Topics on Descriptive Complexity

Armando Matos

Departamento de Ciência de Computadores
Universidade do Porto

June 2010

Outline

On complexity

Complexity classes

Complexity classes

1. Some complexity classes:

$$\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PH} \subseteq \mathrm{PSPACE}$$

   L: log-space, N: non-deterministic, P: polynomial.

   Are the inclusions proper?
   What is known can be summarized by NL$\neq$ PSPACE

Complexity classes

1. Some complexity classes:

$$\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PH} \subseteq \mathrm{PSPACE}$$

   L: log-space, N: non-deterministic, P: polynomial.

   Are the inclusions proper?
   What is known can be summarized by NL$\neq$ PSPACE

2. Polynomial reductions.

Complexity classes

1. Some complexity classes:

   $$L \subseteq NL \subseteq P \subseteq NP \subseteq PH \subseteq PSPACE$$

   L: log-space, N: non-deterministic, P: polynomial.

   Are the inclusions proper?
   What is known can be summarized by NL$\neq$ PSPACE
2. Polynomial reductions.
3. Complete problems in a class.

(Concepts similar to those of Logic and Computability)

A note on problem reductions

$A \leq_P B$:

In complexity a many-to-one polynomial reduction between the languages A and B is a polynomial time computable function f mapping instances of the first problem into instances of the second problem, such that $x \in A$ (the answer to x is YES) iff $f(x) \in B$ (the answer to $f(x)$ is YES).

A note on problem reductions

$A \leq_P B$:
In complexity a many-to-one polynomial reduction between the languages A and B is a polynomial time computable function f mapping instances of the first problem into instances of the second problem, such that $x \in A$ (the answer to x is YES) iff $f(x) \in B$ (the answer to $f(x)$ is YES).

A very weak reduction power seems often enough, say the first or second in the sequence

$$\leq_{\mathrm{FO}},\ \leq_L,\ \leq_P,\ \leq_{TM}$$

A note on problem reductions

$A \leq_P B$:

In complexity a many-to-one polynomial reduction between the languages A and B is a polynomial time computable function f mapping instances of the first problem into instances of the second problem, such that $x \in A$ (the answer to x is YES) iff $f(x) \in B$ (the answer to $f(x)$ is YES).

A very weak reduction power seems often enough, say the first or second in the sequence

$$\leq_{\mathrm{FO}}, \leq_L, \leq_P, \leq_{TM}$$

...it is surprising that the vast majority of natural complete problems remain complete via first-order reductions (Immerman)

Running away from concrete models. . .

Running away from concrete models...

An early example.

Blum axioms for computational complexity.

Running away from concrete models...

An early example.
Blum axioms for computational complexity.

The descriptive complexity approach.

Complexity class $\leftrightarrow$ expressivity in a logic

On logic and model theory

Finite and infinite models

Classical logic on infinite structures arose from paradoxes of the infinite and from the desire to understand the infinite. Central constructions of classical logic yield infinite structures and most of model theory is based on methods that take infiniteness of structures for granted. In that context, finite models are anomalies that deserve only marginal attention.

Finite and infinite models

Classical logic on infinite structures arose from paradoxes of the infinite and from the desire to understand the infinite. Central constructions of classical logic yield infinite structures and most of model theory is based on methods that take infiniteness of structures for granted. In that context, finite models are anomalies that deserve only marginal attention.

Finite model theory arose as an independent field of logic from consideration of problems in theoretical computer science. Basic concepts in this field are finite graphs, databases, computations etc. . . . Many of the problems of complexity theory and database theory can be formulated as problems of mathematical logic, provided that we limit ourselves to finite structures.

(Väänänen, "A short course on FMT")

Finite and infinite models...

An example of two contrasting results.

Let $\mathcal{L}$ be a language containing a non unary relation symbol.

Finite and infinite models...

An example of two contrasting results.
Let $\mathcal{L}$ be a language containing a non unary relation symbol.

- The set of valid first-order sentences over $\mathcal{L}$ is r.e. but not co-r.e.

Finite and infinite models. . .

An example of two contrasting results.
Let $\mathcal{L}$ be a language containing a non unary relation symbol.

- The set of valid first-order sentences over $\mathcal{L}$ is r.e. but not co-r.e.
- The set of first-order sentences over $\mathcal{L}$ that are valid over finite structures is co-r.e. but not r.e.

Concepts. . .

- Vocabulary τ, ρ: relational, functional, constant symbols
  Examples: $\langle E^2 \rangle$, $\langle E^2, x, y \rangle$, $\langle \leq^2, S^1 \rangle$.

Concepts...

- Vocabulary τ, ρ: relational, functional, constant symbols
  Examples: $\langle E^2 \rangle$, $\langle E^2, x, y \rangle$, $\langle \leq^2, S^1 \rangle$.
- Structures $\mathcal{A}$ on a vocabulary:
  universe $|\mathcal{A}|$ (assumed to be $\{0, 1, \ldots, n-1\}$),
  for each R^k in τ a relation of arity k defined on $|\mathcal{A}|$, etc.
  Examples:
  A graph: $\langle \{0, 1, 2\}, \{(1, 2), (2, 1)\} \rangle$,
  Word 1101: $\langle \{0, 1, 2, 3\}, \leq^2 = \{(0, 0), (0, 1), \ldots\},$
  $S^1 = \{0 \to T, 1 \to T, 2 \to F, 3 \to T\} \rangle$

Concepts...

