Number of Variables for Graph Differentiation and the Resolution of Graph Isomorphism Formulas

Jacobo Torán & Florian Wörz

Universität Ulm

GI Meeting Deduction and Logic
Friedrich-Alexander-Universität Erlangen-Nürnberg
April 8, 2022
Take the Graph Isomorphism Problem...

\[ \exists \varphi \in \text{Iso}(G, H)? \]

\[ x_{i,j} = 1 : \iff v_i \text{ is mapped to } w_j \]

Type 1 clauses:

\[ \forall i \in [n]: (x_{i,1} \lor x_{i,2} \lor \cdots \lor x_{i,n}) \]

\[ \forall j \in [n]: (x_{1,j} \lor x_{2,j} \lor \cdots \lor x_{n,j}) \]

Type 2 clauses:

\[ \forall i, j, k \in [n] \text{ with } j \neq k: (x_{i,j} \lor x_{i,k}) \]

\[ \forall i, j, k \in [n] \text{ with } i \neq j: (x_{i,k} \lor x_{j,k}) \]

Type 3 clauses:

\[ \forall i < j \text{ and } k \neq \ell \text{ with } \{v_i, v_j\} \in E_G \iff \{v_k, v_\ell\} \notin E_H: (x_{i,k} \lor x_{j,\ell}) \]
Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem . . .

\[
\exists \varphi \in \text{Iso}(G, H) \quad ?
\]

\[
x_{i,j} = 1 : \iff v_i \text{ is mapped to } w_j
\]

. . . and encode it as the formula ISO(G, H):

- **Type 1 clauses:** consider all vertices
  \[
  \forall i \in [n] : (x_{i,1} \lor x_{i,2} \lor \cdots \lor x_{i,n})
  \]
  \[
  \forall j \in [n] : (x_{1,j} \lor x_{2,j} \lor \cdots \lor x_{n,j})
  \]

- **Type 2 clauses:** function + injective
  \[
  \forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \lor \overline{x_{i,k}})
  \]
  \[
  \forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \lor \overline{x_{j,k}})
  \]

- **Type 3 clauses:** adjacency relation
  \[
  \forall i < j \text{ and } k \neq \ell \text{ with }
  \{v_i, v_j\} \in E_G \iff \{v_k, v_{\ell}\} \notin E_H : (\overline{x_{i,k}} \lor \overline{x_{j,\ell}})
  \]
Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem... and encode it as the formula $\text{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices
  
  $\forall i \in [n] : (x_{i,1} \lor x_{i,2} \lor \cdots \lor x_{i,n})$
  
  $\forall j \in [n] : (x_{1,j} \lor x_{2,j} \lor \cdots \lor x_{n,j})$

- **Type 2 clauses:** function + injective
  
  $\forall i, j, k \in [n]$ with $j \neq k : (\overline{x_{i,j}} \lor \overline{x_{i,k}})$
  
  $\forall i, j, k \in [n]$ with $i \neq j : (\overline{x_{i,k}} \lor \overline{x_{j,k}})$

- **Type 3 clauses:** adjacency relation
  
  $\forall i < j$ and $k \neq \ell$ with
  
  $\{v_i, v_j\} \in E_G \iff \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \lor \overline{x_{j,\ell}})$
Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem. . .

\[ x_{i,j} = 1 :\iff v_i \text{ is mapped to } w_j \]

. . . and encode it as the formula \( \text{ISO}(G, H) \):

- **Type 1 clauses:** consider all vertices
  \[
  \forall i \in [n] : (x_{i,1} \lor x_{i,2} \lor \cdots \lor x_{i,n}) \\
  \forall j \in [n] : (x_{1,j} \lor x_{2,j} \lor \cdots \lor x_{n,j})
  \]

- **Type 2 clauses:** function + injective
  \[
  \forall i, j, k \in [n] \text{ with } j \neq k : (x_{i,j} \lor x_{i,k}) \\
  \forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \lor \overline{x_{j,k}})
  \]

- **Type 3 clauses:** adjacency relation
  \[
  \forall i < j \text{ and } k \neq \ell \text{ with } \\
  \{v_i, v_j\} \in E_G \iff \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \lor \overline{x_{j,\ell}})
  \]
Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem... and encode it as the formula ISO(G, H):

- **Type 1 clauses:** consider all vertices
  \[\forall i \in [n] : (x_{i,1} \lor x_{i,2} \lor \cdots \lor x_{i,n})\]
  \[\forall j \in [n] : (x_{1,j} \lor x_{2,j} \lor \cdots \lor x_{n,j})\]

- **Type 2 clauses:** function + injective
  \[\forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \lor \overline{x_{i,k}})\]
  \[\forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \lor \overline{x_{j,k}})\]

- **Type 3 clauses:** adjacency relation
  \[\forall i < j \text{ and } k \neq \ell \text{ with }\]
  \[\{v_i, v_j\} \in E_G \iff \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \lor \overline{x_{j,\ell}})\]
Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...

