The Fine-Grained And Parallel Complexity of Andersen’s Pointer Analysis

Anders Alnor Mathiasen and Andreas Pavlogiannis
Pointers make static analysis hard

```
...  
*a=42;
*b=84;
c=*a;
// is c 42 or 84?
```
Pointers make static analysis hard

```plaintext
...  
* a = 42;  
* b = 84;  
c = * a;  
// is c 42 or 84?
```

Pointer analysis is typically a prerequisite to other static analyses

- Must be fast
- Determines the quality of the static analysis
Andersen’s Pointer Analysis (APA)

- Flow insensitive
- No nested dereferences
- Inclusion based
Andersen’s Pointer Analysis (APA)
Andersen’s Pointer Analysis (APA)

\[ c = &a \]
Andersen’s Pointer Analysis (APA)

\[
\begin{align*}
&c = &a \\
&e = d
\end{align*}
\]
Andersen’s Pointer Analysis (APA)

\[
\begin{align*}
c &= \&a \\
e &= d \\
b &= \ast e
\end{align*}
\]
Andersen’s Pointer Analysis (APA)

Memory

$c = &a$
$e = d$
$b = *e$
$*d = a$
### Andersen’s Pointer Analysis (APA)

Solve for inclusion constraints

<table>
<thead>
<tr>
<th>Type</th>
<th>Statement</th>
<th>Inclusion Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a = b$</td>
<td>$\text{PointsToSet}(b) \subseteq \text{PointsToSet}(a)$</td>
</tr>
<tr>
<td>2</td>
<td>$a = &amp; b$</td>
<td>$b \in \text{PointsToSet}(a)$</td>
</tr>
<tr>
<td>3</td>
<td>$a = \star b$</td>
<td>$\forall c \in \text{PointsToSet}(b): \text{PointsToSet}(c) \subseteq \text{PointsToSet}(a)$</td>
</tr>
<tr>
<td>4</td>
<td>$\star a = b$</td>
<td>$\forall c \in \text{PointsToSet}(a): \text{PointsToSet}(b) \subseteq \text{PointsToSet}(c)$</td>
</tr>
</tbody>
</table>
Andersen’s Pointer Analysis (APA)

Solve for inclusion constraints

<table>
<thead>
<tr>
<th>Type</th>
<th>Statement</th>
<th>Inclusion Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a = b$</td>
<td>$\text{PointsToSet}(b) \subseteq \text{PointsToSet}(a)$</td>
</tr>
<tr>
<td>2</td>
<td>$a = &amp; b$</td>
<td>$b \in \text{PointsToSet}(a)$</td>
</tr>
<tr>
<td>3</td>
<td>$a = * b$</td>
<td>$\forall c \in \text{PointsToSet}(b): \text{PointsToSet}(c) \subseteq \text{PointsToSet}(a)$</td>
</tr>
<tr>
<td>4</td>
<td>$* a = b$</td>
<td>$\forall c \in \text{PointsToSet}(a): \text{PointsToSet}(b) \subseteq \text{PointsToSet}(c)$</td>
</tr>
</tbody>
</table>

**Input:** $(A, S)$

- $A$ is a set of $n$ pointers
- $S$ is a set of $m$ statements
  - $m \leq 4 \cdot n^2$ unique statements
Andersen’s Pointer Analysis (APA)

Solve for inclusion constraints

<table>
<thead>
<tr>
<th>Type</th>
<th>Statement</th>
<th>Inclusion Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a = b$</td>
<td>$\text{PointsToSet}(b) \subseteq \text{PointsToSet}(a)$</td>
</tr>
<tr>
<td>2</td>
<td>$a = &amp; b$</td>
<td>$b \in \text{PointsToSet}(a)$</td>
</tr>
<tr>
<td>3</td>
<td>$a = * b$</td>
<td>$\forall c \in \text{PointsToSet}(b): \text{PointsToSet}(c) \subseteq \text{PointsToSet}(a)$</td>
</tr>
<tr>
<td>4</td>
<td>$* a = b$</td>
<td>$\forall c \in \text{PointsToSet}(a): \text{PointsToSet}(b) \subseteq \text{PointsToSet}(c)$</td>
</tr>
</tbody>
</table>

