ESO universal Horn logic: Expressions at machine level versus structure level

Prabhu Manyem
Department of Mathematics
Shanghai University, China

(With Nerio Borges, Simon Bolivar University, Venezuela)

Goal, Purpose

Goal, Purpose

- There are Logic expressions at the *machine level* and the *structure level*.

Goal, Purpose

- There are Logic expressions at the *machine level* and the *structure level*.
- We show that these two don't have the same expressive power.

Goal, Purpose

- There are Logic expressions at the *machine level* and the *structure level*.
- We show that these two don't have the same expressive power.
- We consider the case of ESO universal Horn logic (ESO Π_1 Horn).

Goal, Purpose

- There are Logic expressions at the *machine level* and the *structure level*.
- We show that these two don't have the same expressive power.
- We consider the case of ESO universal Horn logic (ESO Π_1 Horn).
- For ESO Π_1 Horn, we'll show that machine level expressions are *more powerful*.

Goal, Purpose

- There are Logic expressions at the *machine level* and the *structure level*.
- We show that these two don't have the same expressive power.
- We consider the case of ESO universal Horn logic (ESO Π_1 Horn).
- For ESO Π_1 Horn, we'll show that machine level expressions are *more powerful*.
- What does *more powerful* mean? It means that it can express more number of properties.

Goal, Purpose

- There are Logic expressions at the *machine level* and the *structure level*.
- We show that these two don't have the same expressive power.
- We consider the case of ESO universal Horn logic (ESO Π_1 Horn).
- For ESO Π_1 Horn, we'll show that machine level expressions are *more powerful*.
- What does *more powerful* mean? It means that it can express more number of properties.
- By a *property*, we refer to a given mathematical structure (the input) possessing this property.

Goal, Purpose

- There are Logic expressions at the *machine level* and the *structure level*.
- We show that these two don't have the same expressive power.
- We consider the case of ESO universal Horn logic (ESO Π_1 Horn).
- For ESO Π_1 Horn, we'll show that machine level expressions are *more powerful*.
- What does *more powerful* mean? It means that it can express more number of properties.
- By a *property*, we refer to a given mathematical structure (the input) possessing this property.
- For instance, an input graph $G = (V, E)$ may contain a Hamiltonian cycle. The *property* here is the existence of an HC (Hamiltonian cycle) in G.

Continuing with the HC example

Continuing with the HC example

- So the set of all graphs G that contain a HC (as opposed to those graphs that don't) is *equivalent* to the HC property.

Continuing with the HC example

- So the set of all graphs G that contain a HC (as opposed to those graphs that don't) is *equivalent* to the HC property.
- We can say,
  $\mathrm{HC} \equiv \{G \mid G \text{ contains a Hamiltonian cycle}\}$.

Continuing with the HC example

- So the set of all graphs G that contain a HC (as opposed to those graphs that don't) is *equivalent* to the HC property.
- We can say,
  HC $\equiv \{G \mid G$ contains a Hamiltonian cycle$\}$.
- In general, a property $\mathcal{P}$ can also refer to a class of isomorphism closed structures that possess the particular property in question.

Continuing with the HC example

- So the set of all graphs G that contain a HC (as opposed to those graphs that don't) is *equivalent* to the HC property.
- We can say,
  HC $\equiv \{G \mid G$ contains a Hamiltonian cycle$\}$.
- In general, a property $\mathcal{P}$ can also refer to a class of isomorphism closed structures that possess the particular property in question.
- In Logic, properties are captured by logic expressions (formulae).

Continuing with the HC example

- So the set of all graphs G that contain a HC (as opposed to those graphs that don't) is *equivalent* to the HC property.
- We can say,
  HC $\equiv \{G \mid G$ contains a Hamiltonian cycle$\}$.
- In general, a property $\mathcal{P}$ can also refer to a class of isomorphism closed structures that possess the particular property in question.
- In Logic, properties are captured by logic expressions (formulae).
- A structure $\mathcal{A}$ possesses a certain property $\mathcal{P}$ iff $\mathcal{A}$ satisfies a certain formula ϕ, that is iff $\mathcal{A} \models \phi$.

Continuing with the HC example

- So the set of all graphs G that contain a HC (as opposed to those graphs that don't) is *equivalent* to the HC property.
- We can say,
  HC $\equiv \{G \mid G$ contains a Hamiltonian cycle$\}$.
- In general, a property $\mathcal{P}$ can also refer to a class of isomorphism closed structures that possess the particular property in question.
- In Logic, properties are captured by logic expressions (formulae).
- A structure $\mathcal{A}$ possesses a certain property $\mathcal{P}$ iff $\mathcal{A}$ satisfies a certain formula ϕ, that is iff $\mathcal{A} \models \phi$.
- $\mathcal{P} \equiv \{\mathcal{A} \mid \mathcal{A} \models \phi\}$.

Machine level and structure level expressions

Machine level and structure level expressions

- Structure level expressions (SLE) are direct.

Machine level and structure level expressions

- Structure level expressions (SLE) are direct. They express the possession of a property by an (input) structure without reference to any computation (nor do they use any computation device).

Machine level and structure level expressions

- Structure level expressions (SLE) are direct. They express the possession of a property by an (input) structure without reference to any computation (nor do they use any computation device).
- A machine level expression (MLE) uses a computation device (such as a Turing machine - TM).

Machine level and structure level expressions

- Structure level expressions (SLE) are direct. They express the possession of a property by an (input) structure without reference to any computation (nor do they use any computation device).
- A machine level expression (MLE) uses a computation device (such as a Turing machine - TM). They encode the steps of a TM.

Machine level and structure level expressions

- Structure level expressions (SLE) are direct. They express the possession of a property by an (input) structure without reference to any computation (nor do they use any computation device).
- A machine level expression (MLE) uses a computation device (such as a Turing machine - TM). They encode the steps of a TM.
- Such an expression says that given a structure $\mathcal{A}$, and if the TM goes through its computation steps and eventually reaches an "accept state", then $\mathcal{A}$ possesses the property in question.