- Vocabulary τ, ρ: relational, functional, constant symbols
  Examples: $\langle E^2 \rangle$, $\langle E^2, x, y \rangle$, $\langle \leq^2, S^1 \rangle$.
- Structures $\mathcal{A}$ on a vocabulary:
  universe $|\mathcal{A}|$ (assumed to be $\{0, 1, \ldots, n-1\}$),
  for each R^k in τ a relation of arity k defined on $|\mathcal{A}|$, etc.
  Examples:
  A graph: $\langle \{0, 1, 2\}, \{(1, 2), (2, 1)\} \rangle$,
  Word 1101: $\langle \{0, 1, 2, 3\}, \leq^2 = \{(0, 0), (0, 1), \ldots\},$
  $S^1 = \{0 \to T, 1 \to T, 2 \to F, 3 \to T\} \rangle$
- (Arity will be often omitted)

More concepts. . .

More concepts...

- STRUCT(τ): set of structures $\mathcal{A}$ with vocabulary τ

More concepts...

- STRUCT(τ): set of structures $\mathcal{A}$ with vocabulary τ
- Truth value of a logical formula ψ (with vocabulary τ) in a structure $\mathcal{A}$ (with vocabulary τ); in symbols $\mathcal{A} \models \psi$ (if TRUE)

More concepts...

- STRUCT(τ): set of structures $\mathcal{A}$ with vocabulary τ
- Truth value of a logical formula ψ (with vocabulary τ) in a structure $\mathcal{A}$ (with vocabulary τ); in symbols $\mathcal{A} \models \psi$ (if TRUE)
- Language L: $L \subseteq$STRUCT(τ)

More concepts...

- STRUCT(τ): set of structures $\mathcal{A}$ with vocabulary τ
- Truth value of a logical formula ψ (with vocabulary τ) in a structure $\mathcal{A}$ (with vocabulary τ); in symbols $\mathcal{A} \models \psi$ (if TRUE)
- Language L: $L \subseteq$STRUCT(τ)
- Language $L(\psi)$ defined by the logical formula ψ: set of structures $\mathcal{A}$ with vocabulary τ for which ψ is TRUE

$$L(\psi) = \{\mathcal{A} : \mathcal{A} \models \psi\}$$

More concepts...

- STRUCT(τ): set of structures $\mathcal{A}$ with vocabulary τ
- Truth value of a logical formula ψ (with vocabulary τ) in a structure $\mathcal{A}$ (with vocabulary τ); in symbols $\mathcal{A} \models \psi$ (if TRUE)
- Language L: $L \subseteq$STRUCT(τ)
- Language $L(\psi)$ defined by the logical formula ψ: set of structures $\mathcal{A}$ with vocabulary τ for which ψ is TRUE

$$L(\psi) = \{\mathcal{A} : \mathcal{A} \models \psi\}$$

- Example. Vocabulary $\tau = \langle E, x, y \rangle$, language $L = \{(V, E) : \text{there is a path } x \rightsquigarrow y\}$

On spectra

Spectra of FO sentences: one of the first studied problems of FMT (by Fagin and others).

On spectra

Spectra of FO sentences: one of the first studied problems of FMT (by Fagin and others).

Spectrum.

Let ρ be a logical sentence; the integer n belongs to the spectrum of ρ iff there is a structure satisfying ρ whose universe has cardinality n.

On spectra

Spectra of FO sentences: one of the first studied problems of FMT (by Fagin and others).

Spectrum.

Let ρ be a logical sentence; the integer n belongs to the spectrum of ρ iff there is a structure satisfying ρ whose universe has cardinality n.

Example.

The spectrum of the (conjunction of) the axioms of field theory is the set of powers of primes, $\{2, 3, 4, 5, 7, 8, 9, 11 \ldots\}$.

On spectra

Spectra of FO sentences: one of the first studied problems of FMT (by Fagin and others).

Spectrum.

Let ρ be a logical sentence; the integer n belongs to the spectrum of ρ iff there is a structure satisfying ρ whose universe has cardinality n.

Example.

The spectrum of the (conjunction of) the axioms of field theory is the set of powers of primes, $\{2, 3, 4, 5, 7, 8, 9, 11 \ldots\}$.

Open problem (Asser problem).

In first order logic, is the class of spectra closed under complement? For the example above, if the answer is YES, $\{1, 6, 10, 12, \ldots\}$ would be a spectrum.

Outline

First order logic (FO)

FO: preview

FO: preview

1. Classical results
2. Examples of properties expressed in FO
3. The 0/1 law

Some classical results

Some classical results

1. Completeness Theorem, Gödel. Ψ is a consequence of Φ iff Ψ is provable from Φ.

Some classical results

1. Completeness Theorem, Gödel. Ψ is a consequence of Φ iff Ψ is provable from Φ.
2. Corollary. The set of valid sentences is r.e.

Some classical results

1. Completeness Theorem, Gödel. Ψ is a consequence of Φ iff Ψ is provable from Φ.
2. Corollary. The set of valid sentences is r.e.
3. Compactness.
   (i) if Ψ is a consequence of Φ, it is a consequence of a finite subset of Φ.
   (ii) If every finite subset of Ψ is satisfiable, then Ψ is satisfiable.