...and encode it as the formula $\text{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices
  
  $\forall i \in [n] : (x_{i,1} \lor x_{i,2} \lor \cdots \lor x_{i,n})$
  
  $\forall j \in [n] : (x_{1,j} \lor x_{2,j} \lor \cdots \lor x_{n,j})$

- **Type 2 clauses:** function + injective
  
  $\forall i, j, k \in [n]$ with $j \neq k : (\overline{x_{i,j}} \lor \overline{x_{i,k}})$
  
  $\forall i, j, k \in [n]$ with $i \neq j : (\overline{x_{i,k}} \lor \overline{x_{j,k}})$

- **Type 3 clauses:** adjacency relation
  
  $\forall i < j$ and $k \neq \ell$ with
  
  $\{v_i, v_j\} \in E_G \iff \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \lor \overline{x_{j,\ell}})$

$x_{i,j} = 1 : \iff v_i$ is mapped to $w_j$
Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem... and encode it as the formula \( \text{ISO}(G, H) \):

- **Type 1 clauses:** consider all vertices
  
  \[
  \forall i \in [n] : (x_{i,1} \lor x_{i,2} \lor \cdots \lor x_{i,n})
  \]
  
  \[
  \forall j \in [n] : (x_{1,j} \lor x_{2,j} \lor \cdots \lor x_{n,j})
  \]

- **Type 2 clauses:** function + injective
  
  \[
  \forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \lor \overline{x_{i,k}})
  \]
  
  \[
  \forall i, j, k \in [n] \text{ with } i \neq j : (x_{i,k} \lor x_{j,k})
  \]

- **Type 3 clauses:** adjacency relation
  
  \[
  \forall i < j \text{ and } k \neq \ell \text{ with } \{v_i, v_j\} \in E_G \iff \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \lor \overline{x_{j,\ell}})
  \]
World 1:
The Resolution Proof System
The Proof System Resolution

Resolution Rule:

\[
\frac{A \lor x \quad B \lor \overline{x}}{A \lor B}
\]

Distinction by Cases: [Galesi & Thapen]

\[
\frac{A_1 \lor \overline{x_1} \quad \ldots \quad A_m \lor \overline{x_m}}{B \lor A_1 \lor \ldots \lor A_m}
\]
if \( (B \lor x_1 \lor \ldots \lor x_m) \in F \)
Complexity Measures for Resolution

**Size**
- \# clauses

**Width**
- \# literals in largest clause

**Narrow Width**
- exclude all axioms

**Space**
- max \# clauses in memory
Complexity Measures for Resolution

**Size**

# clauses (here: 11)

**Width**

# literals in largest clause

**Narrow Width**

exclude all axioms

**Space**

max # clauses in memory
Complexity Measures for Resolution

**Size**

# clauses (here: 11)

**Width**

# literals in largest clause (here: 3)

**Narrow Width**

exclude all axioms

**Space**

max # clauses in memory
Complexity Measures for Resolution

**Size**

# clauses (here: 11)

**Width**

# literals in largest clause (here: 3)

**Narrow Width**

exclude all axioms (here: 2)

**Space**

max # clauses in memory
Complexity Measures for Resolution

Size

# clauses (here: 11)

Width

# literals in largest clause (here: 3)

Narrow Width

exclude all axioms (here: 2)

Space

max # clauses in memory
Complexity Measures for Resolution

Size
    # clauses (here: 11)

Width
    # literals in largest clause (here: 3)

Narrow Width
    exclude all axioms (here: 2)

Space
    max # clauses in memory
Complexity Measures for Resolution

Size
   # clauses (here: 11)

Width
   # literals in largest clause (here: 3)

Narrow Width
   exclude all axioms (here: 2)

Space
   max # clauses in memory (here: 5 at time 8)
Complexity Measures for Resolution—What we really care about

For each complexity measure $\mathcal{C}$:

Take minimum over all refutations $\pi$

$$\mathcal{C}(F \vdash \bot) := \min_{\pi: F \vdash \bot} \mathcal{C}(\pi)$$
World 2:
Descriptive Complexity / First-order logic
Immerman’s $k$-pebble game: Player I wants to show $G \not\cong H$