Input: $(A, S)$
- $A$ is a set of $n$ pointers
- $S$ is a set of $m$ statements
  - $m \leq 4 \cdot n^2$ unique statements

Output:
- **Exhaustive**: Return $\text{PointsToSet}(a)$ for all pointers $a \in A$
- **On-demand**: Return if $b \in \text{PointsToSet}(a)$ for a specific pair $a, b \in A$
This Paper: How fast can we perform Andersen’s Pointer Analysis?

- Part A: Overview of results
- Part B: D₁-Reachability in $O(n^\omega \cdot \log^2 n)$ time
Part A: Overview of results
Cubic complexity of APA
Cubic Complexity of APA

- \((A, S), |A| = n, |S| = m\)
- \(m\) can be as large as \(4 \cdot n^2\)

APA is solvable in cubic time

Cubic means:

- \(O(m^3)\)?
- \(O(n^2 \cdot m)\)?
- \(O(n^4)\)??
Cubic Complexity of APA

- \((A, S), |A| = n, |S| = m\)
- \(m\) can be as large as \(4 \cdot n^2\)

**APA is solvable in cubic time**

Cubic means:

- \(O(m^3)\)?
- \(O(n^2 \cdot m)\)?
- \(O(n^4)\)?

**Theorem**

APA solvable in \(O(n^3)\) time, regardless of \(m\).
Can we do it faster? Is $n^3$ tight?
### Exhaustive vs On-demand

#### Exhaustive
- For all $a$, output $\text{PointsToSet}(a)$
- Output size $\Theta(n^2)$

#### On-demand
- Given $a$, $b$, is it that $b \in \text{PointsToSet}(a)$?
- Output size $O(1)$
Exhaustive vs On-demand

**Exhaustive**
- For all $a$, output $\text{PointsToSet}(a)$
- Output size $\Theta(n^2)$

**On-demand**
- Given $a, b$, is it that $b \in \text{PointsToSet}(a)$?
- Output size $O(1)$

---

**Cubic Hardness for Exhaustive APA**

- Exhaustive APA can encode transitive closure of graph $G = (V, E)$

```
  a
  ↓
b
  ↓
c
```

```
  a
  ↓
\bar{a}
```

```
  b = a
  ↓
b
  ↓
c = b
```

```
  b = \& \bar{b}
  ↓
\bar{b}
```

```
  c = \& \bar{c}
  ↓
\bar{c}
```

- $\Rightarrow$ Combinatorial cubic hardness for exhaustive APA
Cubic Bottleneck for APA

**Theorem**

On-demand APA has no \( (n^{3-\epsilon}) \) time algorithm, for any \( \epsilon > 0 \), under the combinatorial MM hypothesis.

- On-demand not easier than exhaustive
- Hardness does not come from large outputs
Theorem

On-demand APA has no \((n^{3-\epsilon})\) time algorithm, for any \(\epsilon > 0\), under the combinatorial MM hypothesis.

- On-demand not easier than exhaustive
- Hardness does not come from large outputs

Proof.

Fine-grained reduction from finding a triangle in an undirected graph.
Witness Bounding
Witness Bounding

\[
\begin{align*}
&c = \&a \\
&b = \ast e \\
&b = \&d \\
&\ast d = a
\end{align*}
\]
Witness Bounding

\[ c = \&a \quad e = d \quad b = *e \quad *d = a \]

\[ n^2 \quad n^2 \quad n^2 \quad \log n \]
(i, j)-bounded APA

At most j applications of rule i.

1. \( a = b \)
2. \( a = \& b \)
3. \( a = \ast b \)
4. \( \ast a = b \)
(i, j)-bounded APA

At most \( j \) applications of rule \( i \).