Some classical results

1. Completeness Theorem, Gödel. Ψ is a consequence of Φ iff Ψ is provable from Φ.
2. Corollary. The set of valid sentences is r.e.
3. Compactness.
   (i) if Ψ is a consequence of Φ, it is a consequence of a finite subset of Φ.
   (ii) If every finite subset of Ψ is satisfiable, then Ψ is satisfiable.
4. Löwenheim-Skolem. If Ψ has a model, it has a model which is countable or finite (upper bound on the minimum cardinality).

Finite models only ("in the finite")

Finite models only ("in the finite")

1. Trakhtenbrot. The set of sentences of FO that are valid in the finite is not r.e.!

Finite models only ("in the finite")

1. Trakhtenbrot. The set of sentences of FO that are valid in the finite is not r.e.!
2. The Completeness and the Completness Theorems fail in the finite.

Expressing some graph properties in FO

Instance is a graph. Vocabulary E, universe $\{0, 1, \ldots, n-1\}$.

Expressing some graph properties in FO

Instance is a graph. Vocabulary E, universe $\{0, 1, \ldots, n-1\}$.

- Property: (V, E) is an undirected graph.
  FO: $\forall x\, \forall y\, \neg E(x, x) \wedge (E(x, y) \rightarrow E(y, x))$

Expressing some graph properties in FO

Instance is a graph. Vocabulary E, universe $\{0, 1, \ldots, n-1\}$.

- Property: (V, E) is an undirected graph.
  FO: $\forall x\ \forall y\ \neg E(x, x) \wedge (E(x, y) \rightarrow E(y, x))$
- Property: Every vertex has in-degree at least 2.
  FO: $\forall x\ \exists y\ \exists z\ x \neq y \wedge x \neq z \wedge y \neq z \wedge E(y, x) \wedge E(z, x)$

Expressing some graph properties in FO

Instance is a graph. Vocabulary E, universe $\{0, 1, \ldots, n-1\}$.

- Property: (V, E) is an undirected graph.
  FO: $\forall x\, \forall y\, \neg E(x,x) \wedge (E(x,y) \rightarrow E(y,x))$
- Property: Every vertex has in-degree at least 2.
  FO: $\forall x\, \exists y\, \exists z\; x \neq y \wedge x \neq z \wedge y \neq z \wedge E(y,x) \wedge E(z,x)$
- Property: The graph is complete.
  FO: Exercise

Expressing some graph properties in FO

Instance is a graph. Vocabulary E, universe $\{0, 1, \ldots, n-1\}$.

- Property: (V, E) is an undirected graph.
  FO: $\forall x\ \forall y\ \neg E(x, x) \wedge (E(x, y) \rightarrow E(y, x))$
- Property: Every vertex has in-degree at least 2.
  FO: $\forall x\ \exists y\ \exists z\ x \neq y \wedge x \neq z \wedge y \neq z \wedge E(y, x) \wedge E(z, x)$
- Property: The graph is complete.
  FO: EXERCISE
- Property: There is a path with length at most 3.
  FO: EXERCISE

Expressing some graph properties in FO

Instance is a graph. Vocabulary E, universe $\{0, 1, \ldots, n-1\}$.

- Property: (V, E) is an undirected graph.
  FO: $\forall x \, \forall y \, \neg E(x,x) \wedge (E(x,y) \rightarrow E(y,x))$
- Property: Every vertex has in-degree at least 2.
  FO: $\forall x \, \exists y \, \exists z \, x \neq y \wedge x \neq z \wedge y \neq z \wedge E(y,x) \wedge E(z,x)$
- Property: The graph is complete.
  FO: EXERCISE
- Property: There is a path with length at most 3.
  FO: EXERCISE
- Property: The graph is bipartite.
  FO: . . .

Expressing some graph properties in FO

Instance is a graph. Vocabulary E, universe $\{0, 1, \ldots, n-1\}$.

- Property: (V, E) is an undirected graph.
  FO: $\forall x \, \forall y \, \neg E(x,x) \wedge (E(x,y) \rightarrow E(y,x))$
- Property: Every vertex has in-degree at least 2.
  FO: $\forall x \, \exists y \, \exists z \, x \neq y \wedge x \neq z \wedge y \neq z \wedge E(y,x) \wedge E(z,x)$
- Property: The graph is complete.
  FO: EXERCISE
- Property: There is a path with length at most 3.
  FO: EXERCISE
- Property: The graph is bipartite.
  FO: . . . NOT POSSIBLE

Expressing some graph properties in FO

Instance is a graph. Vocabulary E, universe $\{0, 1, \ldots, n-1\}$.

- Property: (V, E) is an undirected graph.
  FO: $\forall x \, \forall y \, \neg E(x,x) \wedge (E(x,y) \rightarrow E(y,x))$
- Property: Every vertex has in-degree at least 2.
  FO: $\forall x \, \exists y \, \exists z \; x \neq y \wedge x \neq z \wedge y \neq z \wedge E(y,x) \wedge E(z,x)$
- Property: The graph is complete.
  FO: EXERCISE
- Property: There is a path with length at most 3.
  FO: EXERCISE
- Property: The graph is bipartite.
  FO: . . . NOT POSSIBLE
- Property: Given $\langle (V, E), x, y \rangle$, there is a path from x to y.
  FO: . . .

