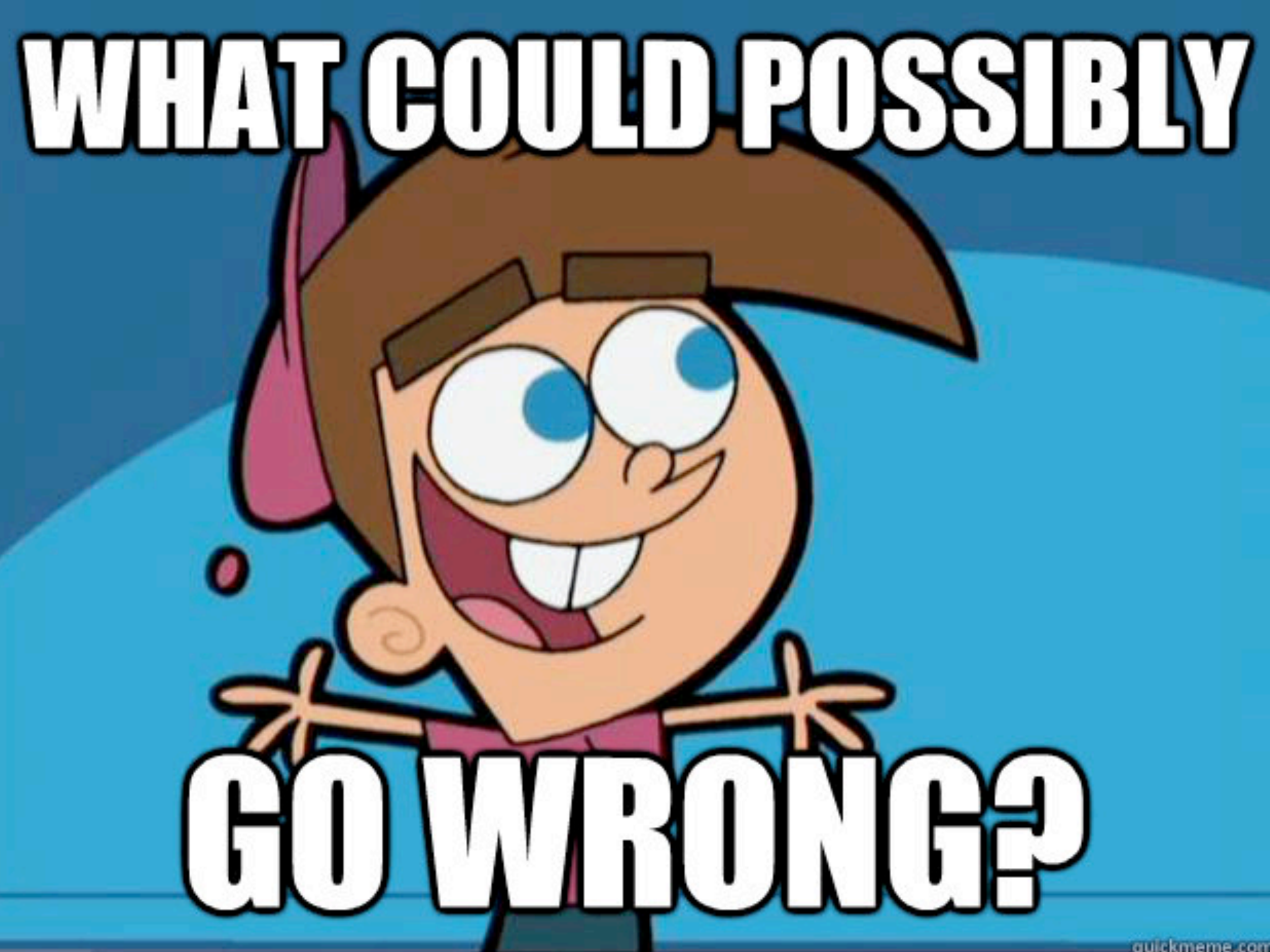
Analyzing Android

Shuying Liang, Matt Might*,
Cambell Christensen, Hao Hou, Petey Aldous,
Celeste Hollenbeck, Will Byrd

A terrible idea

WHAT COULD POSSIBLY



GO WRONG?


DELL

DARPA·APAC

App auditor

Good app

App auditor

App auditor

App auditor

App auditor

Good app

Bad app

App auditor

Good app

App auditor

Good app

App auditor

Good app

Good app

Step 1

Solve the Halting problem

Step 2

Pass the Turing test

App auditor

Good app

Human

App auditor

Good app

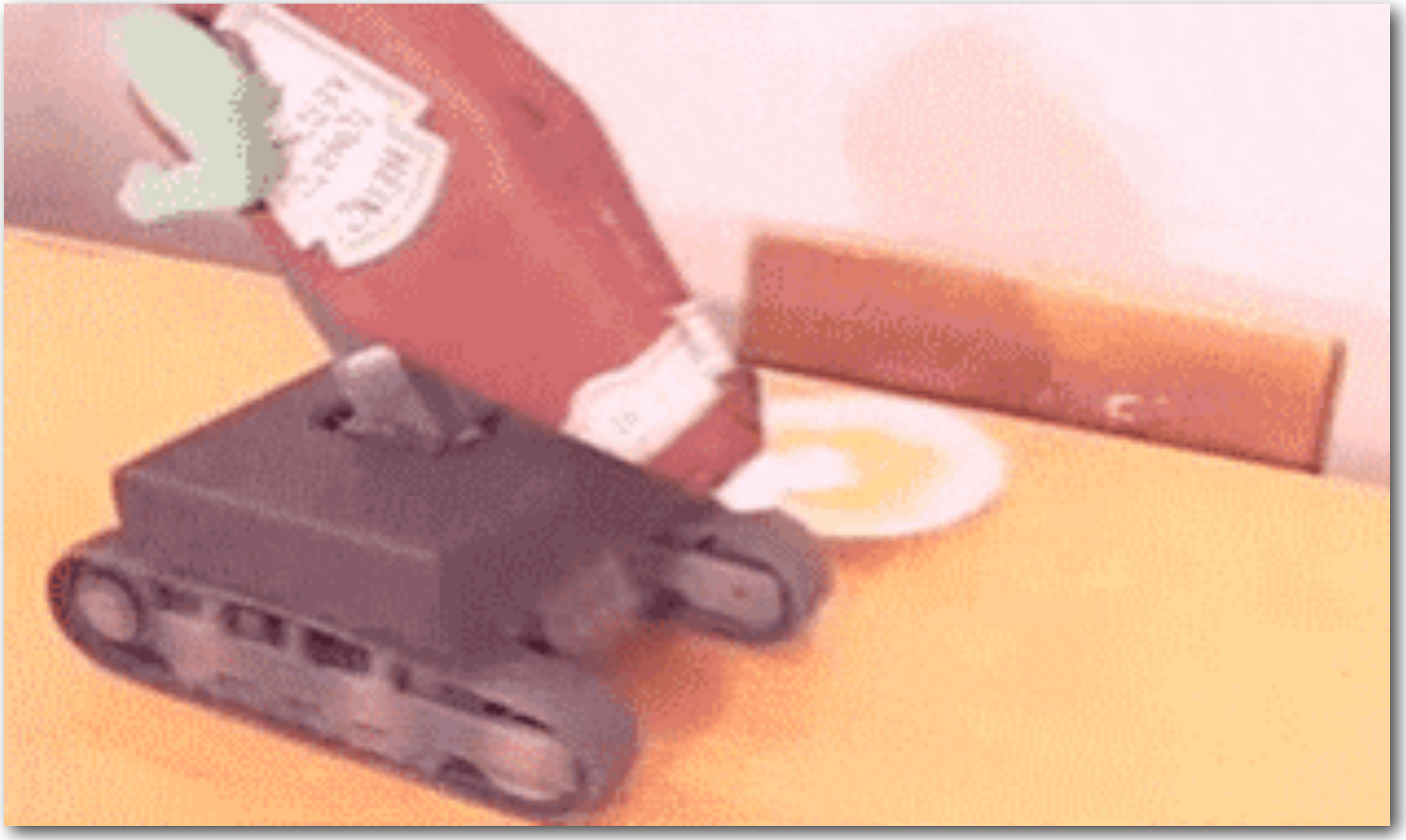
Engagements

In practice?

0%

Vision


LOKI THOR CROSSES

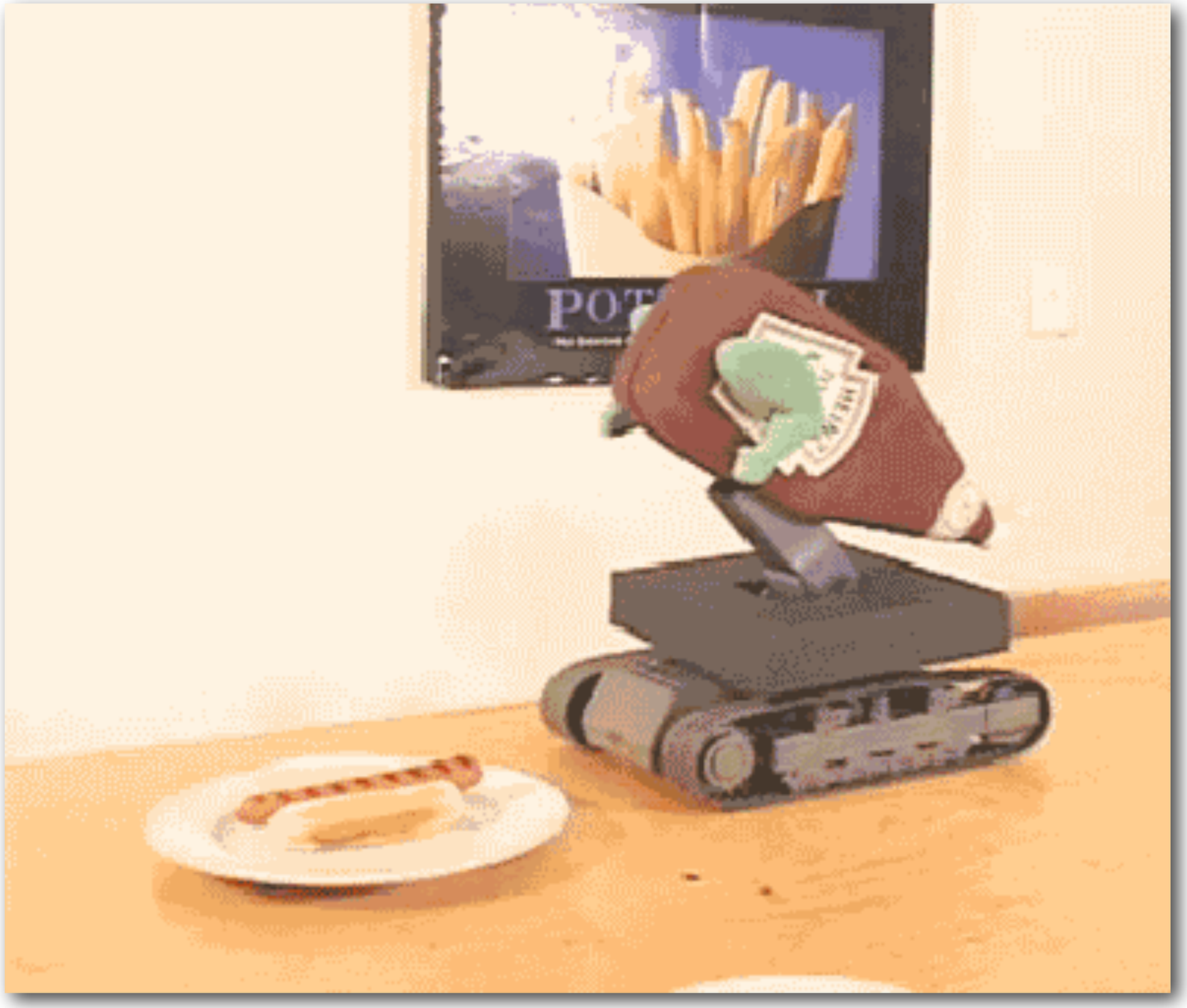


researchinprogress.tumblr.com

50%

90%

93%



How does it work?

MAGIC!?

Small-step analysis

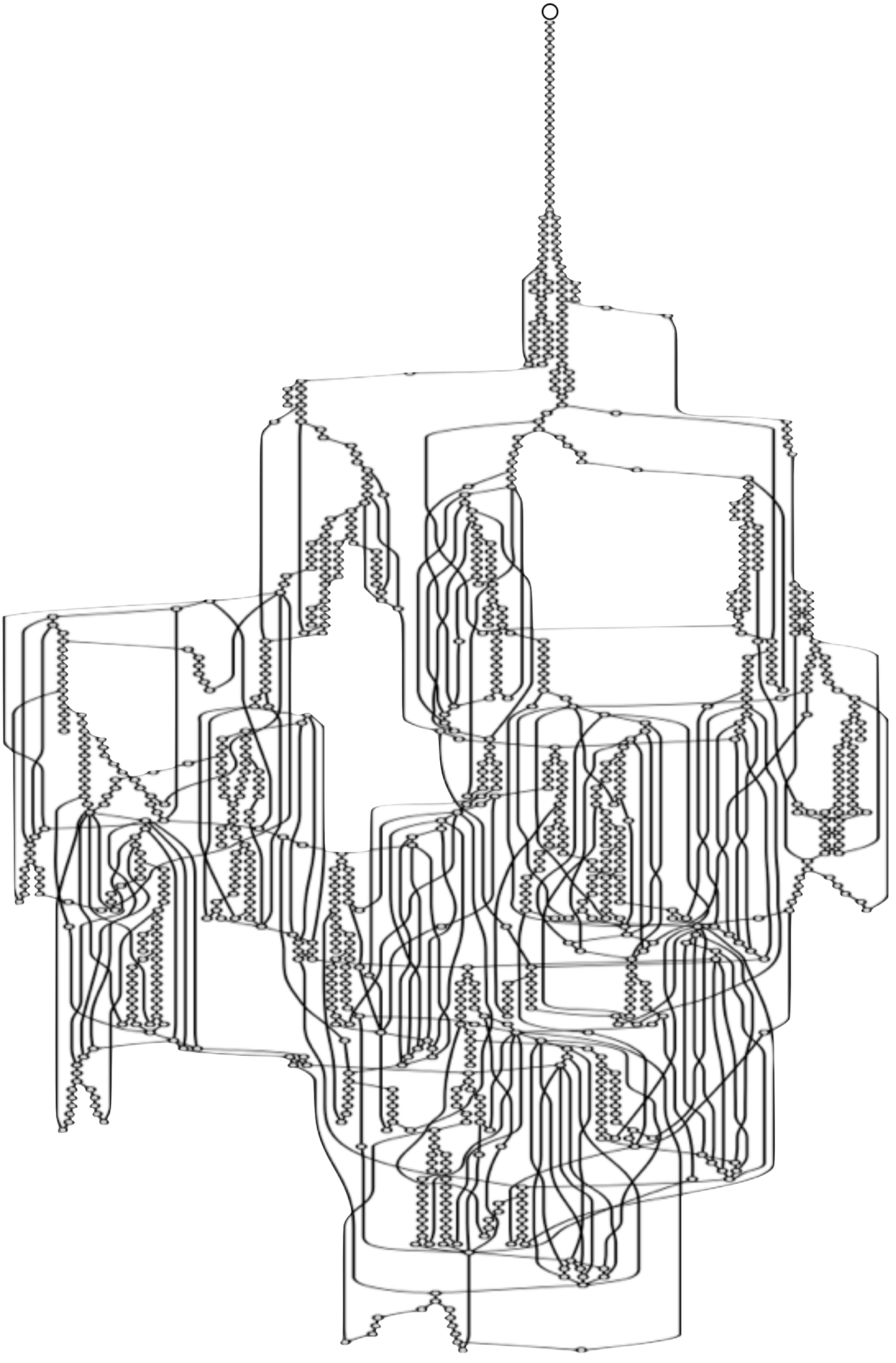
AAM

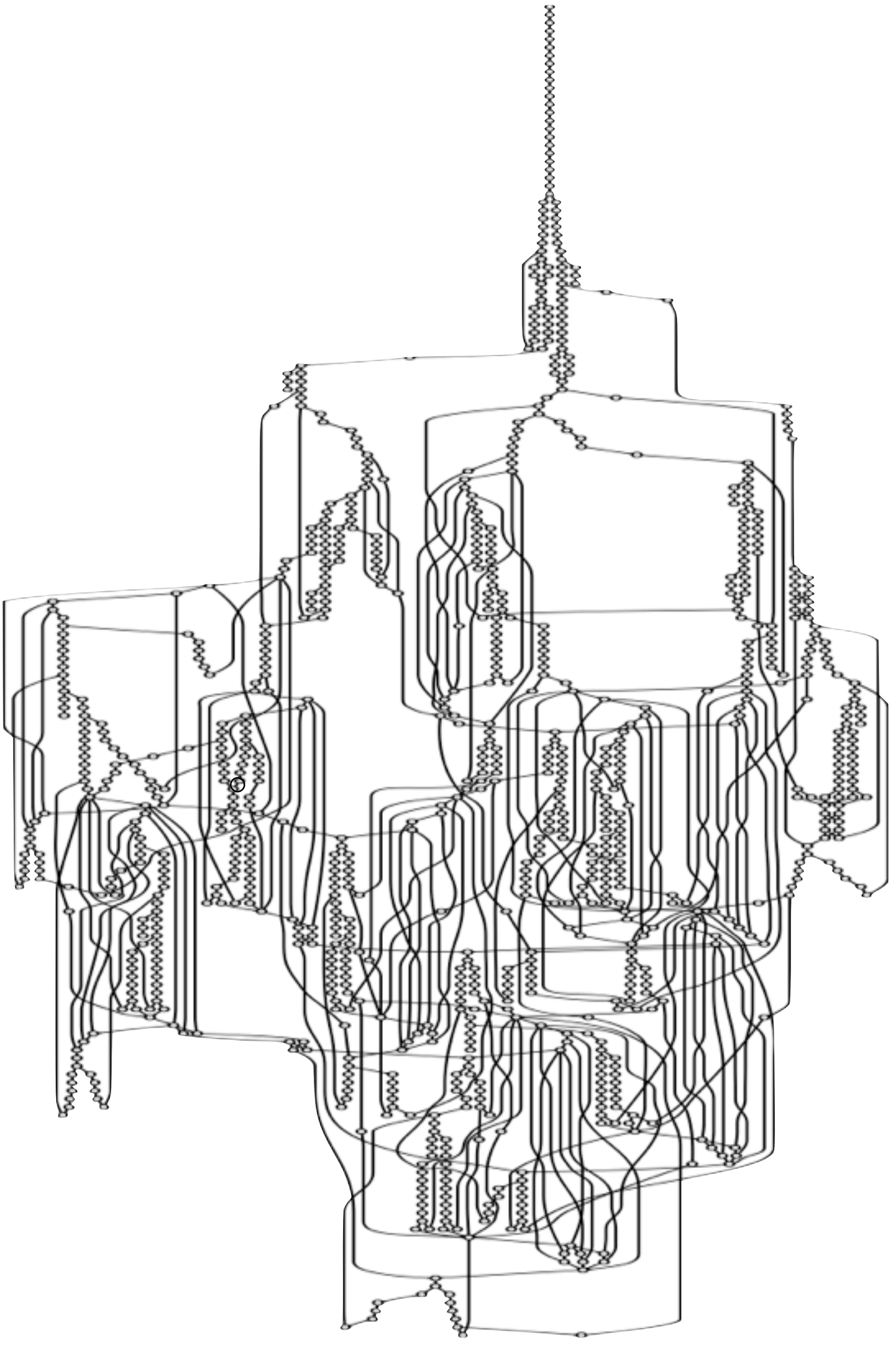
AAM

(Van Horn & Might, 2010, 2011, 2012)

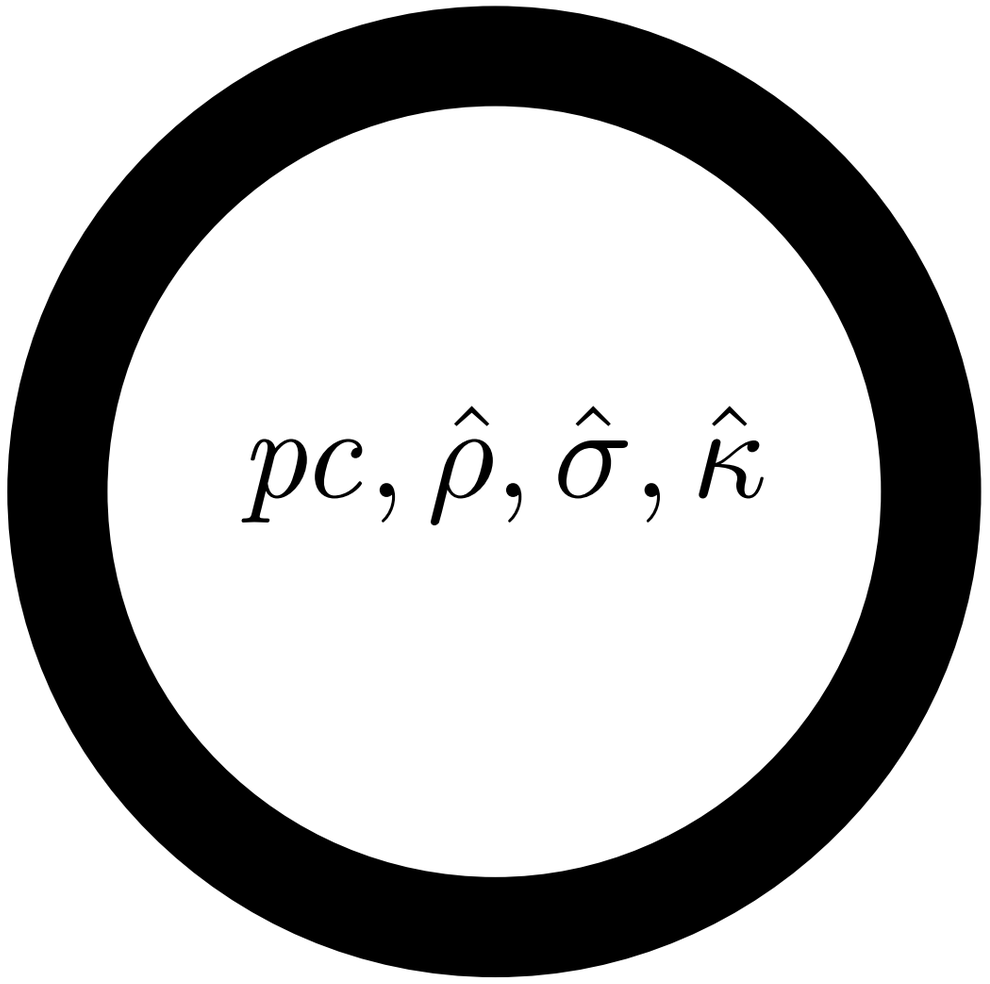
```
class MyActivity {

 public MyActivity() {
  activateMic();
 }


}
```

```
.class MyActivity

 .method public MyActivity
    invokedynamic activateMic
 .end method

.end class
```

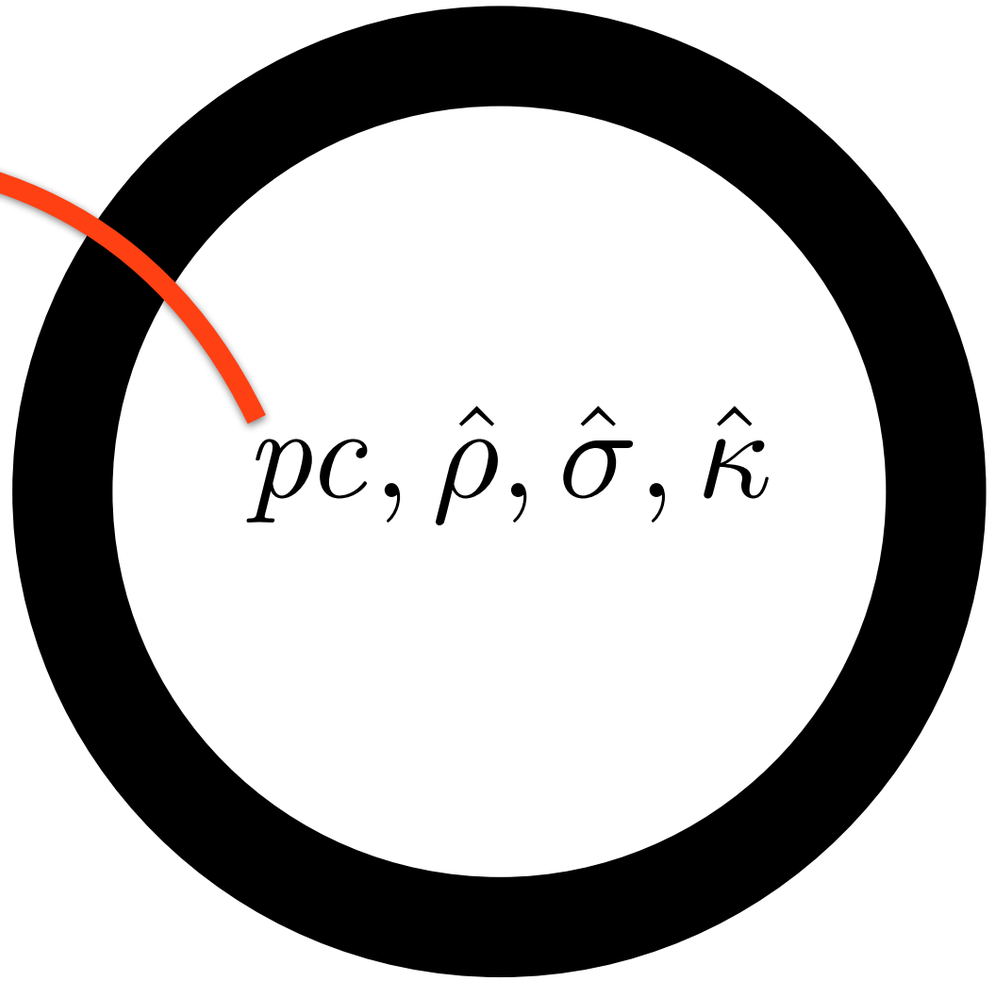




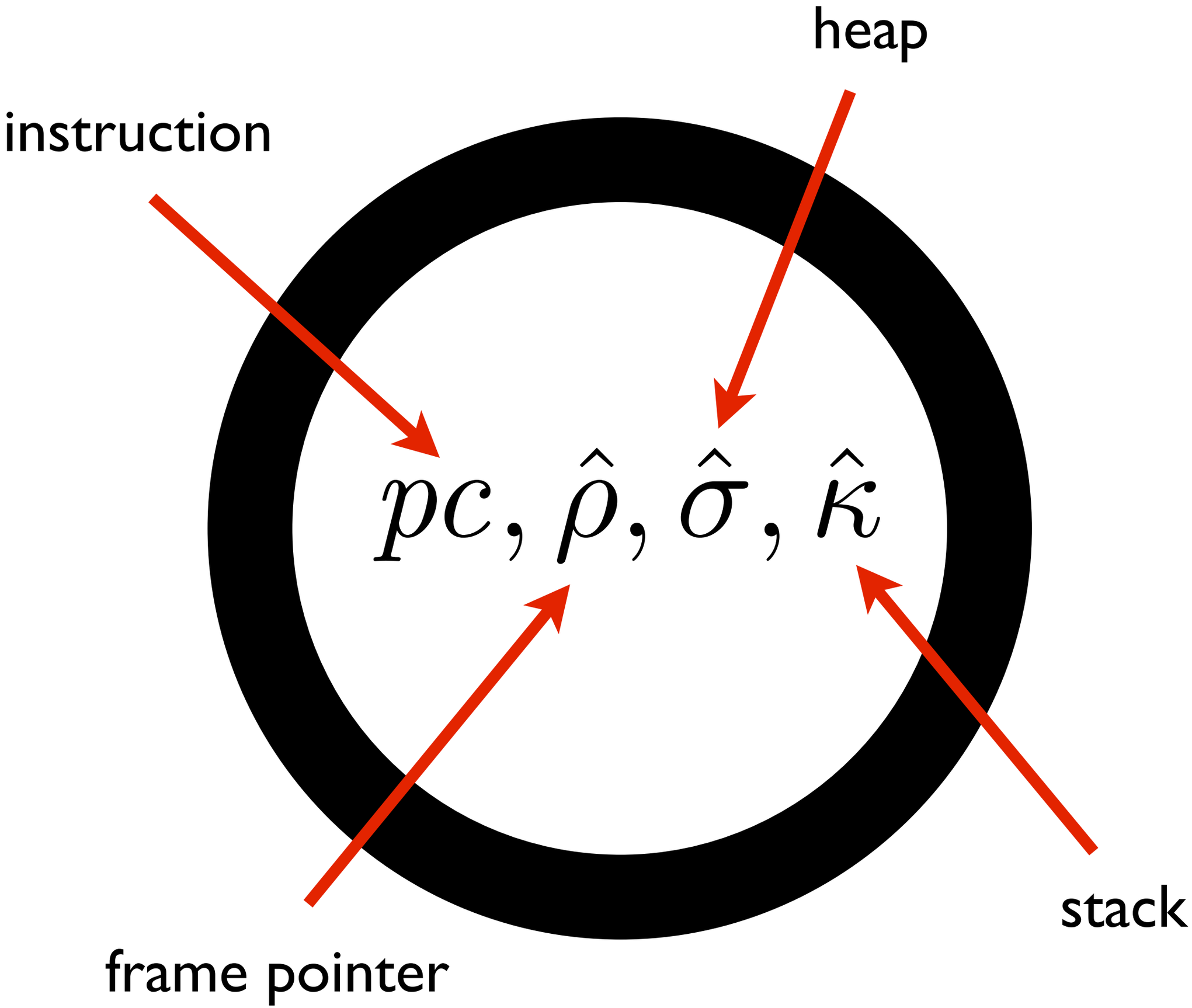
Look, a malware!


$pc, \hat{\rho}, \hat{\sigma}, \hat{\kappa}$

```
.class MyActivity

 .method public MyActivity
    invokedynamic activateMic
 .end method

.end class
```



$$pc, \hat{\rho}, \hat{\sigma}, \hat{\kappa}$$


heap
instruction
$pc, \hat{\rho}, \hat{\sigma}, \hat{\kappa}$
stack
frame pointer

Method:AAM

$$\langle \texttt{nop} :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma, \kappa\rangle$$

$$\langle \texttt{move-object}(r_d, r_s) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto \sigma(r_s, fp)], \kappa\rangle$$

$$\langle \texttt{return-void} :: \vec{stmt}', fp', \sigma, \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle \longmapsto \langle \vec{stmt}, fp, \sigma, \kappa\rangle$$

$$\langle \texttt{return-object}(r) :: \vec{stmt}', fp', \sigma, \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(\texttt{ret}, fp) \mapsto \sigma(n, fp')], \kappa\rangle$$

$$\langle \texttt{const}(r, c) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(r, fp) \mapsto c], \kappa\rangle$$

$$\langle \texttt{throw}^{\ell}(r) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \mathcal{S}(\ell'), fp', \sigma[(\texttt{exn}, fp') \mapsto \sigma(r, fp)], \kappa'\rangle$$
$$\text{where } (\ell', fp', \kappa') = \mathcal{H}(\ell, fp, \kappa)$$

$$\langle \texttt{goto}(\ell) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \mathcal{S}(\ell), fp, \sigma, \kappa\rangle$$

$$\langle \texttt{new-instance}(r, \tau) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(r, fp) \mapsto o], \kappa\rangle$$
$$\text{where } o = new(\varsigma)$$

$$\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \mathcal{S}(\ell), fp, \sigma, \kappa\rangle \text{ if } \sigma(r, fp) = \sigma(r', fp)$$

$$\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma, \kappa\rangle \text{ if } \sigma(r, fp) \neq \sigma(r', fp)$$

$$\langle \texttt{iget}(r_d, r_s, field) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto \sigma(a)], \kappa\rangle$$
$$\text{where } \sigma(r_s, fp) = o \text{ and } o.field = a$$

$$\langle \texttt{iput}(r_v, r_s, field) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[a \mapsto \sigma(r_v, fp)], \kappa\rangle$$
$$\text{where } \sigma(r_s, fp) = o \text{ and } o.field = a$$

$$\langle \texttt{invoke-direct}(r_0, \ldots, r_n, id) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \mathcal{M}(id), fp', \sigma', \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle$$
$$\text{where } \sigma' = \sigma[(0, fp') \mapsto \sigma(r_0, fp), \ldots, (n, fp') \mapsto \sigma(r_n, fp)]$$
$$fp' = alloc(\varsigma)$$

$$\langle \texttt{invoke-virtual}(r_0, \ldots, r_n, id) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \mathcal{V}(id, \sigma(r_0, fp)), fp', \sigma', \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle$$
$$\text{where } \sigma' = \sigma[(0, fp') \mapsto \sigma(r_0, fp), \ldots, (n, fp') \mapsto \sigma(r_n, fp)]$$
$$fp' = alloc(\varsigma)$$

$$\langle \mathit{unop}(r_d, r_s) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto v], \kappa\rangle$$
$$\text{where } v = \delta(\mathit{unop}, \sigma(r_s, fp))$$

$$\langle \mathit{binop}(r_d, r_1, r_2) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto v], \kappa\rangle$$
$$\text{where } v = \delta(\mathit{binop}, \sigma(r_1, fp), \sigma(r_2, fp))$$

$$
\begin{aligned}
\langle \texttt{nop} :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma, \kappa\rangle \\
\langle \texttt{move-object}(r_d, r_s) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto \sigma(r_s, fp)], \kappa\rangle \\
\langle \texttt{return-void} :: \vec{stmt}', fp', \sigma, \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma, \kappa\rangle \\
\langle \texttt{return-object}(r) :: \vec{stmt}', fp', \sigma, \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(\texttt{ret}, fp) \mapsto \sigma(n, fp')], \kappa\rangle \\
\langle \texttt{const}(r, c) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(r, fp) \mapsto c], \kappa\rangle \\
\langle \texttt{throw}^{\ell}(r) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \mathcal{S}(\ell'), fp', \sigma[(\texttt{exn}, fp') \mapsto \sigma(r, fp)], \kappa'\rangle \\
&\quad \text{where } (\ell', fp', \kappa') = \mathcal{H}(\ell, fp, \kappa) \\
\langle \texttt{goto}(\ell) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \mathcal{S}(\ell), fp, \sigma, \kappa\rangle \\
\langle \texttt{new-instance}(r, \tau) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(r, fp) \mapsto o], \kappa\rangle \\
&\quad \text{where } o = new(\varsigma) \\
\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \mathcal{S}(\ell), fp, \sigma, \kappa\rangle \text{ if } \sigma(r, fp) = \sigma(r', fp) \\
\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma, \kappa\rangle \text{ if } \sigma(r, fp) \neq \sigma(r', fp) \\
\langle \texttt{iget}(r_d, r_s, field) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto \sigma(a)], \kappa\rangle \\
&\quad \text{where } \sigma(r_s, fp) = o \text{ and } o.field = a \\
\langle \texttt{iput}(r_v, r_s, field) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[a \mapsto \sigma(r_v, fp)], \kappa\rangle \\
&\quad \text{where } \sigma(r_s, fp) = o \text{ and } o.field = a \\
\langle \texttt{invoke-direct}(r_0, \ldots, r_n, id) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \mathcal{M}(id), fp', \sigma', \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle \\
&\quad \text{where } \sigma' = \sigma[(0, fp') \mapsto \sigma(r_0, fp), \ldots, (n, fp') \mapsto \sigma(r_n, fp)] \\
&\qquad fp' = alloc(\varsigma) \\
\langle \texttt{invoke-virtual}(r_0, \ldots, r_n, id) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \mathcal{V}(id, \sigma(r_0, fp)), fp', \sigma', \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle \\
&\quad \text{where } \sigma' = \sigma[(0, fp') \mapsto \sigma(r_0, fp), \ldots, (n, fp') \mapsto \sigma(r_n, fp)] \\
&\qquad fp' = alloc(\varsigma) \\
\langle \mathit{unop}(r_d, r_s) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto v], \kappa\rangle \\
&\quad \text{where } v = \delta(\mathit{unop}, \sigma(r_s, fp)) \\
\langle \mathit{binop}(r_d, r_1, r_2) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto v], \kappa\rangle \\
&\quad \text{where } v = \delta(\mathit{binop}, \sigma(r_1, fp), \sigma(r_2, fp))
\end{aligned}
$$

$$
\begin{aligned}
\langle \texttt{nop} :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma, \kappa\rangle \\
\langle \texttt{move-object}(r_d, r_s) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto \sigma(r_s, fp)], \kappa\rangle \\
\langle \texttt{return-void} :: \vec{stmt}', fp', \sigma, \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma, \kappa\rangle \\
\langle \texttt{return-object}(r) :: \vec{stmt}', fp', \sigma, \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(\texttt{ret}, fp) \mapsto \sigma(n, fp')], \kappa\rangle \\
\langle \texttt{const}(r, c) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(r, fp) \mapsto c], \kappa\rangle \\
\langle \texttt{throw}^{\ell}(r) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \mathcal{S}(\ell'), fp', \sigma[(\texttt{exn}, fp') \mapsto \sigma(r, fp)], \kappa'\rangle \\
&\quad \text{where } (\ell', fp', \kappa') = \mathcal{H}(\ell, fp, \kappa) \\
\langle \texttt{goto}(\ell) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \mathcal{S}(\ell), fp, \sigma, \kappa\rangle \\
\langle \texttt{new-instance}(r, \tau) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(r, fp) \mapsto o], \kappa\rangle \\
&\quad \text{where } o = new(\varsigma) \\
\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \mathcal{S}(\ell), fp, \sigma, \kappa\rangle \text{ if } \sigma(r, fp) = \sigma(r', fp) \\
\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma, \kappa\rangle \text{ if } \sigma(r, fp) \neq \sigma(r', fp) \\
\langle \texttt{iget}(r_d, r_s, field) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto \sigma(a)], \kappa\rangle \\
&\quad \text{where } \sigma(r_s, fp) = o \text{ and } o.field = a \\
\langle \texttt{iput}(r_v, r_s, field) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[a \mapsto \sigma(r_v, fp)], \kappa\rangle \\
&\quad \text{where } \sigma(r_s, fp) = o \text{ and } o.field = a \\
\langle \texttt{invoke-direct}(r_0, \ldots, r_n, id) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \mathcal{M}(id), fp', \sigma', \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle \\
&\quad \text{where } \sigma' = \sigma[(0, fp') \mapsto \sigma(r_0, fp), \ldots, (n, fp') \mapsto \sigma(r_n, fp)] \\
&\quad\quad fp' = alloc(\varsigma) \\
\langle \texttt{invoke-virtual}(r_0, \ldots, r_n, id) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \mathcal{V}(id, \sigma(r_0, fp)), fp', \sigma', \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle \\
&\quad \text{where } \sigma' = \sigma[(0, fp') \mapsto \sigma(r_0, fp), \ldots, (n, fp') \mapsto \sigma(r_n, fp)] \\
&\quad\quad fp' = alloc(\varsigma) \\
\langle unop(r_d, r_s) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto v], \kappa\rangle \\
&\quad \text{where } v = \delta(unop, \sigma(r_s, fp)) \\
\langle binop(r_d, r_1, r_2) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto v], \kappa\rangle \\
&\quad \text{where } v = \delta(binop, \sigma(r_1, fp), \sigma(r_2, fp))
\end{aligned}
$$

