Introduction

- Define BAPA

- Explain the *decision procedure* and *quantifier elimination* for the BAPA logic

- NP completeness of the QFBAPA
Define BAPA

- **BA** – a variant of ordinary algebra
  - instead of the usual algebra of numbers, BA is the algebra of truth values (0 and 1) or equivalently of subsets of a given set

- **PA** – first order theory of natural numbers
  - contains only addition and equality operations
  - is a decidable theory
Define BAPA

- **BAPA** combines:
  1. boolean algebras of sets of uninterpreted integers
  2. Presburger Arithmetic operations

- It can express the relationship between int vars and cardinalities of unbounded finite sets and verification of data structure consistency properties.
- Was found to be of interest in program verification and constraint DB (ex. Revesz 2004)
Define BAPA

- **BAPA formulas with quantifier alternations** arise when
  - verifying programs with annotations containing quantifiers
  - Proving simulation relation conditions for refinement and equivalence of program fragments

- **BAPA constraints** can be used for proving the termination of programs that manipulate data structures
1. BAPA constraints can be used for program analysis and verification by expressing:
   - DS invariants and the correctness of procedures with respect to their specification
   - Simulation relation between program fragments
   - Termination conditions for programs that manipulate DS

2. Presenting an alg. $\alpha$ that translates BAPA sentences into PA sentences

3. Analyze the alg. $\alpha$ and show that its complexity matches the lower bound for PA and is therefore **optimal**
First–Order theory of BAPA

\[ F ::= A \mid F_1 \land F_2 \mid F_1 \lor F_2 \mid \neg F \mid \exists x.F \mid \forall x.F \]

\[ A ::= B_1 = B_2 \mid B_1 \subseteq B_2 \mid |B| = K \mid |B| \geq K \]

\[ B ::= x \mid 0 \mid 1 \mid B_1 \cup B_2 \mid B_1 \cap B_2 \mid B^c \]

\[ K ::= 0 \mid 1 \mid 2 \mid \ldots \]

*Figure 1. Formulas of Boolean Algebra (BA)*

\[ F ::= A \mid F_1 \land F_2 \mid F_1 \lor F_2 \mid \neg F \mid \exists k.F \mid \forall k.F \]

\[ A ::= T_1 = T_2 \mid T_1 < T_2 \mid K \text{ dvd } T \]

\[ T ::= K \mid T_1 + T_2 \mid K \cdot T \]

\[ K ::= \ldots -2 \mid -1 \mid 0 \mid 1 \mid 2 \ldots \]

*Figure 2. Formulas of Presburger Arithmetic (PA)*

- Our initial motivation for BAPA was the use of BA to reason about data structures in terms of sets.
First–Order theory of BAPA

- BA language allows cardinality constraints of the form $|A| = K$, where $K$ constant integer

- Such constant cardinality constraints are useful and enable quantifier elimination for the resulting language

- But not constraints such as $|A| = |B|

- Also cannot represent constraints on relationships between sizes of sets and integer variables
First–Order theory of BAPA

- If we consider the equicardinality $A \sim B$ that holds iff $|A| = |B|$, and consider BA extended with the relation $A \sim B$

- Define the relation:

  $\text{plus}(A, B, C) \iff (|A|+|B|=|C|)$

  by the formula:

  $$\exists x_1. \exists x_2. x_1 \cap x_2 = \emptyset \land A \sim x_1 \land B \sim x_2 \land x_1 \cup x_2 = C$$
First–Order theory of BAPA

- The relation $\text{plus}(A, B, C)$ allows us to:
  - Express addition using arbitrary sets as representatives for natural numbers
  - $\emptyset$ can represent the natural number 0 and
  - Any singleton set can represent the natural number 1