Machine level and structure level expressions

- Structure level expressions (SLE) are direct. They express the possession of a property by an (input) structure without reference to any computation (nor do they use any computation device).
- A machine level expression (MLE) uses a computation device (such as a Turing machine - TM). They encode the steps of a TM.
- Such an expression says that given a structure $\mathcal{A}$, and if the TM goes through its computation steps and eventually reaches an "accept state", then $\mathcal{A}$ possesses the property in question.
- Hence an MLE encodes (1) the computation steps and (2) the arrival at the *accept state* at the end.

Machine level and structure level expressions

- Structure level expressions (SLE) are direct. They express the possession of a property by an (input) structure without reference to any computation (nor do they use any computation device).
- A machine level expression (MLE) uses a computation device (such as a Turing machine - TM). They encode the steps of a TM.
- Such an expression says that given a structure $\mathcal{A}$, and if the TM goes through its computation steps and eventually reaches an "accept state", then $\mathcal{A}$ possesses the property in question.
- Hence an MLE encodes (1) the computation steps and (2) the arrival at the *accept state* at the end.
- If we only use ESO Π_1 Horn logic and its fragments, then we find that $\mathcal{P}(sle) \subset \mathcal{P}(mle)$.

Machine level, structure level and ESO Π_1 Horn logic

Machine level, structure level and ESO Π_1 Horn logic

- $\mathcal{P}(sle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by structure level expressions.

Machine level, structure level and ESO Π_1 Horn logic

- $\mathcal{P}(sle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by structure level expressions.
- $\mathcal{P}(mle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by machine level expressions.

Machine level, structure level and ESO Π_1 Horn logic

- $\mathcal{P}(sle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by structure level expressions.
- $\mathcal{P}(mle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by machine level expressions.
- We restrict our attention to ordered structures (universe is linearly ordered).

Machine level, structure level and ESO Π_1 Horn logic

- $\mathcal{P}(sle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by structure level expressions.
- $\mathcal{P}(mle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by machine level expressions.
- We restrict our attention to ordered structures (universe is linearly ordered).
- Then, $\mathcal{P}(mle)$ is precisely the well-known class P, the set of properties that can be decided in polynomial time, deterministically (Grädel 1991).

Machine level, structure level and ESO Π_1 Horn logic

- $\mathcal{P}(sle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by structure level expressions.
- $\mathcal{P}(mle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by machine level expressions.
- We restrict our attention to ordered structures (universe is linearly ordered).
- Then, $\mathcal{P}(mle)$ is precisely the well-known class P, the set of properties that can be decided in polynomial time, deterministically (Grädel 1991).
  Given an input A and a property $\mathcal{P}$, it can be decided within polynomial time in the size of A, whether A satisfies $\mathcal{P}$.

Machine level, structure level and ESO Π_1 Horn logic

- $\mathcal{P}(sle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by structure level expressions.
- $\mathcal{P}(mle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by machine level expressions.
- We restrict our attention to ordered structures (universe is linearly ordered).
- Then, $\mathcal{P}(mle)$ is precisely the well-known class P, the set of properties that can be decided in polynomial time, deterministically (Grädel 1991).
  Given an input A and a property $\mathcal{P}$, it can be decided within polynomial time in the size of A, whether A satisfies $\mathcal{P}$.
- So what are the properties (or problems) in $\mathcal{P}(mle) \setminus \mathcal{P}(sle)$?

Machine level, structure level and ESO Π_1 Horn logic

- $\mathcal{P}(sle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by structure level expressions.
- $\mathcal{P}(mle)$: Set of properties that can be expressed (in ESO Π_1 Horn logic) by machine level expressions.
- We restrict our attention to ordered structures (universe is linearly ordered).
- Then, $\mathcal{P}(mle)$ is precisely the well-known class P, the set of properties that can be decided in polynomial time, deterministically (Grädel 1991).
  Given an input A and a property $\mathcal{P}$, it can be decided within polynomial time in the size of A, whether A satisfies $\mathcal{P}$.
- So what are the properties (or problems) in $\mathcal{P}(mle) \setminus \mathcal{P}(sle)$? These are *decision* problems based on optimization (solvable in polynomial time).

Decision problems based on optimization

Decision problems based on optimization

- Optimization problem:

$$\begin{aligned}
&\text{Maximize } f(\mathbf{x}) \text{ (the \textit{objective function}),}\\
&\text{subject to the following constraints:}\\
&h_j(\mathbf{x}) = 0, \quad 1 \le j \le p,\\
&g_i(\mathbf{x}) \le 0, \quad 1 \le i \le m,\\
&\text{and } \mathbf{x} \in X \text{ (for example, } X = \mathbb{R}^n\text{).}
\end{aligned} \tag{1}$$

Decision problems based on optimization

- Optimization problem:

$$\text{Maximize } f(\mathbf{x}) \text{ (the \textit{objective function}),}$$
$$\text{subject to the following constraints:}$$
$$h_j(\mathbf{x}) = 0, \quad 1 \leq j \leq p, \tag{1}$$
$$g_i(\mathbf{x}) \leq 0, \quad 1 \leq i \leq m,$$
$$\text{and } \mathbf{x} \in X \text{ (for example, } X = \mathbb{R}^n\text{).}$$

- Decision problem based on the above:

Decision problems based on optimization

- Optimization problem:

$$\begin{aligned} &\text{Maximize } f(\mathbf{x}) \text{ (the \textit{objective function}),} \\ &\text{subject to the following constraints:} \\ &h_j(\mathbf{x}) = 0, \quad 1 \le j \le p, \\ &g_i(\mathbf{x}) \le 0, \quad 1 \le i \le m, \\ &\text{and } \mathbf{x} \in X \text{ (for example, } X = \mathbb{R}^n\text{).} \end{aligned} \tag{1}$$

- Decision problem based on the above:
  Given a lower bound K, is there an $\mathbf{x} \in X$, such that
  $g_i(\mathbf{x}) \le 0 \quad (1 \le i \le m)$,
  $h_j(\mathbf{x}) = 0 \quad (1 \le j \le p)$, and $f(\mathbf{x}) \ge K$?