1. \( a = b \)
2. \( a = \& b \)
3. \( a = * b \)
4. \( * a = b \)

Theorem

(4, \( \tilde{O}(1) \))-bounded APA solvable in \( \tilde{O}(n^\omega) \) time (\( \omega < 2.372 \ldots \)).

Only polylog \( n \) many applications of statements of the form \( * a = b \)
D₁-Reachability captures \((A, S \setminus S_4)\)

\[
\begin{align*}
    b &= \& a \\
    c &= b \\
    d &= \ast c
\end{align*}
\]
**D₁-Reachability captures** \((A, S \setminus S_4)\)

\[
\begin{align*}
    b &= \& a \\
    c &= b \\
    d &= \ast c
\end{align*}
\]

**Algorithm for \((4, j)\)-bounded APA**

1. Take the Dyck graph \(G = (A, E)\) representing \((A, S \setminus S_4)\)
2. For \(i \in \{1, \ldots, j\}\)
   2.1 Solve D₁-Reachability in \(G\)
   2.2 Find all \(x \stackrel{\&}{\rightarrow} a \stackrel{\ast}{\sim} d\)
   2.3 For every statement \(*d = y\), insert \(y \stackrel{\ast}{\rightarrow} x\) in \(G\)
### Strategy

D₁-Reachability captures \((A, S \setminus S_4)\)

- \(b = \& a\)
- \(c = b\)
- \(d = *c\)

#### Algorithm for \((4, j)\)-bounded APA

1. Take the Dyck graph \(G = (A, E)\) representing \((A, S \setminus S_4)\)
2. For \(i \in \{1, \ldots, j\}\)
   1. Solve D₁-Reachability in \(G\)
   2. Find all \(x \rightarrow a \leadsto d\)
   3. For every statement \(*d = y\), insert \(y \rightarrow x\) in \(G\)

#### Theorem

D₁-Reachability in \(O(n^\omega \cdot \log^2 n)\) time.
### Quadratic Lower Bound

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4, $\tilde{O}(1)$)-bounded APA solvable in $\tilde{O}(n^\omega)$ time.</td>
</tr>
</tbody>
</table>

- $\omega < 2.372\ldots$
- Believed that $\omega = 2 + o(1)$

| Can we do $O(n^{2-\epsilon})$ if we only look for logarithmic-length witnesses? |
Quadratic Lower Bound

Theorem

(4, $\tilde{O}(1)$)-bounded APA solvable in $\tilde{O}(n^\omega)$ time.

- $\omega < 2.372\ldots$
- Believed that $\omega = 2 + o(1)$

Can we do $O(n^{2-\epsilon})$ if we only look for logarithmic-length witnesses?

Theorem

(All, $\tilde{O}(1)$)-bounded on-demand APA has no $O(n^{2-\epsilon})$ time algorithm, for any $\epsilon > 0$, under the Orthogonal Vectors Hypothesis.

Orthogonal Vectors

Input: Sets $A, B \subseteq \{0, 1\}^d$ with $|A| = |B| = n$ and $d = \Omega(\log n)$

Output: $(a, b) \in A \times B$ such that $a \perp b$

Hypothesis: No $O(n^{2-\epsilon})$-time algorithm
Is APA parallelizable?

Can we solve it really fast with polynomial resources?
Theorem

APA is P-complete.
**Theorem**

APA is P-complete.

**Proof.**

Log-space reduction from Monotone Circuit Value Problem (CVP).