(Van Horn & Might, 2012)

$$
\begin{aligned}
\langle \texttt{nop} :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \\
\langle \texttt{move-object}(r_d, r_s) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r_d, \hat{fp}) \mapsto \hat{\sigma}(r_s, \hat{fp})], \hat{\kappa}, \hat{t}\rangle \\
\langle \texttt{return-void} :: \vec{stmt}', \hat{fp}', \hat{\sigma}, \mathbf{fnk}(\vec{stmt}, \hat{fp}, \hat{a}_\kappa)\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}\rangle \text{ if } \hat{\kappa} \in \hat{\sigma}(\hat{a}_\kappa) \\
\langle \texttt{return-object}(r) :: \vec{stmt}', \hat{fp}', \hat{\sigma}, \mathbf{fnk}(\vec{stmt}, \hat{fp}, \hat{a}_\kappa)\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(\texttt{ret}, \hat{fp}) \mapsto \hat{\sigma}(n, \hat{fp}')], \hat{\kappa}\rangle \text{ if } \hat{\kappa} \in \hat{\sigma}(\hat{a}_\kappa) \\
\langle \texttt{const}(r, c) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r, \hat{fp}) \mapsto c], \hat{\kappa}, \hat{t}\rangle \\
\langle \texttt{throw}^{\ell}(r) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}\rangle &\longmapsto \langle \mathcal{S}(\ell'), \hat{fp}', \hat{\sigma} \sqcup [(\texttt{exn}, \hat{fp}') \mapsto \hat{\sigma}(r, \hat{fp})], \hat{\kappa}'\rangle \\
&\quad \text{where } (\ell', \hat{fp}', \hat{\kappa}') \in \hat{\mathcal{H}}_{\hat{\sigma}}(\ell, \hat{fp}, \hat{\kappa}) \\
\langle \texttt{goto}(\ell) :: \vec{stmt}', \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \mathcal{S}(\ell), \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \\
\langle \texttt{new-instance}(r, \tau) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r, \hat{fp}) \mapsto o], \hat{\kappa}, \hat{t}\rangle \\
&\quad \text{where } o = \widehat{new}(\varsigma) \\
\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \mathcal{S}(\ell), \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \\
&\quad \text{if } \exists v_1 \in \hat{\sigma}(r, \hat{fp}), \exists v_2 \in \hat{\sigma}(r', \hat{fp}).v_1 = v_2 \\
\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \\
&\quad \text{if } \exists v_1 \in \hat{\sigma}(r, \hat{fp}), \exists v_2 \in \hat{\sigma}(r', \hat{fp}).v_1 \neq v_2 \\
\langle \texttt{iget}(r_d, r_s, field) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r_d, \hat{fp}) \mapsto \hat{\sigma}(a)], \hat{\kappa}, \hat{t}\rangle \\
&\quad \text{where } \hat{\sigma}(r_s, \hat{fp}) \ni o \text{ and } o.field = a \\
\langle \texttt{iput}(r_v, r_s, field) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [a \mapsto \hat{\sigma}(r_v, \hat{fp})], \hat{\kappa}, \hat{t}\rangle \\
&\quad \text{where } \hat{\sigma}(r_s, \hat{fp}) \ni o \text{ and } o.field = a \\
\langle \texttt{invoke-direct}(r_0, \ldots, r_n, id) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \mathcal{M}(id), \hat{fp}', \hat{\sigma}'', \mathbf{fnk}(\vec{stmt}, \hat{fp}, \hat{a}_\kappa), \hat{t}'\rangle \\
&\quad \text{where } \hat{\sigma}'' = \hat{\sigma}' \sqcup [(0, \hat{fp}') \mapsto \sigma(r_0, \hat{fp}), \ldots, (n, \hat{fp}') \mapsto \sigma(r_n, \hat{fp})] \\
&\quad\quad \hat{\sigma}' = \hat{\sigma} \sqcup [\hat{a}_\kappa \mapsto \hat{\kappa}] \\
&\quad\quad \hat{fp}' = \widehat{alloc}(\hat{\varsigma}) \\
&\quad\quad \hat{a}_\kappa = \widehat{allock}(\hat{\varsigma}) \\
&\quad\quad \hat{t}' = \widehat{tick}(\hat{t}) \\
\langle \texttt{invoke-virtual}(r_0, \ldots, r_n, id) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \mathcal{V}(id, v), \hat{fp}', \hat{\sigma}'', \mathbf{fnk}(\vec{stmt}, \hat{fp}, \hat{\kappa}), \hat{t}'\rangle \text{ if } v \in \hat{\sigma}(r_0, \hat{fp}) \\
&\quad \text{where } \hat{\sigma}'' = \hat{\sigma}' \sqcup [(0, \hat{fp}') \mapsto \hat{\sigma}(r_0, \hat{fp}), \ldots, (n, \hat{fp}') \mapsto \sigma(r_n, \hat{fp})] \\
&\quad\quad \hat{\sigma}' = \hat{\sigma} \sqcup [\hat{a}_\kappa \mapsto \hat{\kappa}] \\
&\quad\quad \hat{fp}' = \widehat{alloc}(\hat{\varsigma}) \\
&\quad\quad \hat{a}_\kappa = \widehat{allock}(\hat{\varsigma}) \\
&\quad\quad \hat{t}' = \widehat{tick}(\hat{t}) \\
\langle unop(r_d, r_s) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r_d, \hat{fp}) \mapsto v], \hat{\kappa}\rangle \\
&\quad \text{where } v \in \hat{\delta}(unop, \sigma(r_s, fp)) \\
\langle binop(r_d, r_1, r_2) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r_d, \hat{fp}) \mapsto v], \hat{\kappa}\rangle \\
&\quad \text{where } v \in \hat{\delta}(binop, \hat{\sigma}(r_1, \hat{fp}), \hat{\sigma}(r_2, \hat{fp}))
\end{aligned}
$$

$$
\begin{aligned}
\langle \texttt{nop} :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \\
\langle \texttt{move-object}(r_d, r_s) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r_d, \hat{fp}) \mapsto \hat{\sigma}(r_s, \hat{fp})], \hat{\kappa}, \hat{t}\rangle \\
\langle \texttt{return-void} :: \vec{stmt}', \hat{fp}', \hat{\sigma}, \mathbf{fnk}(\vec{stmt}, \hat{fp}, \hat{a}_\kappa)\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}\rangle \text{ if } \hat{\kappa} \in \hat{\sigma}(\hat{a}_\kappa) \\
\langle \texttt{return-object}(r) :: \vec{stmt}', \hat{fp}', \hat{\sigma}, \mathbf{fnk}(\vec{stmt}, \hat{fp}, \hat{a}_\kappa)\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(\texttt{ret}, \hat{fp}) \mapsto \hat{\sigma}(n, \hat{fp}')], \hat{\kappa}\rangle \text{ if } \hat{\kappa} \in \hat{\sigma}(\hat{a}_\kappa) \\
\langle \texttt{const}(r, c) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r, \hat{fp}) \mapsto c], \hat{\kappa}, \hat{t}\rangle \\
\langle \texttt{throw}^\ell(r) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}\rangle &\longmapsto \langle \mathcal{S}(\ell'), fp', \hat{\sigma} \sqcup [(\texttt{exn}, \hat{fp}') \mapsto \hat{\sigma}(r, \hat{fp})], \hat{\kappa}'\rangle \\
&\quad \text{where } (\ell', \hat{fp}', \hat{\kappa}') \in \widehat{\mathcal{H}}_{\hat{\sigma}}(\ell, \hat{fp}, \hat{\kappa}) \\
\langle \texttt{goto}(\ell) :: \vec{stmt}', \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \mathcal{S}(\ell), \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \\
\langle \texttt{new-instance}(r, \tau) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r, \hat{fp}) \mapsto o], \hat{\kappa}, \hat{t}\rangle \\
&\quad \text{where } o = \widehat{new}(\varsigma) \\
\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \mathcal{S}(\ell), \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \\
&\quad \text{if } \exists v_1 \in \hat{\sigma}(r, \hat{fp}), \exists v_2 \in \hat{\sigma}(r', \hat{fp}).v_1 = v_2 \\
\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \\
&\quad \text{if } \exists v_1 \in \hat{\sigma}(r, \hat{fp}), \exists v_2 \in \hat{\sigma}(r', \hat{fp}).v_1 \neq v_2 \\
\langle \texttt{iget}(r_d, r_s, field) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r_d, \hat{fp}) \mapsto \hat{\sigma}(a)], \hat{\kappa}, \hat{t}\rangle \\
&\quad \text{where } \hat{\sigma}(r_s, fp) \ni o \text{ and } o.field = a \\
\langle \texttt{iput}(r_v, r_s, field) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [a \mapsto \hat{\sigma}(r_v, \hat{fp})], \hat{\kappa}, \hat{t}\rangle \\
&\quad \text{where } \hat{\sigma}(r_s, \hat{fp}) \ni o \text{ and } o.field = a \\
\langle \texttt{invoke-direct}(r_0, \ldots, r_n, id) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \mathcal{M}(id), \hat{fp}', \hat{\sigma}'', \mathbf{fnk}(\vec{stmt}, \hat{fp}, \hat{a}_\kappa), \hat{t}'\rangle \\
&\quad \text{where } \hat{\sigma}'' = \hat{\sigma}' \sqcup [(0, \hat{fp}') \mapsto \sigma(r_0, \hat{fp}), \ldots, (n, \hat{fp}') \mapsto \sigma(r_n, \hat{fp})] \\
&\qquad \hat{\sigma}' = \hat{\sigma} \sqcup [\hat{a}_\kappa \mapsto \hat{\kappa}] \\
&\qquad \hat{fp}' = \widehat{alloc}(\hat{\varsigma}) \\
&\qquad \hat{a}_\kappa = \widehat{allock}(\hat{\varsigma}) \\
&\qquad \hat{t}' = \widehat{tick}(\hat{t}) \\
\langle \texttt{invoke-virtual}(r_0, \ldots, r_n, id) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle &\longmapsto \langle \mathcal{V}(id, v), \hat{fp}', \hat{\sigma}'', \mathbf{fnk}(\vec{stmt}, \hat{fp}, \hat{\kappa}), \hat{t}'\rangle \text{ if } v \in \hat{\sigma}(r_0, \hat{fp}) \\
&\quad \text{where } \hat{\sigma}'' = \hat{\sigma}' \sqcup [(0, fp') \mapsto \hat{\sigma}(r_0, \hat{fp}), \ldots, (n, \hat{fp}') \mapsto \sigma(r_n, \hat{fp})] \\
&\qquad \hat{\sigma}' = \hat{\sigma} \sqcup [\hat{a}_\kappa \mapsto \hat{\kappa}] \\
&\qquad \hat{fp}' = \widehat{alloc}(\hat{\varsigma}) \\
&\qquad \hat{a}_\kappa = \widehat{allock}(\hat{\varsigma}) \\
&\qquad \hat{t}' = \widehat{tick}(\hat{t}) \\
\langle unop(r_d, r_s) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r_d, \hat{fp}) \mapsto v], \hat{\kappa}\rangle \\
&\quad \text{where } v \in \hat{\delta}(unop, \sigma(r_s, fp)) \\
\langle binop(r_d, r_1, r_2) :: \vec{stmt}, fp, \sigma, \kappa\rangle &\longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r_d, \hat{fp}) \mapsto v], \hat{\kappa}\rangle \\
&\quad \text{where } v \in \hat{\delta}(binop, \hat{\sigma}(r_1, \hat{fp}), \hat{\sigma}(r_2, \hat{fp}))
\end{aligned}
$$

6 months

```
nop

move vx,vy
move/from16 vx,vy
move/16
move-wide
move-wide/from16 vx,vy
move-wide/16
move-object vx,vy
move-object/from16 vx,vy
move-object/16
move-result vx
move-result-wide vx
move-result-object vx
move-exception vx

return-void
return vx
return-wide vx
return-object vx

const/4 vx,lit4
const/16 vx,lit16
const vx, lit32
const/high16 v0, lit16
const-wide/16 vx, lit16
const-wide/32 vx, lit32
const-wide vx, lit64
const-wide/high16 vx,lit16
const-string vx,string_id
const-string-jumbo
const-class vx,type_id

monitor-enter vx
monitor-exit

check-cast vx, type_id

instance-of vx,vy,type_id

array-length vx,vy

new-instance vx,type
new-array vx,vy,type_id
filled-new-array {parameters},type_id
filled-new-array-range {vx..vy},type_id
fill-array-data vx,array_data_offset

throw vx

goto target
goto/16 target
goto/32 target

packed-switch vx,table
sparse-switch vx,table

cmpl-float
cmpg-float vx, vy, vz
cmpl-double vx,vy,vz
cmpg-double vx, vy, vz
cmp-long vx, vy, vz

if-eq vx,vy,target
if-ne vx,vy,target
if-lt vx,vy,target
if-ge vx, vy,target
if-gt vx,vy,target
if-le vx,vy,target
if-eqz vx,target
if-nez vx,target
if-ltz vx,target
if-gez vx,target
if-gtz vx,target
if-lez vx,target

aget vx,vy,vz
aget-wide vx,vy,vz
aget-object vx,vy,vz
aget-boolean vx,vy,vz
aget-byte vx,vy,vz
aget-char vx, vy,vz
aget-short vx,vy,vz

aput vx,vy,vz
aput-wide vx,vy,vz
aput-object vx,vy,vz
aput-boolean vx,vy,vz
aput-byte vx,vy,vz
aput-char vx,vy,vz
aput-short vx,vy,vz

iget vx, vy, field_id
iget-wide vx,vy,field_id
iget-object vx,vy,field_id
iget-boolean vx,vy,field_id
iget-byte vx,vy,field_id
iget-char vx,vy,field_id
iget-short vx,vy,field_id

iput vx,vy, field_id
iput-wide vx,vy, field_id
iput-object vx,vy,field_id
iput-boolean vx,vy, field_id
iput-byte vx,vy,field_id
iput-char vx,vy,field_id
iput-short vx,vy,field_id

sget vx,field_id
sget-wide vx, field_id
sget-object vx,field_id
sget-boolean vx,field_id
sget-byte vx,field_id
sget-char vx,field_id
sget-short vx,field_id

sput vx, field_id
sput-wide vx, field_id
sput-object vx,field_id
sput-boolean vx,field_id
sput-byte vx,field_id
sput-char vx,field_id
sput-short vx,field_id

invoke-virtual { parameters },methodtocall
invoke-super {parameter},methodtocall
invoke-direct { parameters },methodtocall
invoke-static {parameters},methodtocall
invoke-interface {parameters},methodtocall

invoke-virtual/range {vx..vy},methodtocall
invoke-super/range {vx..vy},methodtocall
invoke-direct/range {vx..vy},methodtocall
invoke-static/range {vx..vy},methodtocall
invoke-interface/range {vx..vy},methodtocall

neg-int vx,vy
not-int vx,vy
neg-long vx,vy
not-long vx,vy
neg-float vx,vy
neg-double vx,vy

int-to-long vx, vy
int-to-float vx, vy
int-to-double vx, vy
long-to-int vx,vy
long-to-float vx, vy
long-to-double vx, vy
float-to-int vx, vy
float-to-long vx,vy
float-to-double vx, vy
double-to-int vx, vy
double-to-long vx, vy
double-to-float vx, vy
int-to-byte vx,vy
int-to-char vx,vy
int-to-short vx,vy

add-int vx,vy,vz
sub-int vx,vy,vz
mul-int vx, vy, vz
div-int vx,vy,vz
rem-int vx,vy,vz

and-int vx, vy, vz
or-int vx, vy, vz
xor-int vx, vy, vz
shl-int vx, vy, vz
shr-int vx, vy, vz
ushr-int vx, vy, vz
add-long vx, vy, vz
sub-long vx,vy,vz
mul-long vx,vy,vz
div-long vx, vy, vz
rem-long vx,vy,vz
and-long vx, vy, vz
or-long vx, vy, vz
xor-long vx, vy, vz
shl-long vx, vy, vz
shr-long vx,vy,vz
ushr-long vx, vy, vz
add-float vx,vy,vz
sub-float vx,vy,vz
mul-float vx, vy, vz
div-float vx, vy, vz
rem-float vx,vy,vz
add-double vx,vy,vz
sub-double vx,vy,vz
mul-double vx, vy, vz
div-double vx, vy, vz
rem-double vx,vy,vz
add-int/2addr vx,vy
sub-int/2addr vx,vy
mul-int/2addr vx,vy
div-int/2addr vx,vy
rem-int/2addr vx,vy
and-int/2addr vx, vy
or-int/2addr vx, vy
xor-int/2addr vx, vy
shl-int/2addr vx, vy
shr-int/2addr vx, vy
ushr-int/2addr vx, vy
add-long/2addr vx,vy
sub-long/2addr vx,vy
mul-long/2addr vx,vy
div-long/2addr vx, vy
rem-long/2addr vx,vy
and-long/2addr vx, vy
or-long/2addr vx, vy
xor-long/2addr vx, vy
shl-long/2addr vx, vy
shr-long/2addr vx, vy
ushr-long/2addr vx, vy
add-float/2addr vx,vy
sub-float/2addr vx,vy
mul-float/2addr vx, vy
div-float/2addr vx, vy
rem-float/2addr vx,vy
add-double/2addr vx, vy
sub-double/2addr vx, vy
mul-double/2addr vx, vy
div-double/2addr vx, vy
rem-double/2addr vx,vy
add-int/lit16 vx,vy,lit16
sub-int/lit16 vx,vy,lit16
mul-int/lit16 vx,vy,lit16
div-int/lit16 vx,vy,lit16
rem-int/lit16 vx,vy,lit16
and-int/lit16 vx,vy,lit16
or-int/lit16 vx,vy,lit16
xor-int/lit16 vx,vy,lit16
add-int/lit8 vx,vy,lit8
sub-int/lit8 vx,vy,lit8
mul-int/lit8 vx,vy,lit8
div-int/lit8 vx,vy,lit8
rem-int/lit8 vx,vy,lit8
and-int/lit8 vx,vy,lit8
or-int/lit8 vx, vy, lit8
xor-int/lit8 vx, vy, lit8
shl-int/lit8 vx, vy, lit8
shr-int/lit8 vx, vy, lit8
ushr-int/lit8 vx, vy, lit8
```

```
nop
```

```
move vx,vy
move/from16 vx,vy
move/16
move-wide
move-wide/from16 vx,vy
move-wide/16
move-object vx,vy
move-object/from16 vx,vy
move-object/16
move-result vx
move-result-wide vx
move-result-object vx
move-exception vx
```

```
return-void
return vx
return-wide vx
return-object vx
```

```
const/4 vx,lit4
const/16 vx,lit16
const vx, lit32
const/high16 v0, lit16
const-wide/16 vx, lit16
const-wide/32 vx, lit32
const-wide vx, lit64
const-wide/high16 vx,lit16
const-string vx,string_id
const-string-jumbo
const-class vx,type_id
```

```
monitor-enter vx
monitor-exit
```

```
check-cast vx, type_id
```

```
instance-of vx,vy,type_id
```

```
array-length vx,vy
```

```
new-instance vx,type
new-array vx,vy,type_id
filled-new-array {parameters},type_id
filled-new-array-range {vx..vy},type_id
fill-array-data vx,array_data_offset
```

```
throw vx
```

```
goto target
goto/16 target
goto/32 target
```

```
packed-switch vx,table
sparse-switch vx,table
```

```
cmpl-float
cmpg-float vx, vy, vz
cmpl-double vx,vy,vz
cmpg-double vx, vy, vz
cmp-long vx, vy, vz
```

```
if-eq vx,vy,target
if-ne vx,vy,target
if-lt vx,vy,target
if-ge vx, vy,target
if-gt vx,vy,target
if-le vx,vy,target
if-eqz vx,target
if-nez vx,target
if-ltz vx,target
if-gez vx,target
if-gtz vx,target
if-lez vx,target
```

```
aget vx,vy,vz
aget-wide vx,vy,vz
aget-object vx,vy,vz
aget-boolean vx,vy,vz
aget-byte vx,vy,vz
aget-char vx, vy,vz
aget-short vx,vy,vz
```

```
aput vx,vy,vz
aput-wide vx,vy,vz
aput-object vx,vy,vz
aput-boolean vx,vy,vz
aput-byte vx,vy,vz
aput-char vx,vy,vz
aput-short vx,vy,vz
```

```
iget vx, vy, field_id
iget-wide vx,vy,field_id
iget-object vx,vy,field_id
iget-boolean vx,vy,field_id
iget-byte vx,vy,field_id
iget-char vx,vy,field_id
iget-short vx,vy,field_id
```

```
iput vx,vy, field_id
iput-wide vx,vy, field_id
iput-object vx,vy,field_id
iput-boolean vx,vy, field_id
iput-byte vx,vy,field_id
iput-char vx,vy,field_id
iput-short vx,vy,field_id
```

```
sget vx,field_id
sget-wide vx, field_id
sget-object vx,field_id
sget-boolean vx,field_id
sget-byte vx,field_id
sget-char vx,field_id
sget-short vx,field_id
```

```
sput vx, field_id
sput-wide vx, field_id
sput-object vx,field_id
sput-boolean vx,field_id
sput-byte vx,field_id
sput-char vx,field_id
sput-short vx,field_id
```

```
invoke-virtual { parameters },methodtocall
invoke-super {parameter},methodtocall
invoke-direct { parameters },methodtocall
invoke-static {parameters},methodtocall
invoke-interface {parameters},methodtocall
```

```
invoke-virtual/range {vx..vy},methodtocall
invoke-super/range {vx..vy},methodtocall
invoke-direct/range {vx..vy},methodtocall
invoke-static/range {vx..vy},methodtocall
invoke-interface/range {vx..vy},methodtocall
```

```
neg-int vx,vy
not-int vx,vy
neg-long vx,vy
not-long vx,vy
neg-float vx,vy
neg-double vx,vy
```

```
int-to-long vx, vy
int-to-float vx, vy
int-to-double vx, vy
long-to-int vx,vy
long-to-float vx, vy
long-to-double vx, vy
float-to-int vx, vy
float-to-long vx,vy
float-to-double vx, vy
double-to-int vx, vy
double-to-long vx, vy
double-to-float vx, vy
int-to-byte vx,vy
int-to-char vx,vy
int-to-short vx,vy
```

```
add-int vx,vy,vz
sub-int vx,vy,vz
mul-int vx, vy, vz
div-int vx,vy,vz
rem-int vx,vy,vz
```

```
and-int vx, vy, vz
or-int vx, vy, vz
xor-int vx, vy, vz
shl-int vx, vy, vz
shr-int vx, vy, vz
ushr-int vx, vy, vz
add-long vx, vy, vz
sub-long vx,vy,vz
mul-long vx,vy,vz
div-long vx, vy, vz
rem-long vx,vy,vz
and-long vx, vy, vz
or-long vx, vy, vz
xor-long vx, vy, vz
shl-long vx, vy, vz
shr-long vx,vy,vz
ushr-long vx, vy, vz
add-float vx,vy,vz
sub-float vx,vy,vz
mul-float vx, vy, vz
div-float vx, vy, vz
rem-float vx,vy,vz
add-double vx,vy,vz
sub-double vx,vy,vz
mul-double vx, vy, vz
div-double vx, vy, vz
rem-double vx,vy,vz
add-int/2addr vx,vy
sub-int/2addr vx,vy
mul-int/2addr vx,vy
div-int/2addr vx,vy
rem-int/2addr vx,vy
and-int/2addr vx, vy
or-int/2addr vx, vy
xor-int/2addr vx, vy
shl-int/2addr vx, vy
shr-int/2addr vx, vy
ushr-int/2addr vx, vy
add-long/2addr vx,vy
sub-long/2addr vx,vy
mul-long/2addr vx,vy
div-long/2addr vx, vy
rem-long/2addr vx,vy
and-long/2addr vx, vy
or-long/2addr vx, vy
xor-long/2addr vx, vy
shl-long/2addr vx, vy
shr-long/2addr vx, vy
ushr-long/2addr vx, vy
add-float/2addr vx,vy
sub-float/2addr vx,vy
mul-float/2addr vx, vy
div-float/2addr vx, vy
rem-float/2addr vx,vy
add-double/2addr vx, vy
sub-double/2addr vx, vy
mul-double/2addr vx, vy
div-double/2addr vx, vy
rem-double/2addr vx,vy
add-int/lit16 vx,vy,lit16
sub-int/lit16 vx,vy,lit16
mul-int/lit16 vx,vy,lit16
div-int/lit16 vx,vy,lit16
rem-int/lit16 vx,vy,lit16
and-int/lit16 vx,vy,lit16
or-int/lit16 vx,vy,lit16
xor-int/lit16 vx,vy,lit16
add-int/lit8 vx,vy,lit8
sub-int/lit8 vx,vy,lit8
mul-int/lit8 vx,vy,lit8
div-int/lit8 vx,vy,lit8
rem-int/lit8 vx,vy,lit8
and-int/lit8 vx,vy,lit8
or-int/lit8 vx, vy, lit8
xor-int/lit8 vx, vy, lit8
shl-int/lit8 vx, vy, lit8
shr-int/lit8 vx, vy, lit8
ushr-int/lit8 vx, vy, lit8
```

```
nop
```

```
move vx,vy
move/from16 vx,vy
move/16
move-wide
move-wide/from16 vx,vy
move-wide/16
move-object vx,vy
move-object/from16 vx,vy
move-object/16
move-result vx
move-result-wide vx
move-result-object vx
move-exception vx
```

```
return-void
return vx
return-wide vx
return-object vx
```

```
const/4 vx,lit4
const/16 vx,lit16
const vx, lit32
const/high16 v0, lit16
const-wide/16 vx, lit16
const-wide/32 vx, lit32
const-wide vx, lit64
const-wide/high16 vx,lit16
const-string vx,string_id
const-string-jumbo
const-class vx,type_id
```

```
monitor-enter vx
monitor-exit
```

```
check-cast vx, type_id
```

```
instance-of vx,vy,type_id
```

```
array-length vx,vy
```

```
new-instance vx,type
new-array vx,vy,type_id
filled-new-array {parameters},type_id
filled-new-array-range {vx..vy},type_id
fill-array-data vx,array_data_offset
```

```
throw vx
```

```
goto target
goto/16 target
goto/32 target
```

```
packed-switch vx,table
sparse-switch vx,table
```

```
cmpl-float
cmpg-float vx, vy, vz
cmpl-double vx,vy,vz
cmpg-double vx, vy, vz
cmp-long vx, vy, vz
```

```
if-eq vx,vy,target
if-ne vx,vy,target
if-lt vx,vy,target
if-ge vx, vy,target
if-gt vx,vy,target
if-le vx,vy,target
if-eqz vx,target
if-nez vx,target
if-ltz vx,target
if-gez vx,target
if-gtz vx,target
if-lez vx,target
```

```
aget vx,vy,vz
aget-wide vx,vy,vz
aget-object vx,vy,vz
aget-boolean vx,vy,vz
aget-byte vx,vy,vz
aget-char vx, vy,vz
aget-short vx,vy,vz
```

```
aput vx,vy,vz
aput-wide vx,vy,vz
aput-object vx,vy,vz
aput-boolean vx,vy,vz
aput-byte vx,vy,vz
aput-char vx,vy,vz
aput-short vx,vy,vz
```

```
iget vx, vy, field_id
iget-wide vx,vy,field_id
iget-object vx,vy,field_id
iget-boolean vx,vy,field_id
iget-byte vx,vy,field_id
iget-char vx,vy,field_id
iget-short vx,vy,field_id
```

```
iput vx,vy, field_id
iput-wide vx,vy, field_id
iput-object vx,vy,field_id
iput-boolean vx,vy, field_id
iput-byte vx,vy,field_id
iput-char vx,vy,field_id
iput-short vx,vy,field_id
```

```
sget vx,field_id
sget-wide vx, field_id
sget-object vx,field_id
sget-boolean vx,field_id
sget-byte vx,field_id
sget-char vx,field_id
sget-short vx,field_id
```

```
sput vx, field_id
sput-wide vx, field_id
sput-object vx,field_id
sput-boolean vx,field_id
sput-byte vx,field_id
sput-char vx,field_id
sput-short vx,field_id
```

```
invoke-virtual { parameters },methodtocall
invoke-super {parameter},methodtocall
invoke-direct { parameters },methodtocall
invoke-static {parameters},methodtocall
invoke-interface {parameters},methodtocall
```

```
invoke-virtual/range {vx..vy},methodtocall
invoke-super/range {vx..vy},methodtocall
invoke-direct/range {vx..vy},methodtocall
invoke-static/range {vx..vy},methodtocall
invoke-interface/range {vx..vy},methodtocall
```

```
neg-int vx,vy
not-int vx,vy
neg-long vx,vy
not-long vx,vy
neg-float vx,vy
neg-double vx,vy
```

```
int-to-long vx, vy
int-to-float vx, vy
int-to-double vx, vy
long-to-int vx,vy
long-to-float vx, vy
long-to-double vx, vy
float-to-int vx, vy
float-to-long vx,vy
float-to-double vx, vy
double-to-int vx, vy
double-to-long vx, vy
double-to-float vx, vy
int-to-byte vx,vy
int-to-char vx,vy
int-to-short vx,vy
```

```
add-int vx,vy,vz
sub-int vx,vy,vz
mul-int vx, vy, vz
div-int vx,vy,vz
rem-int vx,vy,vz
```

```
and-int vx, vy, vz
or-int vx, vy, vz
xor-int vx, vy, vz
shl-int vx, vy, vz
shr-int vx, vy, vz
ushr-int vx, vy, vz
add-long vx, vy, vz
sub-long vx,vy,vz
mul-long vx,vy,vz
div-long vx, vy, vz
rem-long vx,vy,vz
and-long vx, vy, vz
or-long vx, vy, vz
xor-long vx, vy, vz
shl-long vx, vy, vz
shr-long vx,vy,vz
ushr-long vx, vy, vz
add-float vx,vy,vz
sub-float vx,vy,vz
mul-float vx, vy, vz
div-float vx, vy, vz
rem-float vx,vy,vz
add-double vx,vy,vz
sub-double vx,vy,vz
mul-double vx, vy, vz
div-double vx, vy, vz
rem-double vx,vy,vz
add-int/2addr vx,vy
sub-int/2addr vx,vy
mul-int/2addr vx,vy
div-int/2addr vx,vy
rem-int/2addr vx,vy
and-int/2addr vx, vy
or-int/2addr vx, vy
xor-int/2addr vx, vy
shl-int/2addr vx, vy
shr-int/2addr vx, vy
ushr-int/2addr vx, vy
add-long/2addr vx,vy
sub-long/2addr vx,vy
mul-long/2addr vx,vy
div-long/2addr vx, vy
rem-long/2addr vx,vy
and-long/2addr vx, vy
or-long/2addr vx, vy
xor-long/2addr vx, vy
shl-long/2addr vx, vy
shr-long/2addr vx, vy
ushr-long/2addr vx, vy
add-float/2addr vx,vy
sub-float/2addr vx,vy
mul-float/2addr vx, vy
div-float/2addr vx, vy
rem-float/2addr vx,vy
add-double/2addr vx, vy
sub-double/2addr vx, vy
mul-double/2addr vx, vy
div-double/2addr vx, vy
rem-double/2addr vx,vy
add-int/lit16 vx,vy,lit16
sub-int/lit16 vx,vy,lit16
mul-int/lit16 vx,vy,lit16
div-int/lit16 vx,vy,lit16
rem-int/lit16 vx,vy,lit16
and-int/lit16 vx,vy,lit16
or-int/lit16 vx,vy,lit16
xor-int/lit16 vx,vy,lit16
add-int/lit8 vx,vy,lit8
sub-int/lit8 vx,vy,lit8
mul-int/lit8 vx,vy,lit8
div-int/lit8 vx,vy,lit8
rem-int/lit8 vx,vy,lit8
and-int/lit8 vx,vy,lit8
or-int/lit8 vx, vy, lit8
xor-int/lit8 vx, vy, lit8
shl-int/lit8 vx, vy, lit8
shr-int/lit8 vx, vy, lit8
ushr-int/lit8 vx, vy, lit8
```

```
nop
```

```
move vx,vy
move/from16 vx,vy
move/16
move-wide
move-wide/from16 vx,vy
move-wide/16
move-object vx,vy
move-object/from16 vx,vy
move-object/16
move-result vx
move-result-wide vx
move-result-object vx
move-exception vx
```

```
return-void
return vx
return-wide vx
return-object vx
```

```
const/4 vx,lit4
const/16 vx,lit16
const vx, lit32
const/high16 v0, lit16
const-wide/16 vx, lit16
const-wide/32 vx, lit32
const-wide vx, lit64
const-wide/high16 vx,lit16
const-string vx,string_id
const-string-jumbo
const-class vx,type_id
```

```
monitor-enter vx
monitor-exit
```

```
check-cast vx, type_id
```

```
instance-of vx,vy,type_id
```

```
array-length vx,vy
```

```
new-instance vx,type
new-array vx,vy,type_id
filled-new-array {parameters},type_id
filled-new-array-range {vx..vy},type_id
fill-array-data vx,array_data_offset
```

```
throw vx
```

```
goto target
goto/16 target
goto/32 target
```

```
packed-switch vx,table
sparse-switch vx,table
```

```
cmpl-float
cmpg-float vx, vy, vz
cmpl-double vx,vy,vz
cmpg-double vx, vy, vz
cmp-long vx, vy, vz
```

```
if-eq vx,vy,target
if-ne vx,vy,target
if-lt vx,vy,target
if-ge vx, vy,target
if-gt vx,vy,target
if-le vx,vy,target
if-eqz vx,target
if-nez vx,target
if-ltz vx,target
if-gez vx,target
if-gtz vx,target
if-lez vx,target
```

```
aget vx,vy,vz
aget-wide vx,vy,vz
aget-object vx,vy,vz
aget-boolean vx,vy,vz
aget-byte vx,vy,vz
aget-char vx, vy,vz
aget-short vx,vy,vz
```

```
aput vx,vy,vz
aput-wide vx,vy,vz
aput-object vx,vy,vz
aput-boolean vx,vy,vz
aput-byte vx,vy,vz
aput-char vx,vy,vz
aput-short vx,vy,vz
```

```
iget vx, vy, field_id
iget-wide vx,vy,field_id
iget-object vx,vy,field_id
iget-boolean vx,vy,field_id
iget-byte vx,vy,field_id
iget-char vx,vy,field_id
iget-short vx,vy,field_id
```

```
iput vx,vy, field_id
iput-wide vx,vy, field_id
iput-object vx,vy,field_id
iput-boolean vx,vy, field_id
iput-byte vx,vy,field_id
iput-char vx,vy,field_id
iput-short vx,vy,field_id
```

```
sget vx,field_id
sget-wide vx, field_id
sget-object vx,field_id
sget-boolean vx,field_id
sget-byte vx,field_id
sget-char vx,field_id
sget-short vx,field_id
```

```
sput vx, field_id
sput-wide vx, field_id
sput-object vx,field_id
sput-boolean vx,field_id
sput-byte vx,field_id
sput-char vx,field_id
sput-short vx,field_id
```

```
invoke-virtual { parameters },methodtocall
invoke-super {parameter},methodtocall
invoke-direct { parameters },methodtocall
invoke-static {parameters},methodtocall
invoke-interface {parameters},methodtocall
```

```
invoke-virtual/range {vx..vy},methodtocall
invoke-super/range {vx..vy},methodtocall
invoke-direct/range {vx..vy},methodtocall
invoke-static/range {vx..vy},methodtocall
invoke-interface/range {vx..vy},methodtocall
```

```
neg-int vx,vy
not-int vx,vy
neg-long vx,vy
not-long vx,vy
neg-float vx,vy
neg-double vx,vy
```

```
int-to-long vx, vy
int-to-float vx, vy
int-to-double vx, vy
long-to-int vx,vy
long-to-float vx, vy
long-to-double vx, vy
float-to-int vx, vy
float-to-long vx,vy
float-to-double vx, vy
double-to-int vx, vy
double-to-long vx, vy
double-to-float vx, vy
int-to-byte vx,vy
int-to-char vx,vy
int-to-short vx,vy
```

```
add-int vx,vy,vz
sub-int vx,vy,vz
mul-int vx, vy, vz
div-int vx,vy,vz
rem-int vx,vy,vz
```

```
and-int vx, vy, vz
or-int vx, vy, vz
xor-int vx, vy, vz
shl-int vx, vy, vz
shr-int vx, vy, vz
ushr-int vx, vy, vz
add-long vx, vy, vz
sub-long vx,vy,vz
mul-long vx,vy,vz
div-long vx, vy, vz
rem-long vx,vy,vz
and-long vx, vy, vz
or-long vx, vy, vz
xor-long vx, vy, vz
shl-long vx, vy, vz
shr-long vx,vy,vz
ushr-long vx, vy, vz
add-float vx,vy,vz
sub-float vx,vy,vz
mul-float vx, vy, vz
div-float vx, vy, vz
rem-float vx,vy,vz
add-double vx,vy,vz
sub-double vx,vy,vz
mul-double vx, vy, vz
div-double vx, vy, vz
rem-double vx,vy,vz
add-int/2addr vx,vy
sub-int/2addr vx,vy
mul-int/2addr vx,vy
div-int/2addr vx,vy
rem-int/2addr vx,vy
and-int/2addr vx, vy
or-int/2addr vx, vy
xor-int/2addr vx, vy
shl-int/2addr vx, vy
shr-int/2addr vx, vy
ushr-int/2addr vx, vy
add-long/2addr vx,vy
sub-long/2addr vx,vy
mul-long/2addr vx,vy
div-long/2addr vx, vy
rem-long/2addr vx,vy
and-long/2addr vx, vy
or-long/2addr vx, vy
xor-long/2addr vx, vy
shl-long/2addr vx, vy
shr-long/2addr vx, vy
ushr-long/2addr vx, vy
add-float/2addr vx,vy
sub-float/2addr vx,vy
mul-float/2addr vx, vy
div-float/2addr vx, vy
rem-float/2addr vx,vy
add-double/2addr vx, vy
sub-double/2addr vx, vy
mul-double/2addr vx, vy
div-double/2addr vx, vy
rem-double/2addr vx,vy
add-int/lit16 vx,vy,lit16
sub-int/lit16 vx,vy,lit16
mul-int/lit16 vx,vy,lit16
div-int/lit16 vx,vy,lit16
rem-int/lit16 vx,vy,lit16
and-int/lit16 vx,vy,lit16
or-int/lit16 vx,vy,lit16
xor-int/lit16 vx,vy,lit16
add-int/lit8 vx,vy,lit8
sub-int/lit8 vx,vy,lit8
mul-int/lit8 vx,vy,lit8
div-int/lit8 vx,vy,lit8
rem-int/lit8 vx,vy,lit8
and-int/lit8 vx,vy,lit8
or-int/lit8 vx, vy, lit8
xor-int/lit8 vx, vy, lit8
shl-int/lit8 vx, vy, lit8
shr-int/lit8 vx, vy, lit8
ushr-int/lit8 vx, vy, lit8
```

```
nop
```

```
move vx,vy
move/from16 vx,vy
move/16
move-wide
move-wide/from16 vx,vy
move-wide/16
move-object vx,vy
move-object/from16 vx,vy
move-object/16
move-result vx
move-result-wide vx
move-result-object vx
move-exception vx
```

```
return-void
return vx
return-wide vx
return-object vx
```

```
const/4 vx,lit4
const/16 vx,lit16
const vx, lit32
const/high16 v0, lit16
const-wide/16 vx, lit16
const-wide/32 vx, lit32
const-wide vx, lit64
const-wide/high16 vx,lit16
const-string vx,string_id
const-string-jumbo
const-class vx,type_id
```

```
monitor-enter vx
monitor-exit
```

```
check-cast vx, type_id
```

```
instance-of vx,vy,type_id
```

```
array-length vx,vy
```

```
new-instance vx,type
new-array vx,vy,type_id
filled-new-array {parameters},type_id
filled-new-array-range {vx..vy},type_id
fill-array-data vx,array_data_offset
```

```
throw vx
```

```
goto target
goto/16 target
goto/32 target
```

```
packed-switch vx,table
sparse-switch vx,table
```

```
cmpl-float
cmpg-float vx, vy, vz
cmpl-double vx,vy,vz
cmpg-double vx, vy, vz
cmp-long vx, vy, vz
```

```
if-eq vx,vy,target
if-ne vx,vy,target
if-lt vx,vy,target
if-ge vx, vy,target
if-gt vx,vy,target
if-le vx,vy,target
if-eqz vx,target
if-nez vx,target
if-ltz vx,target
if-gez vx,target
if-gtz vx,target
if-lez vx,target
```

```
aget vx,vy,vz
aget-wide vx,vy,vz
aget-object vx,vy,vz
aget-boolean vx,vy,vz
aget-byte vx,vy,vz
aget-char vx, vy,vz
aget-short vx,vy,vz
```

```
aput vx,vy,vz
aput-wide vx,vy,vz
aput-object vx,vy,vz
aput-boolean vx,vy,vz
aput-byte vx,vy,vz
aput-char vx,vy,vz
aput-short vx,vy,vz
```

```
iget vx, vy, field_id
iget-wide vx,vy,field_id
iget-object vx,vy,field_id
iget-boolean vx,vy,field_id
iget-byte vx,vy,field_id
iget-char vx,vy,field_id
iget-short vx,vy,field_id
```

```
iput vx,vy, field_id
iput-wide vx,vy, field_id
iput-object vx,vy,field_id
iput-boolean vx,vy, field_id
iput-byte vx,vy,field_id
iput-char vx,vy,field_id
iput-short vx,vy,field_id
```

```
sget vx,field_id
sget-wide vx, field_id
sget-object vx,field_id
sget-boolean vx,field_id
sget-byte vx,field_id
sget-char vx,field_id
sget-short vx,field_id
```

```
sput vx, field_id
sput-wide vx, field_id
sput-object vx,field_id
sput-boolean vx,field_id
sput-byte vx,field_id
sput-char vx,field_id
sput-short vx,field_id
```

```
invoke-virtual { parameters },methodtocall
invoke-super {parameter},methodtocall
invoke-direct { parameters },methodtocall
invoke-static {parameters},methodtocall
invoke-interface {parameters},methodtocall
```

```
invoke-virtual/range {vx..vy},methodtocall
invoke-super/range {vx..vy},methodtocall
invoke-direct/range {vx..vy},methodtocall
invoke-static/range {vx..vy},methodtocall
invoke-interface/range {vx..vy},methodtocall
```

```
neg-int vx,vy
not-int vx,vy
neg-long vx,vy
not-long vx,vy
neg-float vx,vy
neg-double vx,vy
```

```
int-to-long vx, vy
int-to-float vx, vy
int-to-double vx, vy
long-to-int vx,vy
long-to-float vx, vy
long-to-double vx, vy
float-to-int vx, vy
float-to-long vx,vy
float-to-double vx, vy
double-to-int vx, vy
double-to-long vx, vy
double-to-float vx, vy
int-to-byte vx,vy
int-to-char vx,vy
int-to-short vx,vy
```

```
add-int vx,vy,vz
sub-int vx,vy,vz
mul-int vx, vy, vz
div-int vx,vy,vz
rem-int vx,vy,vz
```

```
and-int vx, vy, vz
or-int vx, vy, vz
xor-int vx, vy, vz
shl-int vx, vy, vz
shr-int vx, vy, vz
ushr-int vx, vy, vz
add-long vx, vy, vz
sub-long vx,vy,vz
mul-long vx,vy,vz
div-long vx, vy, vz
rem-long vx,vy,vz
and-long vx, vy, vz
or-long vx, vy, vz
xor-long vx, vy, vz
shl-long vx, vy, vz
shr-long vx,vy,vz
ushr-long vx, vy, vz
add-float vx,vy,vz
sub-float vx,vy,vz
mul-float vx, vy, vz
div-float vx, vy, vz
rem-float vx,vy,vz
add-double vx,vy,vz
sub-double vx,vy,vz
mul-double vx, vy, vz
div-double vx, vy, vz
rem-double vx,vy,vz
add-int/2addr vx,vy
sub-int/2addr vx,vy
mul-int/2addr vx,vy
div-int/2addr vx,vy
rem-int/2addr vx,vy
and-int/2addr vx, vy
or-int/2addr vx, vy
xor-int/2addr vx, vy
shl-int/2addr vx, vy
shr-int/2addr vx, vy
ushr-int/2addr vx, vy
add-long/2addr vx,vy
sub-long/2addr vx,vy
mul-long/2addr vx,vy
div-long/2addr vx, vy
rem-long/2addr vx,vy
and-long/2addr vx, vy
or-long/2addr vx, vy
xor-long/2addr vx, vy
shl-long/2addr vx, vy
shr-long/2addr vx, vy
ushr-long/2addr vx, vy
add-float/2addr vx,vy
sub-float/2addr vx,vy
mul-float/2addr vx, vy
div-float/2addr vx, vy
rem-float/2addr vx,vy
add-double/2addr vx, vy
sub-double/2addr vx, vy
mul-double/2addr vx, vy
div-double/2addr vx, vy
rem-double/2addr vx,vy
add-int/lit16 vx,vy,lit16
sub-int/lit16 vx,vy,lit16
mul-int/lit16 vx,vy,lit16
div-int/lit16 vx,vy,lit16
rem-int/lit16 vx,vy,lit16
and-int/lit16 vx,vy,lit16
or-int/lit16 vx,vy,lit16
xor-int/lit16 vx,vy,lit16
add-int/lit8 vx,vy,lit8
sub-int/lit8 vx,vy,lit8
mul-int/lit8 vx,vy,lit8
div-int/lit8 vx,vy,lit8
rem-int/lit8 vx,vy,lit8
and-int/lit8 vx,vy,lit8
or-int/lit8 vx, vy, lit8
xor-int/lit8 vx, vy, lit8
shl-int/lit8 vx, vy, lit8
shr-int/lit8 vx, vy, lit8
ushr-int/lit8 vx, vy, lit8
```

```
nop
```

```
move vx,vy
move/from16 vx,vy
move/16
move-wide
move-wide/from16 vx,vy
move-wide/16
move-object vx,vy
move-object/from16 vx,vy
move-object/16
move-result vx
move-result-wide vx
move-result-object vx
move-exception vx
```

```
return-void
return vx
return-wide vx
return-object vx
```

```
const/4 vx,lit4
const/16 vx,lit16
const vx, lit32
const/high16 v0, lit16
const-wide/16 vx, lit16
const-wide/32 vx, lit32
const-wide vx, lit64
const-wide/high16 vx,lit16
const-string vx,string_id
const-string-jumbo
const-class vx,type_id
```

```
monitor-enter vx
monitor-exit
```

```
check-cast vx, type_id
```

```
instance-of vx,vy,type_id
```

```
array-length vx,vy
```

```
new-instance vx,type
new-array vx,vy,type_id
filled-new-array {parameters},type_id
filled-new-array-range {vx..vy},type_id
fill-array-data vx,array_data_offset
```

```
throw vx
```

```
goto target
goto/16 target
goto/32 target
```

```
packed-switch vx,table
sparse-switch vx,table
```

```
cmpl-float
cmpg-float vx, vy, vz
cmpl-double vx,vy,vz
cmpg-double vx, vy, vz
cmp-long vx, vy, vz
```

```
if-eq vx,vy,target
if-ne vx,vy,target
if-lt vx,vy,target
if-ge vx, vy,target
if-gt vx,vy,target
if-le vx,vy,target
if-eqz vx,target
if-nez vx,target
if-ltz vx,target
if-gez vx,target
if-gtz vx,target
if-lez vx,target
```

```
aget vx,vy,vz
aget-wide vx,vy,vz
aget-object vx,vy,vz
aget-boolean vx,vy,vz
aget-byte vx,vy,vz
aget-char vx, vy,vz
aget-short vx,vy,vz
```

```
aput vx,vy,vz
aput-wide vx,vy,vz
aput-object vx,vy,vz
aput-boolean vx,vy,vz
aput-byte vx,vy,vz
aput-char vx,vy,vz
aput-short vx,vy,vz
```

```
iget vx, vy, field_id
iget-wide vx,vy,field_id
iget-object vx,vy,field_id
iget-boolean vx,vy,field_id
iget-byte vx,vy,field_id
iget-char vx,vy,field_id
iget-short vx,vy,field_id
```

```
iput vx,vy, field_id
iput-wide vx,vy, field_id
iput-object vx,vy,field_id
iput-boolean vx,vy, field_id
iput-byte vx,vy,field_id
iput-char vx,vy,field_id
iput-short vx,vy,field_id
```

```
sget vx,field_id
sget-wide vx, field_id
sget-object vx,field_id
sget-boolean vx,field_id
sget-byte vx,field_id
sget-char vx,field_id
sget-short vx,field_id
```

```
sput vx, field_id
sput-wide vx, field_id
sput-object vx,field_id
sput-boolean vx,field_id
sput-byte vx,field_id
sput-char vx,field_id
sput-short vx,field_id
```

```
invoke-virtual { parameters },methodtocall
invoke-super {parameter},methodtocall
invoke-direct { parameters },methodtocall
invoke-static {parameters},methodtocall
invoke-interface {parameters},methodtocall
```

```
invoke-virtual/range {vx..vy},methodtocall
invoke-super/range {vx..vy},methodtocall
invoke-direct/range {vx..vy},methodtocall
invoke-static/range {vx..vy},methodtocall
invoke-interface/range {vx..vy},methodtocall
```

```
neg-int vx,vy
not-int vx,vy
neg-long vx,vy
not-long vx,vy
neg-float vx,vy
neg-double vx,vy
```

```
int-to-long vx, vy
int-to-float vx, vy
int-to-double vx, vy
long-to-int vx,vy
long-to-float vx, vy
long-to-double vx, vy
float-to-int vx, vy
float-to-long vx,vy
float-to-double vx, vy
double-to-int vx, vy
double-to-long vx, vy
double-to-float vx, vy
int-to-byte vx,vy
int-to-char vx,vy
int-to-short vx,vy
```

```
add-int vx,vy,vz
sub-int vx,vy,vz
mul-int vx, vy, vz
div-int vx,vy,vz
rem-int vx,vy,vz
```

```
and-int vx, vy, vz
or-int vx, vy, vz
xor-int vx, vy, vz
shl-int vx, vy, vz
shr-int vx, vy, vz
ushr-int vx, vy, vz
add-long vx, vy, vz
sub-long vx,vy,vz
mul-long vx,vy,vz
div-long vx, vy, vz
rem-long vx,vy,vz
and-long vx, vy, vz
or-long vx, vy, vz
xor-long vx, vy, vz
shl-long vx, vy, vz
shr-long vx,vy,vz
ushr-long vx, vy, vz
add-float vx,vy,vz
sub-float vx,vy,vz
mul-float vx, vy, vz
div-float vx, vy, vz
rem-float vx,vy,vz
add-double vx,vy,vz
sub-double vx,vy,vz
mul-double vx, vy, vz
div-double vx, vy, vz
rem-double vx,vy,vz
add-int/2addr vx,vy
sub-int/2addr vx,vy
mul-int/2addr vx,vy
div-int/2addr vx,vy
rem-int/2addr vx,vy
and-int/2addr vx, vy
or-int/2addr vx, vy
xor-int/2addr vx, vy
shl-int/2addr vx, vy
shr-int/2addr vx, vy
ushr-int/2addr vx, vy
add-long/2addr vx,vy
sub-long/2addr vx,vy
mul-long/2addr vx,vy
div-long/2addr vx, vy
rem-long/2addr vx,vy
and-long/2addr vx, vy
or-long/2addr vx, vy
xor-long/2addr vx, vy
shl-long/2addr vx, vy
shr-long/2addr vx, vy
ushr-long/2addr vx, vy
add-float/2addr vx,vy
sub-float/2addr vx,vy
mul-float/2addr vx, vy
div-float/2addr vx, vy
rem-float/2addr vx,vy
add-double/2addr vx, vy
sub-double/2addr vx, vy
mul-double/2addr vx, vy
div-double/2addr vx, vy
rem-double/2addr vx,vy
add-int/lit16 vx,vy,lit16
sub-int/lit16 vx,vy,lit16
mul-int/lit16 vx,vy,lit16
div-int/lit16 vx,vy,lit16
rem-int/lit16 vx,vy,lit16
and-int/lit16 vx,vy,lit16
or-int/lit16 vx,vy,lit16
xor-int/lit16 vx,vy,lit16
add-int/lit8 vx,vy,lit8
sub-int/lit8 vx,vy,lit8
mul-int/lit8 vx,vy,lit8
div-int/lit8 vx,vy,lit8
rem-int/lit8 vx,vy,lit8
and-int/lit8 vx,vy,lit8
or-int/lit8 vx, vy, lit8
xor-int/lit8 vx, vy, lit8
shl-int/lit8 vx, vy, lit8
shr-int/lit8 vx, vy, lit8
ushr-int/lit8 vx, vy, lit8
```