Decision problems based on optimization

Decision problems based on optimization

- Objective function constraint (OFC): $f(\mathbf{x}) \geq K$

Decision problems based on optimization

- Objective function constraint (OFC): $f(\mathbf{x}) \geq K$
- The others, the basic feasibility constraints (BFC): $g_i(\mathbf{x}) \leq 0 \quad (1 \leq i \leq m)$ and $h_j(\mathbf{x}) = 0 \quad (1 \leq j \leq p)$.

Decision problems based on optimization

- Objective function constraint (OFC): $f(\mathbf{x}) \geq K$
- The others, the basic feasibility constraints (BFC): $g_i(\mathbf{x}) \leq 0 \quad (1 \leq i \leq m)$ and $h_j(\mathbf{x}) = 0 \quad (1 \leq j \leq p)$.
- The decision problem is a conjunction of the OFC and the BFC.

Decision problems based on optimization

- Objective function constraint (OFC): $f(\mathbf{x}) \geq K$
- The others, the basic feasibility constraints (BFC): $g_i(\mathbf{x}) \leq 0 \quad (1 \leq i \leq m)$ and $h_j(\mathbf{x}) = 0 \quad (1 \leq j \leq p)$.
- The decision problem is a conjunction of the OFC and the BFC.
- So why is such a (polynomially solvable) problem in $\mathcal{P}(mle) \setminus \mathcal{P}(sle)$?

Decision problems based on optimization

- Objective function constraint (OFC): $f(\mathbf{x}) \geq K$
- The others, the basic feasibility constraints (BFC): $g_i(\mathbf{x}) \leq 0 \quad (1 \leq i \leq m)$ and $h_j(\mathbf{x}) = 0 \quad (1 \leq j \leq p)$.
- The decision problem is a conjunction of the OFC and the BFC.
- So why is such a (polynomially solvable) problem in $\mathcal{P}(mle) \setminus \mathcal{P}(sle)$?
- (recall) $\mathcal{P}(mle)$: Set of properties that can be expressed in ESO Π_1 Horn logic, by machine level expressions.

Decision problems based on optimization

- Objective function constraint (OFC): $f(\mathbf{x}) \geq K$
- The others, the basic feasibility constraints (BFC): $g_i(\mathbf{x}) \leq 0 \quad (1 \leq i \leq m)$ and $h_j(\mathbf{x}) = 0 \quad (1 \leq j \leq p)$.
- The decision problem is a conjunction of the OFC and the BFC.
- So why is such a (polynomially solvable) problem in $\mathcal{P}(mle) \setminus \mathcal{P}(sle)$?
- (recall) $\mathcal{P}(mle)$: Set of properties that can be expressed in ESO Π_1 Horn logic, by machine level expressions.
- Reason: The OFC cannot be expressed in Π_1 Horn form (universal Horn).

Decision problems based on optimization

- Objective function constraint (OFC): $f(\mathbf{x}) \geq K$
- The others, the basic feasibility constraints (BFC): $g_i(\mathbf{x}) \leq 0 \quad (1 \leq i \leq m)$ and $h_j(\mathbf{x}) = 0 \quad (1 \leq j \leq p)$.
- The decision problem is a conjunction of the OFC and the BFC.
- So why is such a (polynomially solvable) problem in $\mathcal{P}(mle) \setminus \mathcal{P}(sle)$?
- (recall) $\mathcal{P}(mle)$: Set of properties that can be expressed in ESO Π_1 Horn logic, by machine level expressions.
- Reason: The OFC cannot be expressed in Π_1 Horn form (universal Horn).

  (Anything to do with cardinality or counting.)

Sentences in ESO universal Horn logic

Sentences in ESO universal Horn logic

- $\exists \mathbf{S} \phi \equiv \exists S_1 \cdots \exists S_r \ \phi$, $\quad \phi$: first order formula.

Sentences in ESO universal Horn logic

- $\exists \mathbf{S} \phi \equiv \exists S_1 \cdots \exists S_r\ \phi$, ϕ: first order formula.
- $\mathbf{S} = (S_1, \cdots, S_r)$:
  second order predicates, not in the first order vocabulary.

Sentences in ESO universal Horn logic

- $\exists \mathbf{S}\phi \equiv \exists S_1 \cdots \exists S_r \ \phi$, ϕ: first order formula.
- $\mathbf{S} = (S_1, \ \cdots, \ S_r)$:
  second order predicates, not in the first order vocabulary.
- The second order predicates capture the solution to the optimization problem. (the unknowns)

Sentences in ESO universal Horn logic

- $\exists \mathbf{S} \phi \equiv \exists S_1 \cdots \exists S_r \ \phi$, ϕ: first order formula.
- $\mathbf{S} = (S_1, \ \cdots, \ S_r)$:
  second order predicates, not in the first order vocabulary.
- The second order predicates capture the solution to the optimization problem. (the unknowns)
  F.O. predicates: Truth values known from the input.

Sentences in ESO universal Horn logic

- $\exists \mathbf{S} \phi \equiv \exists S_1 \cdots \exists S_r \ \phi$, ϕ: first order formula.
- $\mathbf{S} = (S_1, \ \cdots, \ S_r)$:
  second order predicates, not in the first order vocabulary.
- The second order predicates capture the solution to the optimization problem. (the unknowns)
  F.O. predicates: Truth values known from the input.
- ϕ can be expressed in Π_1 form, $\phi \equiv \forall x_1 \cdots \forall x_p \ \eta$,
  $x_1, \cdots, x_p$: first order variables,
  η: quantifier-free, conjunction of Horn clauses.