- Player I and Player II have $k$ pebble pairs

---

Player I and Player II have $k$ pebble pairs
Immerman’s $k$-pebble game: Player I wants to show $G \not\cong H$

- Player I and Player II have $k$ pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
Immerman’s $k$-pebble game: Player I wants to show $G \not\cong H$

- Player I and Player II have $k$ pebble pairs
- In each round:
  - Player I chooses:
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
Immerman’s $k$-pebble game: Player I wants to show $G \not\cong H$

- Player I and Player II have $k$ pebble pairs
- In each round:
  - Player I chooses:
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - Player II simply reacts.
Immerman’s $k$-pebble game: Player I wants to show $G \not\cong H$

- Player I and Player II have $k$ pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic
Immerman’s $k$-pebble game: Player I wants to show $G \not\cong H$

- Player I and Player II have $k$ pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic
Immerman’s $k$-pebble game: Player I wants to show $G \not\cong H$

- Player I and Player II have $k$ pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic $\times$ Player I won!
Immerman's $k$-pebble game: Player I wants to show $G \not\cong H$

- Player I and Player II have $k$ pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic $\times$ Player I won!
Immerman’s $k$-pebble game: Player I wants to show $G \not\cong H$

- Player I and Player II have $k$ pebble pairs.
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic $\times$ Player I won!
Our Main Result:
Combining both worlds
Main Result: Connection between FO and PC

Immerman’s Pebble Game on $G$ and $H$

$G \not\equiv_{L_{k,m}} H$

Narrow Width Refutation of $\text{ISO}(G,H)$

$N\text{-Width}(\pi) \leq k - 1$  
$\text{PosDepth}(\pi) \leq m$
Implications

For every pair of non-isomorphic graphs \((G, H)\) with \(n\) vertices each and for every \(k \in \mathbb{N}\):

1. \(G \not\equiv_{\mathcal{L}_k} H \implies \text{Size}(\text{ISO}(G, H) \vdash \bot) \leq n^{O(k)}\)

2. \(G \equiv_{\mathcal{L}_k} H \implies \begin{cases} 
\text{Tree-Size}(\text{ISO}(G, H) \vdash \bot) \geq 2^k \\
\text{Space}(\text{ISO}(G, H) \vdash \bot) \geq k + 1
\end{cases}\)

3. \((G, \lambda) \equiv_{\mathcal{L}_k} (H, \mu) \implies \text{Size}(\text{ISO}(G, H) \vdash \bot) \geq \exp\left(\Omega\left(\frac{k^2}{\text{sum of color class sizes}}\right)\right)\)
Proof Idea: Use $k$-witnessing game

Immerman's Pebble Game on $G$ and $H$

$G \neq L_{k,m} H$

$k$-witnessing game

Spoiler wins on ISO($G, H$)

Narrow Width Refutation of ISO($G, H$)

N-Width($\pi$) $\leq k - 1$
PosDepth($\pi$) $\leq m$
They compete in the $k$-witnessing game on the formula $\text{ISO}(G, H)$

- Game state is a partial assignment, initially $\alpha_0 = \varepsilon$
- In each round $i$
  - **Spoiler:** Chooses a subset $\alpha' \subseteq \alpha_{i-1}$ of size at most $k - 1$
    - Chooses a Type 1 clause $C$ in $\text{ISO}(G, H)$
  - **Duplicator:** Extends $\alpha_i := \alpha' \cup \{\ell = 1\}$ for some literal $\ell \in C$

- Game ends when Duplicator cannot extend such that
  - $\alpha_i$ satisfies $C$ and
  - does not falsify any other clause in $\text{ISO}(G, H)$
Proof: \( G \not\equiv_{\mathcal{L}_k} H \implies \text{N-Width}(\text{ISO}(G, H) \vdash \bot) \leq k - 1 \)

Convert Strategy Graph of Spoiler into Narrow Width Refutation
Proof: \( G \not\equiv L_k H \iff \text{N-Width}(\text{ISO}(G, H) \vdash \bot) \leq k - 1 \)

Convert Strategy Graph of Spoiler into Narrow Width Refutation
Proof: $G \not\equiv L_k H \implies \text{N-Width}(\text{ISO}(G, H) \vdash \bot) \leq k - 1$

Convert Strategy Graph of Spoiler into Narrow Width Refutation
Proof: \( G \not\equiv_{\mathcal{L}_k} H \implies \text{N-Width}(\text{ISO}(G, H) \vdash \bot) \leq k - 1 \)