![Diagram](attachment:diagram.png)

\[ A_1 = 1 \quad A_2 = 1 \quad A_3 = 0 \]
### The class NC

Recall: given $i \in \mathbb{N}^+$, $\text{NC}^i$ is the class of problems solvable
(a) in parallel time $O(\log^i n)$ and (b) with polynomially many of processors

$$\text{NC} = \bigcup_i \text{NC}^i$$

E.g., matrix multiplication $\in \text{NC}^1$, graph reachability $\in \text{NC}^2$
**Bounded APA is Parallelizable**

### The class $NC$

Recall: given $i \in \mathbb{N}^+$, $NC^i$ is the class of problems solvable (a) in parallel time $O(\log^i n)$ and (b) with polynomially many of processors

$$NC = \bigcup \limits_{i} NC^i$$

E.g., matrix multiplication $\in NC^1$, graph reachability $\in NC^2$

### Theorem

$(4, \log^i n)$-bounded APA is in $NC^{i+2}$.

Witnesses with $\leq \log^i n$ applications of $*a = b$ helps
Conclusion

• Andersen’s Pointer Analysis is one of the most popular static pointer analyses
• Its running time is very important, yet its complexity, so far, poorly understood

Thank you!
Conclusion

- Andersen’s Pointer Analysis is one of the most popular static pointer analyses
- Its running time is very important, yet its complexity, so far, poorly understood

The complexity of Andersen’s Pointer Analysis

- Cubic upper and lower bounds
- Bounding number of $a = b$ statements helps
  - $\tilde{O}(n^\omega)$ upper bound (D1-Reachability solvable in this bound)
  - $\Omega(n^2)$ lower bound
- $P$-complete
- Bounding number of $a = b$ statements brings it in NC
Conclusion

- Andersen’s Pointer Analysis is one of the most popular static pointer analyses
- Its running time is very important, yet its complexity, so far, poorly understood

The complexity of Andersen’s Pointer Analysis

- Cubic upper and lower bounds
- Bounding number of $\star a = b$ statements helps
  - $\tilde{O}(n^\omega)$ upper bound (D$_1$-Reachability solvable in this bound)
  - $\Omega(n^2)$ lower bound
- P-complete
- Bounding number of $\star a = b$ statements brings it in NC

Thank you!
Part B: $D_1$-Reachability in $O(n^\omega \cdot \log^2 n)$ time
\[ \Sigma = \{(1,)_1, \ldots (k,)_k\} \cup \{\epsilon\} \]

\[ S \to SS | (1S)_1 | \ldots | (kS)_k | \epsilon \]

\[ G = (V, E, \lambda : E \to \Sigma) \]
\[ \Sigma = \{(1,)_1, \ldots (k,)_k\} \cup \{\epsilon\} \]

\[ S \rightarrow SS \mid (1\ S)_{1} \mid \ldots \mid (k\ S)_{k} \mid \epsilon \]

\[ G = (V, E, \lambda : E \rightarrow \Sigma) \]
D$_1$-Reachability

D$_k$-Reachability: use $k$ parenthesis types $(i,)_i$

**Computational problem**

Given a Dyck graph $G$, compute all-pairs D$_k$-Reachability.
D\textsubscript{1}-Reachability

D\textsubscript{k}-Reachability: use \( k \) parenthesis types \((i,)_i\)

### Computational problem

Given a Dyck graph \( G \), compute all-pairs D\textsubscript{k}-Reachability.

How fast can we solve it?

- \( k = 0 \): standard graph reachability, \( O(n^\omega) \)
  - Standard transitive closure \( O(n^3) \)
  - \( \omega \simeq 2.37 \)
- \( k \geq 2 \): \( O(n^3) \), believed to be tight
D\textsubscript{k}-Reachability: use \( k \) parenthesis types \((i,)_i\)

**Computational problem**
Given a Dyck graph \( G \), compute all-pairs D\textsubscript{k}-Reachability.

**How fast can we solve it?**
- \( k = 0 \): standard graph reachability, \( O(n^\omega) \)
  - Standard transitive closure \( O(n^3) \)
  - \( \omega \simeq 2.37 \)
- \( k \geq 2 \): \( O(n^3) \), believed to be tight

**Theorem**
All-pairs D\textsubscript{1}-Reachability *can be solved in* \( O(n^\omega \cdot \log^2 n) \) time.
Length of Witness Paths