Expressing some graph properties in FO

Instance is a graph. Vocabulary E, universe $\{0, 1, \ldots, n-1\}$.

- Property: (V, E) is an undirected graph.
  FO: $\forall x\, \forall y\, \neg E(x, x) \wedge (E(x, y) \rightarrow E(y, x))$
- Property: Every vertex has in-degree at least 2.
  FO: $\forall x\, \exists y\, \exists z\; x \neq y \wedge x \neq z \wedge y \neq z \wedge E(y, x) \wedge E(z, x)$
- Property: The graph is complete.
  FO: EXERCISE
- Property: There is a path with length at most 3.
  FO: EXERCISE
- Property: The graph is bipartite.
  FO: ... NOT POSSIBLE
- Property: Given $\langle (V, E), x, y \rangle$, there is a path from x to y.
  FO: ... NOT POSSIBLE

The 0/1 law of FO

ψ: first order sentence.
$\mu_n(\psi)$: fraction of structures with an universe $\{0, 1, \ldots, n-1\}$ that satisfy ψ.

The 0/1 law of FO

ψ: first order sentence.
$\mu_n(\psi)$: fraction of structures with an universe $\{0, 1, \ldots, n-1\}$ that satisfy ψ.

Definition.
ψ is almost surely true if $\lim_{n\to\infty} \mu_n(\psi) = 1$.
ψ is almost surely false if $\lim_{n\to\infty} \mu_n(\psi) = 0$.

The 0/1 law of FO

ψ: first order sentence.
$\mu_n(\psi)$: fraction of structures with an universe $\{0, 1, \ldots, n-1\}$ that satisfy ψ.

Definition.
ψ is almost surely true if $\lim_{n\to\infty} \mu_n(\psi) = 1$.
ψ is almost surely false if $\lim_{n\to\infty} \mu_n(\psi) = 0$.

Theorem (the 0/1 law). Every FO sentence is almost surely true or almost surely false.

The 0/1 law of FO (cont.)

Fact. For the problem a EVEN ("is the cardinality of the universe even?") the limit $\lim_{n \to \infty} \mu_n(\psi)$ does not exist.

The 0/1 law of FO (cont.)

Fact. For the problem a EVEN ("is the cardinality of the universe even?") the limit $\lim_{n\to\infty} \mu_n(\psi)$ does not exist.

Corollary. the predicate EVEN is not expressible in FO

The 0/1 law of FO (cont.)

Fact. For the problem a EVEN ("is the cardinality of the universe even?") the limit $\lim_{n\to\infty} \mu_n(\psi)$ does not exist.

Corollary. the predicate EVEN is not expressible in FO

Theorem. The problem of deciding if a first-order sentence is almost surely false or almost surely true is PSPACE-complete.

The 0/1 law of FO (cont.)

Fact. For the problem a EVEN ("is the cardinality of the universe even?") the limit $\lim_{n\to\infty}\mu_n(\psi)$ does not exist.

Corollary. the predicate EVEN is not expressible in FO

Theorem. The problem of deciding if a first-order sentence is almost surely false or almost surely true is PSPACE-complete.

Summary. A sentence of FO could in principle be

- almost surely false (the limit is 0)
- almost surely true (the limit is 1)
- neither (the limit does not exist or $\notin \{0,1\}$)

The last case never happens!

Outline

First order logic with

“transitive closure” (TC)

and “least fixed point” (LFP)

operators

PREVIEW: *extending FO with TC and LFP*

- FO has very low expressivity
- SO is too powerful
- The power of FO can be increased with:
  - TC, the operator of transitive closure of relations
  - LFP, least fixed point, a more general operator.
- Restriction to positive forms (posTC and posFP) are often used.
- Characterization of complexity classes: NL = (FO+posTC), P=FO+LFP

Weakness of FO

FO has very low expressive power. For instance

- EVEN (cardinality of the universe) can not be expressed
- "Is a graph connected?" can not be expressed
- $\mathrm{FO} \subseteq L$

TC and LFP operators

Vocabulary of graphs with 2 endpoints, $\langle E, x, y \rangle$
Consider a expression involving a relation R

$$\psi(R, x, y) \;\equiv\; (x = y) \;\vee\; \exists z \; (E(x, z) \wedge R(z, y))$$

The expression ψ is monotone on R.
The least fixed point (LFP) of $\psi(R, x, y)$ is the relation:
"there is a path from x to y"

TC and LFP operators

Vocabulary of graphs with 2 endpoints, $\langle E, x, y\rangle$
Consider a expression involving a relation R

$$\psi(R, x, y) \equiv (x = y) \vee \exists z\ (E(x, z) \wedge R(z, y))$$

The expression ψ is monotone on R.
The least fixed point (LFP) of $\psi(R, x, y)$ is the relation:
"there is a path from x to y"

In this example TC (transitive closure) is the same as LFP.
In general, LFP is stronger than TC

TC

Let $\overline{x} \equiv (x_1 \ldots x_k)$ and $\overline{y} \equiv (y_1 \ldots y_k)$ be k-tuples.
Let $\phi(\overline{x}, \overline{y})$ be a relation (arity $2k$).