Convert Strategy Graph of Spoiler into Narrow Width Refutation
Proof: \( G \not\equiv \mathcal{L}_k H \implies \text{N-Width} (\text{ISO}(G, H) \vdash \bot) \leq k - 1 \)

Convert Strategy Graph of Spoiler into Narrow Width Refutation
Proof: \( G \not\equiv \mathcal{L}_k H \implies \text{N-Width}(\text{ISO}(G, H) \vdash \bot) \leq k - 1 \)

Convert Strategy Graph of Spoiler into Narrow Width Refutation
Proof: \( G \not\equiv \mathcal{L}_k \) \( H \implies \text{N-Width}(\text{ISO}(G, H) \vdash \bot) \leq k - 1 \)

Convert Strategy Graph of Spoiler into Narrow Width Refutation
Proof: \( G \not\equiv_{\mathcal{L}_k} H \implies \text{N-Width}(\text{ISO}(G, H) \vdash \bot) \leq k - 1 \)

Convert Strategy Graph of Spoiler into Narrow Width Refutation:

\((\alpha, C) \rightsquigarrow C^\alpha\) (set of literals falsified by \(\alpha\))
Proof: $G \not\equiv_{\mathcal{L}_k} H \implies \text{N-Width}(\text{ISO}(G, H) \vdash \bot) \leq k - 1$

Convert Strategy Graph of Spoiler into Narrow Width Refutation:

$(\alpha, C') \sim C'_\alpha$ (set of literals falsified by $\alpha$)
Proof: \( G \not\equiv \mathcal{L}_k H \Rightarrow N\text{-Width}(\text{ISO}(G, H) \vdash \bot) \leq k - 1 \)

Convert Strategy Graph of Spoiler into Narrow Width Refutation:

\((\alpha, C) \rightsquigarrow C_\alpha\) (set of literals falsified by \(\alpha\))

Diagram:

- \((\varepsilon, x_{1,3} \lor x_{2,3} \lor x_{3,3})\)
- \(x_{1,3} = 1\)
- \(x_{2,3} = 1\)
- \(x_{3,3} = 1\)
- \(x_{1,1} \lor x_{3,3}\)
- \(x_{2,1} \lor x_{3,3}\)
- \(x_{3,1} \lor x_{3,3}\)

with axiom \(x_{1,1} \lor x_{2,1} \lor x_{3,1}\)
Proof: \( G \not\equiv L_k H \Rightarrow N\text{-}\text{Width}(\text{ISO}(G, H) \vdash \bot) \leq k - 1 \)

Convert Strategy Graph of Spoiler into Narrow Width Refutation:

\((\alpha, C) \rightsquigarrow C'_{\alpha}\) (set of literals falsified by \(\alpha\))

with axiom \(x_{1,3} \lor x_{2,3} \lor x_{3,3}\)

\(\begin{align*}
\lnot x_{1,3} & \quad \text{with axiom } x_{1,3} \lor x_{2,3} \lor x_{3,3} \\
\lnot x_{2,3} & \quad \text{with axiom } x_{1,1} \lor x_{2,1} \lor x_{3,1} \\
x_{3,3} & \quad \text{with axiom } x_{1,1} \lor x_{2,1} \lor x_{3,1} \\
x_{1,1} \lor x_{3,3} & \\
x_{2,1} \lor x_{3,3} & \\
x_{3,1} \lor x_{3,3} &
\end{align*}\)
Proof: \( G \not\equiv \mathcal{L}_k H \implies \text{N-Width}(\text{ISO}(G, H) \vdash \bot) \leq k - 1 \)

Convert Strategy Graph of Spoiler into Narrow Width Refutation:

\((\alpha, C) \rightsquigarrow C_\alpha\) (set of literals falsified by \(\alpha\))

with axiom \(x_{1,3} \lor x_{2,3} \lor x_{3,3}\)

with axiom \(x_{1,1} \lor x_{2,1} \lor x_{3,1}\)

with axiom \(x_{1,1} \lor x_{3,3}\)

with axiom \(x_{2,1} \lor x_{3,3}\)

with axiom \(x_{3,1} \lor x_{3,3}\)
Symmetric Resolution
An Exponential GI Lower Bound for SRC-1
Krishnamurthy’s Symmetry Rules

\[
\begin{align*}
\neg b_1 c_1 & \\
\neg a_1 & \\
\neg c_2 & \\
\bot & \\
\end{align*}
\]

\[
\begin{align*}
a_1 b_1 & \\
a_1 c_1 & \\
\neg a_1 & \\
\neg c_1 \neg c_2 & \\
\bot & \\
\end{align*}
\]