After

```
nop
```

```
move vx,vy
move/from16 vx,vy
move/16
move-wide
move-wide/from16 vx,vy
move-wide/16
move-object vx,vy
move-object/from16 vx,vy
move-object/16
move-result vx
move-result-wide vx
move-result-object vx
move-exception vx
```

```
return-void
return vx
return-wide vx
return-object vx
```

```
const/4 vx,lit4
const/16 vx,lit16
const vx, lit32
const/high16 v0, lit16
const-wide/16 vx, lit16
const-wide/32 vx, lit32
const-wide vx, lit64
const-wide/high16 vx,lit16
const-string vx,string_id
const-string-jumbo
const-class vx,type_id
```

```
monitor-enter vx
monitor-exit
```

```
check-cast vx, type_id
```

```
instance-of vx,vy,type_id
```

```
array-length vx,vy
```

```
new-instance vx,type
new-array vx,vy,type_id
filled-new-array {parameters},type_id
filled-new-array-range {vx..vy},type_id
fill-array-data vx,array_data_offset
```

```
throw vx
```

```
goto target
goto/16 target
goto/32 target
```

```
packed-switch vx,table
sparse-switch vx,table
```

```
cmpl-float
cmpg-float vx, vy, vz
cmpl-double vx,vy,vz
cmpg-double vx, vy, vz
cmp-long vx, vy, vz
```

```
if-eq vx,vy,target
if-ne vx,vy,target
if-lt vx,vy,target
if-ge vx, vy,target
if-gt vx,vy,target
if-le vx,vy,target
if-eqz vx,target
if-nez vx,target
if-ltz vx,target
if-gez vx,target
if-gtz vx,target
if-lez vx,target
```

```
aget vx,vy,vz
aget-wide vx,vy,vz
aget-object vx,vy,vz
aget-boolean vx,vy,vz
aget-byte vx,vy,vz
aget-char vx, vy,vz
aget-short vx,vy,vz
```

```
aput vx,vy,vz
aput-wide vx,vy,vz
aput-object vx,vy,vz
aput-boolean vx,vy,vz
aput-byte vx,vy,vz
aput-char vx,vy,vz
aput-short vx,vy,vz
```

```
iget vx, vy, field_id
iget-wide vx,vy,field_id
iget-object vx,vy,field_id
iget-boolean vx,vy,field_id
iget-byte vx,vy,field_id
iget-char vx,vy,field_id
iget-short vx,vy,field_id
```

```
iput vx,vy, field_id
iput-wide vx,vy, field_id
iput-object vx,vy,field_id
iput-boolean vx,vy, field_id
iput-byte vx,vy,field_id
iput-char vx,vy,field_id
iput-short vx,vy,field_id
```

```
sget vx,field_id
sget-wide vx, field_id
sget-object vx,field_id
sget-boolean vx,field_id
sget-byte vx,field_id
sget-char vx,field_id
sget-short vx,field_id
```

```
sput vx, field_id
sput-wide vx, field_id
sput-object vx,field_id
sput-boolean vx,field_id
sput-byte vx,field_id
sput-char vx,field_id
sput-short vx,field_id
```

```
invoke-virtual { parameters },methodtocall
invoke-super {parameter},methodtocall
invoke-direct { parameters },methodtocall
invoke-static {parameters},methodtocall
invoke-interface {parameters},methodtocall
```

```
invoke-virtual/range {vx..vy},methodtocall
invoke-super/range {vx..vy},methodtocall
invoke-direct/range {vx..vy},methodtocall
invoke-static/range {vx..vy},methodtocall
invoke-interface/range {vx..vy},methodtocall
```

```
neg-int vx,vy
not-int vx,vy
neg-long vx,vy
not-long vx,vy
neg-float vx,vy
neg-double vx,vy
```

```
int-to-long vx, vy
int-to-float vx, vy
int-to-double vx, vy
long-to-int vx,vy
long-to-float vx, vy
long-to-double vx, vy
float-to-int vx, vy
float-to-long vx,vy
float-to-double vx, vy
double-to-int vx, vy
double-to-long vx, vy
double-to-float vx, vy
int-to-byte vx,vy
int-to-char vx,vy
int-to-short vx,vy
```

```
add-int vx,vy,vz
sub-int vx,vy,vz
mul-int vx, vy, vz
div-int vx,vy,vz
rem-int vx,vy,vz
```

```
and-int vx, vy, vz
or-int vx, vy, vz
xor-int vx, vy, vz
shl-int vx, vy, vz
shr-int vx, vy, vz
ushr-int vx, vy, vz
add-long vx, vy, vz
sub-long vx,vy,vz
mul-long vx,vy,vz
div-long vx, vy, vz
rem-long vx,vy,vz
and-long vx, vy, vz
or-long vx, vy, vz
xor-long vx, vy, vz
shl-long vx, vy, vz
shr-long vx,vy,vz
ushr-long vx, vy, vz
add-float vx,vy,vz
sub-float vx,vy,vz
mul-float vx, vy, vz
div-float vx, vy, vz
rem-float vx,vy,vz
add-double vx,vy,vz
sub-double vx,vy,vz
mul-double vx, vy, vz
div-double vx, vy, vz
rem-double vx,vy,vz
add-int/2addr vx,vy
sub-int/2addr vx,vy
mul-int/2addr vx,vy
div-int/2addr vx,vy
rem-int/2addr vx,vy
and-int/2addr vx, vy
or-int/2addr vx, vy
xor-int/2addr vx, vy
shl-int/2addr vx, vy
shr-int/2addr vx, vy
ushr-int/2addr vx, vy
add-long/2addr vx,vy
sub-long/2addr vx,vy
mul-long/2addr vx,vy
div-long/2addr vx, vy
rem-long/2addr vx,vy
and-long/2addr vx, vy
or-long/2addr vx, vy
xor-long/2addr vx, vy
shl-long/2addr vx, vy
shr-long/2addr vx, vy
ushr-long/2addr vx, vy
add-float/2addr vx,vy
sub-float/2addr vx,vy
mul-float/2addr vx, vy
div-float/2addr vx, vy
rem-float/2addr vx,vy
add-double/2addr vx, vy
sub-double/2addr vx, vy
mul-double/2addr vx, vy
div-double/2addr vx, vy
rem-double/2addr vx,vy
add-int/lit16 vx,vy,lit16
sub-int/lit16 vx,vy,lit16
mul-int/lit16 vx,vy,lit16
div-int/lit16 vx,vy,lit16
rem-int/lit16 vx,vy,lit16
and-int/lit16 vx,vy,lit16
or-int/lit16 vx,vy,lit16
xor-int/lit16 vx,vy,lit16
add-int/lit8 vx,vy,lit8
sub-int/lit8 vx,vy,lit8
mul-int/lit8 vx,vy,lit8
div-int/lit8 vx,vy,lit8
rem-int/lit8 vx,vy,lit8
and-int/lit8 vx,vy,lit8
or-int/lit8 vx, vy, lit8
xor-int/lit8 vx, vy, lit8
shl-int/lit8 vx, vy, lit8
shr-int/lit8 vx, vy, lit8
ushr-int/lit8 vx, vy, lit8
```

```
nop
```

```
move vx,vy
move/from16 vx,vy
move/16
move-wide
move-wide/from16 vx,vy
move-wide/16
move-object vx,vy
move-object/from16 vx,vy
move-object/16
move-result vx
move-result-wide vx
move-result-object vx
move-exception vx
```

```
return-void
return vx
return-wide vx
return-object vx
```

```
const/4 vx,lit4
const/16 vx,lit16
const vx, lit32
const/high16 v0, lit16
const-wide/16 vx, lit16
const-wide/32 vx, lit32
const-wide vx, lit64
const-wide/high16 vx,lit16
const-string vx,string_id
const-string-jumbo
const-class vx,type_id
```

```
monitor-enter vx
monitor-exit
```

```
check-cast vx, type_id
```

```
instance-of vx,vy,type_id
```

```
array-length vx,vy
```

```
new-instance vx,type
new-array vx,vy,type_id
filled-new-array {parameters},type_id
filled-new-array-range {vx..vy},type_id
fill-array-data vx,array_data_offset
```

```
throw vx
```

```
goto target
goto/16 target
goto/32 target
```

```
packed-switch vx,table
sparse-switch vx,table
```

```
cmpl-float
cmpg-float vx, vy, vz
cmpl-double vx,vy,vz
cmpg-double vx, vy, vz
cmp-long vx, vy, vz
```

```
if-eq vx,vy,target
if-ne vx,vy,target
if-lt vx,vy,target
if-ge vx, vy,target
if-gt vx,vy,target
if-le vx,vy,target
if-eqz vx,target
if-nez vx,target
if-ltz vx,target
if-gez vx,target
if-gtz vx,target
if-lez vx,target
```

```
aget vx,vy,vz
aget-wide vx,vy,vz
aget-object vx,vy,vz
aget-boolean vx,vy,vz
aget-byte vx,vy,vz
aget-char vx, vy,vz
aget-short vx,vy,vz
```

```
aput vx,vy,vz
aput-wide vx,vy,vz
aput-object vx,vy,vz
aput-boolean vx,vy,vz
aput-byte vx,vy,vz
aput-char vx,vy,vz
aput-short vx,vy,vz
```

```
iget vx, vy, field_id
iget-wide vx,vy,field_id
iget-object vx,vy,field_id
iget-boolean vx,vy,field_id
iget-byte vx,vy,field_id
iget-char vx,vy,field_id
iget-short vx,vy,field_id
```

```
iput vx,vy, field_id
iput-wide vx,vy, field_id
iput-object vx,vy,field_id
iput-boolean vx,vy, field_id
iput-byte vx,vy,field_id
iput-char vx,vy,field_id
iput-short vx,vy,field_id
```

```
sget vx,field_id
sget-wide vx, field_id
sget-object vx,field_id
sget-boolean vx,field_id
sget-byte vx,field_id
sget-char vx,field_id
sget-short vx,field_id
```

```
sput vx, field_id
sput-wide vx, field_id
sput-object vx,field_id
sput-boolean vx,field_id
sput-byte vx,field_id
sput-char vx,field_id
sput-short vx,field_id
```

```
invoke-virtual { parameters },methodtocall
invoke-super {parameter},methodtocall
invoke-direct { parameters },methodtocall
invoke-static {parameters},methodtocall
invoke-interface {parameters},methodtocall
```

```
invoke-virtual/range {vx..vy},methodtocall
invoke-super/range {vx..vy},methodtocall
invoke-direct/range {vx..vy},methodtocall
invoke-static/range {vx..vy},methodtocall
invoke-interface/range {vx..vy},methodtocall
```

```
neg-int vx,vy
not-int vx,vy
neg-long vx,vy
not-long vx,vy
neg-float vx,vy
neg-double vx,vy
```

```
int-to-long vx, vy
int-to-float vx, vy
int-to-double vx, vy
long-to-int vx,vy
long-to-float vx, vy
long-to-double vx, vy
float-to-int vx, vy
float-to-long vx,vy
float-to-double vx, vy
double-to-int vx, vy
double-to-long vx, vy
double-to-float vx, vy
int-to-byte vx,vy
int-to-char vx,vy
int-to-short vx,vy
```

```
add-int vx,vy,vz
sub-int vx,vy,vz
mul-int vx, vy, vz
div-int vx,vy,vz
rem-int vx,vy,vz
```

```
and-int vx, vy, vz
or-int vx, vy, vz
xor-int vx, vy, vz
shl-int vx, vy, vz
shr-int vx, vy, vz
ushr-int vx, vy, vz
add-long vx, vy, vz
sub-long vx,vy,vz
mul-long vx,vy,vz
div-long vx, vy, vz
rem-long vx,vy,vz
and-long vx, vy, vz
or-long vx, vy, vz
xor-long vx, vy, vz
shl-long vx, vy, vz
shr-long vx,vy,vz
ushr-long vx, vy, vz
add-float vx,vy,vz
sub-float vx,vy,vz
mul-float vx, vy, vz
div-float vx, vy, vz
rem-float vx,vy,vz
add-double vx,vy,vz
sub-double vx,vy,vz
mul-double vx, vy, vz
div-double vx, vy, vz
rem-double vx,vy,vz
add-int/2addr vx,vy
sub-int/2addr vx,vy
mul-int/2addr vx,vy
div-int/2addr vx,vy
rem-int/2addr vx,vy
and-int/2addr vx, vy
or-int/2addr vx, vy
xor-int/2addr vx, vy
shl-int/2addr vx, vy
shr-int/2addr vx, vy
ushr-int/2addr vx, vy
add-long/2addr vx,vy
sub-long/2addr vx,vy
mul-long/2addr vx,vy
div-long/2addr vx, vy
rem-long/2addr vx,vy
and-long/2addr vx, vy
or-long/2addr vx, vy
xor-long/2addr vx, vy
shl-long/2addr vx, vy
shr-long/2addr vx, vy
ushr-long/2addr vx, vy
add-float/2addr vx,vy
sub-float/2addr vx,vy
mul-float/2addr vx, vy
div-float/2addr vx, vy
rem-float/2addr vx,vy
add-double/2addr vx, vy
sub-double/2addr vx, vy
mul-double/2addr vx, vy
div-double/2addr vx, vy
rem-double/2addr vx,vy
add-int/lit16 vx,vy,lit16
sub-int/lit16 vx,vy,lit16
mul-int/lit16 vx,vy,lit16
div-int/lit16 vx,vy,lit16
rem-int/lit16 vx,vy,lit16
and-int/lit16 vx,vy,lit16
or-int/lit16 vx,vy,lit16
xor-int/lit16 vx,vy,lit16
add-int/lit8 vx,vy,lit8
sub-int/lit8 vx,vy,lit8
mul-int/lit8 vx,vy,lit8
div-int/lit8 vx,vy,lit8
rem-int/lit8 vx,vy,lit8
and-int/lit8 vx,vy,lit8
or-int/lit8 vx, vy, lit8
xor-int/lit8 vx, vy, lit8
shl-int/lit8 vx, vy, lit8
shr-int/lit8 vx, vy, lit8
ushr-int/lit8 vx, vy, lit8
```

```
nop
```

```
move vx,vy
move/from16 vx,vy
move/16
move-wide
move-wide/from16 vx,vy
move-wide/16
move-object vx,vy
move-object/from16 vx,vy
move-object/16
move-result vx
move-result-wide vx
move-result-object vx
move-exception vx
```

```
return-void
return vx
return-wide vx
return-object vx
```

```
const/4 vx,lit4
const/16 vx,lit16
const vx, lit32
const/high16 v0, lit16
const-wide/16 vx, lit16
const-wide/32 vx, lit32
const-wide vx, lit64
const-wide/high16 vx,lit16
const-string vx,string_id
const-string-jumbo
const-class vx,type_id
```

```
monitor-enter vx
monitor-exit
```

```
check-cast vx, type_id
```

```
instance-of vx,vy,type_id
```

```
array-length vx,vy
```

```
new-instance vx,type
new-array vx,vy,type_id
filled-new-array {parameters},type_id
filled-new-array-range {vx..vy},type_id
fill-array-data vx,array_data_offset
```

```
throw vx
```

```
goto target
goto/16 target
goto/32 target
```

```
packed-switch vx,table
sparse-switch vx,table
```

```
cmpl-float
cmpg-float vx, vy, vz
cmpl-double vx,vy,vz
cmpg-double vx, vy, vz
cmp-long vx, vy, vz
```

```
if-eq vx,vy,target
if-ne vx,vy,target
if-lt vx,vy,target
if-ge vx, vy,target
if-gt vx,vy,target
if-le vx,vy,target
if-eqz vx,target
if-nez vx,target
if-ltz vx,target
if-gez vx,target
if-gtz vx,target
if-lez vx,target
```

```
aget vx,vy,vz
aget-wide vx,vy,vz
aget-object vx,vy,vz
aget-boolean vx,vy,vz
aget-byte vx,vy,vz
aget-char vx, vy,vz
aget-short vx,vy,vz
```

```
aput vx,vy,vz
aput-wide vx,vy,vz
aput-object vx,vy,vz
aput-boolean vx,vy,vz
aput-byte vx,vy,vz
aput-char vx,vy,vz
aput-short vx,vy,vz
```

```
iget vx, vy, field_id
iget-wide vx,vy,field_id
iget-object vx,vy,field_id
iget-boolean vx,vy,field_id
iget-byte vx,vy,field_id
iget-char vx,vy,field_id
iget-short vx,vy,field_id
```

```
iput vx,vy, field_id
iput-wide vx,vy, field_id
iput-object vx,vy,field_id
iput-boolean vx,vy, field_id
iput-byte vx,vy,field_id
iput-char vx,vy,field_id
iput-short vx,vy,field_id
```

```
sget vx,field_id
sget-wide vx, field_id
sget-object vx,field_id
sget-boolean vx,field_id
sget-byte vx,field_id
sget-char vx,field_id
sget-short vx,field_id
```

```
sput vx, field_id
sput-wide vx, field_id
sput-object vx,field_id
sput-boolean vx,field_id
sput-byte vx,field_id
sput-char vx,field_id
sput-short vx,field_id
```

```
invoke-virtual { parameters },methodtocall
invoke-super {parameter},methodtocall
invoke-direct { parameters },methodtocall
invoke-static {parameters},methodtocall
invoke-interface {parameters},methodtocall
```

```
invoke-virtual/range {vx..vy},methodtocall
invoke-super/range {vx..vy},methodtocall
invoke-direct/range {vx..vy},methodtocall
invoke-static/range {vx..vy},methodtocall
invoke-interface/range {vx..vy},methodtocall
```

```
neg-int vx,vy
not-int vx,vy
neg-long vx,vy
not-long vx,vy
neg-float vx,vy
neg-double vx,vy
```

```
int-to-long vx, vy
int-to-float vx, vy
int-to-double vx, vy
long-to-int vx,vy
long-to-float vx, vy
long-to-double vx, vy
float-to-int vx, vy
float-to-long vx,vy
float-to-double vx, vy
double-to-int vx, vy
double-to-long vx, vy
double-to-float vx, vy
int-to-byte vx,vy
int-to-char vx,vy
int-to-short vx,vy
```

```
add-int vx,vy,vz
sub-int vx,vy,vz
mul-int vx, vy, vz
div-int vx,vy,vz
rem-int vx,vy,vz
```

```
and-int vx, vy, vz
or-int vx, vy, vz
xor-int vx, vy, vz
shl-int vx, vy, vz
shr-int vx, vy, vz
ushr-int vx, vy, vz
add-long vx, vy, vz
sub-long vx,vy,vz
mul-long vx,vy,vz
div-long vx, vy, vz
rem-long vx,vy,vz
and-long vx, vy, vz
or-long vx, vy, vz
xor-long vx, vy, vz
shl-long vx, vy, vz
shr-long vx,vy,vz
ushr-long vx, vy, vz
add-float vx,vy,vz
sub-float vx,vy,vz
mul-float vx, vy, vz
div-float vx, vy, vz
rem-float vx,vy,vz
add-double vx,vy,vz
sub-double vx,vy,vz
mul-double vx, vy, vz
div-double vx, vy, vz
rem-double vx,vy,vz
add-int/2addr vx,vy
sub-int/2addr vx,vy
mul-int/2addr vx,vy
div-int/2addr vx,vy
rem-int/2addr vx,vy
and-int/2addr vx, vy
or-int/2addr vx, vy
xor-int/2addr vx, vy
shl-int/2addr vx, vy
shr-int/2addr vx, vy
ushr-int/2addr vx, vy
add-long/2addr vx,vy
sub-long/2addr vx,vy
mul-long/2addr vx,vy
div-long/2addr vx, vy
rem-long/2addr vx,vy
and-long/2addr vx, vy
or-long/2addr vx, vy
xor-long/2addr vx, vy
shl-long/2addr vx, vy
shr-long/2addr vx, vy
ushr-long/2addr vx, vy
add-float/2addr vx,vy
sub-float/2addr vx,vy
mul-float/2addr vx, vy
div-float/2addr vx, vy
rem-float/2addr vx,vy
add-double/2addr vx, vy
sub-double/2addr vx, vy
mul-double/2addr vx, vy
div-double/2addr vx, vy
rem-double/2addr vx,vy
add-int/lit16 vx,vy,lit16
sub-int/lit16 vx,vy,lit16
mul-int/lit16 vx,vy,lit16
div-int/lit16 vx,vy,lit16
rem-int/lit16 vx,vy,lit16
and-int/lit16 vx,vy,lit16
or-int/lit16 vx,vy,lit16
xor-int/lit16 vx,vy,lit16
add-int/lit8 vx,vy,lit8
sub-int/lit8 vx,vy,lit8
mul-int/lit8 vx,vy,lit8
div-int/lit8 vx,vy,lit8
rem-int/lit8 vx,vy,lit8
and-int/lit8 vx,vy,lit8
or-int/lit8 vx, vy, lit8
xor-int/lit8 vx, vy, lit8
shl-int/lit8 vx, vy, lit8
shr-int/lit8 vx, vy, lit8
ushr-int/lit8 vx, vy, lit8
```

```
nop
```

```
move vx,vy
move/from16 vx,vy
move/16
move-wide
move-wide/from16 vx,vy
move-wide/16
move-object vx,vy
move-object/from16 vx,vy
move-object/16
move-result vx
move-result-wide vx
move-result-object vx
move-exception vx
```

```
return-void
return vx
return-wide vx
return-object vx
```

```
const/4 vx,lit4
const/16 vx,lit16
const vx, lit32
const/high16 v0, lit16
const-wide/16 vx, lit16
const-wide/32 vx, lit32
const-wide vx, lit64
const-wide/high16 vx,lit16
const-string vx,string_id
const-string-jumbo
const-class vx,type_id
```

```
monitor-enter vx
monitor-exit
```

```
check-cast vx, type_id
```

```
instance-of vx,vy,type_id
```

```
array-length vx,vy
```

```
new-instance vx,type
new-array vx,vy,type_id
filled-new-array {parameters},type_id
filled-new-array-range {vx..vy},type_id
fill-array-data vx,array_data_offset
```

```
throw vx
```

```
goto target
goto/16 target
goto/32 target
```

```
packed-switch vx,table
sparse-switch vx,table
```

```
cmpl-float
cmpg-float vx, vy, vz
cmpl-double vx,vy,vz
cmpg-double vx, vy, vz
cmp-long vx, vy, vz
```

```
if-eq vx,vy,target
if-ne vx,vy,target
if-lt vx,vy,target
if-ge vx, vy,target
if-gt vx,vy,target
if-le vx,vy,target
if-eqz vx,target
if-nez vx,target
if-ltz vx,target
if-gez vx,target
if-gtz vx,target
if-lez vx,target
```

```
aget vx,vy,vz
aget-wide vx,vy,vz
aget-object vx,vy,vz
aget-boolean vx,vy,vz
aget-byte vx,vy,vz
aget-char vx, vy,vz
aget-short vx,vy,vz
```

```
aput vx,vy,vz
aput-wide vx,vy,vz
aput-object vx,vy,vz
aput-boolean vx,vy,vz
aput-byte vx,vy,vz
aput-char vx,vy,vz
aput-short vx,vy,vz
```

```
iget vx, vy, field_id
iget-wide vx,vy,field_id
iget-object vx,vy,field_id
iget-boolean vx,vy,field_id
iget-byte vx,vy,field_id
iget-char vx,vy,field_id
iget-short vx,vy,field_id
```

```
iput vx,vy, field_id
iput-wide vx,vy, field_id
iput-object vx,vy,field_id
iput-boolean vx,vy, field_id
iput-byte vx,vy,field_id
iput-char vx,vy,field_id
iput-short vx,vy,field_id
```

```
sget vx,field_id
sget-wide vx, field_id
sget-object vx,field_id
sget-boolean vx,field_id
sget-byte vx,field_id
sget-char vx,field_id
sget-short vx,field_id
```

```
sput vx, field_id
sput-wide vx, field_id
sput-object vx,field_id
sput-boolean vx,field_id
sput-byte vx,field_id
sput-char vx,field_id
sput-short vx,field_id
```

```
invoke-virtual { parameters },methodtocall
invoke-super {parameter},methodtocall
invoke-direct { parameters },methodtocall
invoke-static {parameters},methodtocall
invoke-interface {parameters},methodtocall
```

```
invoke-virtual/range {vx..vy},methodtocall
invoke-super/range {vx..vy},methodtocall
invoke-direct/range {vx..vy},methodtocall
invoke-static/range {vx..vy},methodtocall
invoke-interface/range {vx..vy},methodtocall
```

```
neg-int vx,vy
not-int vx,vy
neg-long vx,vy
not-long vx,vy
neg-float vx,vy
neg-double vx,vy
```

```
int-to-long vx, vy
int-to-float vx, vy
int-to-double vx, vy
long-to-int vx,vy
long-to-float vx, vy
long-to-double vx, vy
float-to-int vx, vy
float-to-long vx,vy
float-to-double vx, vy
double-to-int vx, vy
double-to-long vx, vy
double-to-float vx, vy
int-to-byte vx,vy
int-to-char vx,vy
int-to-short vx,vy
```

```
add-int vx,vy,vz
sub-int vx,vy,vz
mul-int vx, vy, vz
div-int vx,vy,vz
rem-int vx,vy,vz
```

```
and-int vx, vy, vz
or-int vx, vy, vz
xor-int vx, vy, vz
shl-int vx, vy, vz
shr-int vx, vy, vz
ushr-int vx, vy, vz
add-long vx, vy, vz
sub-long vx,vy,vz
mul-long vx,vy,vz
div-long vx, vy, vz
rem-long vx,vy,vz
and-long vx, vy, vz
or-long vx, vy, vz
xor-long vx, vy, vz
shl-long vx, vy, vz
shr-long vx,vy,vz
ushr-long vx, vy, vz
add-float vx,vy,vz
sub-float vx,vy,vz
mul-float vx, vy, vz
div-float vx, vy, vz
rem-float vx,vy,vz
add-double vx,vy,vz
sub-double vx,vy,vz
mul-double vx, vy, vz
div-double vx, vy, vz
rem-double vx,vy,vz
add-int/2addr vx,vy
sub-int/2addr vx,vy
mul-int/2addr vx,vy
div-int/2addr vx,vy
rem-int/2addr vx,vy
and-int/2addr vx, vy
or-int/2addr vx, vy
xor-int/2addr vx, vy
shl-int/2addr vx, vy
shr-int/2addr vx, vy
ushr-int/2addr vx, vy
add-long/2addr vx,vy
sub-long/2addr vx,vy
mul-long/2addr vx,vy
div-long/2addr vx, vy
rem-long/2addr vx,vy
and-long/2addr vx, vy
or-long/2addr vx, vy
xor-long/2addr vx, vy
shl-long/2addr vx, vy
shr-long/2addr vx, vy
ushr-long/2addr vx, vy
add-float/2addr vx,vy
sub-float/2addr vx,vy
mul-float/2addr vx, vy
div-float/2addr vx, vy
rem-float/2addr vx,vy
add-double/2addr vx, vy
sub-double/2addr vx, vy
mul-double/2addr vx, vy
div-double/2addr vx, vy
rem-double/2addr vx,vy
add-int/lit16 vx,vy,lit16
sub-int/lit16 vx,vy,lit16
mul-int/lit16 vx,vy,lit16
div-int/lit16 vx,vy,lit16
rem-int/lit16 vx,vy,lit16
and-int/lit16 vx,vy,lit16
or-int/lit16 vx,vy,lit16
xor-int/lit16 vx,vy,lit16
add-int/lit8 vx,vy,lit8
sub-int/lit8 vx,vy,lit8
mul-int/lit8 vx,vy,lit8
div-int/lit8 vx,vy,lit8
rem-int/lit8 vx,vy,lit8
and-int/lit8 vx,vy,lit8
or-int/lit8 vx, vy, lit8
xor-int/lit8 vx, vy, lit8
shl-int/lit8 vx, vy, lit8
shr-int/lit8 vx, vy, lit8
ushr-int/lit8 vx, vy, lit8
```

```
nop
```

```
move vx,vy
move/from16 vx,vy
move/16
move-wide
move-wide/from16 vx,vy
move-wide/16
move-object vx,vy
move-object/from16 vx,vy
move-object/16
move-result vx
move-result-wide vx
move-result-object vx
move-exception vx
```

```
return-void
return vx
return-wide vx
return-object vx
```

```
const/4 vx,lit4
const/16 vx,lit16
const vx, lit32
const/high16 v0, lit16
const-wide/16 vx, lit16
const-wide/32 vx, lit32
const-wide vx, lit64
const-wide/high16 vx,lit16
const-string vx,string_id
const-string-jumbo
const-class vx,type_id
```

```
monitor-enter vx
monitor-exit
```

```
check-cast vx, type_id
```

```
instance-of vx,vy,type_id
```

```
array-length vx,vy
```

```
new-instance vx,type
new-array vx,vy,type_id
filled-new-array {parameters},type_id
filled-new-array-range {vx..vy},type_id
fill-array-data vx,array_data_offset
```

```
throw vx
```

```
goto target
goto/16 target
goto/32 target
```

```
packed-switch vx,table
sparse-switch vx,table
```

```
cmpl-float
cmpg-float vx, vy, vz
cmpl-double vx,vy,vz
cmpg-double vx, vy, vz
cmp-long vx, vy, vz
```

```
if-eq vx,vy,target
if-ne vx,vy,target
if-lt vx,vy,target
if-ge vx, vy,target
if-gt vx,vy,target
if-le vx,vy,target
if-eqz vx,target
if-nez vx,target
if-ltz vx,target
if-gez vx,target
if-gtz vx,target
if-lez vx,target
```

```
aget vx,vy,vz
aget-wide vx,vy,vz
aget-object vx,vy,vz
aget-boolean vx,vy,vz
aget-byte vx,vy,vz
aget-char vx, vy,vz
aget-short vx,vy,vz
```

```
aput vx,vy,vz
aput-wide vx,vy,vz
aput-object vx,vy,vz
aput-boolean vx,vy,vz
aput-byte vx,vy,vz
aput-char vx,vy,vz
aput-short vx,vy,vz
```

```
iget vx, vy, field_id
iget-wide vx,vy,field_id
iget-object vx,vy,field_id
iget-boolean vx,vy,field_id
iget-byte vx,vy,field_id
iget-char vx,vy,field_id
iget-short vx,vy,field_id
```

```
iput vx,vy, field_id
iput-wide vx,vy, field_id
iput-object vx,vy,field_id
iput-boolean vx,vy, field_id
iput-byte vx,vy,field_id
iput-char vx,vy,field_id
iput-short vx,vy,field_id
```

```
sget vx,field_id
sget-wide vx, field_id
sget-object vx,field_id
sget-boolean vx,field_id
sget-byte vx,field_id
sget-char vx,field_id
sget-short vx,field_id
```

```
sput vx, field_id
sput-wide vx, field_id
sput-object vx,field_id
sput-boolean vx,field_id
sput-byte vx,field_id
sput-char vx,field_id
sput-short vx,field_id
```

```
invoke-virtual { parameters },methodtocall
invoke-super {parameter},methodtocall
invoke-direct { parameters },methodtocall
invoke-static {parameters},methodtocall
invoke-interface {parameters},methodtocall
```

```
invoke-virtual/range {vx..vy},methodtocall
invoke-super/range {vx..vy},methodtocall
invoke-direct/range {vx..vy},methodtocall
invoke-static/range {vx..vy},methodtocall
invoke-interface/range {vx..vy},methodtocall
```

```
neg-int vx,vy
not-int vx,vy
neg-long vx,vy
not-long vx,vy
neg-float vx,vy
neg-double vx,vy
```

```
int-to-long vx, vy
int-to-float vx, vy
int-to-double vx, vy
long-to-int vx,vy
long-to-float vx, vy
long-to-double vx, vy
float-to-int vx, vy
float-to-long vx,vy
float-to-double vx, vy
double-to-int vx, vy
double-to-long vx, vy
double-to-float vx, vy
int-to-byte vx,vy
int-to-char vx,vy
int-to-short vx,vy
```

```
add-int vx,vy,vz
sub-int vx,vy,vz
mul-int vx, vy, vz
div-int vx,vy,vz
rem-int vx,vy,vz
```

```
and-int vx, vy, vz
or-int vx, vy, vz
xor-int vx, vy, vz
shl-int vx, vy, vz
shr-int vx, vy, vz
ushr-int vx, vy, vz
add-long vx, vy, vz
sub-long vx,vy,vz
mul-long vx,vy,vz
div-long vx, vy, vz
rem-long vx,vy,vz
and-long vx, vy, vz
or-long vx, vy, vz
xor-long vx, vy, vz
shl-long vx, vy, vz
shr-long vx,vy,vz
ushr-long vx, vy, vz
add-float vx,vy,vz
sub-float vx,vy,vz
mul-float vx, vy, vz
div-float vx, vy, vz
rem-float vx,vy,vz
add-double vx,vy,vz
sub-double vx,vy,vz
mul-double vx, vy, vz
div-double vx, vy, vz
rem-double vx,vy,vz
add-int/2addr vx,vy
sub-int/2addr vx,vy
mul-int/2addr vx,vy
div-int/2addr vx,vy
rem-int/2addr vx,vy
and-int/2addr vx, vy
or-int/2addr vx, vy
xor-int/2addr vx, vy
shl-int/2addr vx, vy
shr-int/2addr vx, vy
ushr-int/2addr vx, vy
add-long/2addr vx,vy
sub-long/2addr vx,vy
mul-long/2addr vx,vy
div-long/2addr vx, vy
rem-long/2addr vx,vy
and-long/2addr vx, vy
or-long/2addr vx, vy
xor-long/2addr vx, vy
shl-long/2addr vx, vy
shr-long/2addr vx, vy
ushr-long/2addr vx, vy
add-float/2addr vx,vy
sub-float/2addr vx,vy
mul-float/2addr vx, vy
div-float/2addr vx, vy
rem-float/2addr vx,vy
add-double/2addr vx, vy
sub-double/2addr vx, vy
mul-double/2addr vx, vy
div-double/2addr vx, vy
rem-double/2addr vx,vy
add-int/lit16 vx,vy,lit16
sub-int/lit16 vx,vy,lit16
mul-int/lit16 vx,vy,lit16
div-int/lit16 vx,vy,lit16
rem-int/lit16 vx,vy,lit16
and-int/lit16 vx,vy,lit16
or-int/lit16 vx,vy,lit16
xor-int/lit16 vx,vy,lit16
add-int/lit8 vx,vy,lit8
sub-int/lit8 vx,vy,lit8
mul-int/lit8 vx,vy,lit8
div-int/lit8 vx,vy,lit8
rem-int/lit8 vx,vy,lit8
and-int/lit8 vx,vy,lit8
or-int/lit8 vx, vy, lit8
xor-int/lit8 vx, vy, lit8
shl-int/lit8 vx, vy, lit8
shr-int/lit8 vx, vy, lit8
ushr-int/lit8 vx, vy, lit8
```

```
nop
```

```
move vx,vy
move/from16 vx,vy
move/16
move-wide
move-wide/from16 vx,vy
move-wide/16
move-object vx,vy
move-object/from16 vx,vy
move-object/16
move-result vx
move-result-wide vx
move-result-object vx
move-exception vx
```

```
return-void
return vx
return-wide vx
return-object vx
```

```
const/4 vx,lit4
const/16 vx,lit16
const vx, lit32
const/high16 v0, lit16
const-wide/16 vx, lit16
const-wide/32 vx, lit32
const-wide vx, lit64
const-wide/high16 vx,lit16
const-string vx,string_id
const-string-jumbo
const-class vx,type_id
```

```
monitor-enter vx
monitor-exit
```

```
check-cast vx, type_id
```

```
instance-of vx,vy,type_id
```

```
array-length vx,vy
```

```
new-instance vx,type
new-array vx,vy,type_id
filled-new-array {parameters},type_id
filled-new-array-range {vx..vy},type_id
fill-array-data vx,array_data_offset
```

```
throw vx
```

```
goto target
goto/16 target
goto/32 target
```

```
packed-switch vx,table
sparse-switch vx,table
```

```
cmpl-float
cmpg-float vx, vy, vz
cmpl-double vx,vy,vz
cmpg-double vx, vy, vz
cmp-long vx, vy, vz
```

```
if-eq vx,vy,target
if-ne vx,vy,target
if-lt vx,vy,target
if-ge vx, vy,target
if-gt vx,vy,target
if-le vx,vy,target
if-eqz vx,target
if-nez vx,target
if-ltz vx,target
if-gez vx,target
if-gtz vx,target
if-lez vx,target
```

```
aget vx,vy,vz
aget-wide vx,vy,vz
aget-object vx,vy,vz
aget-boolean vx,vy,vz
aget-byte vx,vy,vz
aget-char vx, vy,vz
aget-short vx,vy,vz
```

```
aput vx,vy,vz
aput-wide vx,vy,vz
aput-object vx,vy,vz
aput-boolean vx,vy,vz
aput-byte vx,vy,vz
aput-char vx,vy,vz
aput-short vx,vy,vz
```

```
iget vx, vy, field_id
iget-wide vx,vy,field_id
iget-object vx,vy,field_id
iget-boolean vx,vy,field_id
iget-byte vx,vy,field_id
iget-char vx,vy,field_id
iget-short vx,vy,field_id
```

```
iput vx,vy, field_id
iput-wide vx,vy, field_id
iput-object vx,vy,field_id
iput-boolean vx,vy, field_id
iput-byte vx,vy,field_id
iput-char vx,vy,field_id
iput-short vx,vy,field_id
```

```
sget vx,field_id
sget-wide vx, field_id
sget-object vx,field_id
sget-boolean vx,field_id
sget-byte vx,field_id
sget-char vx,field_id
sget-short vx,field_id
```

```
sput vx, field_id
sput-wide vx, field_id
sput-object vx,field_id
sput-boolean vx,field_id
sput-byte vx,field_id
sput-char vx,field_id
sput-short vx,field_id
```

```
invoke-virtual { parameters },methodtocall
invoke-super {parameter},methodtocall
invoke-direct { parameters },methodtocall
invoke-static {parameters},methodtocall
invoke-interface {parameters},methodtocall
```