Sentences in ESO universal Horn logic

- $\exists \mathbf{S} \phi \equiv \exists S_1 \cdots \exists S_r \ \phi$, ϕ: first order formula.
- $\mathbf{S} = (S_1, \ \cdots, \ S_r)$:
  second order predicates, not in the first order vocabulary.
- The second order predicates capture the solution to the optimization problem. (the unknowns)
  F.O. predicates: Truth values known from the input.
- ϕ can be expressed in Π_1 form, $\phi \equiv \forall x_1 \cdots \forall x_p \ \eta$,
  $x_1, \cdots, x_p$: first order variables,
  η: quantifier-free, conjunction of Horn clauses.
- So why can't ESO universal Horn capture the OFC, $f(\mathbf{x}) \geq K$?

Sentences in ESO universal Horn logic

- $\exists \mathbf{S}\phi \equiv \exists S_1 \cdots \exists S_r\ \phi$, ϕ: first order formula.
- $\mathbf{S} = (S_1, \cdots, S_r)$:
  second order predicates, not in the first order vocabulary.
- The second order predicates capture the solution to the optimization problem. (the unknowns)
  F.O. predicates: Truth values known from the input.
- ϕ can be expressed in Π_1 form, $\phi \equiv \forall x_1 \cdots \forall x_p\ \eta$,
  $x_1, \cdots, x_p$: first order variables,
  η: quantifier-free, conjunction of Horn clauses.
- So why can't ESO universal Horn capture the OFC, $f(\mathbf{x}) \geq K$?
- Because universal Horn formulae are preserved under substructures!

Sentences in ESO universal Horn logic

- $\exists \mathbf{S}\phi \equiv \exists S_1 \cdots \exists S_r\ \phi$, ϕ: first order formula.
- $\mathbf{S} = (S_1, \cdots, S_r)$:
  second order predicates, not in the first order vocabulary.
- The second order predicates capture the solution to the optimization problem. (the unknowns)
  F.O. predicates: Truth values known from the input.
- ϕ can be expressed in Π_1 form, $\phi \equiv \forall x_1 \cdots \forall x_p\ \eta$,
  $x_1, \cdots, x_p$: first order variables,
  η: quantifier-free, conjunction of Horn clauses.
- So why can't ESO universal Horn capture the OFC, $f(\mathbf{x}) \geq K$?
- Because universal Horn formulae are preserved under substructures! (even for ordered structures)

Universal Horn logic preserved under substructures

Universal Horn logic preserved under substructures

- Why does universal Horn logic fail to capture $f(\mathbf{x}) \geq K$?

Universal Horn logic preserved under substructures

- Why does universal Horn logic fail to capture $f(\mathbf{x}) \geq K$?
- Reason: If you take a substructure of the original structure which is *small enough*, we can reduce $f(\mathbf{x})$ to some value below K.

Universal Horn logic preserved under substructures

- Why does universal Horn logic fail to capture $f(\mathbf{x}) \geq K$?
- Reason: If you take a substructure of the original structure which is *small enough*, we can reduce $f(\mathbf{x})$ to some value below K.
- Thus ESO universal Horn logic cannot capture the OFC.

Universal Horn logic preserved under substructures

- Why does universal Horn logic fail to capture $f(\mathbf{x}) \geq K$?
- Reason: If you take a substructure of the original structure which is *small enough*, we can reduce $f(\mathbf{x})$ to some value below K.
- Thus ESO universal Horn logic cannot capture the OFC.
- Hence ESO universal Horn cannot capture the decision problem, which is the conjunction of OFC and BFC.

Universal Horn logic preserved under substructures

- Why does universal Horn logic fail to capture $f(\mathbf{x}) \geq K$?
- Reason: If you take a substructure of the original structure which is *small enough*, we can reduce $f(\mathbf{x})$ to some value below K.
- Thus ESO universal Horn logic cannot capture the OFC.
- Hence ESO universal Horn cannot capture the decision problem, which is the conjunction of OFC and BFC.
- This means, such decision problems are members of the set $\mathcal{P}(mle) \setminus \mathcal{P}(sle)$.

What if the OFC can be written in ESO universal Horn?

What if the OFC can be written in ESO universal Horn?

- If the OFC can be written in ESO universal Horn, then P = NP!

What if the OFC can be written in ESO universal Horn?

- If the OFC can be written in ESO universal Horn, then P = NP!
- Why? Bcos the BFC of some NP-complete problems can be written in ESO universal Horn.

What if the OFC can be written in ESO universal Horn?

- If the OFC can be written in ESO universal Horn, then P = NP!
- Why? Bcos the BFC of some NP-complete problems can be written in ESO universal Horn.
  Example: Maximum Clique.

What if the OFC can be written in ESO universal Horn?

- If the OFC can be written in ESO universal Horn, then P = NP!
- Why? Bcos the BFC of some NP-complete problems can be written in ESO universal Horn.
  Example: Maximum Clique.
- Then BFC and OFC will both be ESO universal Horn, and so is their conjunction, BFC $\wedge$ OFC, ...

What if the OFC can be written in ESO universal Horn?

- If the OFC can be written in ESO universal Horn, then P = NP!
- Why? Bcos the BFC of some NP-complete problems can be written in ESO universal Horn.
  Example: Maximum Clique.
- Then BFC and OFC will both be ESO universal Horn, and so is their conjunction, BFC $\wedge$ OFC, ...
- .... which means, the problem can be solved in polynomial time!

What if the OFC can be written in ESO universal Horn?

- If the OFC can be written in ESO universal Horn, then P = NP!
- Why? Bcos the BFC of some NP-complete problems can be written in ESO universal Horn.
  Example: Maximum Clique.
- Then BFC and OFC will both be ESO universal Horn, and so is their conjunction, BFC $\wedge$ OFC, ...
- .... which means, the problem can be solved in polynomial time!
- P $\neq$ NP probably, hence the OFC cannot be written in ESO universal Horn, probably.

What if the OFC can be written in ESO universal Horn?

- If the OFC can be written in ESO universal Horn, then P $=$ NP!
- Why? Bcos the BFC of some NP-complete problems can be written in ESO universal Horn.
  Example: Maximum Clique.
- Then BFC and OFC will both be ESO universal Horn, and so is their conjunction, BFC $\wedge$ OFC, ...
- .... which means, the problem can be solved in polynomial time!
- P $\neq$ NP probably, hence the OFC cannot be written in ESO universal Horn, probably.
  Other direction need not be true: "OFC cannot be written in ESO universal Horn" does not mean that P $\neq$ NP!