TC

Let $\overline{x} \equiv (x_1 \ldots x_k)$ and $\overline{y} \equiv (y_1 \ldots y_k)$ be k-tuples.
Let $\phi(\overline{x}, \overline{y})$ be a relation (arity $2k$).

Definition. $TC(\phi)$ is the reflexive symmetric closure of ϕ.

TC

Let $\overline{x} \equiv (x_1 \ldots x_k)$ and $\overline{y} \equiv (y_1 \ldots y_k)$ be k-tuples.
Let $\phi(\overline{x}, \overline{y})$ be a relation (arity $2k$).

Definition. $TC(\phi)$ is the reflexive symmetric closure of ϕ.

Definition. (FO+TC) is the set of properties (complexity class) expressible FO plus the operator TC.
(FO+posTC) similar, but uses only positive applications of the operator TC.

Some results involving TC

Theorem. $\mathsf{NL} = (\mathsf{FO}{+}\mathsf{posTC})$

Some results involving TC

Theorem. NL $=$ (FO+posTC)

Theorem. Some predicates expressible (FO+posTC):

- PLUS$(x, y, z) \equiv (x + y = z)$
- ODD(x)
- ON(w, b): $(b < \log n)$ and the bit b of the word w is 1
- PATH(E, x, y).

LFP

Theorem

If R is a new relation symbol with arity k and if $\psi(R, x_1, \ldots, x_k)$ is a monotone FO expression, then, for any structure $\mathcal{A}$, the least fixed point (LFP) of $\psi^{\mathcal{A}}$ exists and it is equal to the least r such that $(\psi^{\mathcal{A}}(\varnothing))^r = (\psi^{\mathcal{A}}(\varnothing))^{r+1}$ □

LFP

Theorem

If R is a new relation symbol with arity k and if $\psi(R, x_1, \ldots, x_k)$ is a monotone FO expression, then, for any structure $\mathcal{A}$, the least fixed point (LFP) of $\psi^{\mathcal{A}}$ exists and it is equal to the least r such that $(\psi^{\mathcal{A}}(\varnothing))^r = (\psi^{\mathcal{A}}(\varnothing))^{r+1}$ □

Previous example: algorithm to find the LFP → algorithm to solve "is there a path from x to y?"

FO + LFP

FO+LFP: FO + least fixed point of monotone relations (inductive definition):

FO + LFP

FO+LFP: FO + least fixed point of monotone relations (inductive definition):
Let $\psi(R, x_1, \ldots, x_k)$ be a FO+LFP formula, which is monotone (positive) relatively to R (arity k). Then, a new k-ary relation symbol can be used to denote LFP(ψ).

Examples

- $\langle E, x, y \rangle$, is there a path from x to y?
  May be expressed as
  $\mathrm{LFP}(\psi(R, x, y)) \equiv (x = y) \vee \exists z\, (E(x, z) \wedge R(z, y))$

Examples

- $\langle E, x, y\rangle$, is there a path from x to y?
  May be expressed as
  $\text{LFP}(\psi(R, x, y)) \equiv (x = y) \vee \exists z\,(E(x, z) \wedge R(z, y))$
- $\langle E, \alpha, x, y\rangle$, is there a path from x to y in the alternating graph?
  Alternating graph: A vertex z is either “universal” (if $\alpha(z)$ holds) or “existential”. Existential $z \rightarrow$ a path that passes by z may follow any edge (z, w). Universal z: the path must follow all edges (z, w) (and there must be at least one).
  May be expressed as
  $\text{LFP}(\phi(R, x, y)) \equiv (x = y) \vee$
  $[\exists z\,(E(x, z) \wedge R(z, y)) \wedge A(x) \rightarrow \forall z(E(x, z) \rightarrow R(z, y))])$

2 results

- Theorem. (again) NL = FO+posTC
- Theorem. With linear order, P = FO+LFP

Outline

Second order logic (SO)

Second order logic (SO)

Relations can be quantified – second order quantification.

Second order logic (SO)

Relations can be quantified – second order quantification.

- Second order quantifiers can be placed at the beginning of any SO formula (one of the following forms)

$$\exists R_1 \exists R_2 \ldots \forall R'_1 \forall R'_2 \ldots \exists R''_1 \exists R''_2 \ldots \psi(R_1, R_2, R'_1, R'_2, \ldots)$$

$$\forall R_1 \forall R_2 \ldots \exists R'_1 \exists R'_2 \ldots \forall R''_1 \forall R''_2 \ldots \psi(R_1, R_2, R'_1, R'_2, \ldots)$$

where ψ is a FOL formula.

Second order logic (SO)

Relations can be quantified – second order quantification.

- Second order quantifiers can be placed at the beginning of any SO formula (one of the following forms)

  $$\exists R_1 \exists R_2 \ldots \forall R'_1 \forall R'_2 \ldots \exists R''_1 \exists R''_2 \ldots \psi(R_1, R_2, R'_1, R'_2, \ldots)$$

  $$\forall R_1 \forall R_2 \ldots \exists R'_1 \exists R'_2 \ldots \forall R''_1 \forall R''_2 \ldots \psi(R_1, R_2, R'_1, R'_2, \ldots)$$

  where ψ is a FOL formula.
- Special case: SO$\exists$ logic, when there are only existential second order quantifiers, $\exists R_1 \exists R_2 \ldots \psi(R_1, R_2, \ldots)$ where ψ is a FOL formula.