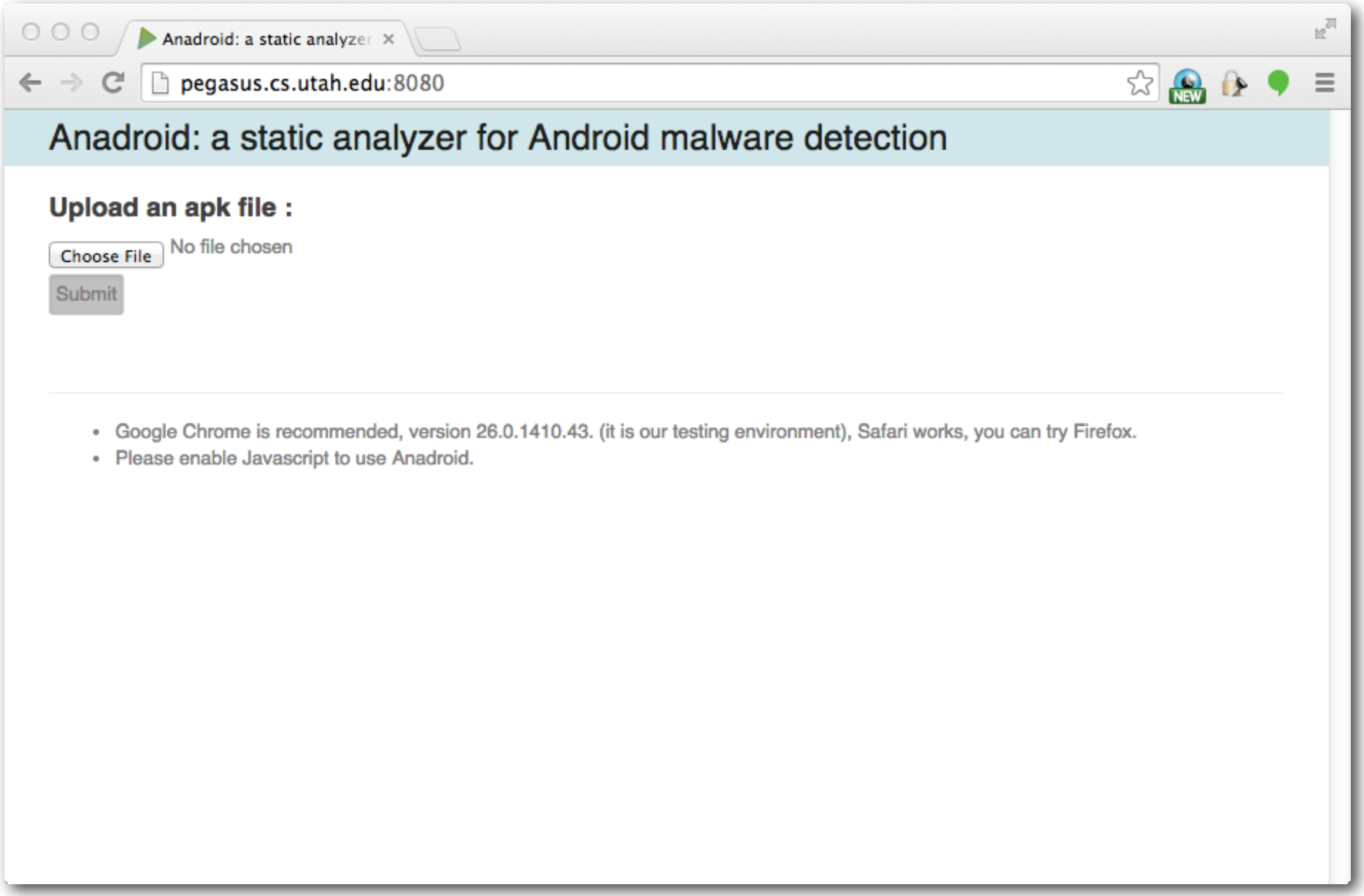
```
invoke-virtual/range {vx..vy},methodtocall
invoke-super/range {vx..vy},methodtocall
invoke-direct/range {vx..vy},methodtocall
invoke-static/range {vx..vy},methodtocall
invoke-interface/range {vx..vy},methodtocall
```

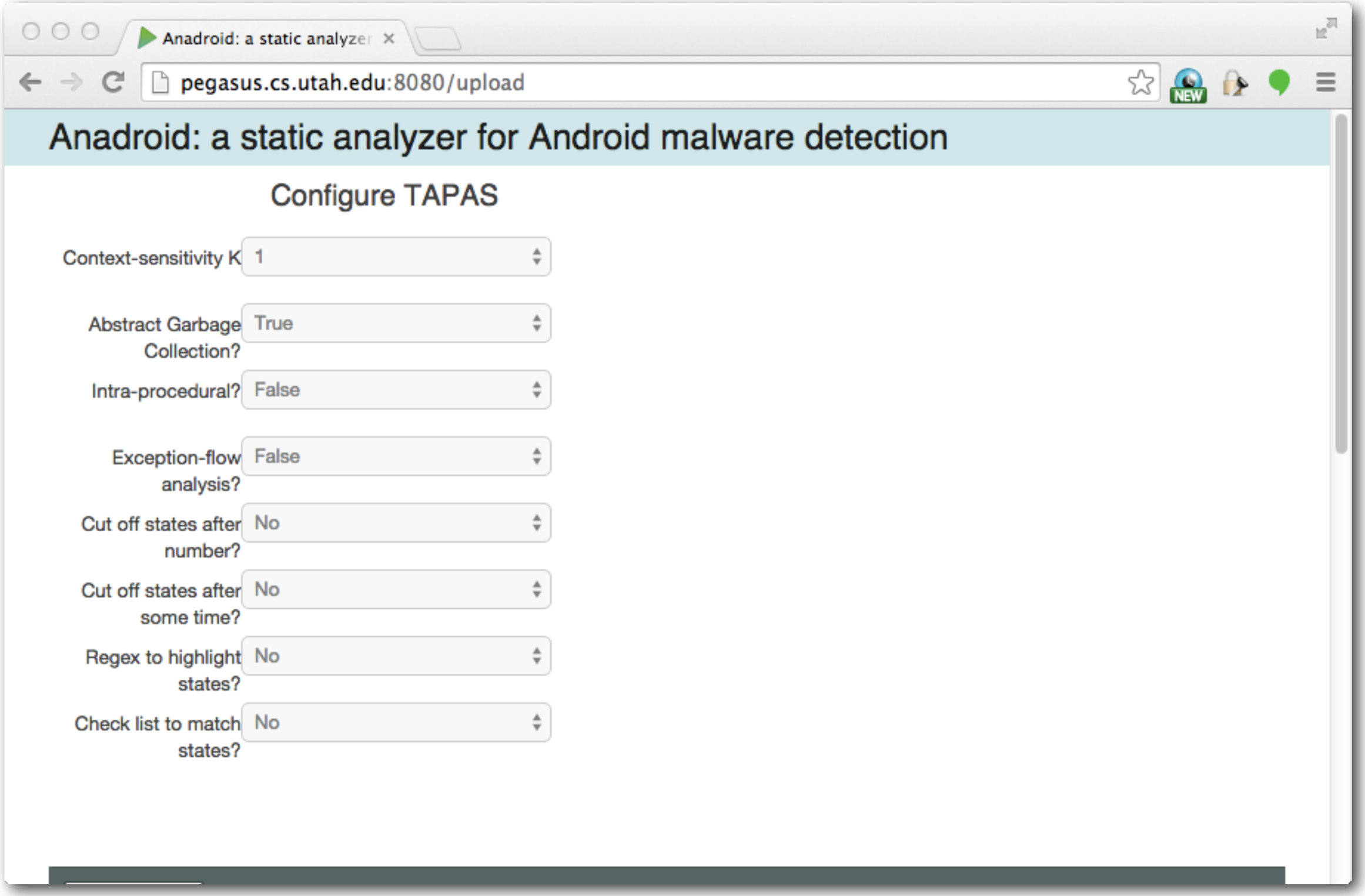
```
neg-int vx,vy
not-int vx,vy
neg-long vx,vy
not-long vx,vy
neg-float vx,vy
neg-double vx,vy
```

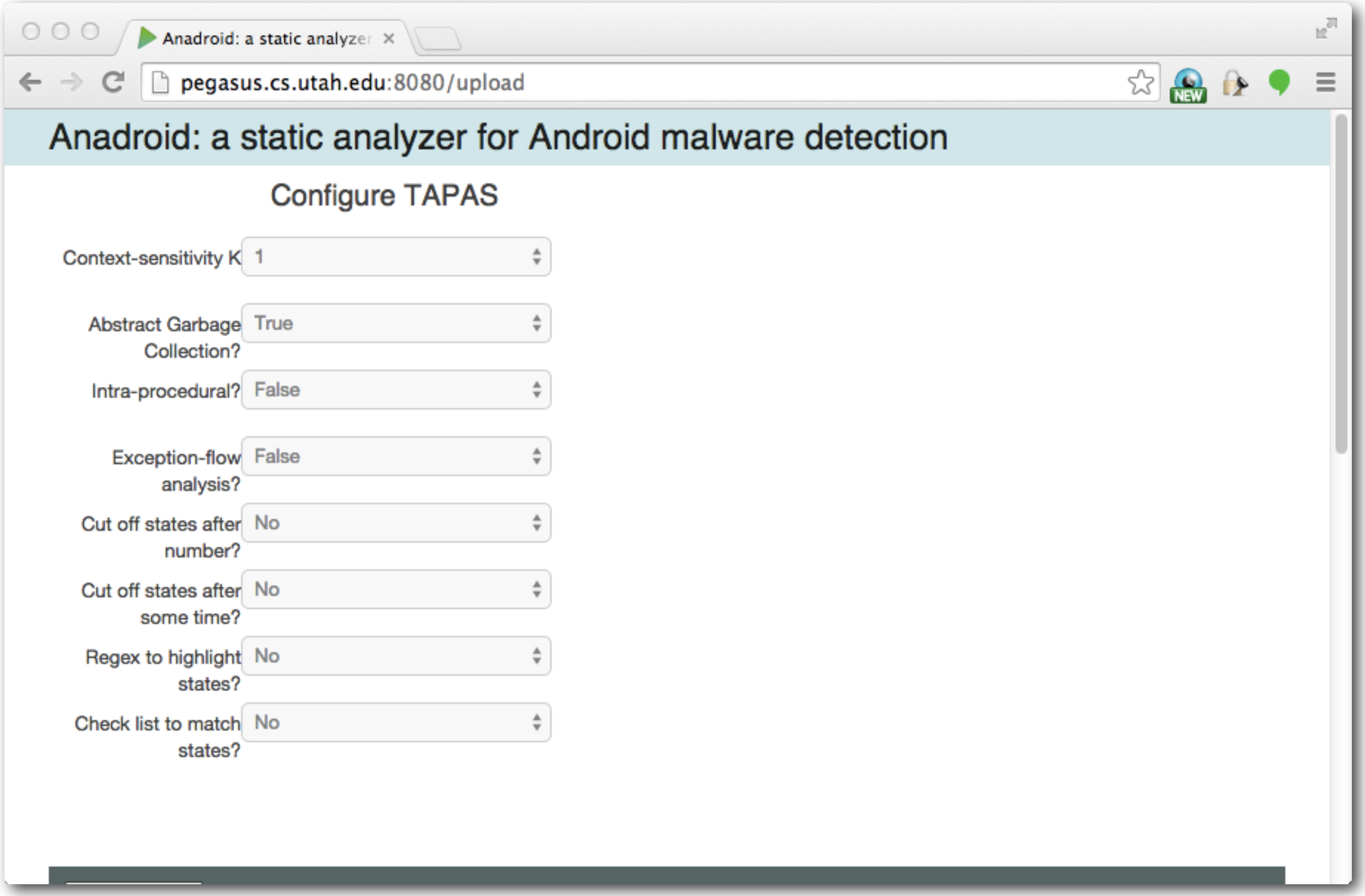
```
int-to-long vx, vy
int-to-float vx, vy
int-to-double vx, vy
long-to-int vx,vy
long-to-float vx, vy
long-to-double vx, vy
float-to-int vx, vy
float-to-long vx,vy
float-to-double vx, vy
double-to-int vx, vy
double-to-long vx, vy
double-to-float vx, vy
int-to-byte vx,vy
int-to-char vx,vy
int-to-short vx,vy
```

```
add-int vx,vy,vz
sub-int vx,vy,vz
mul-int vx, vy, vz
div-int vx,vy,vz
rem-int vx,vy,vz
```

```
and-int vx, vy, vz
or-int vx, vy, vz
xor-int vx, vy, vz
shl-int vx, vy, vz
shr-int vx, vy, vz
ushr-int vx, vy, vz
add-long vx, vy, vz
sub-long vx,vy,vz
mul-long vx,vy,vz
div-long vx, vy, vz
rem-long vx,vy,vz
and-long vx, vy, vz
or-long vx, vy, vz
xor-long vx, vy, vz
shl-long vx, vy, vz
shr-long vx,vy,vz
ushr-long vx, vy, vz
add-float vx,vy,vz
sub-float vx,vy,vz
mul-float vx, vy, vz
div-float vx, vy, vz
rem-float vx,vy,vz
add-double vx,vy,vz
sub-double vx,vy,vz
mul-double vx, vy, vz
div-double vx, vy, vz
rem-double vx,vy,vz
add-int/2addr vx,vy
sub-int/2addr vx,vy
mul-int/2addr vx,vy
div-int/2addr vx,vy
rem-int/2addr vx,vy
and-int/2addr vx, vy
or-int/2addr vx, vy
xor-int/2addr vx, vy
shl-int/2addr vx, vy
shr-int/2addr vx, vy
ushr-int/2addr vx, vy
add-long/2addr vx,vy
sub-long/2addr vx,vy
mul-long/2addr vx,vy
div-long/2addr vx, vy
rem-long/2addr vx,vy
and-long/2addr vx, vy
or-long/2addr vx, vy
xor-long/2addr vx, vy
shl-long/2addr vx, vy
shr-long/2addr vx, vy
ushr-long/2addr vx, vy
add-float/2addr vx,vy
sub-float/2addr vx,vy
mul-float/2addr vx, vy
div-float/2addr vx, vy
rem-float/2addr vx,vy
add-double/2addr vx, vy
sub-double/2addr vx, vy
mul-double/2addr vx, vy
div-double/2addr vx, vy
rem-double/2addr vx,vy
add-int/lit16 vx,vy,lit16
sub-int/lit16 vx,vy,lit16
mul-int/lit16 vx,vy,lit16
div-int/lit16 vx,vy,lit16
rem-int/lit16 vx,vy,lit16
and-int/lit16 vx,vy,lit16
or-int/lit16 vx,vy,lit16
xor-int/lit16 vx,vy,lit16
add-int/lit8 vx,vy,lit8
sub-int/lit8 vx,vy,lit8
mul-int/lit8 vx,vy,lit8
div-int/lit8 vx,vy,lit8
rem-int/lit8 vx,vy,lit8
and-int/lit8 vx,vy,lit8
or-int/lit8 vx, vy, lit8
xor-int/lit8 vx, vy, lit8
shl-int/lit8 vx, vy, lit8
shr-int/lit8 vx, vy, lit8
ushr-int/lit8 vx, vy, lit8
```


Anadroid: a static analyzer
pegasus.cs.utah.edu:8080
Anadroid: a static analyzer for Android malware detection
Upload an apk file :
Choose File
No file chosen
Submit
Google Chrome is recommended, version 26.0.1410.43. (it is our testing environment), Safari works, you can try Firefox.
Please enable Javascript to use Anadroid.


Anadroid: a static analyzer
pegasus.cs.utah.edu:8080/upload
Anadroid: a static analyzer for Android malware detection
Configure TAPAS
Context-sensitivity K
1
Abstract Garbage Collection?
True
Intra-procedural?
False
Exception-flow analysis?
False
Cut off states after number?
No
Cut off states after some time?
No
Regex to highlight states?
No
Check list to match states?
No


Anadroid: a static analyzer
pegasus.cs.utah.edu:8080/upload
Anadroid: a static analyzer for Android malware detection
Configure TAPAS
Context-sensitivity K
1
Abstract Garbage Collection?
True
Intra-procedural?
False
Exception-flow analysis?
False
Cut off states after number?
No
Cut off states after some time?
No
Regex to highlight states?
No
Check list to match states?
No

How to use:

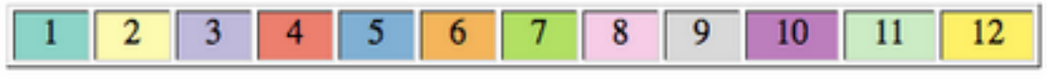
1. **Context sensitivity k**: Last k call sites (0 to up to 2);
2. **Abstract garbage collection**: Garbage collect unreachable abstract addresses, can boost precision and performance. Live register analysis is also turned on when this option is on. It is recommend to turn this option.
3. **Intra-procedural?** : It is specialized to do intra-procedural analysis in the framework. **Turning intra-procedural analysis can be really fast but very imprecise!!! (It is used for a subproject-Intent fuzzer)**
4. : analyze exception-flows. The analyzer may be slowed down. This feature is experimental.
5. **Cut off states after number?**: Stop analyzer after some number of explored states. When this option is on, you will need to input a number in the input text field appeared next line;
6. **Cut off states after some time?**: Stop analyzer after after some minutes or hours. When this option is on, you are required to select the upper bound time limit in the selector appeared in the next line with the default value 5 minutes;
7. **Regex to highlight state?**: The regular expression to search the analysis results, matched states will be highlighted in the color number 8:

   rdpu8 color scheme

   If this option is off, state graph will use default color scheme to color states: 2 for normal states, 4 for tainted states, red for source or sinks states.

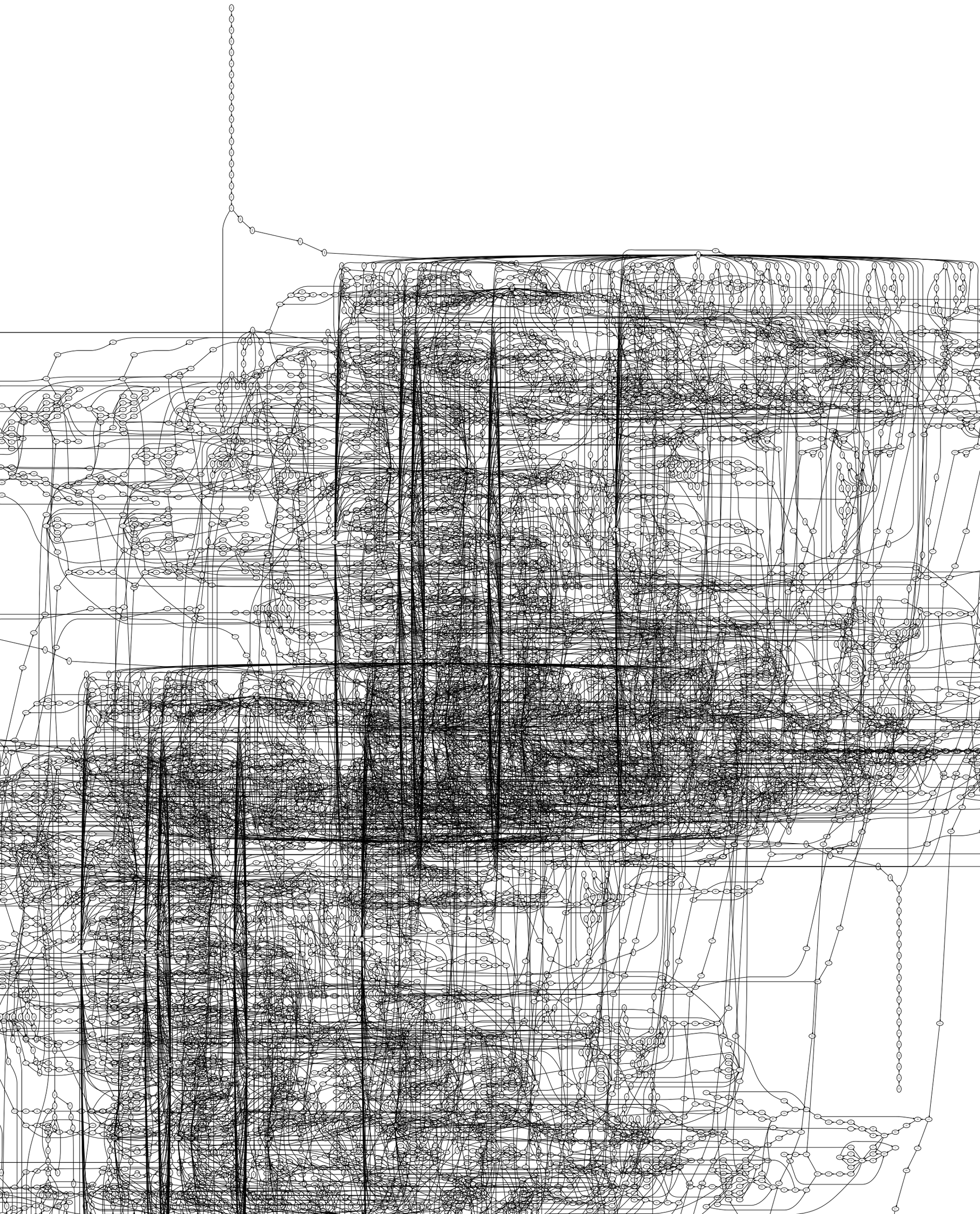
   set312 color scheme

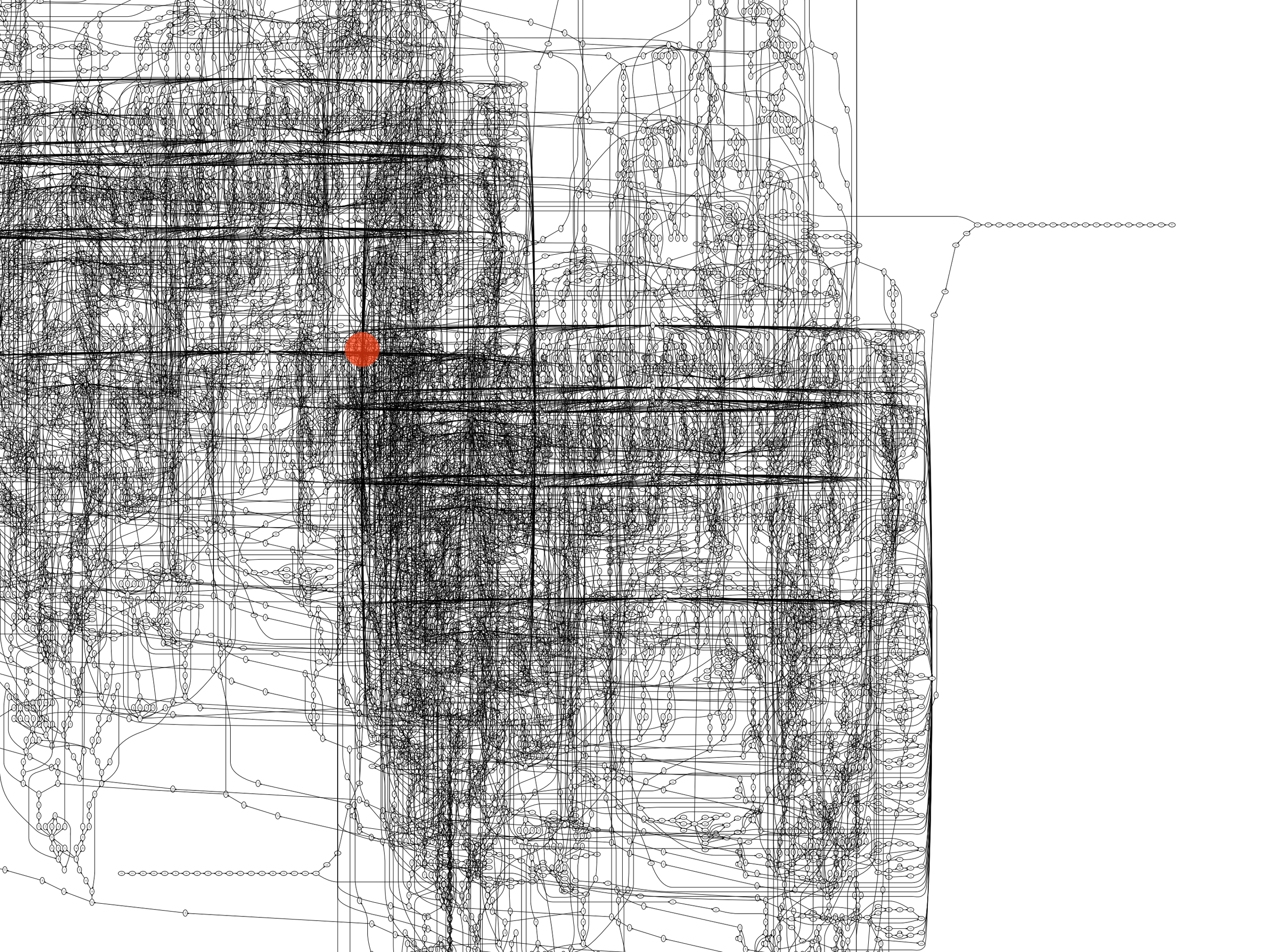
   

8. **Check list to match states?**: Similar to regex matching, this is handy to match analysis result with set of specific kinds of categories. A list of checkboxes will be popped following the option. To reduce confusion, it will use the same color scheme as regex matching do (color number 8 in rdpu8) to color matched states. Otherwise, default color scheme will apply;
9. **Start Anadroid: start analysis.**

Additional note:

1 TB



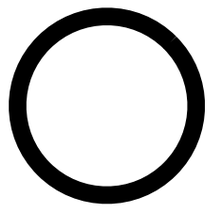
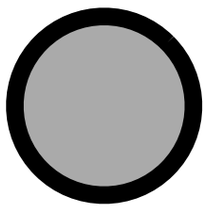
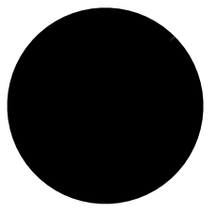


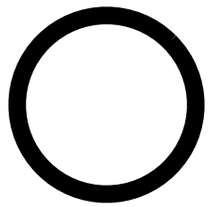
Abstract garbage collection

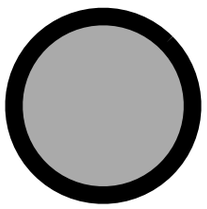
```
o.f ()
```

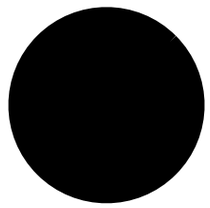
What is ○?

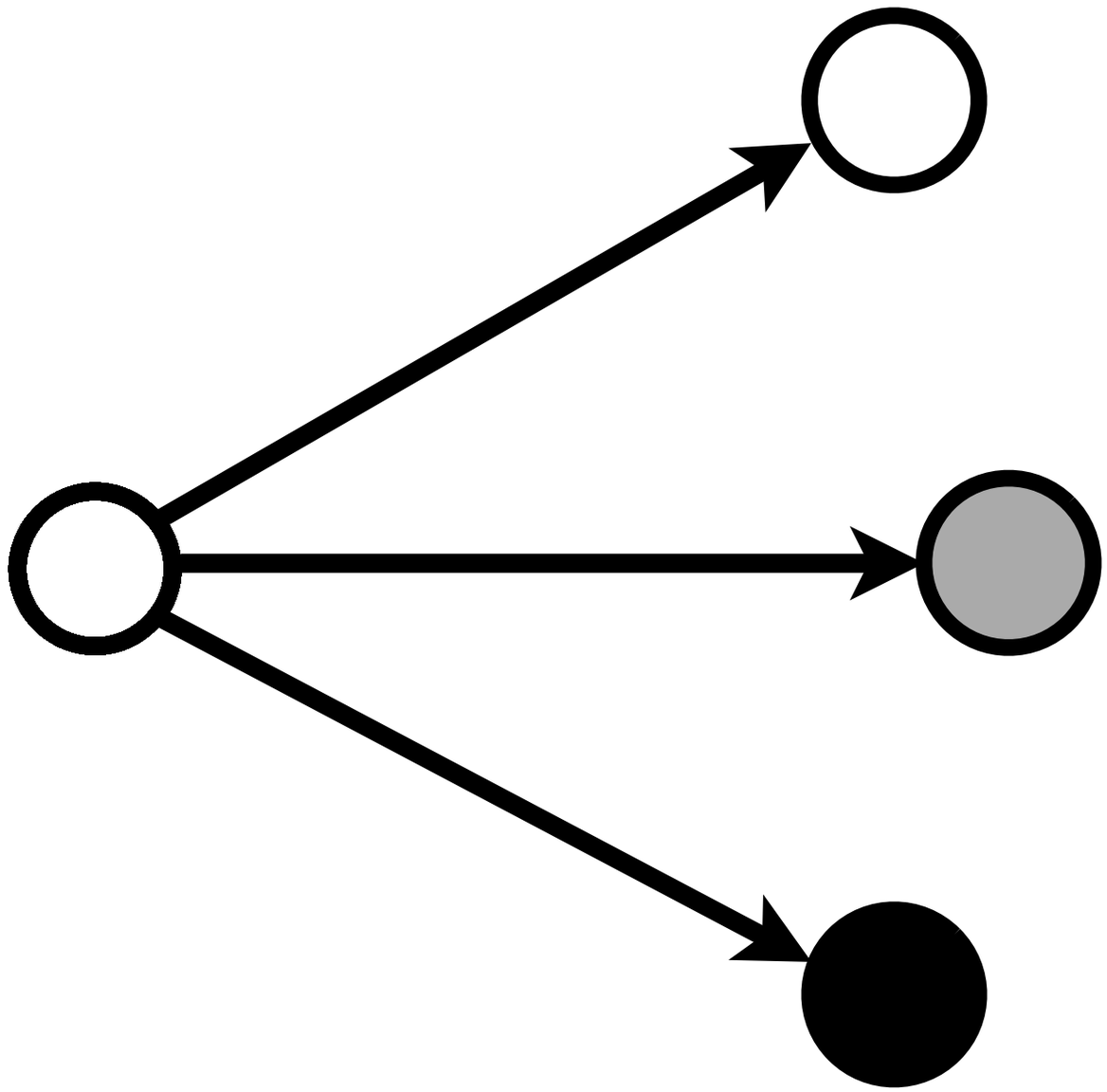
o is or or

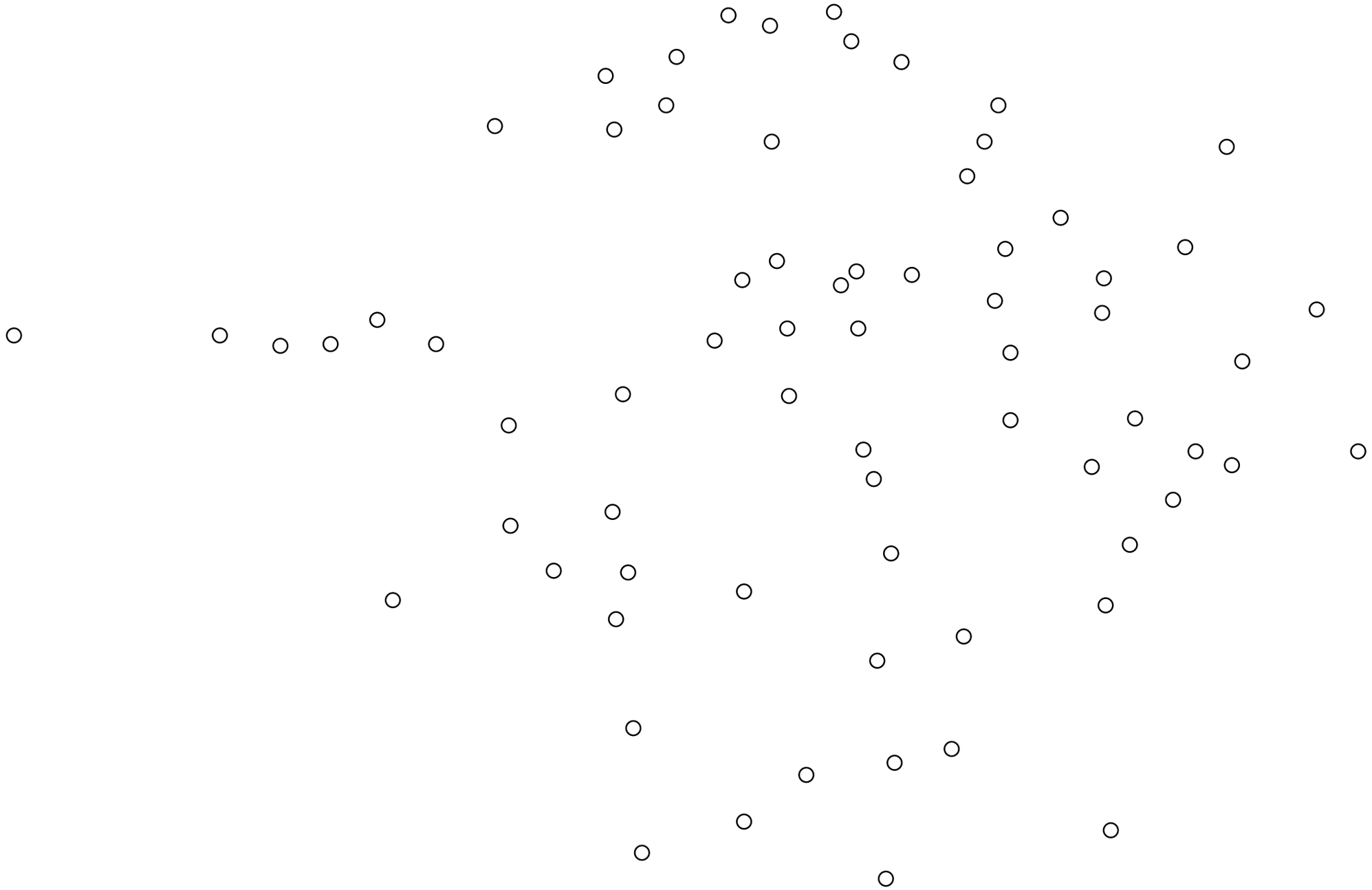
o is  or  or 

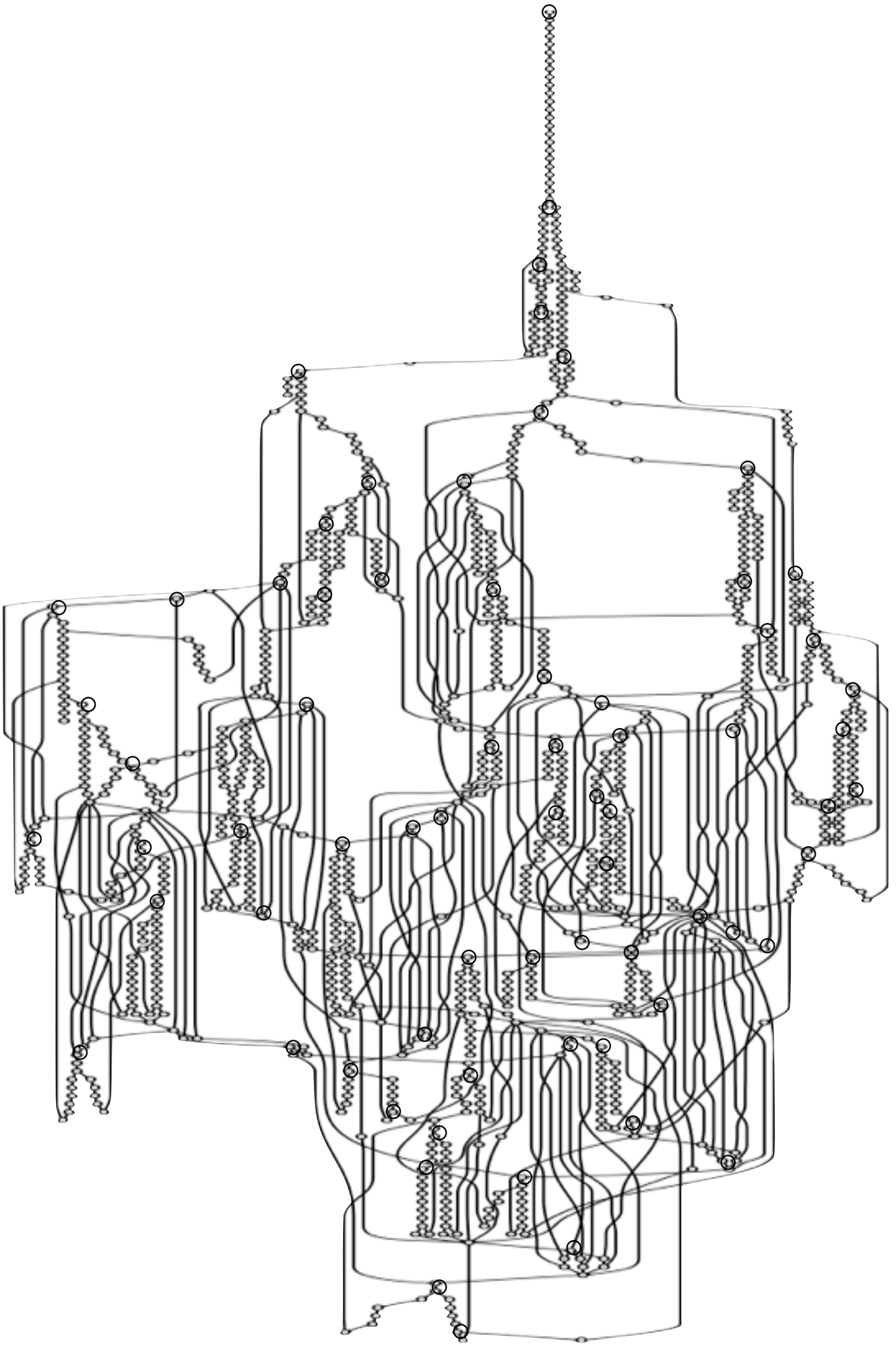










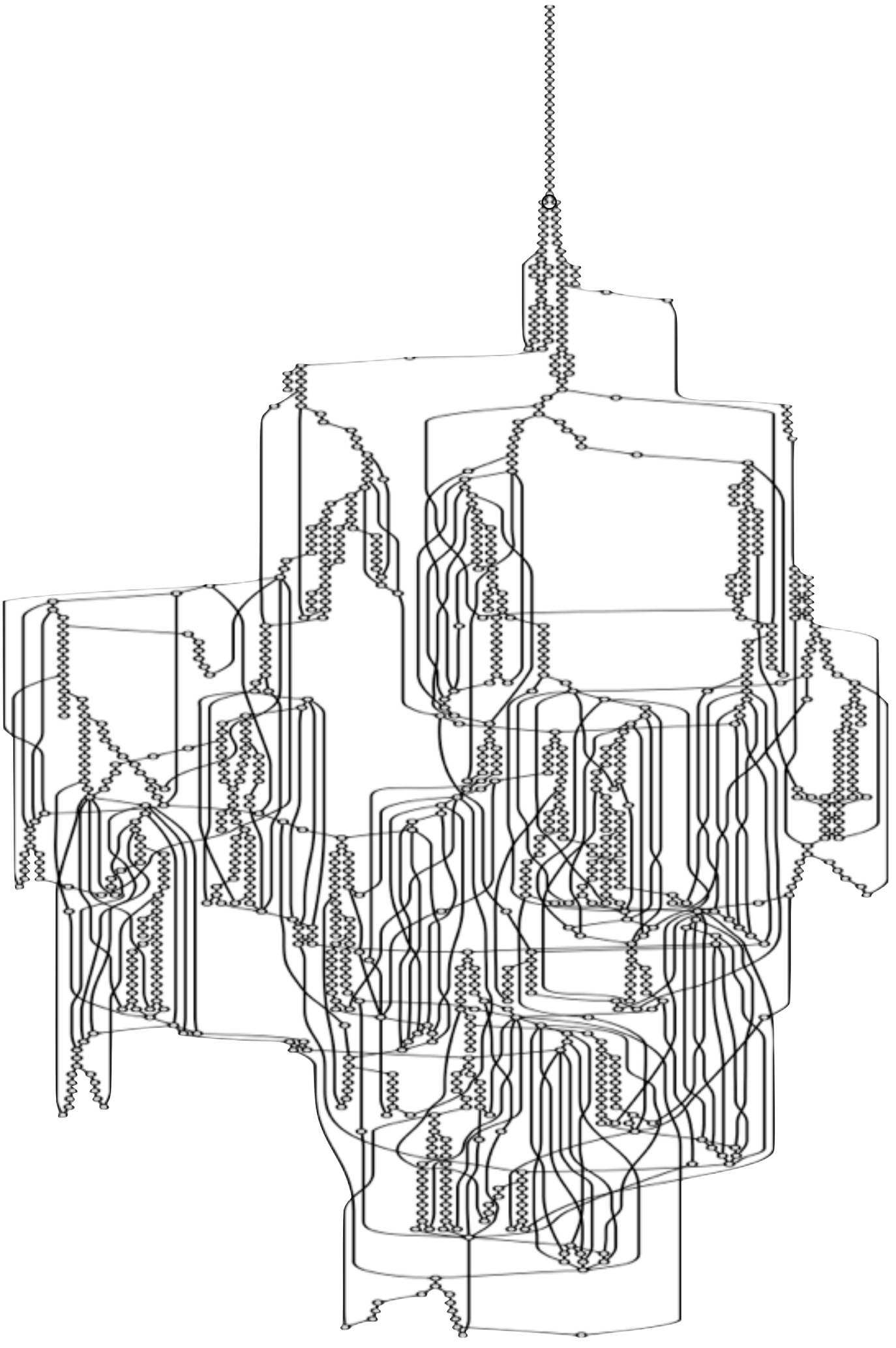


Problem?

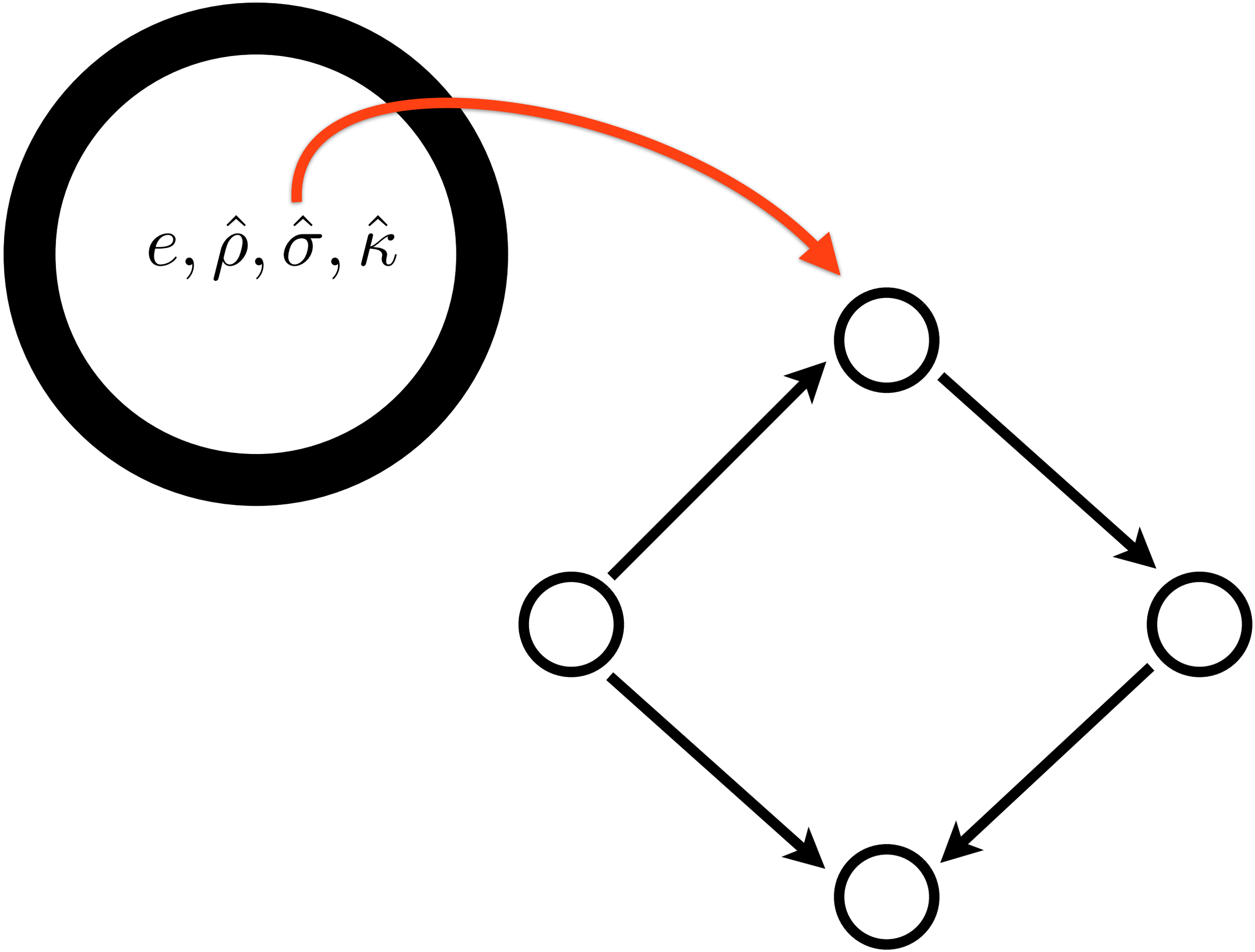
Finite heap.