Therefore a natural closure under definable operations leads to our formulation of the language BAPA which contains both sets and integers.
First-Order theory of BAPA

\[ F ::= A \mid F_1 \land F_2 \mid F_1 \lor F_2 \mid \neg F \mid \exists x.F \mid \forall x.F \mid \exists k.F \mid \forall k.F \]

\[ A ::= B_1 = B_2 \mid B_1 \subseteq B_2 \mid T_1 = T_2 \mid T_1 < T_2 \mid K \text{ dvd } T \]

\[ B ::= x \mid 0 \mid 1 \mid B_1 \cup B_2 \mid B_1 \cap B_2 \mid B^c \]

\[ T ::= k \mid K \mid \text{MAXC} \mid T_1 + T_2 \mid K \cdot T \mid |B| \]

\[ K ::= \ldots -2 \mid -1 \mid 0 \mid 1 \mid 2 \ldots \]

Figure 3. Formulas of Boolean Algebra with Presburger Arithmetic (BAPA)

In the BAPA language we distinguish two kinds of quantifiers; over sets and over integers
The algorithm $\alpha$ which

- Decides the validity of BAPA sentences

- Reduces a BAPA sentence to an equivalent PA sentence with the same number of quantifier alternations and an exponential increase in the total size of the formula
Algorithm procedures

This algorithm describes the steps to transform a BAPA sentence $F_0$ into a PA sentence

First step: Put $F_0$ Into Prenex form

Example:

$$Q_p v_p \ldots Q_1 v_1. F(v_1, \ldots, v_p)$$

$$\exists A. \exists B. |A| = |B| \land A \cup B = C \land |A \cap B| = \emptyset$$
The next step: separate BA from PA by replacing each $b_1 = b_2$ formula with $b_1 \subseteq b_2 \land b_2 \subseteq b_1$ and every $b_1 \subseteq b_2$ with $|b_1 \cap b_2^c| = 0$

$$\exists A. \exists B. |A| = |B| \land |(A \cup B) \cap C^c| = 0 \land |C \cap (A \cup B)^c| = 0 \land |A \cap B| = 0$$
Algorithm procedures

Next step: write each set expression as a union of cubes (regions in the Venn Diagram) - \( \sum_{i=1}^{a} |s_{j_i}| \)

and for each \( S_i \) introduce a new variable \( l_i \)

The resulting formula will be of the form:

\[
Q_p v_p \ldots Q_1 v_1 . \exists^+ l_1, \ldots, l_m . \bigwedge_{i=1}^{m} |s_i| = l_i \land G_1
\]
Algorithm procedures

In our example:

\[ |A| = |S_1| + |S_2| + |S_4| + |S_5| \]
\[ |B| = |S_2| + |S_3| + |S_5| + |S_6| \]
\[ |(A \cup B) \cap C^c| = |S_1| + |S_2| + |S_3| \]
\[ |C \cap (A \cup B)^c| = |S_7| + |S_8| \]
\[ |A \cap B| = |S_2| + |S_5| \]
Algorithm procedures

\[ |A| = |B| \quad \Rightarrow l_1 + \frac{l_2}{4} + \frac{l_5}{5} = \frac{l_2}{2} + \frac{l_3}{3} + \frac{l_5}{5} + l_6 \]
\[ \Rightarrow l_1 + l_4 = l_3 + l_6 \]

\[ |(A \cup B) \cap C^c| = l_1 + l_2 + l_3 = 0 \]
\[ \Rightarrow l_1 = l_2 = l_3 = 0 \]

\[ |C \cap (A \cup B)^c| = l_7 + l_8 = 0 \]
\[ \Rightarrow l_7 = l_8 = 0 \]

\[ |A \cap B| = l_2 + l_5 = 0 \]
\[ \Rightarrow l_2 = l_5 = 0 \]
Algorithm procedures

\[ l_1 = |A \cap B^c \cap C^c| \]
\[ l_2 = |A \cap B \cap C^c| \]
\[ l_3 = |A^c \cap B \cap C^c| \]
\[ l_4 = |A \cap B^c \cap C| \]
\[ l_5 = |A \cap B \cap C| \]
\[ l_6 = |A^c \cap B \cap C| \]
\[ l_7 = |A^c \cap B^c \cap C| \]
\[ l_8 = |A^c \cap B^c \cap C^c| \]
∃A. ∃ B. ∃ l₁. ∃ l₂. ∃ l₃. ∃ l₄. ∃ l₅. ∃ l₆. ∃ l₇. ∃ l₈