Expressing 3-colourability in SO∃

Expressing 3-colourability in SO∃

- Property: (V, E) is 3-colourable.
  Expression in in SO∃: (R, G and B are unary relations)

Expressing 3-colourability in SO∃

- Property: (V, E) is 3-colourable.
  Expression in in SO∃: (R, G and B are unary relations)

$\exists R \; \exists G \; \exists B$
Each vertex has at least one color:
$[\forall x \; R(x) \vee G(x) \vee B(x) \wedge$
But not 2 colors:
$\neg(R(x) \wedge G(x)) \wedge \neg(G(x) \wedge B(x)) \wedge \neg(B(x) \wedge R(x))] \wedge$
Edge condition:
$\forall x \; \forall y \; E(x, y) \rightarrow$
$\neg(R(x) \wedge R(y)) \wedge \neg(G(x) \wedge G(y)) \wedge \neg(B(x) \wedge B(y))$

FO and complexity classes

FO and complexity classes

Theorem (Fagin). NP = SO∃

FO and complexity classes

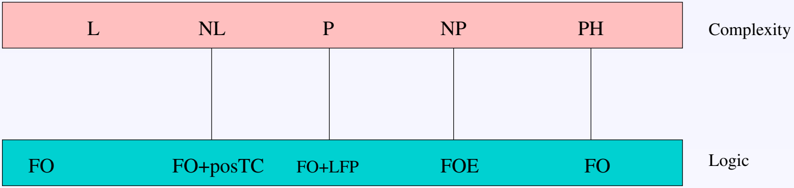
Theorem (Fagin). $\mathsf{NP} = \mathsf{SO}\exists$

Theorem. $\mathsf{PH} = \mathsf{SO}$

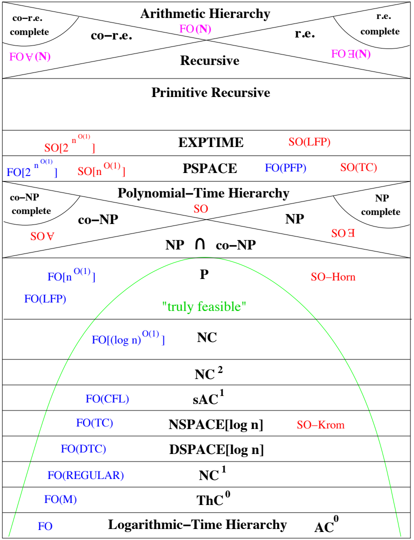
Outline

Finishing

Complexity classes and expressivity in logic



A more detailed picture (Immerman)



Expressing complexity in logic...

Surprisingly, it turns out that, in some cases, we can characterize complexity classes (like NP) in terms of logic, where there is no notion of machine, computation or time. *Fagin*

Expressing complexity in logic...

Surprisingly, it turns out that, in some cases, we can characterize complexity classes (like NP) in terms of logic, where there is no notion of machine, computation or time. *Fagin*

Comments

Expressing complexity in logic...

Surprisingly, it turns out that, in some cases, we can characterize complexity classes (like NP) in terms of logic, where there is no notion of machine, computation or time. *Fagin*

Comments

- Certain concepts are robust relatively to the model of computation used.
- Codification can change the class of a problem; for instance PARTITION is
  - NP-complete when integers are codified in binary (say an unary relation on the universe).
  - P when an integer m is codified as $s^m(0)$ ($s(s(\ldots s(0)\ldots))$, m applications of s).

From Turing machines to logic (part I)

An NP problem π characterized by predicate $p(x, y)$.
Notation:

- TM: (deterministic) Turing machine
- NDTM: non-deterministic Turing machine
- $p(x, y)$: predicate computable in polynomial time (in terms of $|x|$). If y is used, as we assume, $|y|$ must be bounded by a polynomial in $|x|$.
- $q(n)$: a polynomial.

$$x \in L_\pi \leftrightarrow \exists y,\ p(x, y)$$

where the evaluation of $p(x, y)$ can be made in polynomial time in $|x|$.

From Turing machines to logic (part II)

Expressing the NP problem "given x, $\exists y\ p(x, y)$?"

From Turing machines to logic (part II)

Expressing the NP problem "given x, $\exists y\ p(x, y)$?"

- NDTM M: There is an accepting computation of M with input x and time $\leq q(|x|)$.

From Turing machines to logic (part II)

Expressing the NP problem "given x, $\exists y\ p(x, y)$?"

- NDTM M: There is an accepting computation of M with input x and time $\leq q(|x|)$.
- 2-phase NDTM M: the non-deterministic phase (the first part) of M can be such that the deterministic phase (the second part) of M finishes in an accepting state, and the overall time $\leq q(|x|)$.

From Turing machines to logic (part II)

Expressing the NP problem "given x, $\exists y \; p(x, y)$?"

- NDTM M: There is an accepting computation of M with input x and time $\leq q(|x|)$.
- 2-phase NDTM M: the non-deterministic phase (the first part) of M can be such that the deterministic phase (the second part) of M finishes in an accepting state, and the overall time $\leq q(|x|)$.
- TM with a part of the input tape unspecified M: The initial contents of the tape is $\langle y, x \rangle$; where y is an initially unspecified tape segment with length $q(|x|)$. The machine M can (depending on y) finish in an accepting state in time $\leq q(|x|)$.