Horn Formulae Preserved Under Reduced Products

Horn Formulae Preserved Under Reduced Products

- We show that the OFC, $f(\mathbf{x}) \leq K$, will not work under a reduced product.

Horn Formulae Preserved Under Reduced Products

- We show that the OFC, $f(\mathbf{x}) \leq K$, will not work under a reduced product.
- So this means that "$f(\mathbf{x}) \leq K$" cannot be represented as a Horn formula.

- We show that the OFC, $f(\mathbf{x}) \leq K$, will not work under a reduced product.
- So this means that "$f(\mathbf{x}) \leq K$" cannot be represented as a Horn formula.
- If you take 2 structures A and B (under the same signature Λ), then their "product structure" $A \times B$ is defined as follows:

Horn Formulae Preserved Under Reduced Products

- We show that the OFC, $f(\mathbf{x}) \leq K$, will not work under a reduced product.
- So this means that "$f(\mathbf{x}) \leq K$" cannot be represented as a Horn formula.
- If you take 2 structures A and B (under the same signature Λ), then their "product structure" $A \times B$ is defined as follows:
- (1) If $a \in dom(A)$ and $b \in dom(B)$, then $(a, b) \in dom(A \times B)$.

Horn Formulae Preserved Under Reduced Products

- We show that the OFC, $f(\mathbf{x}) \leq K$, will not work under a reduced product.
- So this means that "$f(\mathbf{x}) \leq K$" cannot be represented as a Horn formula.
- If you take 2 structures A and B (under the same signature Λ), then their "product structure" $A \times B$ is defined as follows:
- (1) If $a \in dom(A)$ and $b \in dom(B)$, then $(a, b) \in dom(A \times B)$.
- (2) For every relation $R_i \in \Lambda$ of arity k, let us define a relation $\mathcal{R}_i^{(A \times B)}$ in the product structure. Then

  $\mathcal{R}_i^{(A \times B)}[(a_1, b_1), \cdots, (a_k, b_k)]$ holds iff

  $R_i^{(A)}(a_1, \cdots, a_k)$ and $R_i^{(B)}(b_1, \cdots, b_k)$ hold.

Proof Using Reduced Products

Proof Using Reduced Products

- Consider 2 structures A and B with the same domain, $\{1, \cdots, 2\theta\}$ ($\theta \in \mathbb{N}$) with a linear ordering.

Proof Using Reduced Products

- Consider 2 structures A and B with the same domain, $\{1, \cdots, 2\theta\}$ ($\theta \in \mathbb{N}$) with a linear ordering.
- Every relation in A and B is defined in the same way, including F.

Proof Using Reduced Products

- Consider 2 structures A and B with the same domain, $\{1, \cdots, 2\theta\}$ ($\theta \in \mathbb{N}$) with a linear ordering.
- Every relation in A and B is defined in the same way, including F.

  For the objective function relation F, let

  $F^{(A)} = F^{(B)} = \{1, \cdots, \theta\}$.

Proof Using Reduced Products

- Consider 2 structures A and B with the same domain, $\{1, \cdots, 2\theta\}$ ($\theta \in \mathbb{N}$) with a linear ordering.
- Every relation in A and B is defined in the same way, including F.

  For the objective function relation F, let

  $F^{(A)} = F^{(B)} = \{1, \cdots, \theta\}$.

  Hence $|F^{(A)}| = |F^{(B)}| = \theta$. Let $K = \theta > 1$.

Proof Using Reduced Products

- Consider 2 structures A and B with the same domain, $\{1, \cdots, 2\theta\}$ ($\theta \in \mathbb{N}$) with a linear ordering.
- Every relation in A and B is defined in the same way, including F.

  For the objective function relation F, let

  $F^{(A)} = F^{(B)} = \{1, \cdots, \theta\}$.

  Hence $|F^{(A)}| = |F^{(B)}| = \theta$. Let $K = \theta > 1$.
- Then in the "product structure" $A \times B$,

  $F^{(A \times B)} = \{(1,1), \cdots, (\theta, \theta)\}$.

Proof Using Reduced Products

- Consider 2 structures A and B with the same domain, $\{1, \cdots, 2\theta\}$ ($\theta \in \mathbb{N}$) with a linear ordering.
- Every relation in A and B is defined in the same way, including F.
  For the objective function relation F, let
  $F^{(A)} = F^{(B)} = \{1, \cdots, \theta\}$.
  Hence $|F^{(A)}| = |F^{(B)}| = \theta$. Let $K = \theta > 1$.
- Then in the "product structure" $A \times B$,
  $F^{(A \times B)} = \{(1,1), \cdots, (\theta, \theta)\}$.
  $|F^{(A \times B)}| = \theta^2 > K$.

Proof Using Reduced Products

- Consider 2 structures A and B with the same domain, $\{1, \cdots, 2\theta\}$ ($\theta \in \mathbb{N}$) with a linear ordering.
- Every relation in A and B is defined in the same way, including F.

  For the objective function relation F, let

  $F^{(A)} = F^{(B)} = \{1, \cdots, \theta\}$.

  Hence $|F^{(A)}| = |F^{(B)}| = \theta$. Let $K = \theta > 1$.
- Then in the "product structure" $A \times B$,

  $F^{(A \times B)} = \{(1,1), \cdots, (\theta, \theta)\}$.

  $|F^{(A \times B)}| = \theta^2 > K$.

  Hence the upper bound on the objective function, that $|F| \leq K$, is not preserved in the product structure.

Proof using approximation theory

Proof using approximation theory

- This proof assumes that $P \neq NP$.