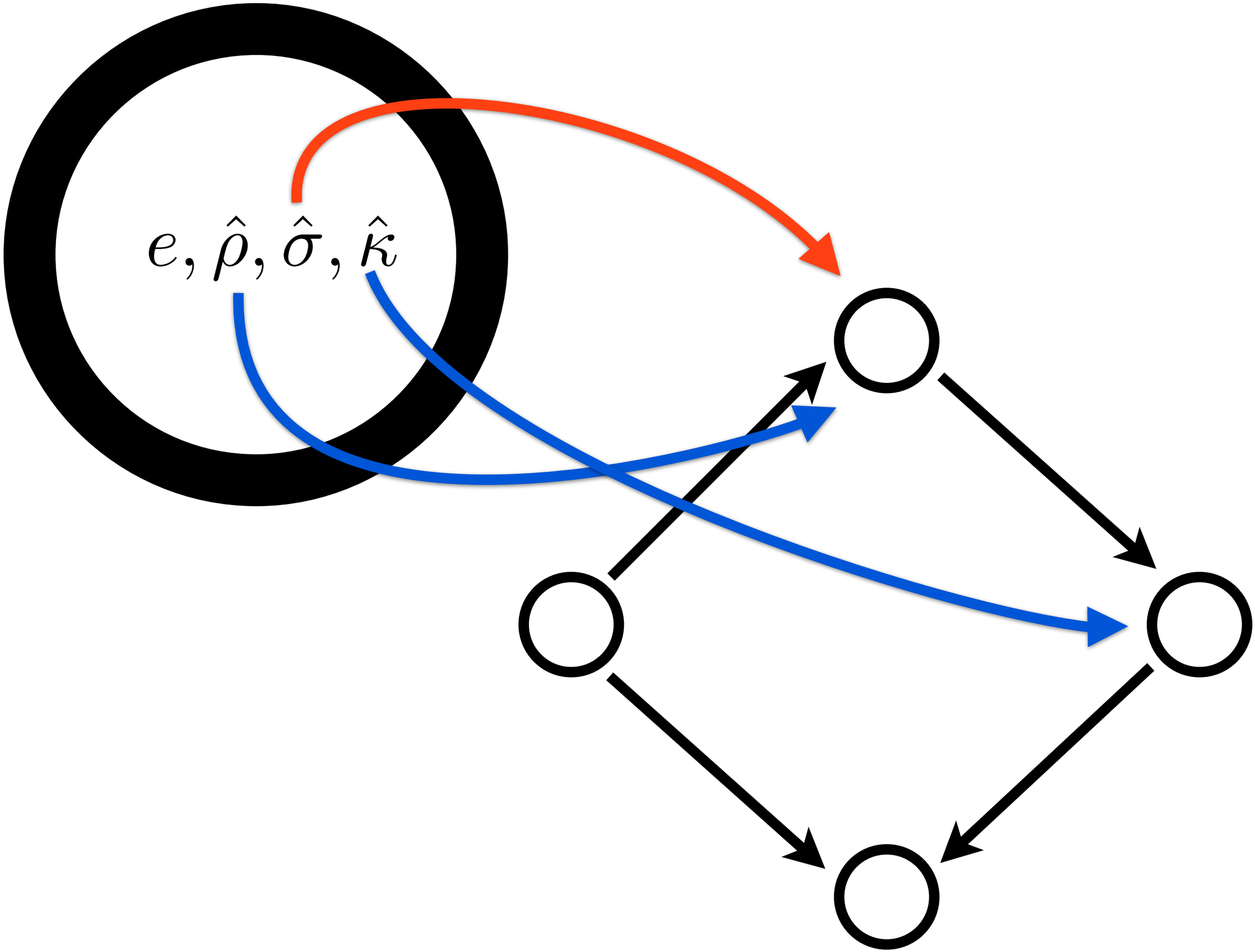
Solution?

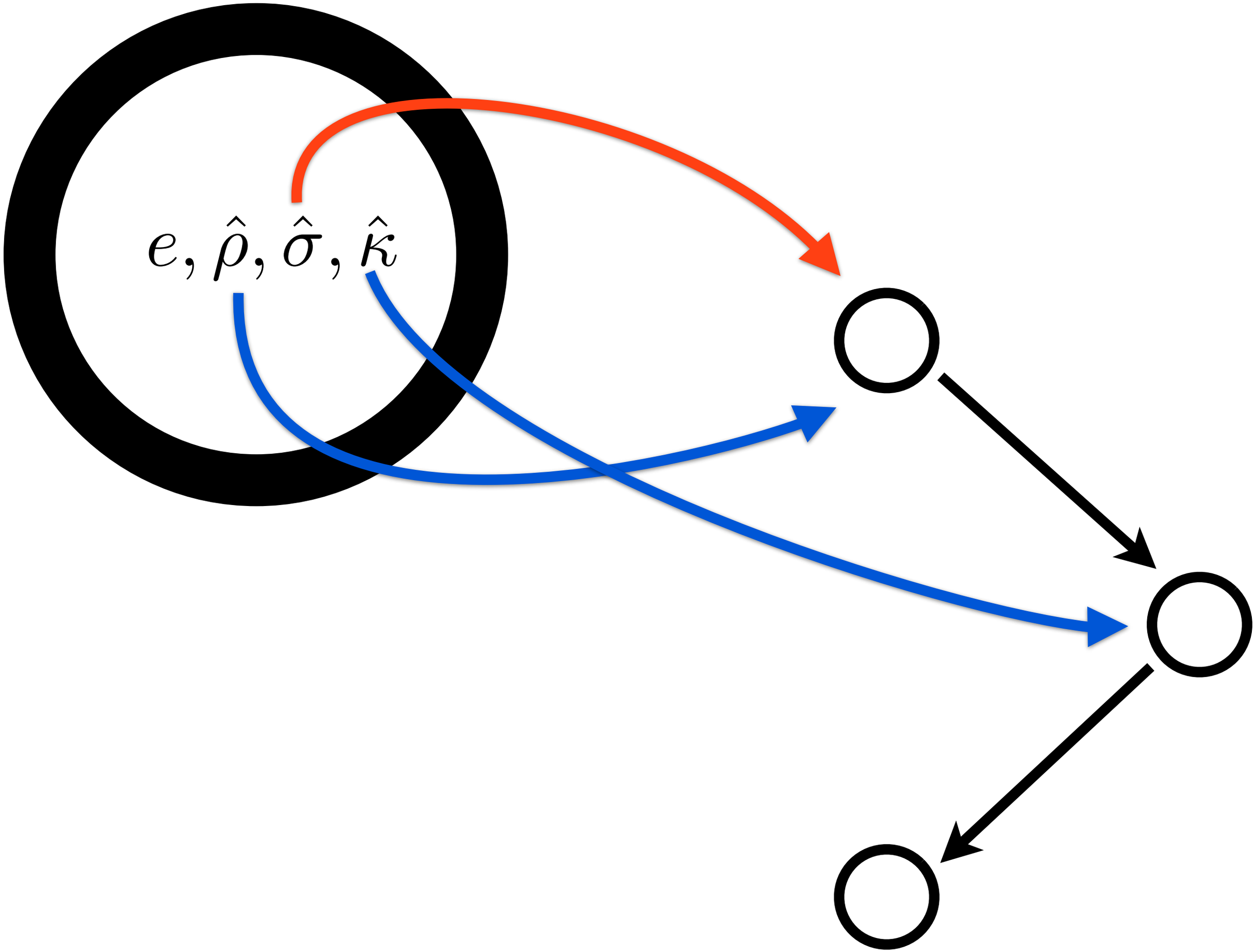
loss garbage.



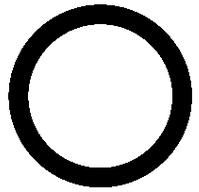

$e, \hat{\rho}, \hat{\sigma}, \hat{\kappa}$

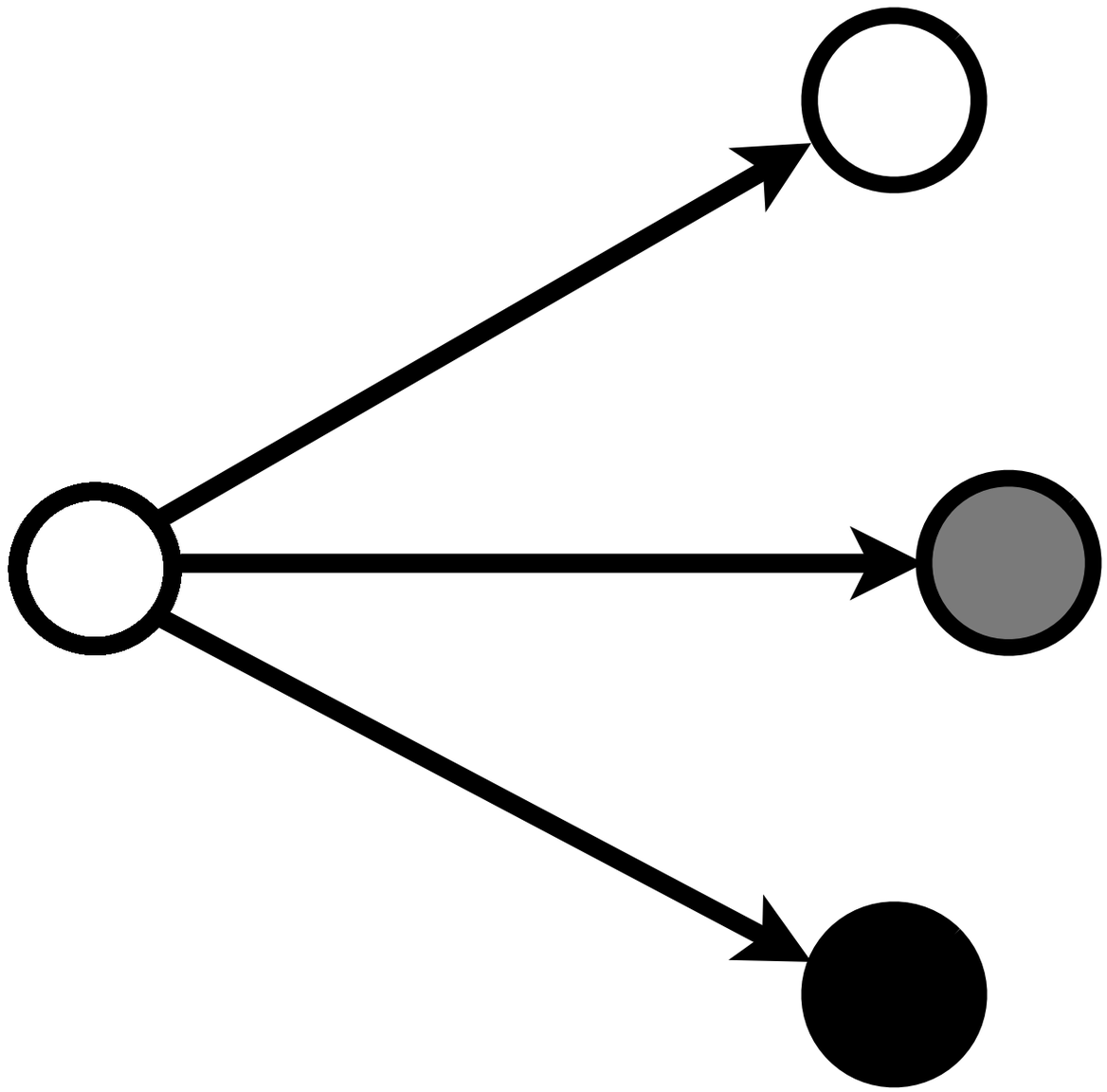

$e, \hat{\rho}, \hat{\sigma}, \hat{\kappa}$

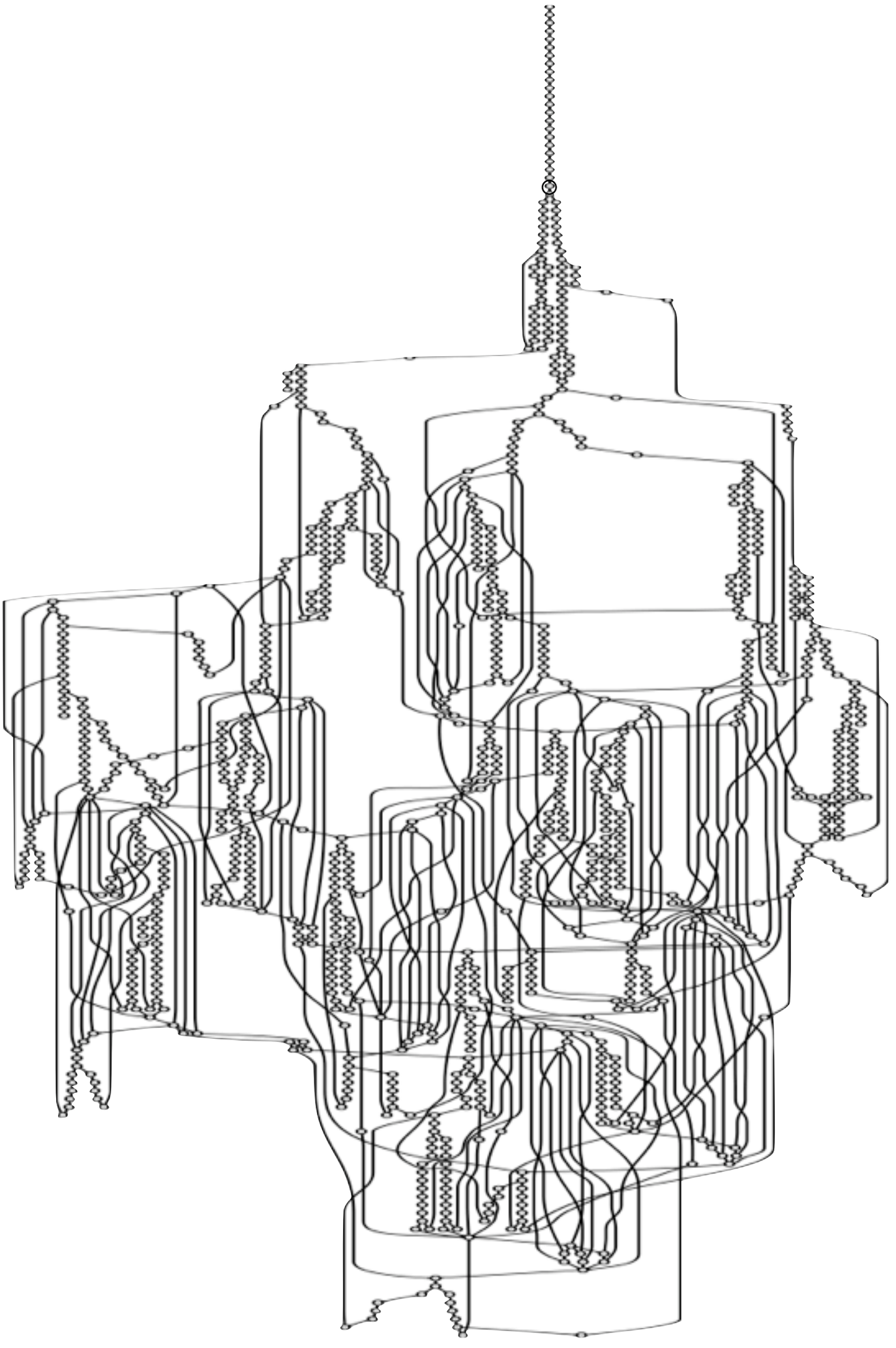

$e, \hat{\rho}, \hat{\sigma}, \hat{\kappa}$

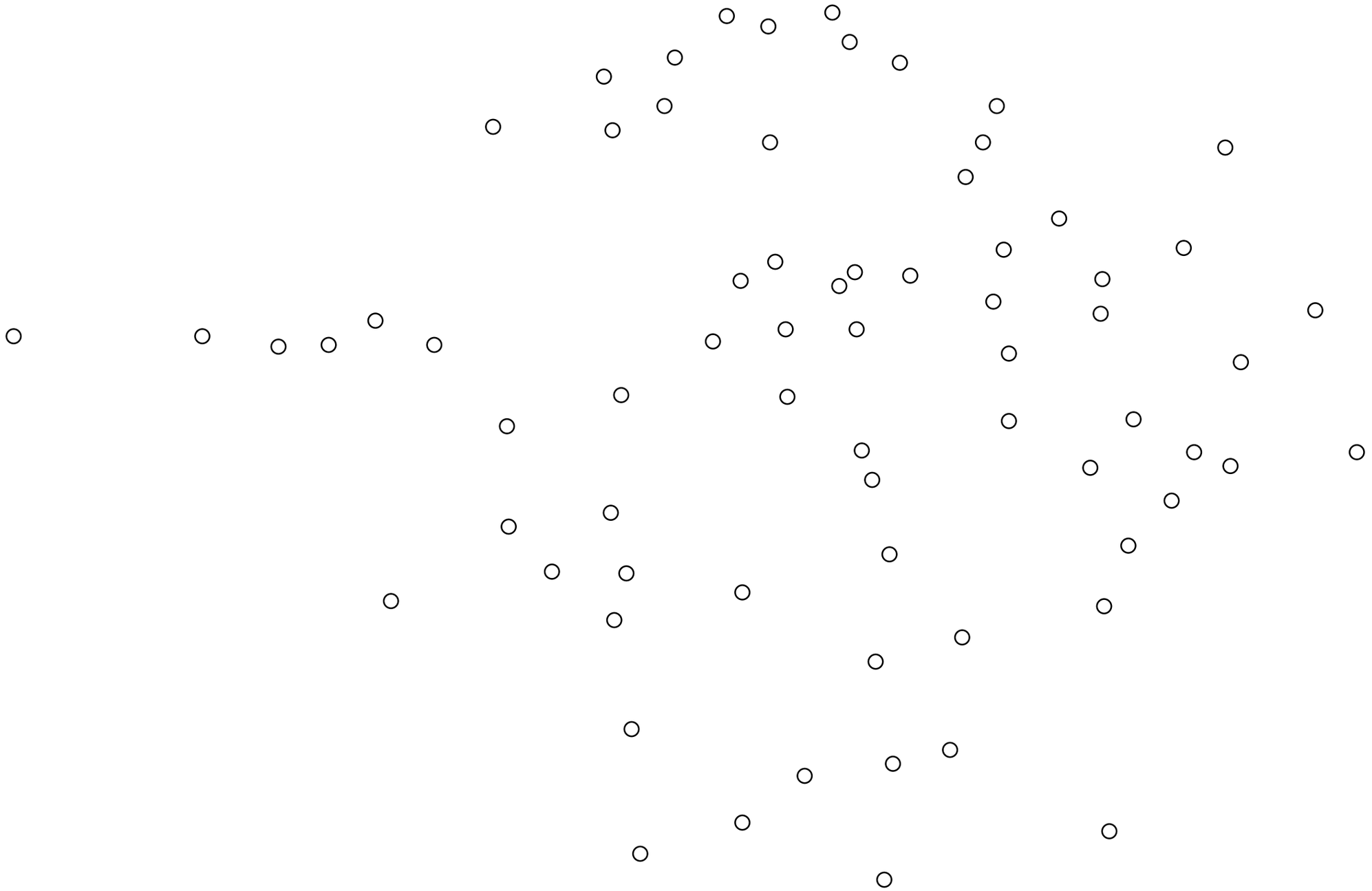

$e, \hat{\rho}, \hat{\sigma}, \hat{\kappa}$

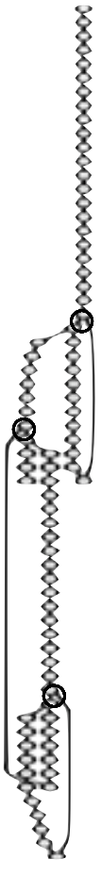

$e, \hat{\rho}, \hat{\sigma}', \hat{\kappa}$



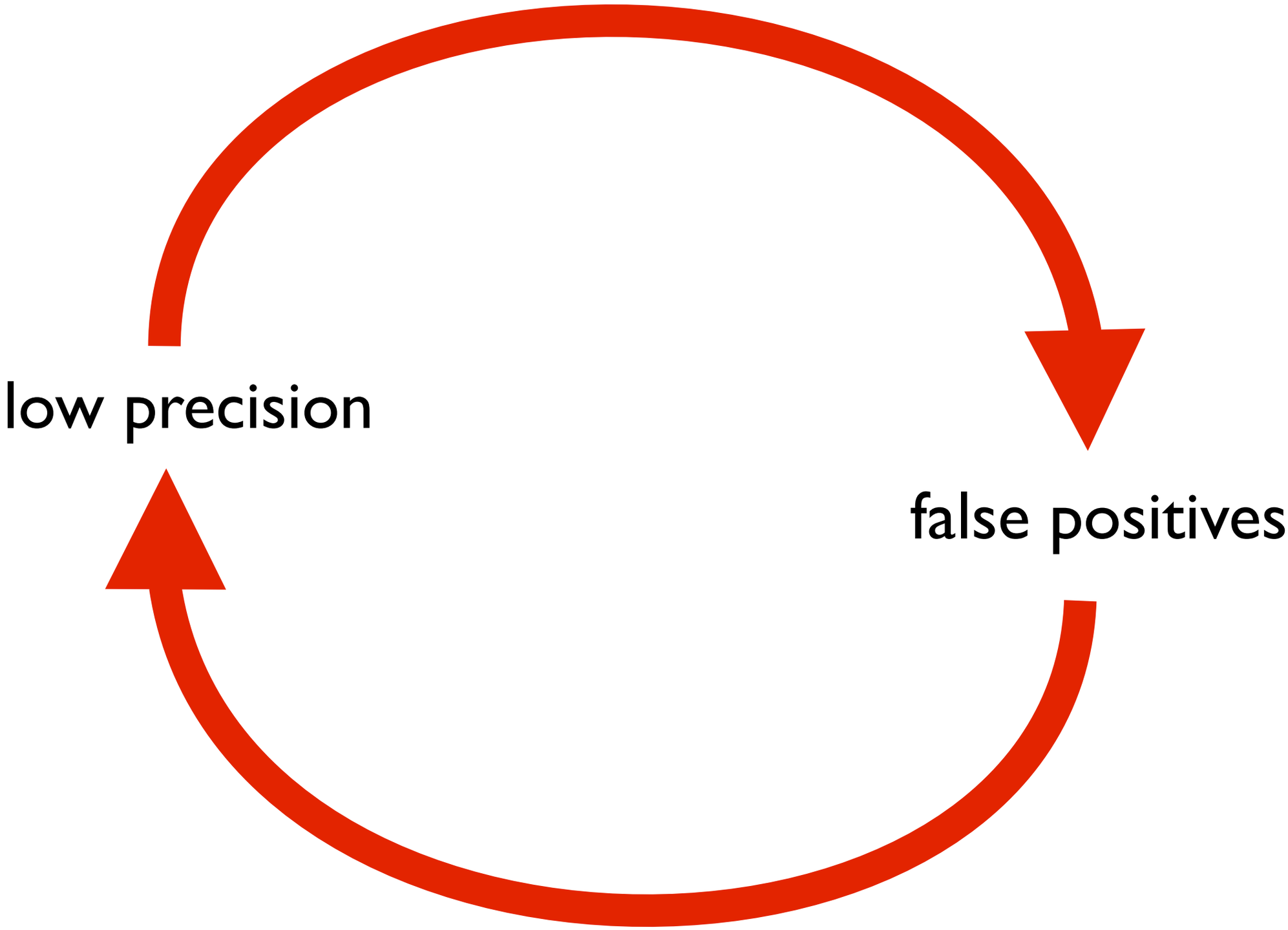




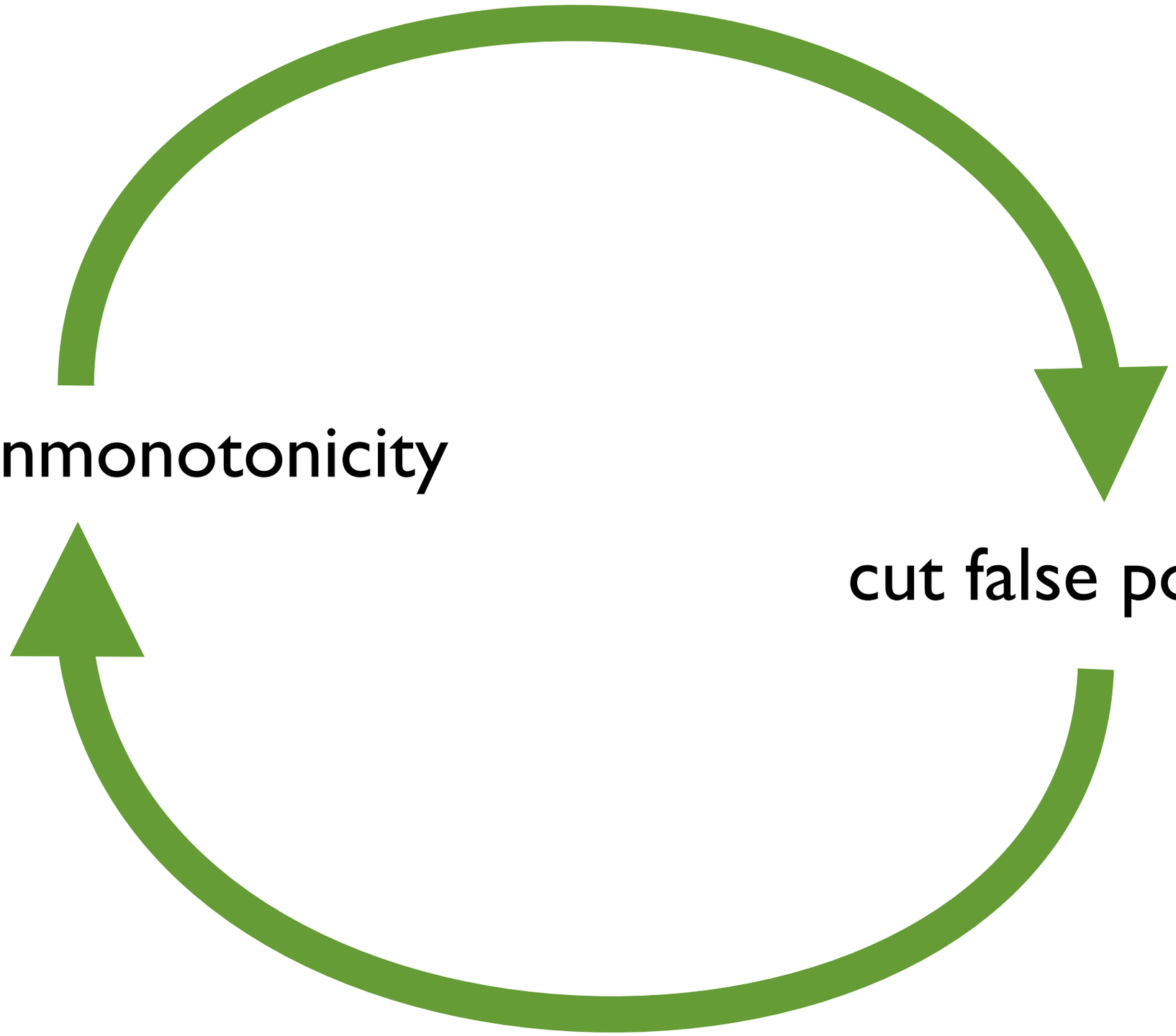




What's happening?


low precision
false positives

local nonmonotonicity

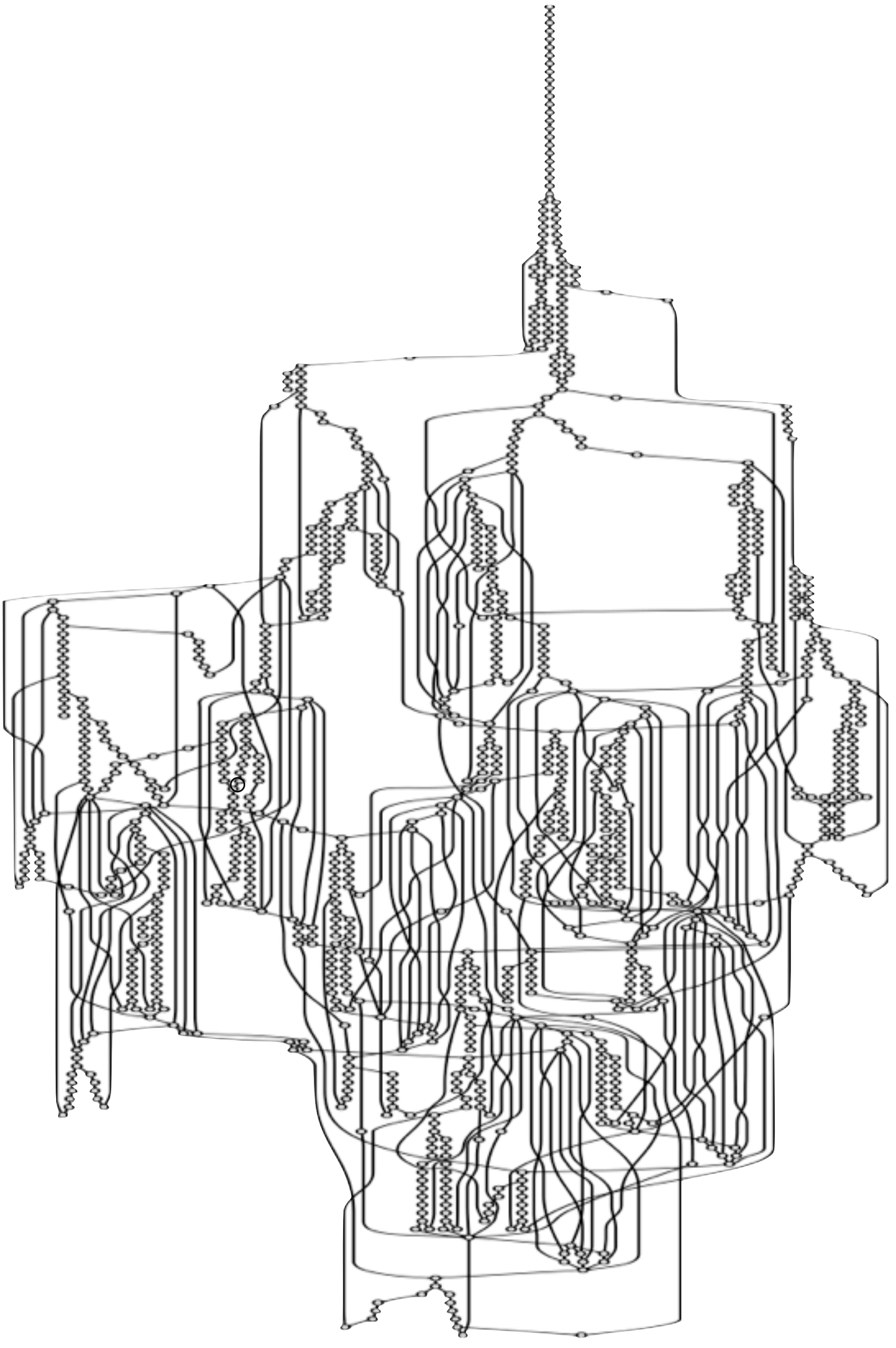


cut false positives

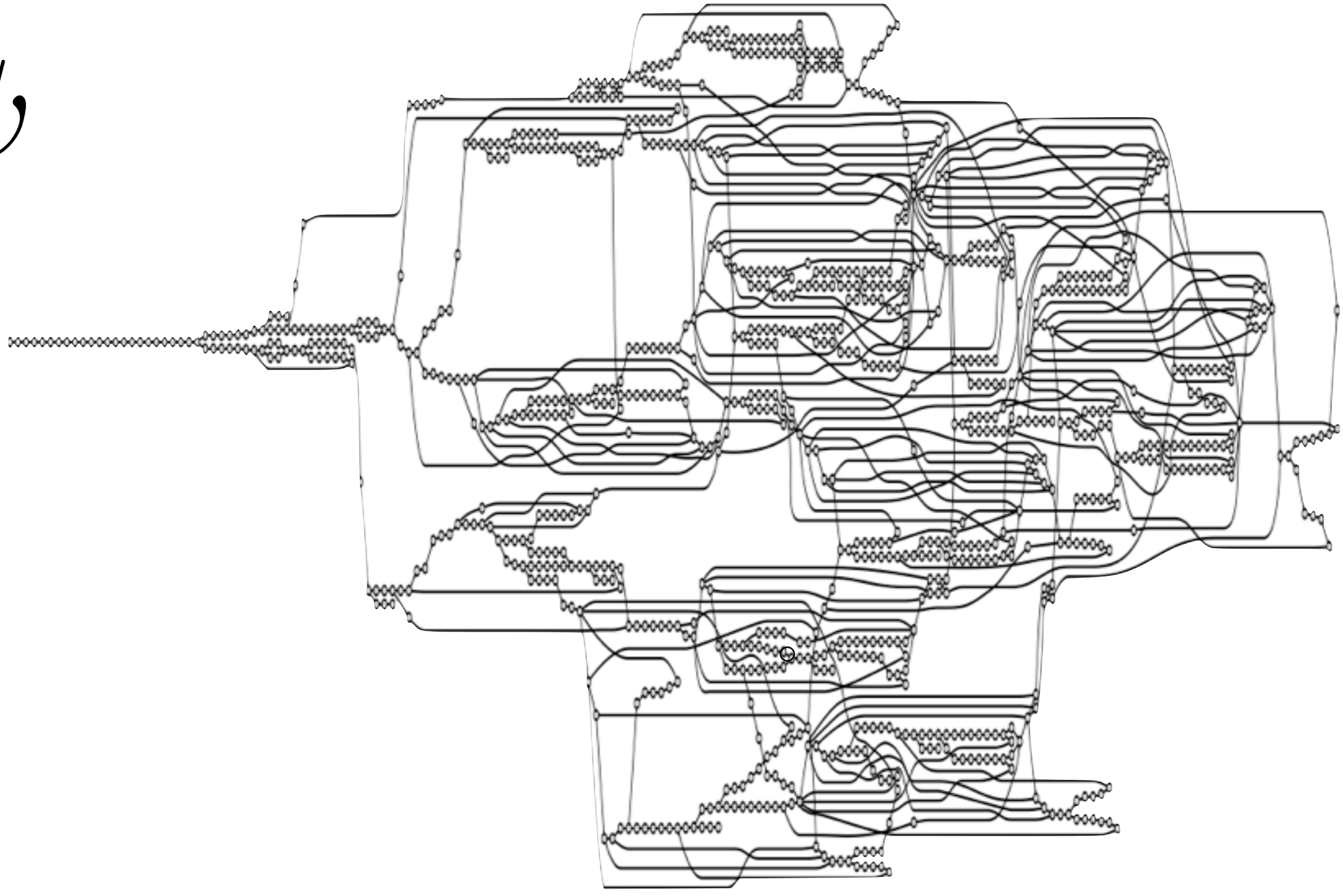
1 GB

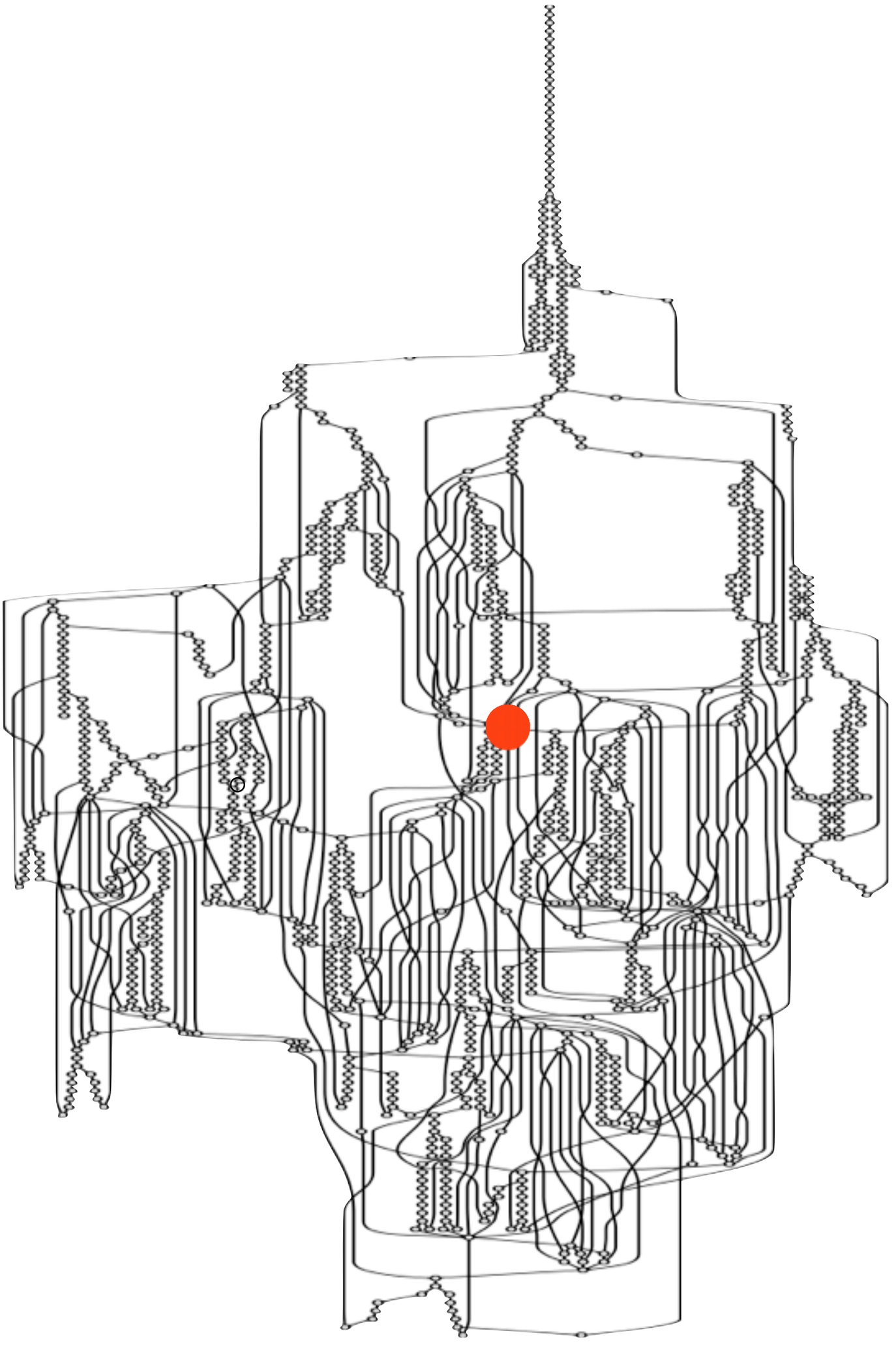

Terminal — zsh — 82×20 — ⌘1
</0>
<282!matt@maptop:~/Copy/APAC-Demo/Anadroid-AWeather>
%
anadroid

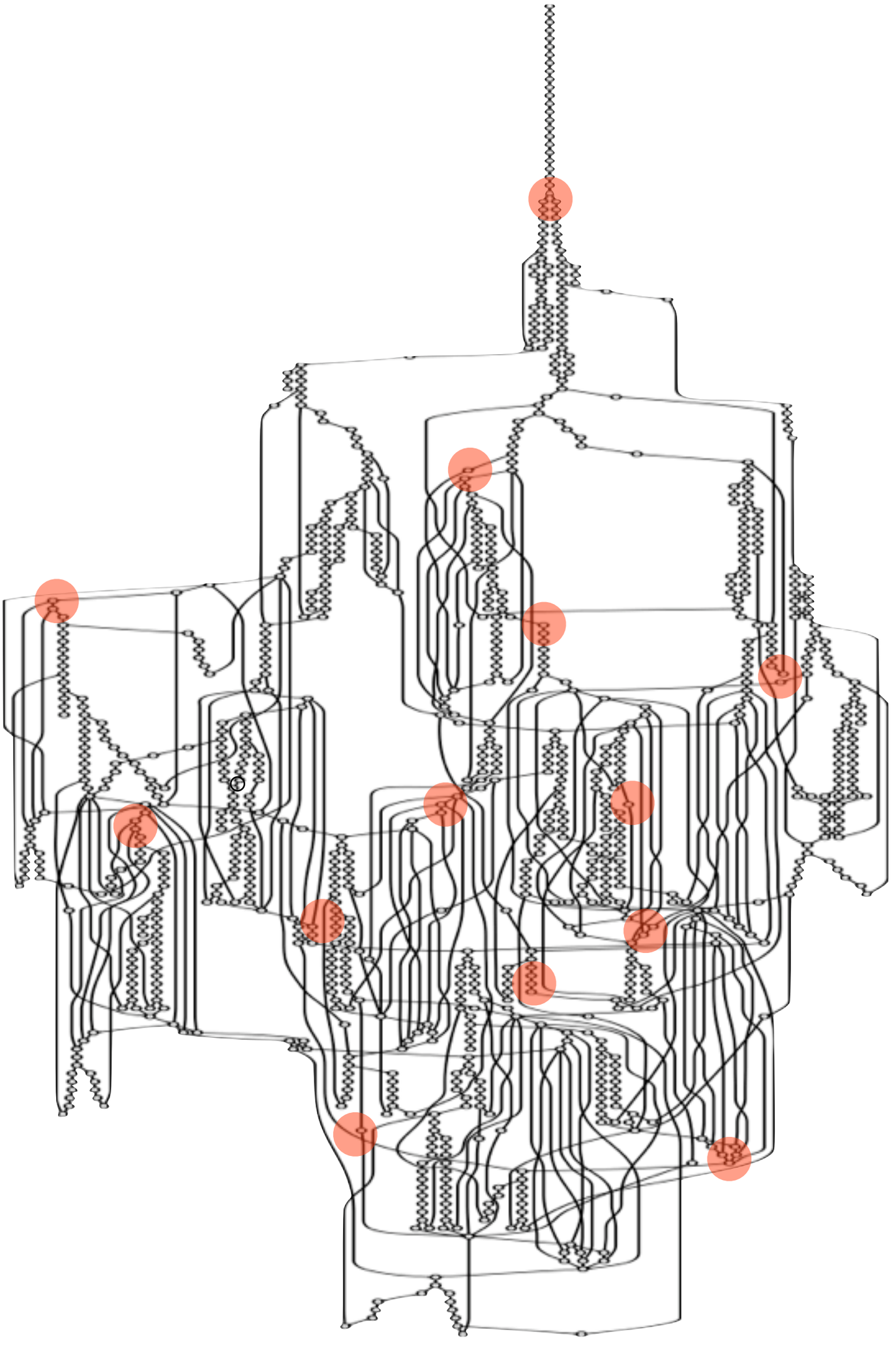
So then...