From Turing machines to logic (part II)

Expressing the NP problem "given x, $\exists y\ p(x, y)$?"

- NDTM M: There is an accepting computation of M with input x and time $\leq q(|x|)$.
- 2-phase NDTM M: the non-deterministic phase (the first part) of M can be such that the deterministic phase (the second part) of M finishes in an accepting state, and the overall time $\leq q(|x|)$.
- TM with a part of the input tape unspecified M: The initial contents of the tape is $\langle y, x \rangle$; where y is an initially unspecified tape segment with length $q(|x|)$. The machine M can (depending on y) finish in an accepting state in time $\leq q(|x|)$.
- FO$\exists$: $\exists Y\ \psi(X, Y)$ where x is represented by the unary relation X, y is represented by the unary relation Y, the formula ψ "corresponds" to the problem in question π.

Where is the execution time in the last case?

Where is the execution time?

I do not know; but the answer may be related to

- Typically, only one universe is used, with a size which is polynomially related with the input length. Thus, possibly, numbers much larger than the input can not be represented.
- A FO formula can be computed in polynomial time: each existential or universal quantifier corresponds to a loop "`for i=0 to n-1`...".
- Testing the existence of a certain relation (FO$\exists$ logic) may, in principle, need exponential time.

Things we didn't talk about:

- Reductions (FO, etc.)
- Complete problems
- The Ehrenfeucht-Fräissé method
- Specific applications
  - Database theory
  - Unary relations only
  - FO + ATC
  - Real numbers
  - FA

Some references

- Fagin, *Finite-model theory – a personal perspective*
- Flum, *Finite Model Theory, (book)*
- Gurevich, *Toward logic tailored for computational complexity*
- Kolaitis, *Combinatorial games in Finite model theory*
- Immerman, *Descriptive Complexity, (book)*
- Immerman, *Languages that capture complexity classes*
- Väänänen, *A Short Course on Finite Model Theory*

The end

Outline

Note on the The Ehrenfeucht-Fräissé method I

The game has m moves. There are 2 players, the Spoiler (S) and the Duplicator (D). There are 2 structures, $\mathcal{A}$ and $\mathcal{B}$, known to the players. In each move

- S selects an element from $\mathcal{A}$ or from $\mathcal{B}$
- D selects an element from $\mathcal{B}$ or from $\mathcal{A}$

After finishing there are 2 sets of elements

$$\{a_1, \ldots, a_m\}, \quad \{b_1, \ldots, b_m\}$$

If the mapping $a_1 \leftrightarrow b_1, \ldots, a_m \leftrightarrow b_m$, is an isomorphism (induced by the structures $\mathcal{A}$ and $\mathcal{B}$), then the player D wins.
Otherwise S wins.

Note on the The Ehrenfeucht-Fräïssé method II

The player D wins iff there is an winning strategy.

The EF game is the main method for the study of the definability of classes of finite structures in FOL.

Unary symbols

Vocabulary $\tau = \{f\}$, only one unary function symbol.
Vocabulary $\tau_k = \{f, R_1, \ldots, R_k\}$, only one unary function symbol and (only) k unary relation symbols..

Some results in: Durand, Fagin, Loesher, "Spectra with only unary function symbols".

Unary symbols

Vocabulary $\tau = \{f\}$, only one unary function symbol.
Vocabulary $\tau_k = \{f, R_1, \ldots, R_k\}$, only one unary function symbol and (only) k unary relation symbols..

Some results in: Durand, Fagin, Loesher, "Spectra with only unary function symbols".

1. Equivalent statements.
   (i) S is the spectra of a FO τ-sentence. (ii) For some fixed $k \in \mathbb{N}$, S is the spectra of a FO τ_k-sentence. (iii) S is ultimately periodic (possibly finite). (iv) S is the set of the lengths of the words belonging to a regular language.

Unary symbols

Vocabulary $\tau = \{f\}$, only one unary function symbol.
Vocabulary $\tau_k = \{f, R_1, \ldots, R_k\}$, only one unary function symbol and (only) k unary relation symbols..

Some results in: Durand, Fagin, Loesher, "Spectra with only unary function symbols".

1. Equivalent statements.
   (i) S is the spectra of a FO τ-sentence. (ii) For some fixed $k \in \mathbb{N}$, S is the spectra of a FO τ_k-sentence. (iii) S is ultimately periodic (possibly finite). (iv) S is the set of the lengths of the words belonging to a regular language.
2. Corollary. The class of spectra involving only one functional symbol is closed under complement (Asser problem).

Unary symbols

Vocabulary $\tau = \{f\}$, only one unary function symbol.
Vocabulary $\tau_k = \{f, R_1, \ldots, R_k\}$, only one unary function symbol and (only) k unary relation symbols..

Some results in: Durand, Fagin, Loesher, "Spectra with only unary function symbols".

1. Equivalent statements.
   (i) S is the spectra of a FO τ-sentence. (ii) For some fixed $k \in \mathbb{N}$, S is the spectra of a FO τ_k-sentence. (iii) S is ultimately periodic (possibly finite). (iv) S is the set of the lengths of the words belonging to a regular language.
2. Corollary. The class of spectra involving only one functional symbol is closed under complement (Asser problem).
3. More than one. There is a spectrum with only 2 unary function symbols, which is not a spectrum corresponding to a vocabulary involving only 1 unary function symbol.