\[
\begin{align*}
a_2 b_2 & \\
a_2 c_2 & \\
\neg a_2 & \\
\neg c_1 \neg c_2 & \\
\bot & \\
\end{align*}
\]
Krishnamurthy’s Symmetry Rules

\[
\begin{align*}
\neg b_1 c_1 & \quad a_1 b_1 \\
\neg a_1 & \quad a_1 c_1 \\
c_1 & \quad \neg c_2 \\
\neg c_1 c_2 & \quad a_2 c_2 \\
a_2 b_2 & \quad \neg b_2 c_2 \\
\neg a_2 & \\
c_2 & \\
\bot &
\end{align*}
\]
Krishnamurthy’s Symmetry Rules

\[ a_1 b_1 \quad \neg b_1 c_1 \quad a_2 b_2 \quad \neg b_2 c_2 \]

\[ a_1 c_1 \quad \neg a_1 \quad a_2 c_2 \quad \neg a_2 \]

\[ c_1 \quad \neg c_1 \neg c_2 \quad \neg c_1 \neg c_2 \quad c_2 \]

\[ \bot \]
Krishnamurthy’s Symmetry Rules

\[ \sigma(F') \subseteq F \]
Krishnamurthy’s Symmetry Rules

\[ [\sigma: \ell_1 \mapsto \ell_2] \in \text{Sym}(F) \]
Krishnamurthy’s Symmetry Rules

\[ \sigma : \ell_1 \mapsto \ell_2 \in \text{Sym}(F) \]

Diagram:

- \( a_1 b_1 \)
- \( \neg b_1 c_1 \)
- \( a_1 c_1 \)
- \( \neg a_1 \)
- \( a_2 b_2 \)
- \( \neg b_2 c_2 \)
- \( a_2 c_2 \)
- \( \neg a_2 \)
- \( c_1 \)
- \( \neg c_1 c_2 \)
- \( c_2 \)
- \( \bot \)
Krishnamurthy’s Symmetry Rules

\[ [\sigma : \ell_1 \mapsto \ell_2] \in \text{Sym}(F) \]
The SRC Proof Systems

Have a derivation $\pi : F' \vdash C$ from a subformula $F' \subseteq F$.
To derive $\sigma(C)$ from $C$ in one step we need a renaming $\sigma$ with

**SRC-1 (Global Symmetries)**

$\sigma(F) = F$

**SRC-2 (Local Symmetries)**

$\sigma(F') \subseteq F$

**SRC-3 (Dynamic Symmetries)**

also allow symmetries in resolvents
Battle SRC-1 With Asymmetric Graphs

Asymmetric Graph $G$: $\text{Aut}(G) = \{\text{id}\}$
Asymmetric Graph $G$: $\text{Aut}(G) = \{\text{id}\}$

Lemma: Asymmetric graphs $\implies$ Asymmetric ISO-formula
Battle SRC-1 With Asymmetric Graphs

**Asymmetric Graph** $G$: $\text{Aut}(G) = \{\text{id}\}$

**Lemma:** Asymmetric graphs $\implies$ Asymmetric ISO-formula

**Lemma:** Asymmetric formula $\implies$ Res-Size $=$ SRC-1-Size [Szeider]
Asymmetric Graphs of Large WL-Dimension

[Dawar and Khan] showed: There are pairs of non-isomorphic graphs that are
- asymmetric (unlike CFI-graphs)
- have small size $O(k)$
- with large WL-dim $k$
- and color classes of size 4

Without looking at ISO-formula:

$$ (G, \lambda) \equiv \mathcal{L}_k (H, \mu) \implies \text{Size}(\text{ISO}(G, H) \vdash \bot) \geq \exp \left( \Omega \left( \frac{k^2}{\text{sum of color class sizes}} \right) \right) $$
An Exponential GI Lower Bound for SRC-1

**Our Result:**

There is a family of non-isomorphic graph pairs \((G_n, H_n)\) with \(O(n)\) vertices each, such that any SRC-1 refutation of \(\text{ISO}(G_n, H_n)\) requires size \(\exp(\Omega(n))\).

CFI graphs (used for Resolution GI lower bound) don’t work! [Schweitzer & Seebach]
Summary and Open Problems

- Number of Variables for Graph Differentiation = Narrow Resolution Width
- Upper and lower bounds for refuting GI in Resolution
- Exponential Size Lower Bound for GI in SRC-1

Q. How does one show “true” exponential lower bounds (for a symmetric formula) in the systems SRC-2 or SRC-3?