Proof using approximation theory

- This proof assumes that $P \neq NP$.
- **Approximation ratio**. Given an instance I of Q and a feasible solution $S \in F_{\mathrm{Q}}(I)$,

Proof using approximation theory

- This proof assumes that $P \neq NP$.
- **Approximation ratio**. Given an instance I of Q and a feasible solution $S \in F_{\mathrm{Q}}(I)$, the approximation ratio is defined as $\frac{f_{\mathrm{Q}}(I,S)}{f_{\mathrm{Q}}^{(opt)}(I)}$ for minimization problems and $\frac{f_{\mathrm{Q}}^{(opt)}(I)}{f_{\mathrm{Q}}(I,S)}$ for maximization problems.

Proof using approximation theory

- This proof assumes that $P \neq NP$.
- **Approximation ratio**. Given an instance I of Q and a feasible solution $S \in F_{\mathrm{Q}}(I)$, the approximation ratio is defined as $\dfrac{f_{\mathrm{Q}}(I, S)}{f_{\mathrm{Q}}^{(opt)}(I)}$ for minimization problems and $\dfrac{f_{\mathrm{Q}}^{(opt)}(I)}{f_{\mathrm{Q}}(I, S)}$ for maximization problems.
- Assume that $f_{\mathrm{Q}}^{(opt)}(I),\ f_{\mathrm{Q}}(I, S) > 0$, where
  $f_{\mathrm{Q}}^{(opt)}(I) =$ the optimal solution value to instance I.
  $f_{\mathrm{Q}}(I, S) =$ value of solution S.

Approximation problems $\mathbf{A}_{\mathrm{Q}}^{\mathbf{T}}$ and $\mathbf{A}_{\mathrm{Q}}^{\mathbf{B}}$

Approximation problems $\mathbf{A}^{\mathbf{T}}_{\mathrm{Q}}$ and $\mathbf{A}^{\mathbf{B}}_{\mathrm{Q}}$

- Problem $\mathbf{A}^{\mathbf{T}}_{\mathrm{Q}}$.
  Given an instance I of problem Q and a parameter K_{Q} that depends on Q,

Approximation problems $\mathbf{A}_{\mathrm{Q}}^{\mathbf{T}}$ and $\mathbf{A}_{\mathrm{Q}}^{\mathbf{B}}$

- Problem $\mathbf{A}_{\mathrm{Q}}^{\mathbf{T}}$.
  Given an instance I of problem Q and a parameter K_{Q} that depends on Q,
  is there a feasible solution $S \in F_{\mathrm{Q}}(I)$, such that the approximation ratio is at most K_{Q}?

Approximation problems $\mathbf{A}_{\mathrm{Q}}^{\mathbf{T}}$ and $\mathbf{A}_{\mathrm{Q}}^{\mathbf{B}}$

- Problem $\mathbf{A}_{\mathrm{Q}}^{\mathbf{T}}$.
  Given an instance I of problem Q and a parameter K_{Q} that depends on Q,
  is there a feasible solution $S \in F_{\mathrm{Q}}(I)$, such that the approximation ratio is at most K_{Q}?
- Problem $\mathbf{A}_{\mathrm{Q}}^{\mathbf{B}}$.
  Given Q, I, and a parameter L_{Q} that depends on Q,
  is there a feasible solution $S \in F_{\mathrm{Q}}(I)$, such that the approximation ratio is at most L_{Q}?
  ($L_{\mathrm{Q}} < K_{\mathrm{Q}}$.)

Approximation problems $\mathbf{A}^{\mathbf{T}}_{\mathrm{Q}}$ and $\mathbf{A}^{\mathbf{B}}_{\mathrm{Q}}$

- Problem $\mathbf{A}^{\mathbf{T}}_{\mathrm{Q}}$.
  Given an instance I of problem Q and a parameter K_{Q} that depends on Q,
  is there a feasible solution $S \in F_{\mathrm{Q}}(I)$, such that the approximation ratio is at most K_{Q}?
- Problem $\mathbf{A}^{\mathbf{B}}_{\mathrm{Q}}$.
  Given Q, I, and a parameter L_{Q} that depends on Q,
  is there a feasible solution $S \in F_{\mathrm{Q}}(I)$, such that the approximation ratio is at most L_{Q}?
  $(L_{\mathrm{Q}} < K_{\mathrm{Q}}.)$
- K_{Q}: Upper bound. L_{Q}: Lower bound.

Approximation problems $\mathbf{A}_{\mathrm{Q}}^{\mathbf{T}}$ and $\mathbf{A}_{\mathrm{Q}}^{\mathbf{B}}$

- Problem $\mathbf{A}_{\mathrm{Q}}^{\mathbf{T}}$.
  Given an instance I of problem Q and a parameter K_{Q} that depends on Q,
  is there a feasible solution $S \in F_{\mathrm{Q}}(I)$, such that the approximation ratio is at most K_{Q}?
- Problem $\mathbf{A}_{\mathrm{Q}}^{\mathbf{B}}$.
  Given Q, I, and a parameter L_{Q} that depends on Q,
  is there a feasible solution $S \in F_{\mathrm{Q}}(I)$, such that the approximation ratio is at most L_{Q}?
  $(L_{\mathrm{Q}} < K_{\mathrm{Q}}.)$
- K_{Q}: Upper bound. L_{Q}: Lower bound.
- Problem $\mathbf{A}_{\mathrm{Q}}^{\mathbf{T}}$: Polynomially solvable.

  Problem $\mathbf{A}_{\mathrm{Q}}^{\mathbf{B}}$: NP-complete.

Vertex Cover (optimization version)

Vertex Cover (optimization version)

- Given: A graph $G = (V, E)$.

  To do: Mark vertices in V in such a way that

  (i) BFC: For any $x, y \in V$ where $x < y$, if $(x, y) \in E$, then either x of y is marked, and

  (ii) The objective function: The number of marked vertices is to be minimized.

Vertex Cover (optimization version)

- Given: A graph $G = (V, E)$.

  To do: Mark vertices in V in such a way that

  (i) BFC: For any $x, y \in V$ where $x < y$, if $(x, y) \in E$, then either x of y is marked, and

  (ii) The objective function: The number of marked vertices is to be minimized.

