vec{Y}, \vec{Y}')$.

$\mathrm{MSO(TC)}$ (without order) simulates $\mathrm{FO(TC)}$ with order

The idea is that natural numbers i are simulated by sets of cardinality i. Recall that $\mathrm{MSO(TC)}$ can express the Härtig quantifier!

The translation $^+ : \mathrm{FO(TC)} \to \mathrm{MSO(TC)}$ is defined as follows:

- For ψ of the form $x_i = x_j$, define $\psi^+ := \mathrm{H}xy\big(X_i(x), X_j(y)\big)$.
- For ψ of the form $s(x_i, x_j)$, define
  $\psi^+ := \exists z\Big(\neg X_i(z) \wedge \mathrm{H}xy\big(X_i(x) \vee x = z, X_j(y)\big)\Big)$.
- For ψ of the form $\exists x_i \varphi$, define $\psi^+ := \exists X_i \varphi^+$.
- For ψ of the form $[\mathrm{TC}_{\vec{x},\vec{x}'}\varphi](\vec{y},\vec{y}')$, define $\psi^+ := [\mathrm{TC}_{\vec{X},\vec{X}'}\varphi^+](\vec{Y},\vec{Y}')$.

$\mathrm{MSO(TC)}$ simulates $\mathrm{CMSO(TC)}$

In $\mathrm{MSO(TC)}$ counter variables are treated as set variables. Define a translation $^* : \mathrm{CMSOTC} \to \mathrm{MSO(TC)}$.

- For an $\mathrm{NLOGSPACE}$ numeric predicate Q_p and ψ of the form $p(\mu_1, \ldots, \mu_k)$, define ψ^* as $\varphi_p^+(\mu_1/X_1, \ldots, \mu_k/X_k)$, where $^+$ is the translation defined above and φ_p the defining $\mathrm{FO(TC)}$ formula of Q_p.
- For ψ of the form $\mu = \#\{x \mid \varphi\}$, ψ^* is $\mathrm{H}xy(\varphi^*, \mu(y))$.
- For ψ of the form $\exists \mu_i \varphi$, define ψ^* as $\exists \mu_i \varphi^*$.

Order invariant MSO

A formula $\varphi \in \mathrm{MSO}$ over $\tau_{\leq}$ is *order-invariant*, if for every τ-structure $\mathfrak{A}$ and expansions $\mathfrak{A}'$ and $\mathfrak{A}^*$ of $\mathfrak{A}$ to the vocabulary $\tau_{\leq}$, in which $\leq^{\mathfrak{A}'}$ and $\leq^{\mathfrak{A}^*}$ are linear orders of A, we have that

$$\mathfrak{A}' \models \varphi \text{ if and only if } \mathfrak{A}^* \models \varphi.$$

A class $\mathcal{C}$ of τ-structures is *definable in order-invariant* MSO if and only if the class

$$\{(\mathfrak{A}, \leq) \mid \mathfrak{A} \in \mathcal{C} \text{ and } \leq \text{ is a linear order of } A\}$$

is definable by some order-invariant MSO-formula.

Order invariant MSO and MSO(TC)

Example

Consider the class

$$\mathcal{C} = \{\mathfrak{A} \mid |A| \text{ is a prime number}\}$$

of $\emptyset$-structures. The language of prime length words over some unary alphabet is not regular and thus it follows via Büchi's theorem that $\mathcal{C}$ is not definable in order-invariant MSO. However the following formula of MSO(TC) defines $\mathcal{C}$.

$$\exists X \forall Y \forall Z \big(\forall x (X(x)) \wedge (\mathrm{size}(Y,1) \vee \mathrm{size}(Z,1) \vee \neg \times (Y,Z,X))\big) \wedge \exists x \exists y \neg x = y.$$

Corollary

For any vocabulary τ, there exists a class $\mathcal{C}$ of τ-structures such that $\mathcal{C}$ is definable in MSO(TC) but it is not definable in order-invariant MSO.

Order invariant MSO and MSO(TC)

Example

Consider the class

$$\mathcal{C} = \{\mathfrak{A} \mid |A| \text{ is a prime number}\}$$

of $\emptyset$-structures. The language of prime length words over some unary alphabet is not regular and thus it follows via Büchi's theorem that $\mathcal{C}$ is not definable in order-invariant MSO. However the following formula of MSO(TC) defines $\mathcal{C}$.

$$\exists X \forall Y \forall Z \big(\forall x(X(x)) \wedge (\mathrm{size}(Y,1) \vee \mathrm{size}(Z,1) \vee \neg \times (Y,Z,X))\big) \wedge \exists x \exists y \neg x = y.$$

Corollary

For any vocabulary τ, there exists a class $\mathcal{C}$ of τ-structures such that $\mathcal{C}$ is definable in MSO(TC) but it is not definable in order-invariant MSO.

Order invariant MSO and MSO(TC)

Theorem

Over finite unary vocabularies MSO(TC) *is strictly more expressive than order-invariant* MSO.

Proof.

The proof is based on Parikh's Theorem (66):
For every regular language L its Parikh image (letter count) $P(L)$ is a finite union of linear sets. □

A subset S of $\mathbb{N}^k$ is a *linear set* if

$$S = \{\vec{v}_0 + \sum_{i=1}^{m} a_i \vec{v}_i \mid a_1, \ldots, a_m \in \mathbb{N}\}$$

for some *offset* $\vec{v}_0 \in \mathbb{N}^k$ and *generators* $\vec{v}_1, \ldots, \vec{v}_m \in \mathbb{N}^k$.

Order invariant MSO and MSO(TC)

Theorem

Over finite unary vocabularies MSO(TC) *is strictly more expressive than order-invariant* MSO.

Proof.

The proof is based on Parikh's Theorem (66):
For every regular language L its Parikh image (letter count) $P(L)$ is a finite union of linear sets. □

A subset S of $\mathbb{N}^k$ is a *linear set* if

$$S = \{\vec{v}_0 + \sum_{i=1}^{m} a_i \vec{v}_i \mid a_1, \ldots, a_m \in \mathbb{N}\}$$

for some *offset* $\vec{v}_0 \in \mathbb{N}^k$ and *generators* $\vec{v}_1, \ldots, \vec{v}_m \in \mathbb{N}^k$.

Open question

- Does the exists a formula of least fixed point logic LFP that is not expressible in MSO(TC). On ordered structures, this would show that there are problems in P that are not in NLIN, which is open (it is only know that the two classes are different).
- Note that EVEN is definable in MSO(TC) but not in LFP (over empty vocabulary).
- What is the relationship of MSO(TC) and order-invariant MSO over vocabularies of higher arity?

Happy Birthday Lauri!