The role of ordering

FO+LFP only captures P in the presence or ordering, (Chandra, Harel, Immerman, Vardi)

To characterize some complexity classes, FO descriptive complexity requires ordering on its universe. Such an ordering, however, is irrelevant to the properties of the graphs or databases that we want to compute. It is easy to prove lower bounds on languages without ordering; on ordered structures these arguments do not work. Furthermore, for databases, the ordering is a low-level implementation issues – how entries are stored in memory or disk – which should be invisible to the person writing queries. For all these reasons, the descriptive characterization of order-independent queries computable in a given complexity class is a fundamental open problem.

Immerman

Generalized spectra

Generalized spectra

1. Relation quantification.

   If all the relations of a FO sentence $\psi(R_1, \ldots, R_k)$ are existentially quantified, $\psi' = \exists R_1 \ldots \exists R_k \psi$, the vocabulary becomes empty.

Generalized spectra

1. Relation quantification.
   If all the relations of a FO sentence $\psi(R_1, \dots, R_k)$ are existentially quantified, $\psi' = \exists R_1 \dots \exists R_k \psi$, the vocabulary becomes empty.
2. Generalized spectra.
   Only some relations of a FO sentence $\psi(P_1, P_2, \dots . P_m, R_1, \dots, R_k)$ are existentially quantified, $\psi' = \exists R_1 \dots \exists R_k \psi$, the vocabulary is $\{P_1, \dots, P_m\}$.

Generalized spectra

1. Relation quantification.

   If all the relations of a FO sentence $\psi(R_1, \ldots, R_k)$ are existentially quantified, $\psi' = \exists R_1 \ldots \exists R_k \psi$, the vocabulary becomes empty.

2. Generalized spectra.

   Only some relations of a FO sentence $\psi(P_1, P_2, \ldots .P_m, R_1, \ldots, R_k)$ are existentially quantified, $\psi' = \exists R_1 \ldots \exists R_k \psi$, the vocabulary is $\{P_1, \ldots, P_m\}$.

3. Example.

   A graph is 3-colourable, $\exists R \exists G \exists B \psi(E, R, G, B)$. the vocabulary is $\{E\}$.

Generalized spectra

1. Relation quantification.
   If all the relations of a FO sentence $\psi(R_1, \ldots, R_k)$ are existentially quantified, $\psi' = \exists R_1 \ldots \exists R_k \psi$, the vocabulary becomes empty.
2. Generalized spectra.
   Only some relations of a FO sentence $\psi(P_1, P_2, \ldots .P_m, R_1, \ldots, R_k)$ are existentially quantified, $\psi' = \exists R_1 \ldots \exists R_k \psi$, the vocabulary is $\{P_1, \ldots, P_m\}$.
3. Example.
   A graph is 3-colourable, $\exists R \exists G \exists B \psi(E, R, G, B)$. the vocabulary is $\{E\}$.
4. Theorem (Fagin)
   The class of generalized spectra is closed under complement, iff NP is closed under complement (NP=co-NP).

Generalized spectra

1. Relation quantification.

   If all the relations of a FO sentence $\psi(R_1, \ldots, R_k)$ are existentially quantified, $\psi' = \exists R_1 \ldots \exists R_k \psi$, the vocabulary becomes empty.

2. Generalized spectra.

   Only some relations of a FO sentence $\psi(P_1, P_2, \ldots .P_m, R_1, \ldots, R_k)$ are existentially quantified, $\psi' = \exists R_1 \ldots \exists R_k \psi$, the vocabulary is $\{P_1, \ldots, P_m\}$.

3. Example.

   A graph is 3-colourable, $\exists R \exists G \exists B \psi(E, R, G, B)$. the vocabulary is $\{E\}$.

4. Theorem (Fagin)

   The class of generalized spectra is closed under complement, iff NP is closed under complement (NP=co-NP).

5. Theorem (Fagin)

   The complement of a generalized spectrum is a generalized spectrum iff the class of graphs that are not 3-colourable is a generalized spectrum (pure logic, no complexity involved!).

FO+ATC (a more general form of TC)

ATC: alternating relation closure

Consider formulas $\phi(\overline{x}, \overline{y})$ where $\overline{x}$ and $\overline{y}$ have arity k and $\alpha(\overline{x})$.

For any structure $\mathcal{A}$, an alternating graph $G_{\psi,\alpha}$ is defined by ϕ and α.

FO+ATC (a more general form of TC)

ATC: alternating relation closure

Consider formulas $\phi(\overline{x}, \overline{y})$ where $\overline{x}$ and $\overline{y}$ have arity k and $\alpha(\overline{x})$.

For any structure $\mathcal{A}$, an alternating graph $G_{\psi,\alpha}$ is defined by ϕ and α.

ATC operator: is there a path $\overline{x} \to \overline{y}$?

Formally: $\mathrm{ATC}(\phi, \alpha) = \mathrm{LFP}(\Psi(\phi, \alpha, R))$ where

$$\Psi(\phi, \alpha, R) \equiv (x = y) \lor$$

$$\exists z(\phi(z, x) \land R(z, y)) \land (\alpha(x) \to \forall z(\phi(x, z) \to R(z, y)))$$