- For Vertex Cover, the upper bound K_{Q} $(K_{vc}) = 2$ and the lower bound L_{Q} $(L_{vc}) = 7/6$.

Vertex Cover (optimization version)

- Given: A graph $G = (V, E)$.

  To do: Mark vertices in V in such a way that

  (i) BFC: For any $x, y \in V$ where $x < y$, if $(x, y) \in E$, then either x of y is marked, and

  (ii) The objective function: The number of marked vertices is to be minimized.
- For Vertex Cover, the upper bound K_{Q} $(K_{vc}) = 2$ and the lower bound L_{Q} $(L_{vc}) = 7/6$.
- The upper bound problem, $\mathbf{A}^{\mathbf{T}}_{\mathbf{vc}}$, is polynomially solvable.

Vertex Cover (optimization version)

- Given: A graph $G = (V, E)$.

  To do: Mark vertices in V in such a way that

  (i) BFC: For any $x, y \in V$ where $x < y$, if $(x, y) \in E$, then either x of y is marked, and

  (ii) The objective function: The number of marked vertices is to be minimized.

- For Vertex Cover, the upper bound K_{Q} $(K_{vc}) = 2$ and the lower bound L_{Q} $(L_{vc}) = 7/6$.
- The upper bound problem, $\mathbf{A}^{\mathbf{T}}_{\mathbf{vc}}$, is polynomially solvable. $\Rightarrow$ can be expressed in ESO universal Horn (Sentence α_1).

Vertex Cover (optimization version)

Vertex Cover (optimization version)

- In α_1, replace "2" with "7/6" (the lower bound).
  (Use 1.16 if you like!)

Vertex Cover (optimization version)

- In α_1, replace "2" with "7/6" (the lower bound).
(Use 1.16 if you like!)
- We get the sentence α_2, which is also in ESO universal Horn form, and this represents the lower bound problem $\mathbf{A}_{\mathbf{vc}}^{\mathbf{B}}$.

Vertex Cover (optimization version)

- In α_1, replace "2" with "7/6" (the lower bound).
  (Use 1.16 if you like!)
- We get the sentence α_2, which is also in ESO universal Horn form, and this represents the lower bound problem $\mathbf{A}_{\mathbf{vc}}^{\mathbf{B}}$.
- $\Rightarrow$ $\mathbf{A}_{\mathbf{vc}}^{\mathbf{B}}$ is polynomially solvable.

Vertex Cover (optimization version)

- In α_1, replace "2" with "7/6" (the lower bound).
  (Use 1.16 if you like!)
- We get the sentence α_2, which is also in ESO universal Horn form, and this represents the lower bound problem $\mathbf{A}^{\mathbf{B}}_{\mathbf{vc}}$.
- $\Rightarrow$ $\mathbf{A}^{\mathbf{B}}_{\mathbf{vc}}$ is polynomially solvable.
- But we know that $\mathbf{A}^{\mathbf{B}}_{\mathbf{vc}}$ is NP-complete.
- This contradicts our assumption that $P \neq NP$.

Vertex Cover (optimization version)

- In α_1, replace "2" with "7/6" (the lower bound). (Use 1.16 if you like!)
- We get the sentence α_2, which is also in ESO universal Horn form, and this represents the lower bound problem $\mathbf{A}^{\mathbf{B}}_{\mathbf{vc}}$.
- $\Rightarrow$ $\mathbf{A}^{\mathbf{B}}_{\mathbf{vc}}$ is polynomially solvable.
- But we know that $\mathbf{A}^{\mathbf{B}}_{\mathbf{vc}}$ is NP-complete.
- This contradicts our assumption that $P \neq NP$.
- Hence our assumption — $\mathbf{A}^{\mathbf{T}}_{\mathbf{vc}}$ (which is known to be polynomially solvable) can be represented in ESO universal Horn logic — must be wrong.

Vertex Cover (optimization version)

- In α_1, replace "2" with "7/6" (the lower bound).
  (Use 1.16 if you like!)
- We get the sentence α_2, which is also in ESO universal Horn form, and this represents the lower bound problem $\mathbf{A}^{\mathbf{B}}_{\mathbf{vc}}$.
- $\Rightarrow$ $\mathbf{A}^{\mathbf{B}}_{\mathbf{vc}}$ is polynomially solvable.
- But we know that $\mathbf{A}^{\mathbf{B}}_{\mathbf{vc}}$ is NP-complete.
- This contradicts our assumption that $P \neq NP$.
- Hence our assumption — $\mathbf{A}^{\mathbf{T}}_{\mathbf{vc}}$ (which is known to be polynomially solvable) can be represented in ESO universal Horn logic — must be wrong.
- NOT every problem in P can be expressed in ESO universal Horn form, at the structure level,

Vertex Cover (optimization version)

- In α_1, replace "2" with "7/6" (the lower bound). (Use 1.16 if you like!)
- We get the sentence α_2, which is also in ESO universal Horn form, and this represents the lower bound problem $\mathbf{A}^{\mathbf{B}}_{\mathbf{vc}}$.
- $\Rightarrow$ $\mathbf{A}^{\mathbf{B}}_{\mathbf{vc}}$ is polynomially solvable.
- But we know that $\mathbf{A}^{\mathbf{B}}_{\mathbf{vc}}$ is NP-complete.
- This contradicts our assumption that $P \neq NP$.
- Hence our assumption — $\mathbf{A}^{\mathbf{T}}_{\mathbf{vc}}$ (which is known to be polynomially solvable) can be represented in ESO universal Horn logic — must be wrong.
- NOT every problem in P can be expressed in ESO universal Horn form, at the structure level, so we obtain a separation between $\mathcal{P}(sle)$ and $\mathcal{P}(mle)$.

Immerman-Vardi Theorem

Immerman-Vardi Theorem

- *I-V Theorem*. Least fixed point logic (LFP) characterizes P for the class of ordered finite structures, is true at the machine level. Structure level?