- For $k = 1$, the stack alphabet is unary
- Just a counter
Length of Witness Paths

- For $k = 1$, the stack alphabet is unary
- Just a counter

$k = 2$

**Lemma**

*Distances can be exponential*

- at least $\Omega(2^n)$
Length of Witness Paths

- For $k = 1$, the stack alphabet is unary
- Just a counter

$k = 2$

<table>
<thead>
<tr>
<th>Lemma</th>
<th>Distances can be exponential</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>at least $\Omega(2^n)$</td>
</tr>
</tbody>
</table>

$k = 1$

<table>
<thead>
<tr>
<th>Lemma</th>
<th>Distances are polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>at most $O(n^2)$</td>
</tr>
</tbody>
</table>
Bell-Shape Paths

Counter
Bell-Shape Paths

Counter

Bell-shape

Counter
Bell-Shape Reachability

- What if we focus on paths with counter $\leq 1$?
- At most one $+1$
Bell-Shape Reachability

• What if we focus on paths with counter \( \leq 1 \)?
• At most one +1
Bell-Shape Reachability

- What if we focus on paths with counter $\leq 1$?
- At most one $+1$

\[ \begin{align*}
+1 & \quad +1 & \quad -1 & \quad -1 \\
-1 & \quad +1 & \quad -1 & \\
\end{align*} \]
Bell-Shape Reachability

- What if we focus on paths with counter \( \leq 1 \)?
- At most one +1

Larger counter values? Iterative doubling
Bell-Shape Reachability

- $a_1 ightarrow a_2 ightarrow a_3$
- $b_1 ightarrow b_2 ightarrow b_3$
- $c_1 ightarrow c_2 ightarrow c_3$
- $d_1 ightarrow d_2 ightarrow d_3$
- $e_1 ightarrow e_2 ightarrow e_3$

- $+1$ for $a ightarrow d$
- $+1$ for $c ightarrow e$
- $-1$ for $b ightarrow e$
- $-1$ for $a ightarrow c$

Diagram:

- Nodes: $a, b, c, d, e$
- Edges with labels $+1$ and $-1$
Bell-Shape Reachability
Bell-Shape Reachability
Bell-Shape Reachability

![Diagram of Bell-Shape Reachability]

- \( a \) to \( b \): +1
- \( a \) to \( d \): +1
- \( b \) to \( d \): -1
- \( d \) to \( e \): -1
- \( c \) to \( d \): +1

Additional diagrams showing different scenarios with +1, -1, and +2 transitions between nodes.
Bell-Shape Reachability
Bell-Shape Reachability

Diagram:

- Nodes: a, b, c, d, e
- Edges:
  - a → d
  - b → e
  - c → b

Weights:
- +1: a → b, c → a, d → c
- -1: b → c, d → b

Additional edges:
- a1 → a2, a2 → a3
- b1 → b2, b2 → b3
- c1 → c2, c2 → c3
- d1 → d2, d2 → d3
- e1 → e2, e2 → e3

Weights for additional edges:
- +1: a1 → a2, b1 → b2, c1 → c2, d1 → d2, e1 → e2
- +2: a2 → b1, c2 → b2, d2 → c1, e2 → d1
- -2: a2 → a1, b2 → b1, c2 → c1, d2 → d1, e2 → e1
Bell-Shape Reachability

\[ a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \]

\[ +1 \rightarrow +1 \rightarrow -1 \rightarrow -1 \]

\[ +1 \rightarrow +2 \rightarrow -1 \rightarrow -2 \]