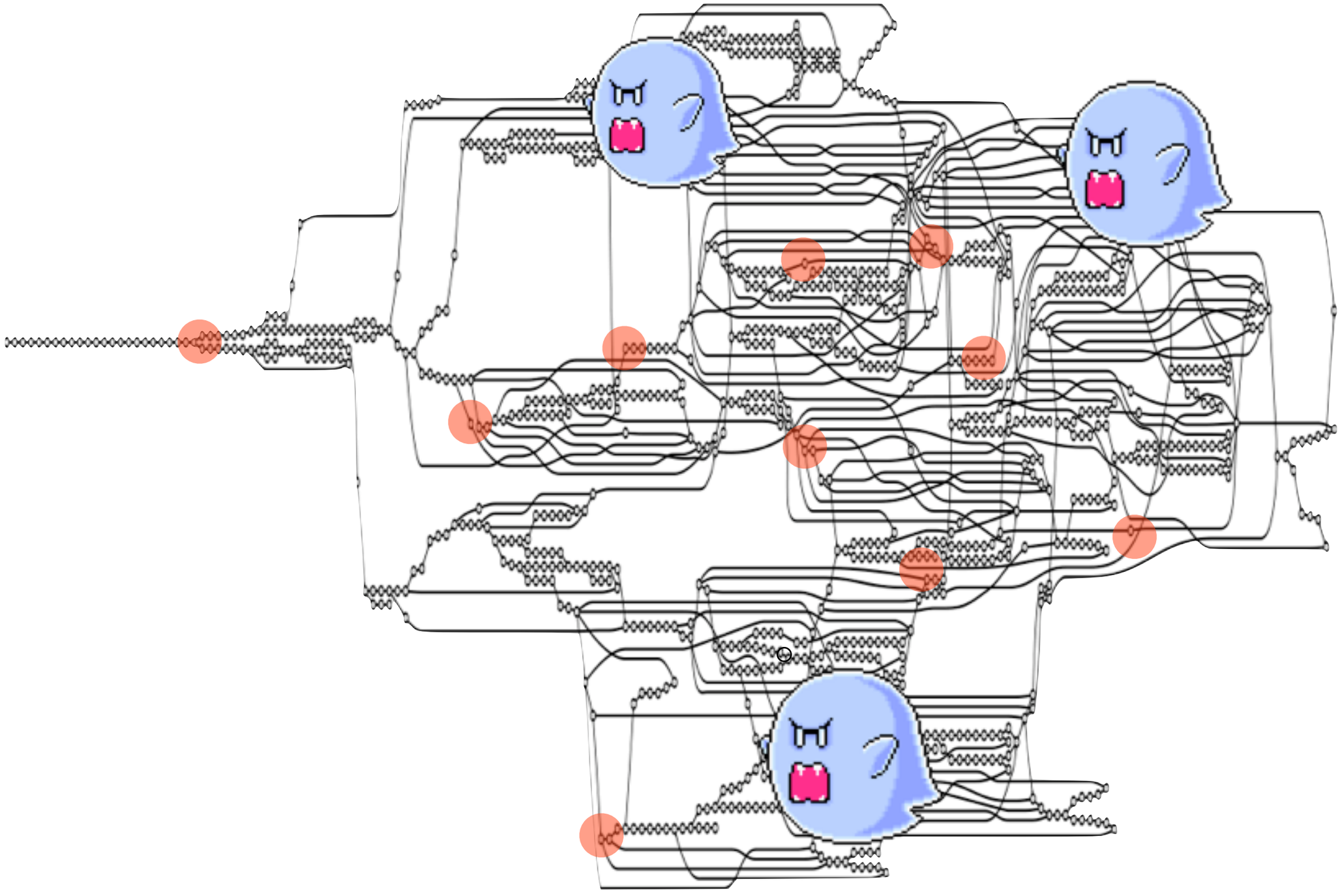


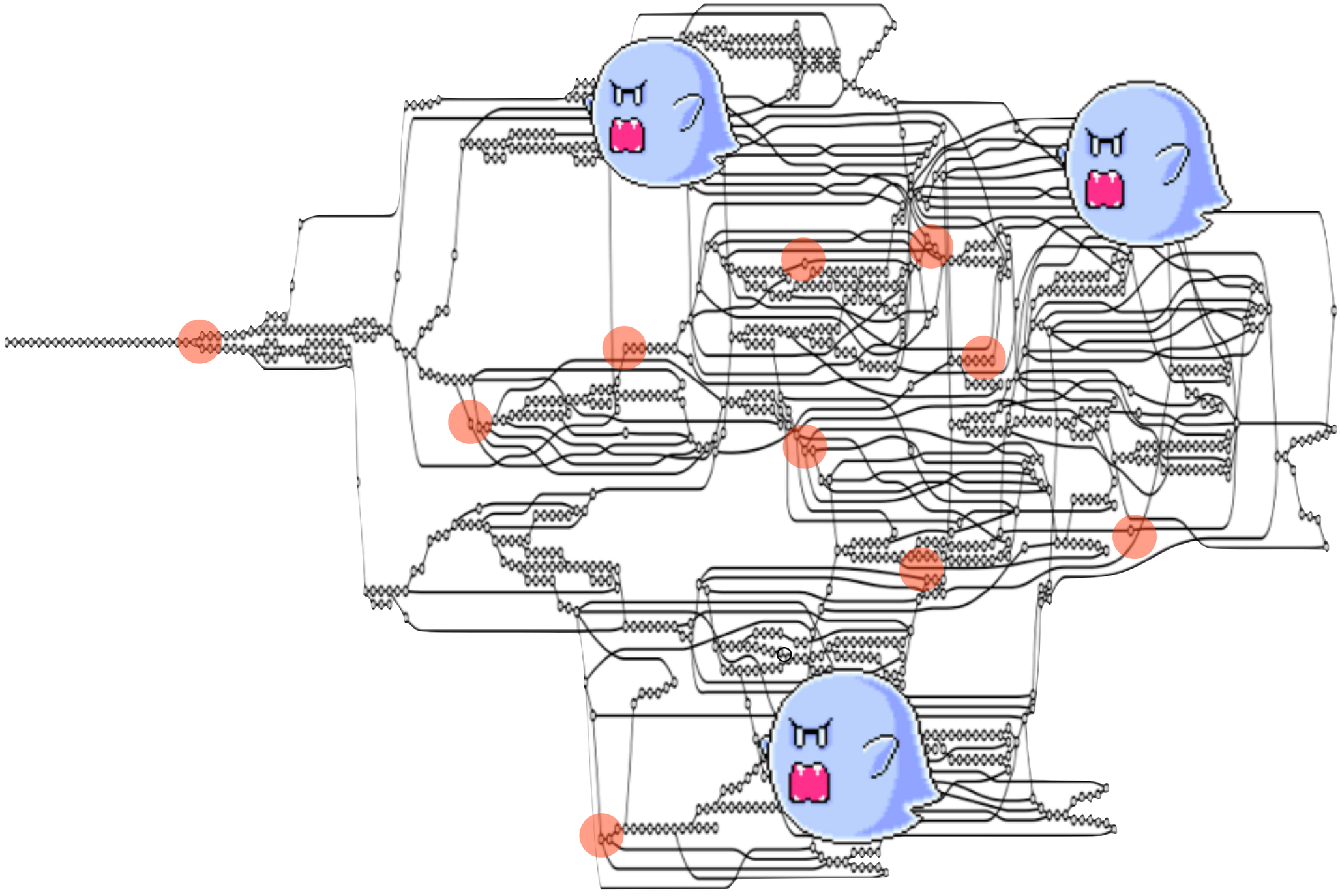
ψ

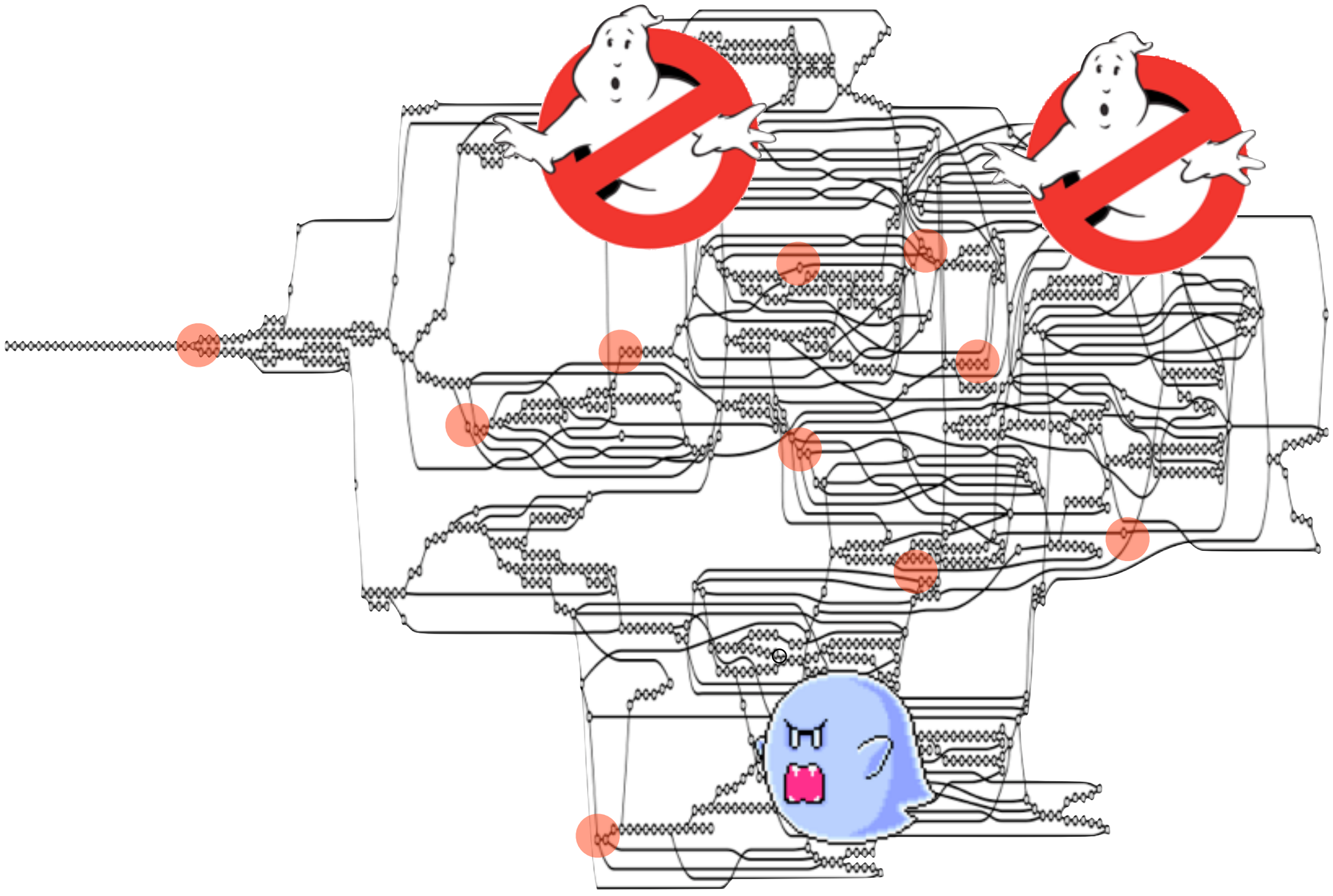


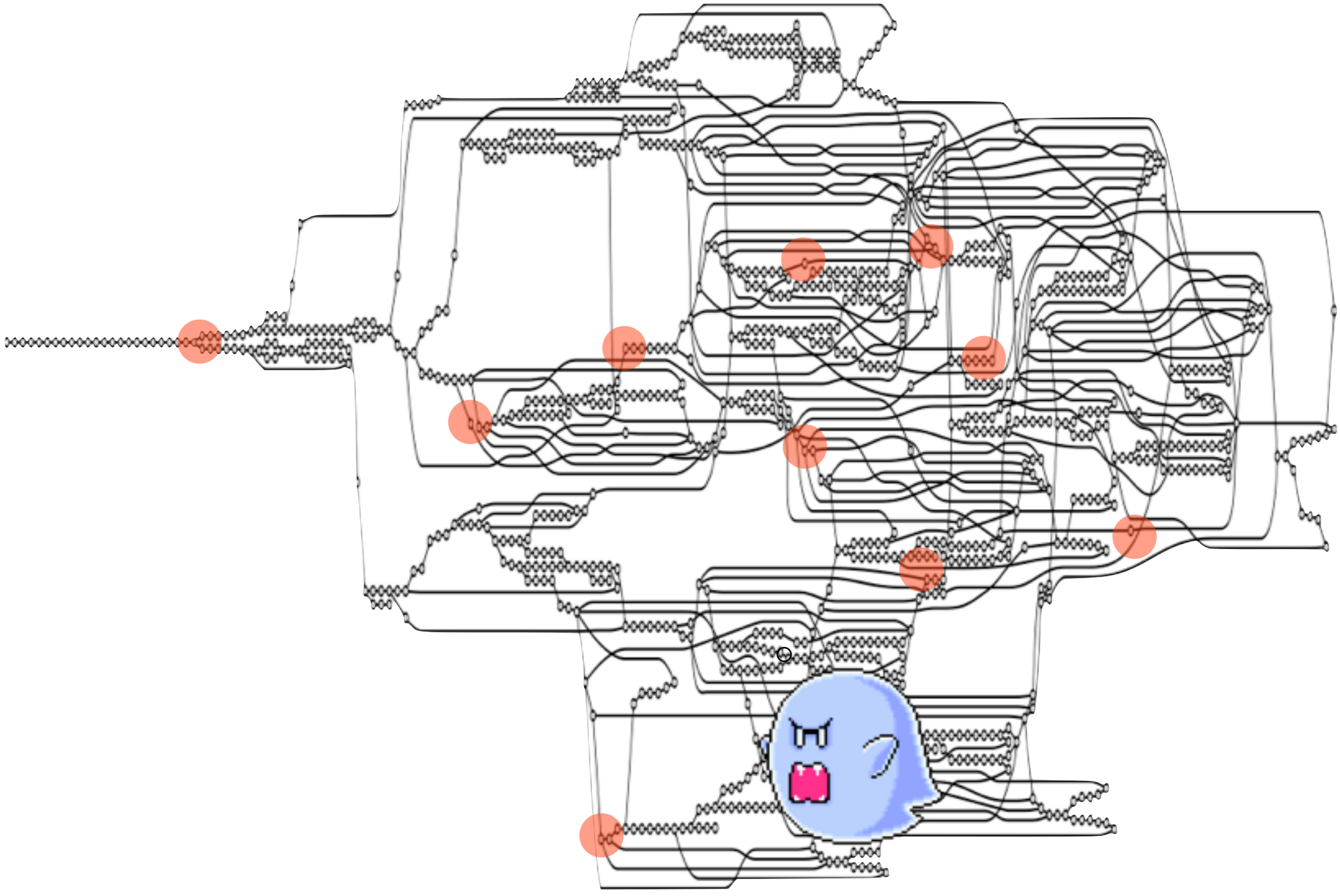


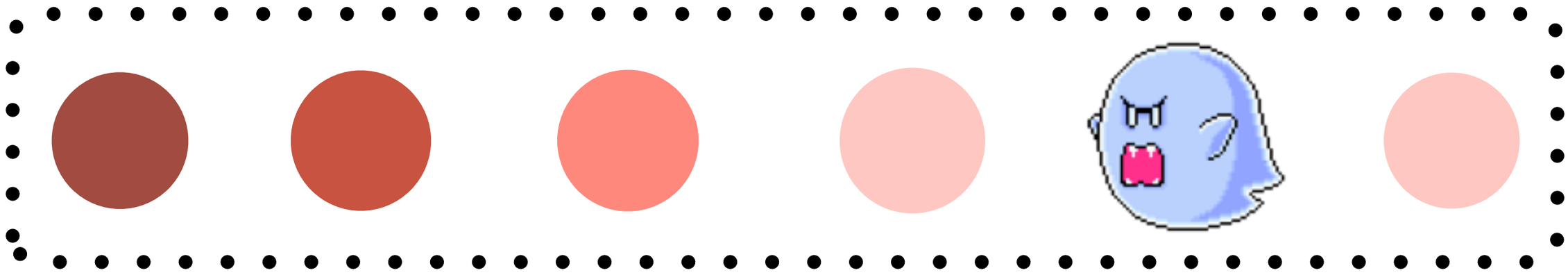


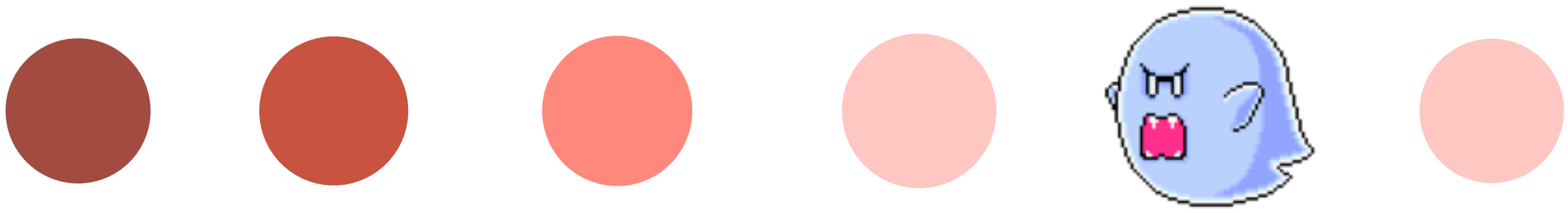












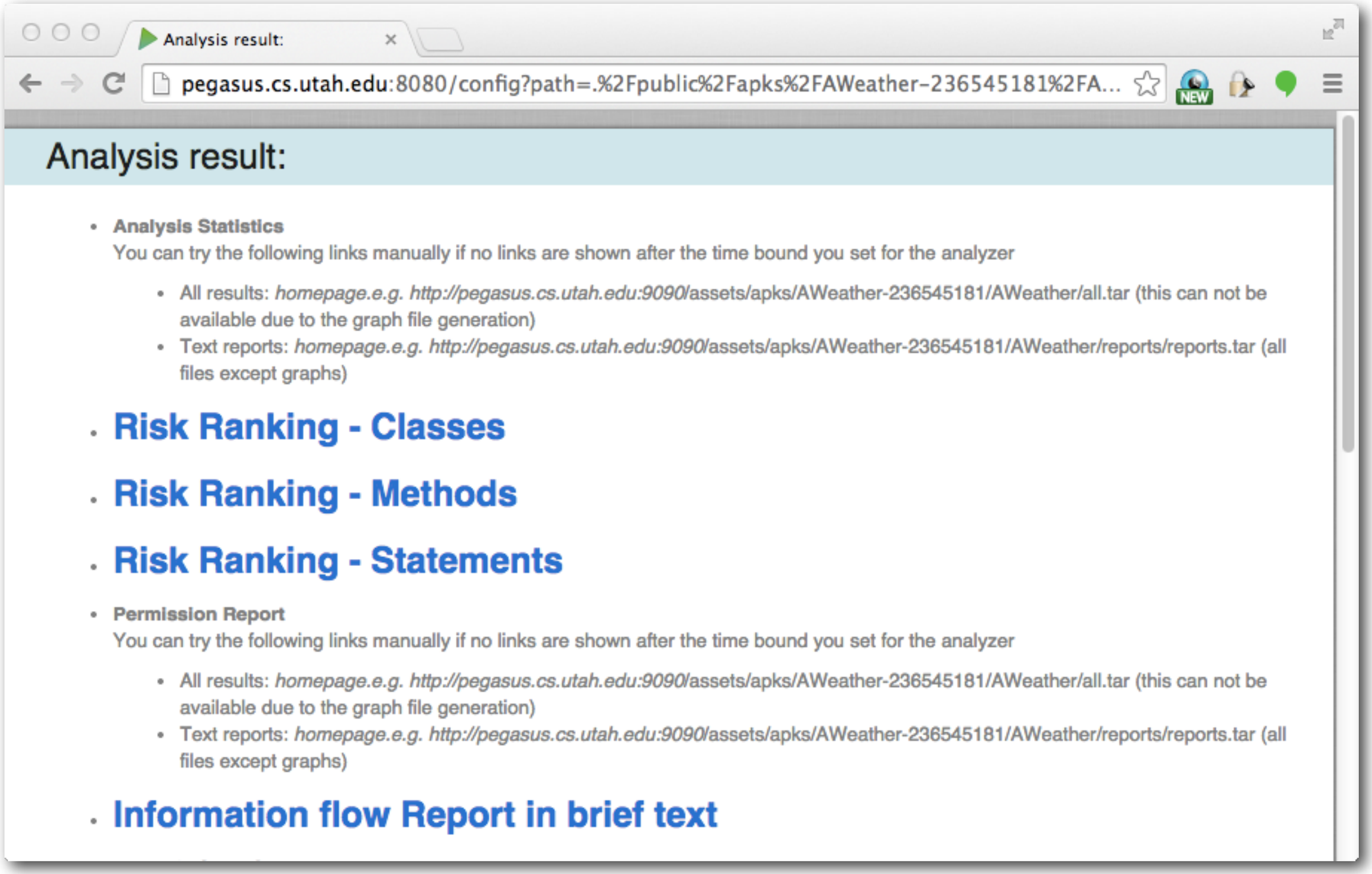
Suspicious?

network(location)

$$\mathit{reflection}(\top)$$

Observation

“Once you see it…”


Analysis result:
pegasus.cs.utah.edu:8080/config?path=.%2Fpublic%2Fapks%2FAWeather-236545181%2FA...
NEW
Analysis result:
Analysis Statistics
You can try the following links manually if no links are shown after the time bound you set for the analyzer
All results: homepage.e.g. http://pegasus.cs.utah.edu:9090/assets/apks/AWeather-236545181/AWeather/all.tar (this can not be available due to the graph file generation)
Text reports: homepage.e.g. http://pegasus.cs.utah.edu:9090/assets/apks/AWeather-236545181/AWeather/reports/reports.tar (all files except graphs)
Risk Ranking - Classes
Risk Ranking - Methods
Risk Ranking - Statements
Permission Report
You can try the following links manually if no links are shown after the time bound you set for the analyzer
All results: homepage.e.g. http://pegasus.cs.utah.edu:9090/assets/apks/AWeather-236545181/AWeather/all.tar (this can not be available due to the graph file generation)
Text reports: homepage.e.g. http://pegasus.cs.utah.edu:9090/assets/apks/AWeather-236545181/AWeather/reports/reports.tar (all files except graphs)
Information flow Report in brief text

Terminal — zsh — 82×20 — ⌘1

```
</0>


<282!matt@maptop:~/Copy/APAC-Demo/Anadroid-AWeather>
%
```