Immerman-Vardi Theorem

- *I-V Theorem*. Least fixed point logic (LFP) characterizes P for the class of ordered finite structures, is true at the machine level. Structure level?
- (Atserias, Bulatov, Dawar, 2009) FEASIBLE cannot be expressed in $\mathbf{C}^{\omega}_{\infty\omega}$.

Immerman-Vardi Theorem

- *I-V Theorem*. Least fixed point logic (LFP) characterizes P for the class of ordered finite structures, is true at the machine level. Structure level?
- (Atserias, Bulatov, Dawar, 2009) FEASIBLE cannot be expressed in $\mathbf{C}^{\omega}_{\infty\omega}$.
  This means, it cannot be expressed in LFP+C.

Immerman-Vardi Theorem

- *I-V Theorem*. Least fixed point logic (LFP) characterizes P for the class of ordered finite structures, is true at the machine level. Structure level?
- (Atserias, Bulatov, Dawar, 2009) FEASIBLE cannot be expressed in $\mathbf{C}^{\omega}_{\infty\omega}$.
  This means, it cannot be expressed in LFP+C.
  Hence it cannot be expressed in LFP.

Immerman-Vardi Theorem

- *I-V Theorem*. Least fixed point logic (LFP) characterizes P for the class of ordered finite structures, is true at the machine level. Structure level?
- (Atserias, Bulatov, Dawar, 2009) FEASIBLE cannot be expressed in $\mathbf{C}^{\omega}_{\infty\omega}$.
  This means, it cannot be expressed in LFP+C.
  Hence it cannot be expressed in LFP.
- But FEASIBLE can be decided in PTIME.

Immerman-Vardi Theorem

- *I-V Theorem*. Least fixed point logic (LFP) characterizes P for the class of ordered finite structures, is true at the machine level. Structure level?
- (Atserias, Bulatov, Dawar, 2009) FEASIBLE cannot be expressed in $\mathbf{C}^{\omega}_{\infty\omega}$.
  This means, it cannot be expressed in LFP+C.
  Hence it cannot be expressed in LFP.
- But FEASIBLE can be decided in PTIME.
  Hence LFP+C and LFP are insufficient to capture P at the structure level.

Immerman-Vardi Theorem

- *I-V Theorem*. Least fixed point logic (LFP) characterizes P for the class of ordered finite structures, is true at the machine level. Structure level?
- (Atserias, Bulatov, Dawar, 2009) FEASIBLE cannot be expressed in $\mathbf{C}^{\omega}_{\infty\omega}$.
  This means, it cannot be expressed in LFP+C.
  Hence it cannot be expressed in LFP.
- But FEASIBLE can be decided in PTIME.
  Hence LFP+C and LFP are insufficient to capture P at the structure level.
- I-V Theorem does not hold at the structure level.

Implications for Fagin's Theorem

Implications for Fagin's Theorem

- Under ordered structures, let $\mathcal{P}_3$ ($\mathcal{P}_4$) be the class of NP decidable properties that can be expressed in ESO universal logic at the machine (structure) level respectively.

Implications for Fagin's Theorem

- Under ordered structures, let $\mathcal{P}_3$ ($\mathcal{P}_4$) be the class of NP decidable properties that can be expressed in ESO universal logic at the machine (structure) level respectively.
  (FO part need not be Horn.)

Implications for Fagin's Theorem

- Under ordered structures, let $\mathcal{P}_3$ ($\mathcal{P}_4$) be the class of NP decidable properties that can be expressed in ESO universal logic at the machine (structure) level respectively.
  (FO part need not be Horn.)
- $\mathcal{P}_4 \subseteq \mathcal{P}_3 = \text{NP}$. (Fagin's theorem)

Implications for Fagin's Theorem

- Under ordered structures, let $\mathcal{P}_3$ ($\mathcal{P}_4$) be the class of NP decidable properties that can be expressed in ESO universal logic at the machine (structure) level respectively. (FO part need not be Horn.)
- $\mathcal{P}_4 \subseteq \mathcal{P}_3 = \text{NP}$. (Fagin's theorem)
- However, no problem in NP derived from a maximization problem can be expressed with a universal first order part, since the OFC cannot be so expressed.

Implications for Fagin's Theorem

- Under ordered structures, let $\mathcal{P}_3$ ($\mathcal{P}_4$) be the class of NP decidable properties that can be expressed in ESO universal logic at the machine (structure) level respectively.
  (FO part need not be Horn.)
- $\mathcal{P}_4 \subseteq \mathcal{P}_3 = \text{NP}$. (Fagin's theorem)
- However, no problem in NP derived from a maximization problem can be expressed with a universal first order part, since the OFC cannot be so expressed.
- This means that $\mathcal{P}_4 \subset \mathcal{P}_3 = \text{NP}$.

Implications for Fagin's Theorem

- Under ordered structures, let $\mathcal{P}_3$ ($\mathcal{P}_4$) be the class of NP decidable properties that can be expressed in ESO universal logic at the machine (structure) level respectively.
(FO part need not be Horn.)
- $\mathcal{P}_4 \subseteq \mathcal{P}_3 = \text{NP}$. (Fagin's theorem)
- However, no problem in NP derived from a maximization problem can be expressed with a universal first order part, since the OFC cannot be so expressed.
- This means that $\mathcal{P}_4 \subset \mathcal{P}_3 = \text{NP}$.
Under ordered structures, Fagin's theorem does not hold?

Implications for Fagin's Theorem

- Under ordered structures, let $\mathcal{P}_3$ ($\mathcal{P}_4$) be the class of NP decidable properties that can be expressed in ESO universal logic at the machine (structure) level respectively.
  (FO part need not be Horn.)
- $\mathcal{P}_4 \subseteq \mathcal{P}_3 = \text{NP}$. (Fagin's theorem)
- However, no problem in NP derived from a maximization problem can be expressed with a universal first order part, since the OFC cannot be so expressed.
- This means that $\mathcal{P}_4 \subset \mathcal{P}_3 = \text{NP}$.
  Under ordered structures, Fagin's theorem does not hold?
- What if the structures are not ordered?