\[ +4 \rightarrow +4 \rightarrow -4 \rightarrow -4 \]
Bell-Shape Reachability

+1

a

+1

b

−1

c

−1
d

+1

e

a_1 \rightarrow b_1 \rightarrow c_1 \rightarrow d_1 \rightarrow e_1

a_2 \rightarrow b_2 \rightarrow c_2 \rightarrow d_2 \rightarrow e_2

a_3 \rightarrow b_3 \rightarrow c_3 \rightarrow d_3 \rightarrow e_3

+2

a_1 \rightarrow b_1 \rightarrow c_1 \rightarrow d_1 \rightarrow e_1

a_2 \rightarrow b_2 \rightarrow c_2 \rightarrow d_2 \rightarrow e_2

a_3 \rightarrow b_3 \rightarrow c_3 \rightarrow d_3 \rightarrow e_3

−2

a_1 \rightarrow b_1 \rightarrow c_1 \rightarrow d_1 \rightarrow e_1

a_2 \rightarrow b_2 \rightarrow c_2 \rightarrow d_2 \rightarrow e_2

a_3 \rightarrow b_3 \rightarrow c_3 \rightarrow d_3 \rightarrow e_3

−4

a_1 \rightarrow b_1 \rightarrow c_1 \rightarrow d_1 \rightarrow e_1

a_2 \rightarrow b_2 \rightarrow c_2 \rightarrow d_2 \rightarrow e_2

a_3 \rightarrow b_3 \rightarrow c_3 \rightarrow d_3 \rightarrow e_3
Bell-Shape Reachability

A diagram illustrating the concept of Bell-Shape Reachability with nodes labeled a, b, c, d, e and edges labeled with +1, -1, +2, -2, +4, -4.
Bell-Shape Reachability

How long did it take?

In each iteration
- Standard transitive closure
- $O(n^\omega)$ time
- How many iterations?
Bell-Shape Reachability

How long did it take?

In each iteration

- Standard transitive closure
- $O(n^\omega)$ time
- How many iterations?

- Every iteration doubles the counter value
- Max counter value $\leq$ path length $\leq n^2$
- Only $2 \log n$ iterations
How long did it take?

In each iteration
- Standard transitive closure
- $O(n^\omega)$ time
- How many iterations?

- Every iteration doubles the counter value
- Max counter value $\leq$ path length $\leq n^2$
- Only $2\log n$ iterations

Total time $O(n^\omega \cdot \log n)$
General Case

Stack Height

Repeat!

How many iterations?

30
General Case

Stack Height

How many iterations?
30
General Case

Repeat!

Stack Height

How many iterations? 30
General Case

Repeat!

How many iterations?
Counting Bells

- #bells halves in each iteration
- initial #bells ≤ path length = $O(n^2)$
- $O(\log n)$ iterations
- $O(n^\omega \cdot \log n)$ per iteration
- Total time $n^\omega \cdot \log 2n$
Counting Bells

- Initial number of bells ≤ path length = \(O(n^2)\)
- \(O(\log n)\) iterations
- \(O(n^{\omega \cdot \log n})\) per iteration

Total time \(n^{\omega \cdot \log 2^n}\)
Counting Bells

Stack Height

• #bells halves in each iteration
• initial #bells \leq \text{path length} = O(n^2)
• \text{O}(\log n) \text{ iterations}
• \text{O}(n^\omega \cdot \log n) \text{ per iteration}

Total time \frac{n^\omega}{\log^2 n}
Counting Bells

- #bells halves in each iteration
- initial #bells ≤ path length = \( O(n^2) \)
- \( O(\log n) \) iterations
- \( O(n^{\omega} \cdot \log n) \) per iteration

Total time \( n^{\omega} \cdot \log^2 n \)
• #bells halves in each iteration
• initial #bells $\leq$ path length $= O(n^2)$
Counting Bells

- #bells halves in each iteration
- initial #bells ≤ path length = $O(n^2)$

- $O(\log n)$ iterations
- $O(n^\omega \cdot \log n)$ per iteration
Counting Bells

- #bells halves in each iteration
- initial #bells ≤ path length = $O(n^2)$
- $O(\log n)$ iterations
- $O(n^\omega \cdot \log n)$ per iteration

Total time $n^\omega \cdot \log^2 n$