rank report


package com.nws.nwsserver;
import java.io.Serializable;
public class Info implements Serializable {
private static final long serialVersionUID = 1L;
public String data;
public int id;
public String lat;
public String lon;
@Override
public String toString() {
return String.format("id: %d, lat=%s, long=%s, data=%s", id, lat, lon, data);
}
}
package org.nws.aweather;
import java.util.TreeMap;
import android.content.Context;
import android.location.Address;
import android.location.Geocoder;
import android.os.Build;
import android.os.Handler;
import android.text.Editable;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnKeyListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
public class AddLocation implements OnClickListener, OnKeyListener, OnItemSelectedListener {
private final class GeocodeResultsHandler implements Runnable {
@Override
public void run() {
geocodeFinished();
}
}
public static final int GEOCODE_STATUS_LOOKUP_FAIL = 2;
public static final int GEOCODE_STATUS_NETWORK_ERROR = 1;
public static final int GEOCODE_STATUS_SUCCESS = 3;
private final Context context;
private final EditText enterLocation;
private final boolean geocoding;
private final TextView log;
private final Button lookupLocation;
private final AWeather owner;
private final Button showweather;
protected Address geocoded;
protected int geocodeStatus;
protected final GeocodeResultsHandler grh = new GeocodeResultsHandler();
protected Handler handler = new Handler();

```java
package com.nws.nwsserver;

import java.io.Serializable;

public class Info implements Serializable {
  private static final long serialVersionUID = 1L;
  public String data;
  public int id;
  public String lat;
  public String lon;

  @Override
  public String toString() {
    return String.format("id: %d, lat=%s, long=%s, data=%s", id, lat, lon,
        data);
  }
}
package org.nws.aweather;

import java.util.TreeMap;

import android.content.Context;
import android.location.Address;
import android.location.Geocoder;
import android.os.Build;
import android.os.Handler;
import android.text.Editable;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnKeyListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

public class AddLocation implements OnClickListener, OnKeyListener,
    OnItemSelectedListener {
  private final class GeocodeResultsHandler implements Runnable {
    @Override
    public void run() {
      geocodeFinished();
    }
  }

  public static final int GEOCODE_STATUS_LOOKUP_FAIL = 2;
  public static final int GEOCODE_STATUS_NETWORK_ERROR = 1;
  public static final int GEOCODE_STATUS_SUCCESS = 3;
  private final Context context;
  private final EditText enterLocation;
  private final boolean geocoding;
  private final TextView log;
  private final Button lookupLocation;
  private final AWeather owner;
  private final Button showweather;
  protected Address geocoded;
  protected int geocodeStatus;
  protected final GeocodeResultsHandler grh = new GeocodeResultsHandler();
  protected Handler handler = new Handler();

  private synchronized void geocode(final String place) {
    if (geocoding)
      return;
    lookupLocation.setEnabled(false);
    showweather.setEnabled(false);
    log(owner.getString(R.string.searchingForLocation));
    final GeocodePlace gp = new GeocodePlace(place, this, context);
    new Thread(gp).start();
  }

  public void geocodeFinished() {
    lookupLocation.setEnabled(true);
    showweather.setEnabled(true);
    if (geocodeStatus == GEOCODE_STATUS_SUCCESS) {
      final StringBuilder sb = new StringBuilder(
          owner.getString(R.string.foundLocation));
      sb.append(geocoded.getLocality()).append(" (")
          .append(geocoded.getAdminArea()).append(")");
      log(sb.toString());
      owner.getLocationManager().storeLocation(geocoded);
    } else if (geocodeStatus == GEOCODE_STATUS_NETWORK_ERROR)
      log(owner.getString(R.string.geocodeNetworkError));
    else
      // geocode status will be lookup fail
      log(owner.getString(R.string.geocodeLookupFailure));
    geocoded = null;
    geocodeStatus = 0;
  }

  public void log(final String text) {
    log.setText(text);
  }

  @Override
  public void onClick(final View v) {
    if (v == lookupLocation) {
      final Editable editable = enterLocation.getText();
      final String s = editable.toString();
      if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.GINGERBREAD
          && Geocoder.isPresent())
        geocode(s);
      else
        log("Geocoder not present");
    } else if (v == showweather) {
      owner.restart();
    }

  }

  @Override
  public void onItemSelected(final AdapterView<?> adapter, final View arg1,
      final int position, final long id) {
    showweather.setEnabled(true);
    owner.getLocationManager().storeSelected(position);
  }

  @Override
  public boolean onKey(final View v, final int keyCode, final KeyEvent event) {
    if (v == enterLocation) {
      final Editable editable = enterLocation.getText();
      final String s = editable.toString();
      if (s.length() == 0)
        lookupLocation.setEnabled(false);
      else
        lookupLocation.setEnabled(true);
      return false;
    }
    if (event.getKeyCode() == KeyEvent.KEYCODE_ENTER) {
      onClick(lookupLocation);
      return true;
    }
    return false;
  }

  @Override
  public void onNothingSelected(final AdapterView<?> arg0) {
    // don't do anything...
  }

  public AddLocation(final AWeather owner) {
    this.owner = owner;
    context = owner.getApplicationContext();
    log = (TextView) owner.findViewById(R.id.log);
    enterLocation = (EditText) owner.findViewById(R.id.enterLocation);
    enterLocation.setOnKeyListener(this);
    lookupLocation = (Button) owner.findViewById(R.id.fireLocationLookup);
    lookupLocation.setOnClickListener(this);
    lookupLocation.setEnabled(false);
    showweather = (Button) owner.findViewById(R.id.buttonShowWeather);
    showweather.setOnClickListener(this);
    showweather.setEnabled(false);
    geocoding = false;
    final Spinner spinner = (Spinner) owner
        .findViewById(R.id.locationSpinner);
    final ArrayAdapter<String> adapter = new ArrayAdapter<String>(owner,
        android.R.layout.simple_spinner_item);
    adapter.setDropDownViewResource(android.R.layout.simple_dropdown_item_1line);
    spinner.setAdapter(adapter);
    final LocationManager locationManager = new LocationManager(owner);
    final TreeMap<String, String[]> locations = locationManager
        .getLocations();
    for (final String id : locations.keySet()) {
      final String[] vals = locations.get(id);
      adapter.add(vals[0]);
    }
    int selected = locationManager.getSelected();
    spinner.setOnItemSelectedListener(this);
    spinner.setSelection(selected);
    log(owner.getString(R.string.addLocationExample));
  }
}
package org.nws.aweather;

import org.nws.aweather.forecast.FetchForecast;

import android.app.Activity;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.Handler;
import android.os.IBinder;
import android.os.Message;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;

public class AWeather extends Activity implements OnClickListener {
  public static final int MENU_ITEM_MANAGE_LOCATIONS = 0;
  public static final int MENU_ITEM_SET_SERVER = 2;
  public static final int MENU_ITEM_START_SERVICE = 1;
  private AddLocation addLocation;
  private LocationManager locationManager;
  private ServiceConnection m_connection = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName className, IBinder service) {
      m_service = ((AWeatherService.LocalBinder) service).getService();
      m_service.setWeatherGallery(weatherGallery);
    }

    @Override
    public void onServiceDisconnected(ComponentName className) {
      m_service = null;
    }
  };
  private final Handler m_handler = new Handler() {
    @Override
    public void handleMessage(final Message msg) {
      final Msg m = (Msg) msg.obj;
      ;
      switch (msg.what) {
      case AWeatherService.MESSAGE_DISCONNECTED:
        m_service.handleDisconnected(m);
        break;

      case AWeatherService.MESSAGE_CONNECTED:
        m_service.handleConnected(m);
        break;

      case AWeatherService.MESSAGE_ERROR:
        m_service.handleError(m);
        break;

      case AWeatherService.MESSAGE_INFO:
        m_service.handleInfo(m);
        break;

      case AWeatherService.MESSAGE_RECEIVED:
        m_service.handleReceived(m);
        break;
      }
    }
  };
  private static final int SET_SERVER = 10;
      private boolean m_isbound = false;

  private AWeatherService m_service;

  private WeatherGallery weatherGallery = null;

  void doBindService() {
    bindService(new Intent(AWeather.this, AWeatherService.class),
        m_connection, Context.BIND_AUTO_CREATE);
    m_isbound = true;
  }

  void doUnbindService() {
    if (m_isbound) {
      unbindService(m_connection);
      m_isbound = false;
    }
  }

  protected LocationManager getLocationManager() {
    return locationManager;
  }

  @Override
  protected void onDestroy() {
    super.onDestroy();
    m_service.setWeatherGallery(null);
    doUnbindService();
  }

  public void forwardMsg(final int type, final Object s) {
    final Message m = Message.obtain(m_handler, type);
    m.obj = new Msg(s);
    m_handler.sendMessage(m);
  }

  @Override
  public void onClick(View view) {
    weatherGallery.refresh();
    //m_service.m_networklistener.push();
  }

  @Override
  public void onCreate(final Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    locationManager = new LocationManager(this);
    if (locationManager.getLocationCount() == 0) {
      setContentView(R.layout.addlocation);
      addLocation = new AddLocation(this);
    } else {
      setContentView(R.layout.main);
      weatherGallery = new WeatherGallery(this);
    }
    doBindService();
    startService(new Intent(this, AWeatherService.class));
  }

  @Override
  public boolean onCreateOptionsMenu(final Menu menu) {
    super.onCreateOptionsMenu(menu);
    menu.add(0, MENU_ITEM_MANAGE_LOCATIONS, 0, R.string.manageLocations);
    menu.add(0, MENU_ITEM_START_SERVICE, 0, R.string.manageService);
    menu.add(0, MENU_ITEM_SET_SERVER, 0, R.string.setserver);
    return true;
  }

  @Override
  public boolean onOptionsItemSelected(final MenuItem item) {
    switch (item.getItemId()) {
    case MENU_ITEM_MANAGE_LOCATIONS:
      setContentView(R.layout.addlocation);
      if (addLocation == null)
        addLocation = new AddLocation(this);
      return true;
    case MENU_ITEM_START_SERVICE:
      startActivity(new Intent(this, ServiceLauncher.class));
      return true;
    case MENU_ITEM_SET_SERVER:
      startActivityForResult(new Intent(this, SetServer.class), SET_SERVER);
      return true;
    }
    return false;
  }

  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent data) {
      if ( requestCode == SET_SERVER && resultCode == Activity.RESULT_OK ){
        String s = FetchForecast.getInstance().getUrl();
        locationManager.storeServer(s);
        restart();
      }
    super.onActivityResult(requestCode, resultCode, data);
  }

  public void restart() {
    stopService(new Intent(this, AWeatherService.class));
    Intent i = getBaseContext().getPackageManager()
        .getLaunchIntentForPackage(getBaseContext().getPackageName());
    i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(i);
  }
}package org.nws.aweather;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.SocketException;
import java.util.Calendar;
import java.util.Timer;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Intent;
import android.os.Binder;
import android.os.IBinder;
import android.widget.Toast;

public class AWeatherService extends Service implements NetworkTaskEvents {

  public class LocalBinder extends Binder {
    AWeatherService getService() {
      return AWeatherService.this;
    }
  }

  private static boolean showToasts = false;
  public static final int MESSAGE_CONNECTED = 0;
  public static final int MESSAGE_DISCONNECTED = 1;
  public static final int MESSAGE_ERROR = 2;
  public static final int MESSAGE_INFO = 3;

  public static final int MESSAGE_RECEIVED = 4;

  private Long m_counter = 0L;

  public NetworkListener m_networklistener;

  private Responder m_responder;

  private WeatherGallery m_weathergallery = null;

  private final IBinder mBinder = new LocalBinder();

  private NotificationManager nm;

  private final Calendar time = Calendar.getInstance();

  private final Timer timer = new Timer();

  private void connect() {
    try {
      m_responder = new Responder(this);
      m_responder.begin();
      m_networklistener = new NetworkListener(this, 6001);
      m_networklistener.start();
    } catch (SocketException e) {
    }
  }

  private void destroyTimer() {
    if (timer != null)
      timer.cancel();
  }

  private void disconnect() {
    if (m_networklistener != null)
      m_networklistener.close();
    if (m_responder != null)
      m_responder.close();
  }

  /** Show a notification while this service is running. */
  @SuppressWarnings("deprecation")
  private void showNotification() {
    // Use the same text for the ticker and the expanded notification
    final CharSequence text = getText(R.string.service_started);
    // Set the icon, scrolling text and timestamp
    final Notification notification = new Notification(R.drawable.android,
        text, System.currentTimeMillis());
    // The PendingIntent to launch our activity if the user selects this
    // notification
    final PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
        new Intent(this, ServiceLauncher.class), 0);
    // Set the info for the views that show in the notification panel.
    notification.setLatestEventInfo(this, getText(R.string.service_label),
        text, contentIntent);
    // Send the notification.
    // We use a layout id because it is a unique number. We use it later to
    // cancel.
    nm.notify(R.string.service_started, notification);
  }

  @Override
  public void handleConnected(Msg o) {
    Object ob = o.getInfo();
    String s = "Connected";
    if (ob instanceof String)
      s = (String) (ob);
    if (showToasts)
      Toast.makeText(this, s, Toast.LENGTH_LONG).show();
  }

  @Override
  public void handleDisconnected(Msg o) {
    Object ob = o.getInfo();
    String s = "Disconnected";
    if (ob instanceof String)
      s = (String) (ob);
    if (showToasts)
      Toast.makeText(this, s, Toast.LENGTH_LONG).show();
  }

  @Override
  public void handleError(Msg o) {
    Object ob = o.getInfo();
    if (ob instanceof String) {
      String s = (String) (ob);
      if (showToasts)
        Toast.makeText(this, s, Toast.LENGTH_LONG).show();
    } else if (ob instanceof DatagramPacket) {
      try {
        m_responder.respond(ob);
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    } else if (ob instanceof byte[]) {
      byte bytes[] = (byte[]) ob;
      try {
        m_responder.respond(ConvertUtils.deserializePacket(bytes));
      } catch (InterruptedException e) {
        e.printStackTrace();
      } catch (IOException e) {
        e.printStackTrace();
      }
    }
  }

  @Override
  public void handleInfo(final Msg o) {
    Object ob = o.getInfo();
    String s = "Info";
    if (ob instanceof String)
      s = (String) (ob);
    if (showToasts)
      Toast.makeText(this, s, Toast.LENGTH_LONG).show();
  }

  @Override
  public void handleReceived(Msg o) {
    Object ob = o.getInfo();
    String s = "Received";
    if (ob instanceof String)
      s = (String) (ob);
    if (s.equals("update"))
      if (m_weathergallery != null)
        m_weathergallery.refresh();
    if (showToasts)
      Toast.makeText(this, s, Toast.LENGTH_LONG).show();
  }

  @Override
  public IBinder onBind(final Intent intent) {
    return mBinder;
  }

  @Override
  public void onCreate() {
    super.onCreate();
    nm = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
    if (showToasts)
      Toast.makeText(this, "Service created at " + time.getTime(),
          Toast.LENGTH_LONG).show();
    showNotification();
    // createTimer();
    connect();
  }

  @Override
  public void onDestroy() {
    super.onDestroy();
    // Cancel the persistent notification.
    destroyTimer();
    disconnect();
    nm.cancel(R.string.service_started);
    if (showToasts)
      Toast.makeText(
          this,
          "Service destroyed at " + time.getTime()
              + "; m_counter is at: " + m_counter,
          Toast.LENGTH_LONG).show();
    m_counter = null;
  }

  @Override
  public int onStartCommand(Intent intent, int flags, int startId) {
    return START_STICKY;
  }

  public void setWeatherGallery(WeatherGallery a) {
    m_weathergallery = a;
  }

}
package org.nws.aweather;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.ObjectInput;
import java.io.ObjectInputStream;
import java.io.ObjectOutput;
import java.io.ObjectOutputStream;
import java.net.DatagramPacket;
import java.net.InetAddress;

public class ConvertUtils {

  static public DatagramPacket deserializePacket(byte[] packetBytes)
      throws IOException {
    DatagramPacket udp = new DatagramPacket(new byte[1024], 1024);

    ByteArrayInputStream bis = new ByteArrayInputStream(packetBytes);
    ObjectInput in = null;
    try {
      in = new ObjectInputStream(bis);
      udp.setAddress((InetAddress) in.readObject());
      udp.setPort((Integer) in.readObject());
      udp.setData((byte[]) in.readObject());
    } catch (IOException e) {
      e.printStackTrace();
      return null;
    } catch (ClassNotFoundException e) {
      e.printStackTrace();
      return null;
    } finally {
      in.close();
      bis.close();
    }

    return udp;
  }

  public int numPackets = 0;

  public void serializePacket(byte[] packetBytes) {
    // record the number of packets and the time between packets received
    // avgInterval
    numPackets += 1;
  }

  public void serializePacket(DatagramPacket udp) throws IOException {
    // byte[] packetBytes = new byte[4096];
    //
    // byte[] tmp = udp.getAddress().getAddress();
    // int offset = 0;
    // System.arraycopy( tmp, 0, packetBytes, offset, tmp.length);
    // offset += tmp.length;
    //
    // ByteBuffer bb = ByteBuffer.allocate(4);
    // bb.putInt(udp.getPort());
    // System.arraycopy( bb.array(), 0, packetBytes, offset, 4);
    // offset += 4;
    //
    // bb.putInt(udp.getLength());
    // System.arraycopy( bb.array(), 0, packetBytes, offset, 4);
    // offset += 4;
    //
    // System.arraycopy( udp.getData(), 0, packetBytes, offset,
    // udp.getLength());
    // serializePacket(packetBytes);

    ByteArrayOutputStream bos = new ByteArrayOutputStream();
    ObjectOutput out = null;
    try {
      out = new ObjectOutputStream(bos);
      out.writeObject(udp.getAddress());
      out.write(udp.getPort());
      out.writeObject(udp.getData());
      serializePacket(bos.toByteArray());
    } catch (IOException e) {
      e.printStackTrace();
    } finally {
      out.close();
      bos.close();
    }
  }
}package org.nws.aweather.forecast;

import java.io.BufferedReader;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.ObjectInput;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.Socket;
import java.net.SocketException;
import java.net.UnknownHostException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Random;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.TimeUnit;

import org.nws.aweather.WeatherGallery;
import org.xml.sax.helpers.DefaultHandler;

import com.nws.nwsserver.Info;

public class FetchForecast extends DefaultHandler implements Runnable {

  private static final String BEGIN = "&begin=";
  private static final DateFormat DATE_FORMAT = new SimpleDateFormat(
      "yyyy-MM-dd'T'hh':00:00'");
  private static final String END = "&end=";
  private static final String HOST = "graphical.weather.gov";
  private static final String LAT = "&lat=";
  private static final String LON = "&lon=";
  private static final int MAX_RETRIES = 3;
  private static final int PORT = 5559;
  private static FetchForecast theInstance = null;

  private static final long TIMEOUT_SECS = 10;

  private static final int UDP_WC_PORT = 10000;
  private static final String UDP_LOCAL = "www.weatherpushupdates.com";
  private static final int UDP_WS_PORT = 5559;
  //public static String udpWsHost = "192.168.19.207";
  public static String udpWsHost = UDP_LOCAL;
  private static final String URL_PREFIX = "/xml/sample_products/browser_interface/ndfdXMLclient.php?"
      + "&product=time-series&icons=icons";
  public static WeatherGallery parent = null;
  public static FetchForecast getInstance() {
    if (theInstance == null) {
      theInstance = new FetchForecast();
      new Thread(theInstance).start();
    }
    return theInstance;
  }

  private final boolean activeFlag = true;
  private Info currentRequestMsg;
  private final BlockingQueue<Info> recvQueue = new LinkedBlockingQueue<Info>(
      1024);
  private int requestId = (new Random()).nextInt();
  private InetAddress serverAddress;

  private DatagramSocket socket;
  private String urlString;

  public boolean useUDPProtocol = true;

  private void getDirectNWS(String lat, String lon) {

    final StringBuilder url = new StringBuilder(URL_PREFIX);
    url.append(LAT).append(lat).append(LON).append(lon);
    final Calendar start = Calendar.getInstance();
    final Calendar end = Calendar.getInstance();
    end.add(Calendar.DAY_OF_MONTH, 7);
    url.append(BEGIN).append(DATE_FORMAT.format(start.getTime()));
    url.append(END).append(DATE_FORMAT.format(end.getTime()));
    urlString = url.toString();

    try {
      System.out.println("Ripping: " + urlString);
      final Socket s = new Socket(HOST, PORT);
      final OutputStream os = s.getOutputStream();
      final PrintWriter out = new PrintWriter(new OutputStreamWriter(os));
      final StringBuilder request = new StringBuilder("GET ").append(
          urlString).append("HTTP/1.1");
      request.append("\n");
      request.append("Host: ").append(HOST).append("\n");
      request.append("\n");
      out.print(request);
      out.flush();
      final InputStream in = s.getInputStream();
      // String t = getIn(in);
      //
      // int n = t.length( );
      final NWSWeatherXMLParser parser = new NWSWeatherXMLParser(in);
      parser.parse(parent);
      System.out.println("Finished parsing");
    } catch (final Exception e) {
      throw new RuntimeException(e);
    }
  }

  @SuppressWarnings("unused")
  private String getIn(final InputStream in) {
    final InputStreamReader is = new InputStreamReader(in);
    final StringBuilder sb = new StringBuilder();
    final BufferedReader br = new BufferedReader(is);
    String read = null;
    try {
      read = br.readLine();
    } catch (final IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }
    while (read != null) {
      // System.out.println(read);
      sb.append(read);
      try {
        read = br.readLine();
      } catch (final IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
      }
    }
    return sb.toString();
  }

  private void getUDPWeatherServce(String lat, String lon) {
    currentRequestMsg = new Info();
    currentRequestMsg.data = "REQW_GETFORCAST";
    currentRequestMsg.lat = lat;
    currentRequestMsg.lon = lon;

    int nTries = MAX_RETRIES;
    currentRequestMsg.id = requestId++;
    while (nTries > 0) {
      nTries--;
      // currentRequestMsg.id = requestId++;

      byte[] sendBuf = null;
      try {
        ByteArrayOutputStream bos = new ByteArrayOutputStream();
        ObjectOutputStream out = new ObjectOutputStream(bos);
        out.writeObject(currentRequestMsg);
        out.close();
        sendBuf = bos.toByteArray();
        bos.close();
      } catch (IOException e) {
        parent.logError(e.getMessage());
        return;
      }

      DatagramPacket sendPkt = new DatagramPacket(sendBuf,
          sendBuf.length, serverAddress, UDP_WS_PORT);

      try {
        if ((socket != null) && (serverAddress != null)) {
          socket.send(sendPkt);
        }
      } catch (IOException e1) {
        parent.logError(e1.getMessage());
        return;
      }

      Info recvMsg;
      do {
        try {
          recvMsg = recvQueue.poll(TIMEOUT_SECS, TimeUnit.SECONDS);
          if (recvMsg != null)
            if (isValid(recvMsg, currentRequestMsg)) {
              parent.gotWeather(recvMsg.data);
              return;
            }

        } catch (InterruptedException e) {
          // TODO Auto-generated catch block
          e.printStackTrace();
        } catch (Exception e) {
          // TODO Auto-generated catch block
          e.printStackTrace();
        }
      } while (!recvQueue.isEmpty());
    }
  }

  private boolean isValid(Info info_in, Info info_out) {
    return info_in.id == info_out.id;
  }

  public String getDefaultUrl() {
    return UDP_LOCAL;
  }

  public String getUrl() {
    return udpWsHost;
  }

  public void initializeUDP() {
    try {
      serverAddress = InetAddress.getByName(udpWsHost);
    } catch (UnknownHostException e) {
      parent.logError(e.getMessage());
      return;
    }
    try {
      socket = new DatagramSocket(UDP_WC_PORT);
    } catch (SocketException e) {
      parent.logError(e.getMessage());
    }
  }

  public void requestForecast(String lat, String lon) {
    if (!useUDPProtocol)
      getDirectNWS(lat, lon);
    else
      getUDPWeatherServce(lat, lon);
  }

  @Override
  public void run() {
    initializeUDP();
    while (activeFlag) {
      byte recvBuf[] = new byte[4096];
      DatagramPacket recvPkt = new DatagramPacket(recvBuf, recvBuf.length);

      try {
        socket.receive(recvPkt);
      } catch (IOException e) {
        parent.logError(e.getMessage());
        continue;
      }
      Info recvInfo = null;
      try {
        ByteArrayInputStream bis = new ByteArrayInputStream(
            recvPkt.getData());
        ObjectInput in = new ObjectInputStream(bis);
        recvInfo = (Info) in.readObject();
      } catch (Exception e) {
        parent.logError(recvPkt.getData());
        continue;
      }

      if (isValid(recvInfo, currentRequestMsg)) {
        recvQueue.offer(recvInfo);
      } else {
        parent.logError("Invalid id.");
      }
    }
  }

  public void setUrl(String url) {
    udpWsHost = url;
  }
}
package org.nws.aweather.forecast;

import java.util.ArrayList;
import java.util.Date;
import java.util.LinkedHashMap;
import java.util.List;

public class NWSWeatherObject {
  private final List<String> conditionIconLinks;
  private String conditionIconTimeLayout;
  private Date createdOn;
  private String moreInfoLink;
  private String mostRecentTimeLayout;
  private final LinkedHashMap<String, List<TimeLayoutBand>> timeLayouts;

  public void addConditionIconLink(final String link) {
    conditionIconLinks.add(link);
  }

  public void addNewTimeBand(final Date date) {
    final List<TimeLayoutBand> l = timeLayouts.get(mostRecentTimeLayout);
    final TimeLayoutBand tlb = new TimeLayoutBand(date);
    l.add(tlb);
  }

  public void addTimeLayout(final String name) {
    mostRecentTimeLayout = name;
    timeLayouts.put(name, new ArrayList<TimeLayoutBand>());
  }

  public void closeCurrentTimeBand(final Date end) {
    final List<TimeLayoutBand> l = timeLayouts.get(mostRecentTimeLayout);
    final TimeLayoutBand last = l.get(l.size() - 1);
    last.setEndDate(end);
  }

  public List<String> getConditionIconLinks() {
    return conditionIconLinks;
  }

  public String getConditionIconTimeLayout() {
    return conditionIconTimeLayout;
  }

  public Date getCreatedOn() {
    return createdOn;
  }

  public String getMoreInfoLink() {
    return moreInfoLink;
  }

  public void setConditionIconTimeLayout(final String name) {
    conditionIconTimeLayout = name;
  }

  public void setCreatedOn(final Date date) {
    createdOn = date;
  }

  public void setMoreInfoLink(final String moreInfoLink) {
    this.moreInfoLink = moreInfoLink;
  }

  public NWSWeatherObject() {
    timeLayouts = new LinkedHashMap<String, List<TimeLayoutBand>>();
    conditionIconLinks = new ArrayList<String>();
  }
}
package org.nws.aweather.forecast;

import java.io.InputStream;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import org.nws.aweather.WeatherGallery;
import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

/**
 * Responsible for parsing through the XML recovered from an NWS hit. There are
 * two core sections to look for:
 * <ul>
 * <li>time-layout: which contains sections/spans of time</li>
 * <li>parameters: which contains specific blocks of weather data</li>
 * </ul>
 *
 * @author keerat
 */
public class NWSWeatherXMLParser extends DefaultHandler {
  // condition icons
  private static final String CONDITIONS = "conditions-icon";
  private static final SimpleDateFormat CREATED_DATE_FORMATTER = new SimpleDateFormat(
      "yyyy-MM-dd'T'HH:mm:ss'Z'");
  // <creation-date
  // refresh-frequency="PT1H">2009-03-14T20:58:51Z</creation-date>
  private static final String CREATION_DATE = "creation-date";
  private static final String END_VALID_TIME = "end-valid-time";
  private static final String ICON_LINK = "icon-link";
  private static final String LAYOUT_KEY = "layout-key";
  // <moreWeatherInformation applicable-location="point1">
  // http://forecast.weather.gov/MapClick.php?textField1=38.99&textField2=-77.01
  // </moreWeatherInformation>
  private static final String MORE_INFO = "moreWeatherInformation";
  private static final String START_VALID_TIME = "start-valid-time";
  private static final String TIME_LAYOUT = "time-layout";
  // <start-valid-time>2009-03-14T08:00:00-04:00</start-valid-time>
  private static final SimpleDateFormat TIME_LAYOUT_DATE_FORMATTER = new SimpleDateFormat(
      "yyyy-MM-dd'T'HH:mm:ssZ");
  private String currentTag;
  private final InputStream in;
  private final NWSWeatherObject nwo;
  private WeatherGallery parent;

  private void handleCreationDate(final String dateString) {
    try {
      final Date date = CREATED_DATE_FORMATTER.parse(dateString);
      nwo.setCreatedOn(date);
    } catch (final ParseException e) {
    }
  }

  private void handleEndValidTime(final String endTimeString) {
    try {
      final Date date = TIME_LAYOUT_DATE_FORMATTER.parse(endTimeString);
      nwo.closeCurrentTimeBand(date);
    } catch (final ParseException e) {
    }
  }

  private void handleStartValidTime(final String startTimeString) {
    try {
      final Date date = TIME_LAYOUT_DATE_FORMATTER.parse(startTimeString);
      nwo.addNewTimeBand(date);
    } catch (final ParseException e) {
    }
  }

  @Override
  public void characters(final char[] ch, final int start, final int length)
      throws SAXException {
    if (currentTag.equals(CREATION_DATE))
      handleCreationDate(new String(ch, start, length));
    if (currentTag.equals(MORE_INFO))
      nwo.setMoreInfoLink(new String(ch, start, length));
    if (currentTag.equals(LAYOUT_KEY))
      nwo.addTimeLayout(new String(ch, start, length));
    if (currentTag.equals(START_VALID_TIME))
      handleStartValidTime(new String(ch, start, length));
    if (currentTag.equals(END_VALID_TIME))
      handleEndValidTime(new String(ch, start, length));
    if (currentTag.equals(ICON_LINK))
      nwo.addConditionIconLink(new String(ch, start, length));
  }

  @Override
  public void endDocument() throws SAXException {
    parent.gotWeather(nwo);
  }

  @Override
  public void endElement(final String uri, final String localName,
      final String name) throws SAXException {
    currentTag = "";
  }

  public void parse(final WeatherGallery parent) throws Exception {
    final SAXParserFactory saxParserFactory = SAXParserFactory
        .newInstance();
    final SAXParser parser = saxParserFactory.newSAXParser();
    this.parent = parent;
    parser.parse(in, this);
  }

  @Override
  public void startElement(final String uri, final String localName,
      final String name, final Attributes attributes) throws SAXException {
    currentTag = localName;
    if (currentTag.equals(CONDITIONS)) {
      // rip the time layout
      final String timeLayoutName = attributes.getValue(TIME_LAYOUT);
      nwo.setConditionIconTimeLayout(timeLayoutName);
    }
  }

  public NWSWeatherXMLParser(final InputStream in) {
    this.in = in;
    currentTag = "";
    nwo = new NWSWeatherObject();
  }
}
package org.nws.aweather.forecast;

import java.util.Date;

public class TimeLayoutBand {
  private Date end;
  private final Date start;

  public Date getEnd() {
    return end;
  }

  public Date getStart() {
    return start;
  }

  public boolean isPointInTime() {
    if (start.equals(end))
      return true;
    else
      return false;
  }

  public void setEndDate(final Date end) {
    this.end = end;
  }

  public TimeLayoutBand(final Date start) {
    this.start = start;
    end = start;
  }
}
package org.nws.aweather;

import java.io.IOException;
import java.util.List;
import java.util.Locale;

import android.content.Context;
import android.location.Address;
import android.location.Geocoder;

public class GeocodePlace implements Runnable {
  private final Context context;
  private final AddLocation owner;
  private final String place;

  @Override
  public void run() {
    final Geocoder geocoder = new Geocoder(context, Locale.getDefault());
    List<Address> addresses = null;
    boolean found = false;
    try {
      addresses = geocoder.getFromLocationName(place, 1);
      if (addresses != null && addresses.size() > 0) {
        found = true;
        owner.geocodeStatus = AddLocation.GEOCODE_STATUS_SUCCESS;
      } else
        owner.geocodeStatus = AddLocation.GEOCODE_STATUS_LOOKUP_FAIL;
    } catch (final IOException e) {
      owner.geocodeStatus = AddLocation.GEOCODE_STATUS_NETWORK_ERROR;
      found = false;
    }
    if (found) {
      final Address a = addresses.get(0);
      owner.geocoded = a;
    }
    owner.handler.post(owner.grh);
  }

  GeocodePlace(final String place, final AddLocation owner,
      final Context context) {
    this.place = place;
    this.owner = owner;
    this.context = context;
  }
}
package org.nws.aweather;

import java.util.Map;
import java.util.Set;
import java.util.TreeMap;

import org.nws.aweather.forecast.FetchForecast;

import android.content.Context;
import android.content.SharedPreferences;
import android.location.Address;

public class LocationManager {
  private static final String LOCAL_MAP_LOCATION = "llm.";
  private static final String LOCATION_LATITUDE = "location.lat.";
  private static final String LOCATION_LONGITUDE = "location.lon.";
  private static final String LOCATION_NAME = "location.name.";
  private static final String SELECTED = "selected";
  private static final String SERVER = "server";
  private int locationCount;
  private final TreeMap<String, String[]> locations;
  private int selected = 0;
  private String server = FetchForecast.udpWsHost;
  private final SharedPreferences sharedPreferences;

  public String[] getLocation(final int location) {
    final String[] loc = locations.get(LOCAL_MAP_LOCATION + location);
    return loc;
  }

  public int getLocationCount() {
    return locationCount;
  }

  public TreeMap<String, String[]> getLocations() {
    return locations;
  }

  public int getSelected() {
    return selected;
  }

  public String getServer() {
    return server;
  }

  public synchronized void storeLocation(final Address address) {
    if (address == null)
      System.out.println("Could not geocode location!");
    final SharedPreferences.Editor editor = sharedPreferences.edit();
    final String adminArea = address.getAdminArea();
    final String locality = address.getLocality();
    final StringBuilder sb = new StringBuilder(locality).append(" (")
        .append(adminArea).append(")");
    final String latitude = Double.toString(address.getLatitude());
    final String longitude = Double.toString(address.getLongitude());
    final String nameKey = LOCATION_NAME + locationCount;
    final String latKey = LOCATION_LATITUDE + locationCount;
    final String lonKey = LOCATION_LONGITUDE + locationCount;
    editor.putString(nameKey, sb.toString());
    editor.putString(latKey, latitude);
    editor.putString(lonKey, longitude);
    editor.commit();
    locations.put(LOCAL_MAP_LOCATION + locationCount,
        new String[] { sb.toString(), latitude, longitude });
    locationCount++;
  }

  public synchronized void storeSelected(final int i) {
    final SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putInt(SELECTED, i);
    editor.commit();
    selected = i;
  }

  public synchronized void storeServer(final String s) {
    final SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString(SERVER, s);
    editor.commit();
    server = s;
  }

  public LocationManager(final AWeather owner) {
    sharedPreferences = owner.getPreferences(Context.MODE_WORLD_READABLE);
    final Map<String, ?> allParams = sharedPreferences.getAll();
    final Set<String> keys = allParams.keySet();
    locations = new TreeMap<String, String[]>();
    locationCount = 0;
    for (final String key : keys)
      if (key.startsWith(LOCATION_NAME)) {
        final String name = sharedPreferences.getString(LOCATION_NAME
            + locationCount, "Corrupted");
        final String latitude = sharedPreferences.getString(
            LOCATION_LATITUDE + locationCount, "0");
        final String longitude = sharedPreferences.getString(
            LOCATION_LONGITUDE + locationCount, "0");
        locations.put(LOCAL_MAP_LOCATION + locationCount, new String[] {
            name, latitude, longitude });
        locationCount++;
      } else if (key.startsWith(SELECTED)) {
        selected = sharedPreferences.getInt(SELECTED, 0);
      } else if (key.startsWith(SERVER)) {
        server = sharedPreferences.getString(SERVER,  FetchForecast.udpWsHost);
      }

  }
}
package org.nws.aweather;

public class Msg {
  private final Object info;

  public String getDescription() {
    if (info instanceof String)
      return String.format("InfoMsg: info = %s", info);
    return info.toString();
  }

  public Object getInfo() {
    return info;
  }

  public Msg(final Object info) {
    this.info = info;
  }
}package org.nws.aweather;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Date;

import org.nws.aweather.forecast.FetchForecast;

import android.os.Handler;
import android.os.Message;

public class NetworkListener extends Thread {
  private boolean domore = true;

  private final Handler m_handler = new Handler() {
    @Override
    public void handleMessage(final Message msg) {
      final Msg m = (Msg) msg.obj;
      ;
      switch (msg.what) {
      case AWeatherService.MESSAGE_DISCONNECTED:
        m_service.handleDisconnected(m);
        break;

      case AWeatherService.MESSAGE_CONNECTED:
        m_service.handleConnected(m);
        break;

      case AWeatherService.MESSAGE_ERROR:
        m_service.handleError(m);
        break;

      case AWeatherService.MESSAGE_INFO:
        m_service.handleInfo(m);
        break;

      case AWeatherService.MESSAGE_RECEIVED:
        m_service.handleReceived(m);
        break;
      }
    }
  };

  private int m_port;
  private AWeatherService m_service;
  private DatagramSocket socket = null;

  private void forwardMsg(final int type, final String s) {
    final Message m = Message.obtain(m_handler, type);
    m.obj = new Msg(s);
    m_handler.sendMessage(m);

  }

  public void close() {
    socket.close();
    System.out.println("Receiver closed.");
  }

  public void push() {
    byte[] greet = "Hello".getBytes();
    DatagramPacket p;
    try {
      p = new DatagramPacket(greet, greet.length,
          InetAddress.getByName(FetchForecast.udpWsHost), m_port);
      socket.send(p);
      System.out.println(new Date().toString() + " :  Forward: "
          + p.getAddress());
    } catch (Exception e) {
      System.out.print(String.format("Exception: %s\n", e.getMessage()));
    }
  }

  @Override
  public void run() {
    forwardMsg(AWeatherService.MESSAGE_CONNECTED,
        String.format("Receiver started on port %d.", m_port));
    push();
    String s;
    while (domore) {
      try {
        byte[] buf = new byte[256];

        // receive request
        DatagramPacket packet = new DatagramPacket(buf, buf.length);
        socket.receive(packet);
        s = new String(packet.getData()).trim();
        forwardMsg(AWeatherService.MESSAGE_RECEIVED,
            new Date().toString() + " : " + packet.getAddress()
                + ": " + s);
        if (s.equals("update"))
          forwardMsg(AWeatherService.MESSAGE_RECEIVED, s);

      } catch (IOException e) {
        // gets here when ReceiverThread::close() is called.
        domore = false;
      }
    }
    socket.close();
  }

  public NetworkListener(final AWeatherService a, int port)
      throws SocketException {
    super("NetworkListener");
    m_service = a;
    socket = new DatagramSocket(port);
    m_port = port;
  }
}
package org.nws.aweather;

public interface NetworkTaskEvents {
  public void handleConnected(Msg o);

  public void handleDisconnected(Msg o);

  public void handleError(Msg o);

  public void handleInfo(Msg o);

  public void handleReceived(Msg o);
}
package org.nws.aweather;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Date;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;

import android.os.Handler;
import android.os.Message;

import com.nws.nwsserver.Info;

public class Responder {
  private final Handler m_handler = new Handler() {
    @Override
    public void handleMessage(final Message msg) {
      final Msg m = (Msg) msg.obj;
      ;
      switch (msg.what) {
      case AWeatherService.MESSAGE_DISCONNECTED:
        m_service.handleDisconnected(m);
        break;

      case AWeatherService.MESSAGE_CONNECTED:
        m_service.handleConnected(m);
        break;

      case AWeatherService.MESSAGE_ERROR:
        m_service.handleError(m);
        break;

      case AWeatherService.MESSAGE_INFO:
        m_service.handleInfo(m);
        break;

      case AWeatherService.MESSAGE_RECEIVED:
        m_service.handleReceived(m);
        break;
      }
    }
  };
  private AWeatherService m_service;
  private final BlockingQueue<Object> queue = new LinkedBlockingQueue<Object>(
      1024);
  protected DatagramSocket socket = null;

  void begin() {
    try {
      socket = new DatagramSocket();
    } catch (SocketException e1) {
      // TODO Auto-generated catch block
      e1.printStackTrace();
    }
    new Thread(new Runnable() {
      @Override
      public void run() {
        while (true) {
          try {
            Object o = queue.take();
            process(o);
            Thread.sleep(750);
          } catch (InterruptedException e) {
            e.printStackTrace();
          }
        }
      }
    }, "Consumer").start();
  }

  private void forwardMsg(final int type, final String s) {
    final Message m = Message.obtain(m_handler, type);
    m.obj = new Msg(s);
    m_handler.sendMessage(m);
  }

  private void process(Object o) {
    if (o instanceof DatagramPacket)
      processDGP(o);
    else if (o instanceof String)
      processStr(o);
    else if (o instanceof Info)
      processInfo(o);
  }

  private void processDGP(Object o) {
    DatagramPacket packet = (DatagramPacket) o;
    InetAddress address = packet.getAddress();
    int port = packet.getPort();
    byte[] data = packet.getData();
    try {
      packet = new DatagramPacket(data, data.length, address, port);
    } catch (Exception e) {
      e.printStackTrace();
      return;
    }
    try {
      socket.send(packet);
    } catch (IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }
  }

  private void processInfo(Object o) {
    forwardMsg(AWeatherService.MESSAGE_INFO, new Date().toString() + " : "
        + ((Info) o).toString());
  }

  private void processStr(Object o) {
    String s = ((String) (o)).trim();
    if (s.equals("update"))
      forwardMsg(AWeatherService.MESSAGE_RECEIVED, s);
    else forwardMsg(AWeatherService.MESSAGE_INFO, new Date().toString() + " : "
        + ((String) (o)).trim());

  }

  public void close() {
    socket.close();
    System.out.println("Responder closed.");
  }

  public void respond(Object o) throws InterruptedException {
    if (o != null)
      queue.put(o);
  }

  public Responder(final AWeatherService a) {
    m_service = a;
  }
}
package org.nws.aweather;

import android.app.Activity;
import android.app.ActivityManager;
import android.app.ActivityManager.RunningServiceInfo;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class ServiceLauncher extends Activity {
  private Button start;
  private final OnClickListener startListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      startService(new Intent(ServiceLauncher.this, AWeatherService.class));
      update();
    }
  };

  private Button stop;

  private final OnClickListener stopListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      stopService(new Intent(ServiceLauncher.this, AWeatherService.class));
      update();
    }
  };

  private boolean serviceIsRunning() {
    ActivityManager manager = (ActivityManager) getSystemService(Context.ACTIVITY_SERVICE);
    for (RunningServiceInfo service : manager
        .getRunningServices(Integer.MAX_VALUE)) {
      if (AWeatherService.class.getName().equals(
          service.service.getClassName())) {
        return true;
      }
    }
    return false;
  }

  private void update() {
    boolean f = serviceIsRunning();
    start.setEnabled(!f);
    stop.setEnabled(f);
  }

  /** Called when the activity is first created. */
  @Override
  public void onCreate(final Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.servicelauncher);
    start = (Button) findViewById(R.id.startButton);
    stop = (Button) findViewById(R.id.stopButton);
    start.setOnClickListener(startListener);
    stop.setOnClickListener(stopListener);
    update();
  }

}
package org.nws.aweather;

import java.net.InetAddress;
import java.net.UnknownHostException;

import org.nws.aweather.forecast.FetchForecast;

import android.app.Activity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class SetServer extends Activity {
  private Button cancelButton;
  private final OnClickListener cancelListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      finish();
    }
  };
  private Button defaultButton;
  private final OnClickListener defaultListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      String s = FetchForecast.getInstance().getDefaultUrl();
      url.setText(s);
    }
  };
  private Button setUrlButton;

  private final OnClickListener setUrlListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      if (isvalidurl()){
        String s = url.getText().toString();
        FetchForecast.getInstance().setUrl(s);
            setResult( RESULT_OK, getIntent( ) );
        finish();
      }
      else
        Toast.makeText(SetServer.this, "Invalid Url", Toast.LENGTH_LONG)
            .show();
    }
  };

  private EditText url;
  private boolean isvalidurl() {
    String s = url.getText().toString();
    try {
      @SuppressWarnings("unused")
      InetAddress serverAddress = InetAddress.getByName(s);
    } catch (UnknownHostException e) {
      return false;
    }
    return true;
  }

  private void update()
  {
    setUrlButton.setEnabled(isvalidurl());
  }

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.setserver);
      setResult( RESULT_CANCELED, getIntent( ) );
    url = (EditText) findViewById(R.id.enterurl);
    String s = FetchForecast.getInstance().getUrl();
    url.setText(s);
    setUrlButton = (Button) findViewById(R.id.setUrlButton);
    defaultButton = (Button) findViewById(R.id.defaultButton);
    cancelButton = (Button) findViewById(R.id.CancelButton);
    setUrlButton.setOnClickListener(setUrlListener);
    defaultButton.setOnClickListener(defaultListener);
    cancelButton.setOnClickListener(cancelListener);
    url.addTextChangedListener(new TextWatcher() {

      @Override
      public void afterTextChanged(Editable arg0) {

      }

      @Override
      public void beforeTextChanged(CharSequence arg0, int arg1,
          int arg2, int arg3) {

      }

      @Override
      public void onTextChanged(CharSequence arg0, int arg1, int arg2,
          int arg3) {
        update();

      }


      });
  }

}
package org.nws.aweather;

import java.io.BufferedInputStream;
import java.io.IOException;
import java.net.URL;
import java.net.URLConnection;
import java.util.List;

import org.nws.aweather.forecast.FetchForecast;
import org.nws.aweather.forecast.NWSWeatherObject;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Handler;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;

public class WeatherGallery {
  private final Handler handler;
  private final ArrayAdapter<String> locationAdapter;
  private final Spinner locationChooser;
  private final LocationManager locationManager;
  private final AWeather owner;
  private final Button refreshButton;

  private Bitmap getImage(final String httpLocation) throws IOException {
    Bitmap result = null;
    final URL url = new URL(httpLocation);
    final URLConnection con = url.openConnection();
    final BufferedInputStream in = new BufferedInputStream(
        con.getInputStream());
    result = BitmapFactory.decodeStream(in);
    in.close();
    return result;
  }

  private void updateWeather(final NWSWeatherObject nwo) {
    final ImageView i = (ImageView) owner.findViewById(R.id.image1);
    if (i == null)
      // view has bailed on us- just fail fast
      return;
    try {
      final List<String> conditions = nwo.getConditionIconLinks();
      final String first = conditions.get(0);
      final Bitmap bm = getImage(first);
      i.setImageBitmap(bm);
    } catch (final IOException e) {
      throw new RuntimeException(e);
    }
    final TextView t = (TextView) owner.findViewById(R.id.day);
    t.setText("Tomorrow");
  }

  private void updateWeather(final String s) {
    final ImageView i = (ImageView) owner.findViewById(R.id.image1);
    if (i == null)
      // view has bailed on us- just fail fast
      return;
    try {
      final Bitmap bm = getImage(s);
      i.setImageBitmap(bm);
    } catch (final IOException e) {
      throw new RuntimeException(e);
    }
    final TextView t = (TextView) owner.findViewById(R.id.day);
    t.setText("Tomorrow");
  }

  public void gotWeather(final NWSWeatherObject nwo) {
    final Thread X = new Thread() {
      @Override
      public void run() {
        updateWeather(nwo);
      }
    };
    handler.post(X);
  }

  public void gotWeather(final String s) {
    final Thread X = new Thread() {
      @Override
      public void run() {
        updateWeather(s);
      }
    };
    handler.post(X);
  }

  public void logError(Object p) {
    owner.forwardMsg(AWeatherService.MESSAGE_ERROR, p);
  }

  public void refresh() {
    final String[] location = locationManager.getLocation(0);
    FetchForecast.getInstance().requestForecast(location[1], location[2]);
  }

  public WeatherGallery(final AWeather owner) {
    this.owner = owner;
    FetchForecast.parent = this;
    locationManager = owner.getLocationManager();

    locationChooser = (Spinner) owner.findViewById(R.id.locationChooser);
    refreshButton = (Button) owner.findViewById(R.id.refreshButton);
    refreshButton.setOnClickListener(owner);
    handler = new Handler();
    locationAdapter = new ArrayAdapter<String>(
        owner.getApplicationContext(),
        android.R.layout.simple_spinner_item);
    final String[] location = locationManager.getLocation(locationManager
        .getSelected());
    locationAdapter.add(location[0]);
    String s = locationManager.getServer();
    FetchForecast.getInstance().setUrl(s);
    FetchForecast.getInstance().requestForecast(location[1], location[2]);
    locationChooser.setAdapter(locationAdapter);
  }
}
```

```java
package com.nws.nwsserver;

import java.io.Serializable;

public class Info implements Serializable {
  private static final long serialVersionUID = 1L;
  public String data;
  public int id;
  public String lat;
  public String lon;

  @Override
  public String toString() {
    return String.format("id: %d, lat=%s, long=%s, data=%s", id, lat, lon,
        data);
  }
}
package org.nws.aweather;

import java.util.TreeMap;

import android.content.Context;
import android.location.Address;
import android.location.Geocoder;
import android.os.Build;
import android.os.Handler;
import android.text.Editable;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnKeyListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

public class AddLocation implements OnClickListener, OnKeyListener,
    OnItemSelectedListener {
  private final class GeocodeResultsHandler implements Runnable {
    @Override
    public void run() {
      geocodeFinished();
    }
  }

  public static final int GEOCODE_STATUS_LOOKUP_FAIL = 2;
  public static final int GEOCODE_STATUS_NETWORK_ERROR = 1;
  public static final int GEOCODE_STATUS_SUCCESS = 3;
  private final Context context;
  private final EditText enterLocation;
  private final boolean geocoding;
  private final TextView log;
  private final Button lookupLocation;
  private final AWeather owner;
  private final Button showweather;
  protected Address geocoded;
  protected int geocodeStatus;
  protected final GeocodeResultsHandler grh = new GeocodeResultsHandler();
  protected Handler handler = new Handler();

  private synchronized void geocode(final String place) {
    if (geocoding)
      return;
    lookupLocation.setEnabled(false);
    showweather.setEnabled(false);
    log(owner.getString(R.string.searchingForLocation));
    final GeocodePlace gp = new GeocodePlace(place, this, context);
    new Thread(gp).start();
  }

  public void geocodeFinished() {
    lookupLocation.setEnabled(true);
    showweather.setEnabled(true);
    if (geocodeStatus == GEOCODE_STATUS_SUCCESS) {
      final StringBuilder sb = new StringBuilder(
          owner.getString(R.string.foundLocation));
      sb.append(geocoded.getLocality()).append(" (")
          .append(geocoded.getAdminArea()).append(")");
      log(sb.toString());
      owner.getLocationManager().storeLocation(geocoded);
    } else if (geocodeStatus == GEOCODE_STATUS_NETWORK_ERROR)
      log(owner.getString(R.string.geocodeNetworkError));
    else
      // geocode status will be lookup fail
      log(owner.getString(R.string.geocodeLookupFailure));
    geocoded = null;
    geocodeStatus = 0;
  }

  public void log(final String text) {
    log.setText(text);
  }

  @Override
  public void onClick(final View v) {
    if (v == lookupLocation) {
      final Editable editable = enterLocation.getText();
      final String s = editable.toString();
      if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.GINGERBREAD
          && Geocoder.isPresent())
        geocode(s);
      else
        log("Geocoder not present");
    } else if (v == showweather) {
      owner.restart();
    }
  }

  @Override
  public void onItemSelected(final AdapterView<?> adapter, final View arg1,
      final int position, final long id) {
    showweather.setEnabled(true);
    owner.getLocationManager().storeSelected(position);
  }

  @Override
  public boolean onKey(final View v, final int keyCode, final KeyEvent event) {
    if (v == enterLocation) {
      final Editable editable = enterLocation.getText();
      final String s = editable.toString();
      if (s.length() == 0)
        lookupLocation.setEnabled(false);
      else
        lookupLocation.setEnabled(true);
      return false;
    }
    if (event.getKeyCode() == KeyEvent.KEYCODE_ENTER) {
      onClick(lookupLocation);
      return true;
    }
    return false;
  }

  @Override
  public void onNothingSelected(final AdapterView<?> arg0) {
    // don't do anything...
  }

  public AddLocation(final AWeather owner) {
    this.owner = owner;
    context = owner.getApplicationContext();
    log = (TextView) owner.findViewById(R.id.log);
    enterLocation = (EditText) owner.findViewById(R.id.enterLocation);
    enterLocation.setOnKeyListener(this);
    lookupLocation = (Button) owner.findViewById(R.id.fireLocationLookup);
    lookupLocation.setOnClickListener(this);
    lookupLocation.setEnabled(false);
    showweather = (Button) owner.findViewById(R.id.buttonShowWeather);
    showweather.setOnClickListener(this);
    showweather.setEnabled(false);
    geocoding = false;
    final Spinner spinner = (Spinner) owner
        .findViewById(R.id.locationSpinner);
    final ArrayAdapter<String> adapter = new ArrayAdapter<String>(owner,
        android.R.layout.simple_spinner_item);
    adapter.setDropDownViewResource(android.R.layout.simple_dropdown_item_1line);
    spinner.setAdapter(adapter);
    final LocationManager locationManager = new LocationManager(owner);
    final TreeMap<String, String[]> locations = locationManager
        .getLocations();
    for (final String id : locations.keySet()) {
      final String[] vals = locations.get(id);
      adapter.add(vals[0]);
    }
    int selected = locationManager.getSelected();
    spinner.setOnItemSelectedListener(this);
    spinner.setSelection(selected);
    log(owner.getString(R.string.addLocationExample));
  }
}
package org.nws.aweather;

import org.nws.aweather.forecast.FetchForecast;

import android.app.Activity;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.Handler;
import android.os.IBinder;
import android.os.Message;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;

public class AWeather extends Activity implements OnClickListener {
  public static final int MENU_ITEM_MANAGE_LOCATIONS = 0;
  public static final int MENU_ITEM_SET_SERVER = 2;
  public static final int MENU_ITEM_START_SERVICE = 1;
  private AddLocation addLocation;
  private LocationManager locationManager;