l₁ = |A ∩ Bᶜ ∩ Cᶜ|
∧ l₂ = |A ∩ B ∩ Cᶜ|
∧ l₃ = |Aᶜ ∩ B ∩ Cᶜ|
∧ l₄ = |A ∩ Bᶜ ∩ C|
∧ l₅ = |A ∩ B ∩ C|
∧ l₆ = |Aᶜ ∩ B ∩ C|
∧ l₇ = |Aᶜ ∩ Bᶜ ∩ C|
∧ l₈ = |Aᶜ ∩ Bᶜ ∩ Cᶜ|
∧ l₁ + l₄ = l₃ + l₆ ∧ l₁ = l₂ = l₃ = 0 ∧ l₇ = l₈ = 0 ∧ l₂ = l₅ = 0
Since: $l_1 = l_2 = l_3 = 0$ ; $l_7 = l_8 = 0$ and $l_2 = l_5 = 0$

We simplify our formula to:

$\exists A. \exists B. \exists l_4. \exists l_6. \quad l_1 = |A \cap B^c \cap C^c|$

$\land \quad l_2 = |A \cap B \cap C^c|$

$\land \quad l_3 = |A^c \cap B \cap C^c|$

$\land \quad l_4 = |A \cap B^c \cap C|$

$\land \quad l_5 = |A \cap B \cap C|$

$\land \quad l_6 = |A^c \cap B \cap C|$

$\land \quad l_7 = |A^c \cap B^c \cap C|$

$\land \quad l_8 = |A^c \cap B^c \cap C^c| \quad \land \quad l_4 = l_6$
Algorithm procedures
Removing quantifiers

- Consider first the **case** $\exists k$. Because $k$ does not occur in $\bigwedge_{i=1}^{q} |s_i| = l_i$, simply move the existential quantifier to $G_r$ and let $G_{r+1} = \exists k. G_r$, which completes the step.

- **Case** $\forall k$, it suffices to let $G_{r+1} = \forall k. G_r$, again because $k$ does not occur in $\bigwedge_{i=1}^{q} |s_i| = l_i$. 
Algorithm procedures
Removing set quantifiers

- To remove \( \exists y \):
- **LEMMA 1:** Let \( b_1, \ldots, b_n \) be finite disjoint sets, and \( l_1, \ldots, l_n, k_1, \ldots, k_n \) be natural numbers. Then the following two statements are equivalent:

  1. There exists a finite set \( y \) such that
     \[
     \bigwedge_{i=1}^{n} |b_i \cap y| = k_i \land |b_i \cap y^c| = l_i
     \]  
     \[ (1) \]

  2. \[
     \bigwedge_{i=1}^{n} |b_i| = k_i + l_i
     \]  
     \[ (2) \]
Algorithm procedures
Removing set quantifiers

To remove ∃ B we have to consider following cases:

\[ A \cap C \quad A \cap C^c \quad A^c \cap C \quad A^c \cap C^c \]

Regarding the statements (1) and (2) of the previous Lemma:

\[ |A \cap C| = |A \cap B^c \cap C| \land |A \cap B \cap C| = l_4 + l_5 \]
\[ |A \cap C^c| = |A \cap B^c \cap C^c| \land |A \cap B \cap C^c| = l_1 + l_2 \]
\[ |A^c \cap C| = |A^c \cap B \cap C| \land |A^c \cap B^c \cap C| = l_6 + l_7 \]
\[ |A^c \cap C^c| = |A^c \cap B \cap C^c| \land |A^c \cap B^c \cap C^c| = l_3 + l_8 \]
After removing $\exists B$ our example takes this form:

$$\exists A. \exists l_4 . \exists l_6 . \quad |A \cap C| = l_4$$

$$\quad \land |A \cap C^c| = 0$$

$$\quad \land |A^c \cap C| = l_6$$

$$\quad \land |A^c \cap C^c| = 0$$

$$\quad \land l_4 = l_6$$
Algorithm procedures
Removing set quantifiers

To remove $\exists A$ we consider the following two cases: $C$ and $C^c$

Again regarding to Lemma1:

$|C| = |A \cap C| \land |A^c \cap C| = l_4 + l_6$

$|C^c| = |A \cap C^c| \land |A^c \cap C^c| = 0$

We achieve the final step providing the PA formula:

$\exists l_4 \cdot \exists l_6 \cdot |C| = l_4 + l_6$

$|C^c| = 0$

$\land l_4 = l_6$
Algorithm procedures
Removing set quantifiers

• In the cases when we have only one set variable \( y \) and its negation \( y^c \) we can write \(|y|\) and \(|y^c|\) as \(|1 \cap y|\) and \(|1 \cap y^c|\) and apply the algorithm one more time.

• Finally we obtain a formula of the form:

\[ \exists^+ l. \; |1| = l \land G_{p+1}(l) \]

which is the last step and the algorithm terminates.
Algorithm procedures
Removing set quantifiers

- Case $\forall y$:

$$\neg(\exists^+ l_1 \ldots l_q. \bigwedge_{i=1}^{q} |s_i| = l_i \land G_r)$$

is equivalent to

$$\exists^+ l_1 \ldots l_q. \bigwedge_{i=1}^{q} |s_i| = l_i \land \neg G_r,$$

because existential quantifiers over $l_i$ together with the conjuncts $|S_i| = l_i$ act as definitions for $l_i$

- So by expressing $\forall y$ as $\neg \exists y \neg$ we show that elimination of $\forall y$ is analogous to elimination of $\exists k$
NP Completeness

\[ F ::= A \mid F_1 \land F_2 \mid F_1 \lor F_2 \mid \neg F \]

\[ A ::= B_1 = B_2 \mid B_1 \subseteq B_2 \mid T_1 = T_2 \mid T_1 < T_2 \mid K \text{ dvd } T \]

\[ B ::= x \mid 0 \mid 1 \mid B_1 \cup B_2 \mid B_1 \cap B_2 \mid B^c \]

\[ T ::= k \mid K \mid \text{MAXC} \mid T_1 + T_2 \mid K \cdot T \mid |B| \]

\[ K ::= \ldots, -2 \mid -1 \mid 0 \mid 1 \mid 2 \ldots \]

Quantifier-Free Boolean Algebra with Presburger Arithmetic (QFBAPA)

- 1) arbitrary BA expressions denoting sets
- 2) arbitrary quantifier-free PA expressions
- 3) cardinality operator
- The constant MAXC denotes the size of the universal set, so \(|1| = \text{MAXC}|.}
NP Completeness

• To be NP – complete:
  - NP – hard (QFBAPA is since it has propositional operators on formulas)
  - NP membership (the challenge)

• QFPA formula is in NP because it has a small model property implying that satisfiable formulas have solutions whose binary representation is polynomial
We show that:

- It is not necessary to generate all Venn region variables (drop them which are empty)
- Instead of using *exponentially* many Venn region cardinality vars to encode sets, we can use *polinomially* many generic variables
- New QFPA formula is equisatisfiable with the original one
NP Completeness

Starting with an example

From a QFBAPA formula construct a polynomially large QFPA formula which is equisatisfiable with the original one

Again separate BA from PA by replacements:

\[ b_1 = b_2 \implies b_1 \subseteq b_2 \land b_2 \subseteq b_1 \]

and each \( |b_1 \cap b_2^c| = 0 \)

ex: \( |A| = |B| \land |A \cap B| = \emptyset \)
$G \land F$ where:

- $G$ – a quantifier–free PA formula
- $F$ – of the form

\[ \bigwedge_{i=0}^{p} |b_i| = k_i \]

\[ |A| = |S_1| + |S_2| \]

\[ |B| = |S_2| + |S_3| \]

\[ |A \cap B| = |S_2| \]
NP Completeness

\[ k_1 = k_2 \quad \land \quad |A| = k_1 \]
\[ k_3 = 0 \quad \land \quad |B| = k_2 \]
\[ |A \cap B| = k_3 \]

• for \( \beta = (p_1, \ldots, p_e) \)
  where \( p_i \in \{0, 1\} \) let \([b_i]_\beta \in \{0, 1\}\)
  express each \( |b_i| = \sum_{\beta \models b_i} |S_\beta| \)
  and for each \( \beta \in \{0, 1\}^e \) introduce a non-negative integer variable \( l_\beta \) denoting \( |S_\beta| \)
NP Completeness

\[ |b_1| = l_{(1, 0)} + l_{(1, 1)} \]
\[ |b_2| = l_{(0, 1)} + l_{(1, 1)} \]
\[ |b_3| = l_{(1, 1)} \]

Finally:

\[ k_1 = k_2 \quad \land \quad |b_1| = l_{(1, 0)} + l_{(1, 1)} \]
\[ k_3 = 0 \quad \land \quad |b_2| = l_{(0, 1)} + l_{(1, 1)} \]
\[ |b_3| = l_{(1, 1)} \]
NP Completeness

Our formula is of the form:

\[ \bigwedge_{i=0}^{p} \sum \{ l_\beta | \beta \in \{0, 1\}^e \land [b_i]_\beta = 1 \} = k_i \]

But instead of using this exponentially large formula where \( \beta \) ranges over all \( 2^e \) propositional assignments, we will check the satisfiability of a smaller formula where \( \beta \) ranges over \( N \) assignments:

\[ G \land \bigwedge_{i=0}^{p} \sum \{ l_\beta | \beta \in \{\beta_1, \ldots, \beta_N\} \land [b_i]_\beta = 1 \} = k_i \]
• $\beta$ ranges over a set of $N$ assignments $\beta_1, \ldots, \beta_N$ for $\beta_i = (p_{i1}, \ldots, p_{ie})$ and

• $p_{ij}$ are fresh free variables ranging over $\{0, 1\}$.

We are interested in the best upper bound $N(d)$ on the number of non-zero Venn regions over all possible systems of equations
Fact 1 (Eisenbrand, Shmonina (2005)) Let $X \subseteq \mathbb{Z}^d$ be a finite set of integer vectors and $M = \max\{(\max_{i=1}^d |x_j^i|) \mid (x_1^1, \ldots, x_d^d) \in X\}$ be the bound on the coordinates of vectors in $X$. If $b \in \text{int}\_\text{cone}(X)$, then there exists a subset $\tilde{X} \subseteq X$ such that $b \in \text{int}\_\text{cone}(\tilde{X})$ and $|\tilde{X}| \leq 2d \log(4dM)$. 
NP Completeness

\[ |b_1| = \beta_{(1, 0)} + \beta_{(1, 1)} \]
\[ |b_2| = \beta_{(0, 1)} + \beta_{(1, 1)} \]
\[ |b_3| = \beta_{(1, 1)} \]

\[ (k_1, k_2, k_3) = (\beta_{(1, 0)}, 0, 0) + (0, \beta_{(0, 1)}, 0) + (\beta_{(1, 1)}, \beta_{(1, 1)}, \beta_{(1, 1)}) \]
We will use a set of integer vectors \( con(X) \) where \( X = \{x_\beta \mid \beta \in \{0, 1\}^e \} \) and \( x_\beta \in \{0, 1\}^e \) is given by \( x_\beta = ([b_0]_\beta, [b_1]_\beta, \ldots, [b_e]_\beta) \)

A direct application of Fact 1 yields \( N = 2d \log (4d) \)
Which is sufficient to show that QFBAPA is in NP
Thank you