  private ServiceConnection m_connection = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName className, IBinder service) {
      m_service = ((AWeatherService.LocalBinder) service).getService();
      m_service.setWeatherGallery(weatherGallery);
    }

    @Override
    public void onServiceDisconnected(ComponentName className) {
      m_service = null;
    }
  };
  private final Handler m_handler = new Handler() {
    @Override
    public void handleMessage(final Message msg) {
      final Msg m = (Msg) msg.obj;
      ;
      switch (msg.what) {
      case AWeatherService.MESSAGE_DISCONNECTED:
        m_service.handleDisconnected(m);
        break;

      case AWeatherService.MESSAGE_CONNECTED:
        m_service.handleConnected(m);
        break;

      case AWeatherService.MESSAGE_ERROR:
        m_service.handleError(m);
        break;

      case AWeatherService.MESSAGE_INFO:
        m_service.handleInfo(m);
        break;

      case AWeatherService.MESSAGE_RECEIVED:
        m_service.handleReceived(m);
        break;
      }
    }
  };
  private static final int SET_SERVER = 10;
      private boolean m_isbound = false;

  private AWeatherService m_service;

  private WeatherGallery weatherGallery = null;

  void doBindService() {
    bindService(new Intent(AWeather.this, AWeatherService.class),
        m_connection, Context.BIND_AUTO_CREATE);
    m_isbound = true;
  }

  void doUnbindService() {
    if (m_isbound) {
      unbindService(m_connection);
      m_isbound = false;
    }
  }

  protected LocationManager getLocationManager() {
    return locationManager;
  }

  @Override
  protected void onDestroy() {
    super.onDestroy();
    m_service.setWeatherGallery(null);
    doUnbindService();
  }

  public void forwardMsg(final int type, final Object s) {
    final Message m = Message.obtain(m_handler, type);
    m.obj = new Msg(s);
    m_handler.sendMessage(m);
  }

  @Override
  public void onClick(View view) {
    weatherGallery.refresh();
    //m_service.m_networklistener.push();
  }

  @Override
  public void onCreate(final Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    locationManager = new LocationManager(this);
    if (locationManager.getLocationCount() == 0) {
      setContentView(R.layout.addlocation);
      addLocation = new AddLocation(this);
    } else {
      setContentView(R.layout.main);
      weatherGallery = new WeatherGallery(this);
    }
    doBindService();
    startService(new Intent(this, AWeatherService.class));
  }

  @Override
  public boolean onCreateOptionsMenu(final Menu menu) {
    super.onCreateOptionsMenu(menu);
    menu.add(0, MENU_ITEM_MANAGE_LOCATIONS, 0, R.string.manageLocations);
    menu.add(0, MENU_ITEM_START_SERVICE, 0, R.string.manageService);
    menu.add(0, MENU_ITEM_SET_SERVER, 0, R.string.setserver);
    return true;
  }

  @Override
  public boolean onOptionsItemSelected(final MenuItem item) {
    switch (item.getItemId()) {
    case MENU_ITEM_MANAGE_LOCATIONS:
      setContentView(R.layout.addlocation);
      if (addLocation == null)
        addLocation = new AddLocation(this);
      return true;
    case MENU_ITEM_START_SERVICE:
      startActivity(new Intent(this, ServiceLauncher.class));
      return true;
    case MENU_ITEM_SET_SERVER:
      startActivityForResult(new Intent(this, SetServer.class), SET_SERVER);
      return true;
    }
    return false;
  }

  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent data) {
      if ( requestCode == SET_SERVER && resultCode == Activity.RESULT_OK ){
        String s = FetchForecast.getInstance().getUrl();
        locationManager.storeServer(s);
        restart();
      }
    super.onActivityResult(requestCode, resultCode, data);
  }

  public void restart() {
    stopService(new Intent(this, AWeatherService.class));
    Intent i = getBaseContext().getPackageManager()
        .getLaunchIntentForPackage(getBaseContext().getPackageName());
    i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(i);
  }
}package org.nws.aweather;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.SocketException;
import java.util.Calendar;
import java.util.Timer;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Intent;
import android.os.Binder;
import android.os.IBinder;
import android.widget.Toast;

public class AWeatherService extends Service implements NetworkTaskEvents {

  public class LocalBinder extends Binder {
    AWeatherService getService() {
      return AWeatherService.this;
    }
  }

  private static boolean showToasts = false;
  public static final int MESSAGE_CONNECTED = 0;
  public static final int MESSAGE_DISCONNECTED = 1;
  public static final int MESSAGE_ERROR = 2;
  public static final int MESSAGE_INFO = 3;

  public static final int MESSAGE_RECEIVED = 4;

  private Long m_counter = 0L;

  public NetworkListener m_networklistener;

  private Responder m_responder;

  private WeatherGallery m_weathergallery = null;

  private final IBinder mBinder = new LocalBinder();

  private NotificationManager nm;

  private final Calendar time = Calendar.getInstance();

  private final Timer timer = new Timer();

  private void connect() {
    try {
      m_responder = new Responder(this);
      m_responder.begin();
      m_networklistener = new NetworkListener(this, 6001);
      m_networklistener.start();
    } catch (SocketException e) {
    }
  }

  private void destroyTimer() {
    if (timer != null)
      timer.cancel();
  }

  private void disconnect() {
    if (m_networklistener != null)
      m_networklistener.close();
    if (m_responder != null)
      m_responder.close();
  }

  /** Show a notification while this service is running. */
  @SuppressWarnings("deprecation")
  private void showNotification() {
    // Use the same text for the ticker and the expanded notification
    final CharSequence text = getText(R.string.service_started);
    // Set the icon, scrolling text and timestamp
    final Notification notification = new Notification(R.drawable.android,
        text, System.currentTimeMillis());
    // The PendingIntent to launch our activity if the user selects this
    // notification
    final PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
        new Intent(this, ServiceLauncher.class), 0);
    // Set the info for the views that show in the notification panel.
    notification.setLatestEventInfo(this, getText(R.string.service_label),
        text, contentIntent);
    // Send the notification.
    // We use a layout id because it is a unique number. We use it later to
    // cancel.
    nm.notify(R.string.service_started, notification);
  }

  @Override
  public void handleConnected(Msg o) {
    Object ob = o.getInfo();
    String s = "Connected";
    if (ob instanceof String)
      s = (String) (ob);
    if (showToasts)
      Toast.makeText(this, s, Toast.LENGTH_LONG).show();
  }

  @Override
  public void handleDisconnected(Msg o) {
    Object ob = o.getInfo();
    String s = "Disconnected";
    if (ob instanceof String)
      s = (String) (ob);
    if (showToasts)
      Toast.makeText(this, s, Toast.LENGTH_LONG).show();
  }

  @Override
  public void handleError(Msg o) {
    Object ob = o.getInfo();
    if (ob instanceof String) {
      String s = (String) (ob);
      if (showToasts)
        Toast.makeText(this, s, Toast.LENGTH_LONG).show();
    } else if (ob instanceof DatagramPacket) {
      try {
        m_responder.respond(ob);
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    } else if (ob instanceof byte[]) {
      byte bytes[] = (byte[]) ob;
      try {
        m_responder.respond(ConvertUtils.deserializePacket(bytes));
      } catch (InterruptedException e) {
        e.printStackTrace();
      } catch (IOException e) {
        e.printStackTrace();
      }
    }
  }

  @Override
  public void handleInfo(final Msg o) {
    Object ob = o.getInfo();
    String s = "Info";
    if (ob instanceof String)
      s = (String) (ob);
    if (showToasts)
      Toast.makeText(this, s, Toast.LENGTH_LONG).show();
  }

  @Override
  public void handleReceived(Msg o) {
    Object ob = o.getInfo();
    String s = "Received";
    if (ob instanceof String)
      s = (String) (ob);
    if (s.equals("update"))
      if (m_weathergallery != null)
        m_weathergallery.refresh();
    if (showToasts)
      Toast.makeText(this, s, Toast.LENGTH_LONG).show();
  }

  @Override
  public IBinder onBind(final Intent intent) {
    return mBinder;
  }

  @Override
  public void onCreate() {
    super.onCreate();
    nm = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
    if (showToasts)
      Toast.makeText(this, "Service created at " + time.getTime(),
          Toast.LENGTH_LONG).show();
    showNotification();
    // createTimer();
    connect();
  }

  @Override
  public void onDestroy() {
    super.onDestroy();
    // Cancel the persistent notification.
    destroyTimer();
    disconnect();
    nm.cancel(R.string.service_started);
    if (showToasts)
      Toast.makeText(
          this,
          "Service destroyed at " + time.getTime()
              + "; m_counter is at: " + m_counter,
          Toast.LENGTH_LONG).show();
    m_counter = null;
  }

  @Override
  public int onStartCommand(Intent intent, int flags, int startId) {
    return START_STICKY;
  }

  public void setWeatherGallery(WeatherGallery a) {
    m_weathergallery = a;
  }

}
package org.nws.aweather;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.ObjectInput;
import java.io.ObjectInputStream;
import java.io.ObjectOutput;
import java.io.ObjectOutputStream;
import java.net.DatagramPacket;
import java.net.InetAddress;

public class ConvertUtils {

  static public DatagramPacket deserializePacket(byte[] packetBytes)
      throws IOException {
    DatagramPacket udp = new DatagramPacket(new byte[1024], 1024);

    ByteArrayInputStream bis = new ByteArrayInputStream(packetBytes);
    ObjectInput in = null;
    try {
      in = new ObjectInputStream(bis);
      udp.setAddress((InetAddress) in.readObject());
      udp.setPort((Integer) in.readObject());
      udp.setData((byte[]) in.readObject());
    } catch (IOException e) {
      e.printStackTrace();
      return null;
    } catch (ClassNotFoundException e) {
      e.printStackTrace();
      return null;
    } finally {
      in.close();
      bis.close();
    }

    return udp;
  }

  public int numPackets = 0;

  public void serializePacket(byte[] packetBytes) {
    // record the number of packets and the time between packets received
    // avgInterval
    numPackets += 1;
  }

  public void serializePacket(DatagramPacket udp) throws IOException {
    // byte[] packetBytes = new byte[4096];
    //
    // byte[] tmp = udp.getAddress().getAddress();
    // int offset = 0;
    // System.arraycopy( tmp, 0, packetBytes, offset, tmp.length);
    // offset += tmp.length;
    //
    // ByteBuffer bb = ByteBuffer.allocate(4);
    // bb.putInt(udp.getPort());
    // System.arraycopy( bb.array(), 0, packetBytes, offset, 4);
    // offset += 4;
    //
    // bb.putInt(udp.getLength());
    // System.arraycopy( bb.array(), 0, packetBytes, offset, 4);
    // offset += 4;
    //
    // System.arraycopy( udp.getData(), 0, packetBytes, offset,
    // udp.getLength());
    // serializePacket(packetBytes);

    ByteArrayOutputStream bos = new ByteArrayOutputStream();
    ObjectOutput out = null;
    try {
      out = new ObjectOutputStream(bos);
      out.writeObject(udp.getAddress());
      out.write(udp.getPort());
      out.writeObject(udp.getData());
      serializePacket(bos.toByteArray());
    } catch (IOException e) {
      e.printStackTrace();
    } finally {
      out.close();
      bos.close();
    }
  }
}package org.nws.aweather.forecast;

import java.io.BufferedReader;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.ObjectInput;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.Socket;
import java.net.SocketException;
import java.net.UnknownHostException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Random;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.TimeUnit;

import org.nws.aweather.WeatherGallery;
import org.xml.sax.helpers.DefaultHandler;

import com.nws.nwsserver.Info;

public class FetchForecast extends DefaultHandler implements Runnable {

  private static final String BEGIN = "&begin=";
  private static final DateFormat DATE_FORMAT = new SimpleDateFormat(
      "yyyy-MM-dd'T'hh':00:00'");
  private static final String END = "&end=";
  private static final String HOST = "graphical.weather.gov";
  private static final String LAT = "&lat=";
  private static final String LON = "&lon=";
  private static final int MAX_RETRIES = 3;
  private static final int PORT = 5559;
  private static FetchForecast theInstance = null;

  private static final long TIMEOUT_SECS = 10;

  private static final int UDP_WC_PORT = 10000;
  private static final String UDP_LOCAL = "www.weatherpushupdates.com";
  private static final int UDP_WS_PORT = 5559;
  //public static String udpWsHost = "192.168.19.207";
  public static String udpWsHost = UDP_LOCAL;
  private static final String URL_PREFIX = "/xml/sample_products/browser_interface/ndfdXMLclient.php?"
      + "&product=time-series&icons=icons";
  public static WeatherGallery parent = null;
  public static FetchForecast getInstance() {
    if (theInstance == null) {
      theInstance = new FetchForecast();
      new Thread(theInstance).start();
    }
    return theInstance;
  }

  private final boolean activeFlag = true;
  private Info currentRequestMsg;
  private final BlockingQueue<Info> recvQueue = new LinkedBlockingQueue<Info>(
      1024);
  private int requestId = (new Random()).nextInt();
  private InetAddress serverAddress;

  private DatagramSocket socket;
  private String urlString;

  public boolean useUDPProtocol = true;

  private void getDirectNWS(String lat, String lon) {

    final StringBuilder url = new StringBuilder(URL_PREFIX);
    url.append(LAT).append(lat).append(LON).append(lon);
    final Calendar start = Calendar.getInstance();
    final Calendar end = Calendar.getInstance();
    end.add(Calendar.DAY_OF_MONTH, 7);
    url.append(BEGIN).append(DATE_FORMAT.format(start.getTime()));
    url.append(END).append(DATE_FORMAT.format(end.getTime()));
    urlString = url.toString();

    try {
      System.out.println("Ripping: " + urlString);
      final Socket s = new Socket(HOST, PORT);
      final OutputStream os = s.getOutputStream();
      final PrintWriter out = new PrintWriter(new OutputStreamWriter(os));
      final StringBuilder request = new StringBuilder("GET ").append(
          urlString).append("HTTP/1.1");
      request.append("\n");
      request.append("Host: ").append(HOST).append("\n");
      request.append("\n");
      out.print(request);
      out.flush();
      final InputStream in = s.getInputStream();
      // String t = getIn(in);
      //
      // int n = t.length( );
      final NWSWeatherXMLParser parser = new NWSWeatherXMLParser(in);
      parser.parse(parent);
      System.out.println("Finished parsing");
    } catch (final Exception e) {
      throw new RuntimeException(e);
    }
  }

  @SuppressWarnings("unused")
  private String getIn(final InputStream in) {
    final InputStreamReader is = new InputStreamReader(in);
    final StringBuilder sb = new StringBuilder();
    final BufferedReader br = new BufferedReader(is);
    String read = null;
    try {
      read = br.readLine();
    } catch (final IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }
    while (read != null) {
      // System.out.println(read);
      sb.append(read);
      try {
        read = br.readLine();
      } catch (final IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
      }
    }
    return sb.toString();
  }

  private void getUDPWeatherServce(String lat, String lon) {
    currentRequestMsg = new Info();
    currentRequestMsg.data = "REQW_GETFORCAST";
    currentRequestMsg.lat = lat;
    currentRequestMsg.lon = lon;

    int nTries = MAX_RETRIES;
    currentRequestMsg.id = requestId++;
    while (nTries > 0) {
      nTries--;
      // currentRequestMsg.id = requestId++;

      byte[] sendBuf = null;
      try {
        ByteArrayOutputStream bos = new ByteArrayOutputStream();
        ObjectOutputStream out = new ObjectOutputStream(bos);
        out.writeObject(currentRequestMsg);
        out.close();
        sendBuf = bos.toByteArray();
        bos.close();
      } catch (IOException e) {
        parent.logError(e.getMessage());
        return;
      }

      DatagramPacket sendPkt = new DatagramPacket(sendBuf,
          sendBuf.length, serverAddress, UDP_WS_PORT);

      try {
        if ((socket != null) && (serverAddress != null)) {
          socket.send(sendPkt);
        }
      } catch (IOException e1) {
        parent.logError(e1.getMessage());
        return;
      }

      Info recvMsg;
      do {
        try {
          recvMsg = recvQueue.poll(TIMEOUT_SECS, TimeUnit.SECONDS);
          if (recvMsg != null)
            if (isValid(recvMsg, currentRequestMsg)) {
              parent.gotWeather(recvMsg.data);
              return;
            }

        } catch (InterruptedException e) {
          // TODO Auto-generated catch block
          e.printStackTrace();
        } catch (Exception e) {
          // TODO Auto-generated catch block
          e.printStackTrace();
        }
      } while (!recvQueue.isEmpty());
    }
  }

  private boolean isValid(Info info_in, Info info_out) {
    return info_in.id == info_out.id;
  }

  public String getDefaultUrl() {
    return UDP_LOCAL;
  }

  public String getUrl() {
    return udpWsHost;
  }

  public void initializeUDP() {
    try {
      serverAddress = InetAddress.getByName(udpWsHost);
    } catch (UnknownHostException e) {
      parent.logError(e.getMessage());
      return;
    }
    try {
      socket = new DatagramSocket(UDP_WC_PORT);
    } catch (SocketException e) {
      parent.logError(e.getMessage());
    }
  }

  public void requestForecast(String lat, String lon) {
    if (!useUDPProtocol)
      getDirectNWS(lat, lon);
    else
      getUDPWeatherServce(lat, lon);
  }

  @Override
  public void run() {
    initializeUDP();
    while (activeFlag) {
      byte recvBuf[] = new byte[4096];
      DatagramPacket recvPkt = new DatagramPacket(recvBuf, recvBuf.length);

      try {
        socket.receive(recvPkt);
      } catch (IOException e) {
        parent.logError(e.getMessage());
        continue;
      }
      Info recvInfo = null;
      try {
        ByteArrayInputStream bis = new ByteArrayInputStream(
            recvPkt.getData());
        ObjectInput in = new ObjectInputStream(bis);
        recvInfo = (Info) in.readObject();
      } catch (Exception e) {
        parent.logError(recvPkt.getData());
        continue;
      }

      if (isValid(recvInfo, currentRequestMsg)) {
        recvQueue.offer(recvInfo);
      } else {
        parent.logError("Invalid id.");
      }
    }
  }

  public void setUrl(String url) {
    udpWsHost = url;
  }
}
package org.nws.aweather.forecast;

import java.util.ArrayList;
import java.util.Date;
import java.util.LinkedHashMap;
import java.util.List;

public class NWSWeatherObject {
  private final List<String> conditionIconLinks;
  private String conditionIconTimeLayout;
  private Date createdOn;
  private String moreInfoLink;
  private String mostRecentTimeLayout;
  private final LinkedHashMap<String, List<TimeLayoutBand>> timeLayouts;

  public void addConditionIconLink(final String link) {
    conditionIconLinks.add(link);
  }

  public void addNewTimeBand(final Date date) {
    final List<TimeLayoutBand> l = timeLayouts.get(mostRecentTimeLayout);
    final TimeLayoutBand tlb = new TimeLayoutBand(date);
    l.add(tlb);
  }

  public void addTimeLayout(final String name) {
    mostRecentTimeLayout = name;
    timeLayouts.put(name, new ArrayList<TimeLayoutBand>());
  }

  public void closeCurrentTimeBand(final Date end) {
    final List<TimeLayoutBand> l = timeLayouts.get(mostRecentTimeLayout);
    final TimeLayoutBand last = l.get(l.size() - 1);
    last.setEndDate(end);
  }

  public List<String> getConditionIconLinks() {
    return conditionIconLinks;
  }

  public String getConditionIconTimeLayout() {
    return conditionIconTimeLayout;
  }

  public Date getCreatedOn() {
    return createdOn;
  }

  public String getMoreInfoLink() {
    return moreInfoLink;
  }

  public void setConditionIconTimeLayout(final String name) {
    conditionIconTimeLayout = name;
  }

  public void setCreatedOn(final Date date) {
    createdOn = date;
  }

  public void setMoreInfoLink(final String moreInfoLink) {
    this.moreInfoLink = moreInfoLink;
  }

  public NWSWeatherObject() {
    timeLayouts = new LinkedHashMap<String, List<TimeLayoutBand>>();
    conditionIconLinks = new ArrayList<String>();
  }
}
package org.nws.aweather.forecast;

import java.io.InputStream;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import org.nws.aweather.WeatherGallery;
import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

/**
 * Responsible for parsing through the XML recovered from an NWS hit. There are
 * two core sections to look for:
 * <ul>
 * <li>time-layout: which contains sections/spans of time</li>
 * <li>parameters: which contains specific blocks of weather data</li>
 * </ul>
 *
 * @author keerat
 */
public class NWSWeatherXMLParser extends DefaultHandler {
  // condition icons
  private static final String CONDITIONS = "conditions-icon";
  private static final SimpleDateFormat CREATED_DATE_FORMATTER = new SimpleDateFormat(
      "yyyy-MM-dd'T'HH:mm:ss'Z'");
  // <creation-date
  // refresh-frequency="PT1H">2009-03-14T20:58:51Z</creation-date>
  private static final String CREATION_DATE = "creation-date";
  private static final String END_VALID_TIME = "end-valid-time";
  private static final String ICON_LINK = "icon-link";
  private static final String LAYOUT_KEY = "layout-key";
  // <moreWeatherInformation applicable-location="point1">
  // http://forecast.weather.gov/MapClick.php?textField1=38.99&textField2=-77.01
  // </moreWeatherInformation>
  private static final String MORE_INFO = "moreWeatherInformation";
  private static final String START_VALID_TIME = "start-valid-time";
  private static final String TIME_LAYOUT = "time-layout";
  // <start-valid-time>2009-03-14T08:00:00-04:00</start-valid-time>
  private static final SimpleDateFormat TIME_LAYOUT_DATE_FORMATTER = new SimpleDateFormat(
      "yyyy-MM-dd'T'HH:mm:ssZ");
  private String currentTag;
  private final InputStream in;
  private final NWSWeatherObject nwo;
  private WeatherGallery parent;

  private void handleCreationDate(final String dateString) {
    try {
      final Date date = CREATED_DATE_FORMATTER.parse(dateString);
      nwo.setCreatedOn(date);
    } catch (final ParseException e) {
    }
  }

  private void handleEndValidTime(final String endTimeString) {
    try {
      final Date date = TIME_LAYOUT_DATE_FORMATTER.parse(endTimeString);
      nwo.closeCurrentTimeBand(date);
    } catch (final ParseException e) {
    }
  }

  private void handleStartValidTime(final String startTimeString) {
    try {
      final Date date = TIME_LAYOUT_DATE_FORMATTER.parse(startTimeString);
      nwo.addNewTimeBand(date);
    } catch (final ParseException e) {
    }
  }

  @Override
  public void characters(final char[] ch, final int start, final int length)
      throws SAXException {
    if (currentTag.equals(CREATION_DATE))
      handleCreationDate(new String(ch, start, length));
    if (currentTag.equals(MORE_INFO))
      nwo.setMoreInfoLink(new String(ch, start, length));
    if (currentTag.equals(LAYOUT_KEY))
      nwo.addTimeLayout(new String(ch, start, length));
    if (currentTag.equals(START_VALID_TIME))
      handleStartValidTime(new String(ch, start, length));
    if (currentTag.equals(END_VALID_TIME))
      handleEndValidTime(new String(ch, start, length));
    if (currentTag.equals(ICON_LINK))
      nwo.addConditionIconLink(new String(ch, start, length));
  }

  @Override
  public void endDocument() throws SAXException {
    parent.gotWeather(nwo);
  }

  @Override
  public void endElement(final String uri, final String localName,
      final String name) throws SAXException {
    currentTag = "";
  }

  public void parse(final WeatherGallery parent) throws Exception {
    final SAXParserFactory saxParserFactory = SAXParserFactory
        .newInstance();
    final SAXParser parser = saxParserFactory.newSAXParser();
    this.parent = parent;
    parser.parse(in, this);
  }

  @Override
  public void startElement(final String uri, final String localName,
      final String name, final Attributes attributes) throws SAXException {
    currentTag = localName;
    if (currentTag.equals(CONDITIONS)) {
      // rip the time layout
      final String timeLayoutName = attributes.getValue(TIME_LAYOUT);
      nwo.setConditionIconTimeLayout(timeLayoutName);
    }
  }

  public NWSWeatherXMLParser(final InputStream in) {
    this.in = in;
    currentTag = "";
    nwo = new NWSWeatherObject();
  }
}
package org.nws.aweather.forecast;

import java.util.Date;

public class TimeLayoutBand {
  private Date end;
  private final Date start;

  public Date getEnd() {
    return end;
  }

  public Date getStart() {
    return start;
  }

  public boolean isPointInTime() {
    if (start.equals(end))
      return true;
    else
      return false;
  }

  public void setEndDate(final Date end) {
    this.end = end;
  }

  public TimeLayoutBand(final Date start) {
    this.start = start;
    end = start;
  }
}
package org.nws.aweather;

import java.io.IOException;
import java.util.List;
import java.util.Locale;

import android.content.Context;
import android.location.Address;
import android.location.Geocoder;

public class GeocodePlace implements Runnable {
  private final Context context;
  private final AddLocation owner;
  private final String place;

  @Override
  public void run() {
    final Geocoder geocoder = new Geocoder(context, Locale.getDefault());
    List<Address> addresses = null;
    boolean found = false;
    try {
      addresses = geocoder.getFromLocationName(place, 1);
      if (addresses != null && addresses.size() > 0) {
        found = true;
        owner.geocodeStatus = AddLocation.GEOCODE_STATUS_SUCCESS;
      } else
        owner.geocodeStatus = AddLocation.GEOCODE_STATUS_LOOKUP_FAIL;
    } catch (final IOException e) {
      owner.geocodeStatus = AddLocation.GEOCODE_STATUS_NETWORK_ERROR;
      found = false;
    }
    if (found) {
      final Address a = addresses.get(0);
      owner.geocoded = a;
    }
    owner.handler.post(owner.grh);
  }

  GeocodePlace(final String place, final AddLocation owner,
      final Context context) {
    this.place = place;
    this.owner = owner;
    this.context = context;
  }
}
package org.nws.aweather;

import java.util.Map;
import java.util.Set;
import java.util.TreeMap;

import org.nws.aweather.forecast.FetchForecast;

import android.content.Context;
import android.content.SharedPreferences;
import android.location.Address;

public class LocationManager {
  private static final String LOCAL_MAP_LOCATION = "llm.";
  private static final String LOCATION_LATITUDE = "location.lat.";
  private static final String LOCATION_LONGITUDE = "location.lon.";
  private static final String LOCATION_NAME = "location.name.";
  private static final String SELECTED = "selected";
  private static final String SERVER = "server";
  private int locationCount;
  private final TreeMap<String, String[]> locations;
  private int selected = 0;
  private String server = FetchForecast.udpWsHost;
  private final SharedPreferences sharedPreferences;

  public String[] getLocation(final int location) {
    final String[] loc = locations.get(LOCAL_MAP_LOCATION + location);
    return loc;
  }

  public int getLocationCount() {
    return locationCount;
  }

  public TreeMap<String, String[]> getLocations() {
    return locations;
  }

  public int getSelected() {
    return selected;
  }

  public String getServer() {
    return server;
  }

  public synchronized void storeLocation(final Address address) {
    if (address == null)
      System.out.println("Could not geocode location!");
    final SharedPreferences.Editor editor = sharedPreferences.edit();
    final String adminArea = address.getAdminArea();
    final String locality = address.getLocality();
    final StringBuilder sb = new StringBuilder(locality).append(" (")
        .append(adminArea).append(")");
    final String latitude = Double.toString(address.getLatitude());
    final String longitude = Double.toString(address.getLongitude());
    final String nameKey = LOCATION_NAME + locationCount;
    final String latKey = LOCATION_LATITUDE + locationCount;
    final String lonKey = LOCATION_LONGITUDE + locationCount;
    editor.putString(nameKey, sb.toString());
    editor.putString(latKey, latitude);
    editor.putString(lonKey, longitude);
    editor.commit();
    locations.put(LOCAL_MAP_LOCATION + locationCount,
        new String[] { sb.toString(), latitude, longitude });
    locationCount++;
  }

  public synchronized void storeSelected(final int i) {
    final SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putInt(SELECTED, i);
    editor.commit();
    selected = i;
  }

  public synchronized void storeServer(final String s) {
    final SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString(SERVER, s);
    editor.commit();
    server = s;
  }

  public LocationManager(final AWeather owner) {
    sharedPreferences = owner.getPreferences(Context.MODE_WORLD_READABLE);
    final Map<String, ?> allParams = sharedPreferences.getAll();
    final Set<String> keys = allParams.keySet();
    locations = new TreeMap<String, String[]>();
    locationCount = 0;
    for (final String key : keys)
      if (key.startsWith(LOCATION_NAME)) {
        final String name = sharedPreferences.getString(LOCATION_NAME
            + locationCount, "Corrupted");
        final String latitude = sharedPreferences.getString(
            LOCATION_LATITUDE + locationCount, "0");
        final String longitude = sharedPreferences.getString(
            LOCATION_LONGITUDE + locationCount, "0");
        locations.put(LOCAL_MAP_LOCATION + locationCount, new String[] {
            name, latitude, longitude });
        locationCount++;
      } else if (key.startsWith(SELECTED)) {
        selected = sharedPreferences.getInt(SELECTED, 0);
      } else if (key.startsWith(SERVER)) {
        server = sharedPreferences.getString(SERVER,  FetchForecast.udpWsHost);
      }

  }
}
package org.nws.aweather;

public class Msg {
  private final Object info;

  public String getDescription() {
    if (info instanceof String)
      return String.format("InfoMsg: info = %s", info);
    return info.toString();
  }

  public Object getInfo() {
    return info;
  }

  public Msg(final Object info) {
    this.info = info;
  }
}package org.nws.aweather;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Date;

import org.nws.aweather.forecast.FetchForecast;

import android.os.Handler;
import android.os.Message;

public class NetworkListener extends Thread {
  private boolean domore = true;

  private final Handler m_handler = new Handler() {
    @Override
    public void handleMessage(final Message msg) {
      final Msg m = (Msg) msg.obj;
      ;
      switch (msg.what) {
      case AWeatherService.MESSAGE_DISCONNECTED:
        m_service.handleDisconnected(m);
        break;

      case AWeatherService.MESSAGE_CONNECTED:
        m_service.handleConnected(m);
        break;

      case AWeatherService.MESSAGE_ERROR:
        m_service.handleError(m);
        break;

      case AWeatherService.MESSAGE_INFO:
        m_service.handleInfo(m);
        break;

      case AWeatherService.MESSAGE_RECEIVED:
        m_service.handleReceived(m);
        break;
      }
    }
  };

  private int m_port;
  private AWeatherService m_service;
  private DatagramSocket socket = null;

  private void forwardMsg(final int type, final String s) {
    final Message m = Message.obtain(m_handler, type);
    m.obj = new Msg(s);
    m_handler.sendMessage(m);

  }

  public void close() {
    socket.close();
    System.out.println("Receiver closed.");
  }

  public void push() {
    byte[] greet = "Hello".getBytes();
    DatagramPacket p;
    try {
      p = new DatagramPacket(greet, greet.length,
          InetAddress.getByName(FetchForecast.udpWsHost), m_port);
      socket.send(p);
      System.out.println(new Date().toString() + " :  Forward: "
          + p.getAddress());
    } catch (Exception e) {
      System.out.print(String.format("Exception: %s\n", e.getMessage()));
    }
  }

  @Override
  public void run() {
    forwardMsg(AWeatherService.MESSAGE_CONNECTED,
        String.format("Receiver started on port %d.", m_port));
    push();
    String s;
    while (domore) {
      try {
        byte[] buf = new byte[256];

        // receive request
        DatagramPacket packet = new DatagramPacket(buf, buf.length);
        socket.receive(packet);
        s = new String(packet.getData()).trim();
        forwardMsg(AWeatherService.MESSAGE_RECEIVED,
            new Date().toString() + " : " + packet.getAddress()
                + ": " + s);
        if (s.equals("update"))
          forwardMsg(AWeatherService.MESSAGE_RECEIVED, s);

      } catch (IOException e) {
        // gets here when ReceiverThread::close() is called.
        domore = false;
      }
    }
    socket.close();
  }

  public NetworkListener(final AWeatherService a, int port)
      throws SocketException {
    super("NetworkListener");
    m_service = a;
    socket = new DatagramSocket(port);
    m_port = port;
  }
}
package org.nws.aweather;

public interface NetworkTaskEvents {
  public void handleConnected(Msg o);

  public void handleDisconnected(Msg o);

  public void handleError(Msg o);

  public void handleInfo(Msg o);

  public void handleReceived(Msg o);
}
package org.nws.aweather;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Date;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;

import android.os.Handler;
import android.os.Message;

import com.nws.nwsserver.Info;

public class Responder {
  private final Handler m_handler = new Handler() {
    @Override
    public void handleMessage(final Message msg) {
      final Msg m = (Msg) msg.obj;
      ;
      switch (msg.what) {
      case AWeatherService.MESSAGE_DISCONNECTED:
        m_service.handleDisconnected(m);
        break;

      case AWeatherService.MESSAGE_CONNECTED:
        m_service.handleConnected(m);
        break;

      case AWeatherService.MESSAGE_ERROR:
        m_service.handleError(m);
        break;

      case AWeatherService.MESSAGE_INFO:
        m_service.handleInfo(m);
        break;

      case AWeatherService.MESSAGE_RECEIVED:
        m_service.handleReceived(m);
        break;
      }
    }
  };
  private AWeatherService m_service;
  private final BlockingQueue<Object> queue = new LinkedBlockingQueue<Object>(
      1024);
  protected DatagramSocket socket = null;

  void begin() {
    try {
      socket = new DatagramSocket();
    } catch (SocketException e1) {
      // TODO Auto-generated catch block
      e1.printStackTrace();
    }
    new Thread(new Runnable() {
      @Override
      public void run() {
        while (true) {
          try {
            Object o = queue.take();
            process(o);
            Thread.sleep(750);
          } catch (InterruptedException e) {
            e.printStackTrace();
          }
        }
      }
    }, "Consumer").start();
  }

  private void forwardMsg(final int type, final String s) {
    final Message m = Message.obtain(m_handler, type);
    m.obj = new Msg(s);
    m_handler.sendMessage(m);
  }

  private void process(Object o) {
    if (o instanceof DatagramPacket)
      processDGP(o);
    else if (o instanceof String)
      processStr(o);
    else if (o instanceof Info)
      processInfo(o);
  }

  private void processDGP(Object o) {
    DatagramPacket packet = (DatagramPacket) o;
    InetAddress address = packet.getAddress();
    int port = packet.getPort();
    byte[] data = packet.getData();
    try {
      packet = new DatagramPacket(data, data.length, address, port);
    } catch (Exception e) {
      e.printStackTrace();
      return;
    }
    try {
      socket.send(packet);
    } catch (IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }
  }

  private void processInfo(Object o) {
    forwardMsg(AWeatherService.MESSAGE_INFO, new Date().toString() + " : "
        + ((Info) o).toString());
  }

  private void processStr(Object o) {
    String s = ((String) (o)).trim();
    if (s.equals("update"))
      forwardMsg(AWeatherService.MESSAGE_RECEIVED, s);
    else forwardMsg(AWeatherService.MESSAGE_INFO, new Date().toString() + " : "
        + ((String) (o)).trim());

  }

  public void close() {
    socket.close();
    System.out.println("Responder closed.");
  }

  public void respond(Object o) throws InterruptedException {
    if (o != null)
      queue.put(o);
  }

  public Responder(final AWeatherService a) {
    m_service = a;
  }
}
package org.nws.aweather;

import android.app.Activity;
import android.app.ActivityManager;
import android.app.ActivityManager.RunningServiceInfo;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class ServiceLauncher extends Activity {
  private Button start;
  private final OnClickListener startListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      startService(new Intent(ServiceLauncher.this, AWeatherService.class));
      update();
    }
  };

  private Button stop;

  private final OnClickListener stopListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      stopService(new Intent(ServiceLauncher.this, AWeatherService.class));
      update();
    }
  };

  private boolean serviceIsRunning() {
    ActivityManager manager = (ActivityManager) getSystemService(Context.ACTIVITY_SERVICE);
    for (RunningServiceInfo service : manager
        .getRunningServices(Integer.MAX_VALUE)) {
      if (AWeatherService.class.getName().equals(
          service.service.getClassName())) {
        return true;
      }
    }
    return false;
  }

  private void update() {
    boolean f = serviceIsRunning();
    start.setEnabled(!f);
    stop.setEnabled(f);
  }

  /** Called when the activity is first created. */
  @Override
  public void onCreate(final Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.servicelauncher);
    start = (Button) findViewById(R.id.startButton);
    stop = (Button) findViewById(R.id.stopButton);
    start.setOnClickListener(startListener);
    stop.setOnClickListener(stopListener);
    update();
  }

}
package org.nws.aweather;

import java.net.InetAddress;
import java.net.UnknownHostException;

import org.nws.aweather.forecast.FetchForecast;

import android.app.Activity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class SetServer extends Activity {
  private Button cancelButton;
  private final OnClickListener cancelListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      finish();
    }
  };
  private Button defaultButton;
  private final OnClickListener defaultListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      String s = FetchForecast.getInstance().getDefaultUrl();
      url.setText(s);
    }
  };
  private Button setUrlButton;

  private final OnClickListener setUrlListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      if (isvalidurl()){
        String s = url.getText().toString();
        FetchForecast.getInstance().setUrl(s);
            setResult( RESULT_OK, getIntent( ) );
        finish();
      }
      else
        Toast.makeText(SetServer.this, "Invalid Url", Toast.LENGTH_LONG)
            .show();
    }
  };

  private EditText url;
  private boolean isvalidurl() {
    String s = url.getText().toString();
    try {
      @SuppressWarnings("unused")
      InetAddress serverAddress = InetAddress.getByName(s);
    } catch (UnknownHostException e) {
      return false;
    }
    return true;
  }

  private void update()
  {
    setUrlButton.setEnabled(isvalidurl());
  }

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.setserver);
      setResult( RESULT_CANCELED, getIntent( ) );
    url = (EditText) findViewById(R.id.enterurl);
    String s = FetchForecast.getInstance().getUrl();
    url.setText(s);
    setUrlButton = (Button) findViewById(R.id.setUrlButton);
    defaultButton = (Button) findViewById(R.id.defaultButton);
    cancelButton = (Button) findViewById(R.id.CancelButton);
    setUrlButton.setOnClickListener(setUrlListener);
    defaultButton.setOnClickListener(defaultListener);
    cancelButton.setOnClickListener(cancelListener);
    url.addTextChangedListener(new TextWatcher() {

      @Override
      public void afterTextChanged(Editable arg0) {

      }

      @Override
      public void beforeTextChanged(CharSequence arg0, int arg1,
          int arg2, int arg3) {

      }

      @Override
      public void onTextChanged(CharSequence arg0, int arg1, int arg2,
          int arg3) {
        update();

      }


      });
  }

}
package org.nws.aweather;

import java.io.BufferedInputStream;
import java.io.IOException;
import java.net.URL;
import java.net.URLConnection;
import java.util.List;

import org.nws.aweather.forecast.FetchForecast;
import org.nws.aweather.forecast.NWSWeatherObject;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Handler;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;

public class WeatherGallery {
  private final Handler handler;
  private final ArrayAdapter<String> locationAdapter;
  private final Spinner locationChooser;
  private final LocationManager locationManager;
  private final AWeather owner;
  private final Button refreshButton;

  private Bitmap getImage(final String httpLocation) throws IOException {
    Bitmap result = null;
    final URL url = new URL(httpLocation);
    final URLConnection con = url.openConnection();
    final BufferedInputStream in = new BufferedInputStream(
        con.getInputStream());
    result = BitmapFactory.decodeStream(in);
    in.close();
    return result;
  }

  private void updateWeather(final NWSWeatherObject nwo) {
    final ImageView i = (ImageView) owner.findViewById(R.id.image1);
    if (i == null)
      // view has bailed on us- just fail fast
      return;
    try {
      final List<String> conditions = nwo.getConditionIconLinks();
      final String first = conditions.get(0);
      final Bitmap bm = getImage(first);
      i.setImageBitmap(bm);
    } catch (final IOException e) {
      throw new RuntimeException(e);
    }
    final TextView t = (TextView) owner.findViewById(R.id.day);
    t.setText("Tomorrow");
  }

  private void updateWeather(final String s) {
    final ImageView i = (ImageView) owner.findViewById(R.id.image1);
    if (i == null)
      // view has bailed on us- just fail fast
      return;
    try {
      final Bitmap bm = getImage(s);
      i.setImageBitmap(bm);
    } catch (final IOException e) {
      throw new RuntimeException(e);
    }
    final TextView t = (TextView) owner.findViewById(R.id.day);
    t.setText("Tomorrow");
  }

  public void gotWeather(final NWSWeatherObject nwo) {
    final Thread X = new Thread() {
      @Override
      public void run() {
        updateWeather(nwo);
      }
    };
    handler.post(X);
  }

  public void gotWeather(final String s) {
    final Thread X = new Thread() {
      @Override
      public void run() {
        updateWeather(s);
      }
    };
    handler.post(X);
  }

  public void logError(Object p) {
    owner.forwardMsg(AWeatherService.MESSAGE_ERROR, p);
  }

  public void refresh() {
    final String[] location = locationManager.getLocation(0);
    FetchForecast.getInstance().requestForecast(location[1], location[2]);
  }

  public WeatherGallery(final AWeather owner) {
    this.owner = owner;
    FetchForecast.parent = this;
    locationManager = owner.getLocationManager();

    locationChooser = (Spinner) owner.findViewById(R.id.locationChooser);
    refreshButton = (Button) owner.findViewById(R.id.refreshButton);
    refreshButton.setOnClickListener(owner);
    handler = new Handler();
    locationAdapter = new ArrayAdapter<String>(
        owner.getApplicationContext(),
        android.R.layout.simple_spinner_item);
    final String[] location = locationManager.getLocation(locationManager
        .getSelected());
    locationAdapter.add(location[0]);
    String s = locationManager.getServer();
    FetchForecast.getInstance().setUrl(s);
    FetchForecast.getInstance().requestForecast(location[1], location[2]);
    locationChooser.setAdapter(locationAdapter);
  }
}
```

```java
package com.nws.nwsserver;

import java.io.Serializable;

public class Info implements Serializable {
  private static final long serialVersionUID = 1L;
  public String data;
  public int id;
  public String lat;
  public String lon;

  @Override
  public String toString() {
    return String.format("id: %d, lat=%s, long=%s, data=%s", id, lat, lon,
        data);
  }
}
package org.nws.aweather;

import java.util.TreeMap;

import android.content.Context;
import android.location.Address;
import android.location.Geocoder;
import android.os.Build;
import android.os.Handler;
import android.text.Editable;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnKeyListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

public class AddLocation implements OnClickListener, OnKeyListener,
    OnItemSelectedListener {
  private final class GeocodeResultsHandler implements Runnable {
    @Override
    public void run() {
      geocodeFinished();
    }
  }

  public static final int GEOCODE_STATUS_LOOKUP_FAIL = 2;
  public static final int GEOCODE_STATUS_NETWORK_ERROR = 1;
  public static final int GEOCODE_STATUS_SUCCESS = 3;
  private final Context context;
  private final EditText enterLocation;
  private final boolean geocoding;
  private final TextView log;
  private final Button lookupLocation;
  private final AWeather owner;
  private final Button showweather;
  protected Address geocoded;
  protected int geocodeStatus;
  protected final GeocodeResultsHandler grh = new GeocodeResultsHandler();
  protected Handler handler = new Handler();

  private synchronized void geocode(final String place) {
    if (geocoding)
      return;
    lookupLocation.setEnabled(false);
    showweather.setEnabled(false);
    log(owner.getString(R.string.searchingForLocation));
    final GeocodePlace gp = new GeocodePlace(place, this, context);
    new Thread(gp).start();
  }

  public void geocodeFinished() {
    lookupLocation.setEnabled(true);
    showweather.setEnabled(true);
    if (geocodeStatus == GEOCODE_STATUS_SUCCESS) {
      final StringBuilder sb = new StringBuilder(
          owner.getString(R.string.foundLocation));
      sb.append(geocoded.getLocality()).append(" (")
          .append(geocoded.getAdminArea()).append(")");
      log(sb.toString());
      owner.getLocationManager().storeLocation(geocoded);
    } else if (geocodeStatus == GEOCODE_STATUS_NETWORK_ERROR)
      log(owner.getString(R.string.geocodeNetworkError));
    else
      // geocode status will be lookup fail
      log(owner.getString(R.string.geocodeLookupFailure));
    geocoded = null;
    geocodeStatus = 0;
  }

  public void log(final String text) {
    log.setText(text);
  }

  @Override
  public void onClick(final View v) {
    if (v == lookupLocation) {
      final Editable editable = enterLocation.getText();
      final String s = editable.toString();
      if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.GINGERBREAD
          && Geocoder.isPresent())
        geocode(s);
      else
        log("Geocoder not present");
    } else if (v == showweather) {
      owner.restart();
    }

  }

  @Override
  public void onItemSelected(final AdapterView<?> adapter, final View arg1,
      final int position, final long id) {
    showweather.setEnabled(true);
    owner.getLocationManager().storeSelected(position);
  }

  @Override
  public boolean onKey(final View v, final int keyCode, final KeyEvent event) {
    if (v == enterLocation) {
      final Editable editable = enterLocation.getText();
      final String s = editable.toString();
      if (s.length() == 0)
        lookupLocation.setEnabled(false);
      else
        lookupLocation.setEnabled(true);
      return false;
    }
    if (event.getKeyCode() == KeyEvent.KEYCODE_ENTER) {
      onClick(lookupLocation);
      return true;
    }
    return false;
  }

  @Override
  public void onNothingSelected(final AdapterView<?> arg0) {
    // don't do anything...
  }

  public AddLocation(final AWeather owner) {
    this.owner = owner;
    context = owner.getApplicationContext();
    log = (TextView) owner.findViewById(R.id.log);
    enterLocation = (EditText) owner.findViewById(R.id.enterLocation);
    enterLocation.setOnKeyListener(this);
    lookupLocation = (Button) owner.findViewById(R.id.fireLocationLookup);
    lookupLocation.setOnClickListener(this);
    lookupLocation.setEnabled(false);
    showweather = (Button) owner.findViewById(R.id.buttonShowWeather);
    showweather.setOnClickListener(this);
    showweather.setEnabled(false);
    geocoding = false;
    final Spinner spinner = (Spinner) owner
        .findViewById(R.id.locationSpinner);
    final ArrayAdapter<String> adapter = new ArrayAdapter<String>(owner,
        android.R.layout.simple_spinner_item);
    adapter.setDropDownViewResource(android.R.layout.simple_dropdown_item_1line);
    spinner.setAdapter(adapter);
    final LocationManager locationManager = new LocationManager(owner);
    final TreeMap<String, String[]> locations = locationManager
        .getLocations();
    for (final String id : locations.keySet()) {
      final String[] vals = locations.get(id);
      adapter.add(vals[0]);
    }
    int selected = locationManager.getSelected();
    spinner.setOnItemSelectedListener(this);
    spinner.setSelection(selected);
    log(owner.getString(R.string.addLocationExample));
  }
}
package org.nws.aweather;

import org.nws.aweather.forecast.FetchForecast;

import android.app.Activity;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.Handler;
import android.os.IBinder;
import android.os.Message;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;

public class AWeather extends Activity implements OnClickListener {
  public static final int MENU_ITEM_MANAGE_LOCATIONS = 0;
  public static final int MENU_ITEM_SET_SERVER = 2;
  public static final int MENU_ITEM_START_SERVICE = 1;
  private AddLocation addLocation;
  private LocationManager locationManager;
  private ServiceConnection m_connection = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName className, IBinder service) {
      m_service = ((AWeatherService.LocalBinder) service).getService();
      m_service.setWeatherGallery(weatherGallery);
    }

    @Override
    public void onServiceDisconnected(ComponentName className) {
      m_service = null;
    }
  };
  private final Handler m_handler = new Handler() {
    @Override
    public void handleMessage(final Message msg) {
      final Msg m = (Msg) msg.obj;
      ;
      switch (msg.what) {
      case AWeatherService.MESSAGE_DISCONNECTED:
        m_service.handleDisconnected(m);
        break;

      case AWeatherService.MESSAGE_CONNECTED:
        m_service.handleConnected(m);
        break;

      case AWeatherService.MESSAGE_ERROR:
        m_service.handleError(m);
        break;

      case AWeatherService.MESSAGE_INFO:
        m_service.handleInfo(m);
        break;

      case AWeatherService.MESSAGE_RECEIVED:
        m_service.handleReceived(m);
        break;
      }
    }
  };
  private static final int SET_SERVER = 10;
    private boolean m_isbound = false;

  private AWeatherService m_service;

  private WeatherGallery weatherGallery = null;

  void doBindService() {
    bindService(new Intent(AWeather.this, AWeatherService.class),
        m_connection, Context.BIND_AUTO_CREATE);
    m_isbound = true;
  }

  void doUnbindService() {
    if (m_isbound) {
      unbindService(m_connection);
      m_isbound = false;
    }
  }

  protected LocationManager getLocationManager() {
    return locationManager;
  }

  @Override
  protected void onDestroy() {
    super.onDestroy();
    m_service.setWeatherGallery(null);
    doUnbindService();
  }

  public void forwardMsg(final int type, final Object s) {
    final Message m = Message.obtain(m_handler, type);
    m.obj = new Msg(s);
    m_handler.sendMessage(m);
  }

  @Override
  public void onClick(View view) {
    weatherGallery.refresh();
    //m_service.m_networklistener.push();
  }

  @Override
  public void onCreate(final Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    locationManager = new LocationManager(this);
    if (locationManager.getLocationCount() == 0) {
      setContentView(R.layout.addlocation);
      addLocation = new AddLocation(this);
    } else {
      setContentView(R.layout.main);
      weatherGallery = new WeatherGallery(this);
    }
    doBindService();
    startService(new Intent(this, AWeatherService.class));
  }

  @Override
  public boolean onCreateOptionsMenu(final Menu menu) {
    super.onCreateOptionsMenu(menu);
    menu.add(0, MENU_ITEM_MANAGE_LOCATIONS, 0, R.string.manageLocations);
    menu.add(0, MENU_ITEM_START_SERVICE, 0, R.string.manageService);
    menu.add(0, MENU_ITEM_SET_SERVER, 0, R.string.setserver);
    return true;
  }

  @Override
  public boolean onOptionsItemSelected(final MenuItem item) {
    switch (item.getItemId()) {
    case MENU_ITEM_MANAGE_LOCATIONS:
      setContentView(R.layout.addlocation);
      if (addLocation == null)
        addLocation = new AddLocation(this);
      return true;
    case MENU_ITEM_START_SERVICE:
      startActivity(new Intent(this, ServiceLauncher.class));
      return true;
    case MENU_ITEM_SET_SERVER:
      startActivityForResult(new Intent(this, SetServer.class), SET_SERVER);
      return true;
    }
    return false;
  }

  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent data) {
      if ( requestCode == SET_SERVER && resultCode == Activity.RESULT_OK ){
        String s = FetchForecast.getInstance().getUrl();
        locationManager.storeServer(s);
        restart();
      }
    super.onActivityResult(requestCode, resultCode, data);
  }

  public void restart() {
    stopService(new Intent(this, AWeatherService.class));
    Intent i = getBaseContext().getPackageManager()
        .getLaunchIntentForPackage(getBaseContext().getPackageName());
    i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(i);
  }
}package org.nws.aweather;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.SocketException;
import java.util.Calendar;
import java.util.Timer;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Intent;
import android.os.Binder;
import android.os.IBinder;
import android.widget.Toast;

public class AWeatherService extends Service implements NetworkTaskEvents {

  public class LocalBinder extends Binder {
    AWeatherService getService() {
      return AWeatherService.this;
    }
  }

  private static boolean showToasts = false;
  public static final int MESSAGE_CONNECTED = 0;
  public static final int MESSAGE_DISCONNECTED = 1;
  public static final int MESSAGE_ERROR = 2;
  public static final int MESSAGE_INFO = 3;

  public static final int MESSAGE_RECEIVED = 4;

  private Long m_counter = 0L;

  public NetworkListener m_networklistener;

  private Responder m_responder;

  private WeatherGallery m_weathergallery = null;

  private final IBinder mBinder = new LocalBinder();

  private NotificationManager nm;

  private final Calendar time = Calendar.getInstance();

  private final Timer timer = new Timer();

  private void connect() {
    try {
      m_responder = new Responder(this);
      m_responder.begin();
      m_networklistener = new NetworkListener(this, 6001);
      m_networklistener.start();
    } catch (SocketException e) {
    }
  }

  private void destroyTimer() {
    if (timer != null)
      timer.cancel();
  }

  private void disconnect() {
    if (m_networklistener != null)
      m_networklistener.close();
    if (m_responder != null)
      m_responder.close();
  }

  /** Show a notification while this service is running. */
  @SuppressWarnings("deprecation")
  private void showNotification() {
    // Use the same text for the ticker and the expanded notification
    final CharSequence text = getText(R.string.service_started);
    // Set the icon, scrolling text and timestamp
    final Notification notification = new Notification(R.drawable.android,
        text, System.currentTimeMillis());
    // The PendingIntent to launch our activity if the user selects this
    // notification
    final PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
        new Intent(this, ServiceLauncher.class), 0);
    // Set the info for the views that show in the notification panel.
    notification.setLatestEventInfo(this, getText(R.string.service_label),
        text, contentIntent);
    // Send the notification.
    // We use a layout id because it is a unique number. We use it later to
    // cancel.
    nm.notify(R.string.service_started, notification);
  }

  @Override
  public void handleConnected(Msg o) {
    Object ob = o.getInfo();
    String s = "Connected";
    if (ob instanceof String)
      s = (String) (ob);
    if (showToasts)
      Toast.makeText(this, s, Toast.LENGTH_LONG).show();
  }

  @Override
  public void handleDisconnected(Msg o) {
    Object ob = o.getInfo();
    String s = "Disconnected";
    if (ob instanceof String)
      s = (String) (ob);
    if (showToasts)
      Toast.makeText(this, s, Toast.LENGTH_LONG).show();
  }

  @Override
  public void handleError(Msg o) {
    Object ob = o.getInfo();
    if (ob instanceof String) {
      String s = (String) (ob);
      if (showToasts)
        Toast.makeText(this, s, Toast.LENGTH_LONG).show();
    } else if (ob instanceof DatagramPacket) {
      try {
        m_responder.respond(ob);
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    } else if (ob instanceof byte[]) {
      byte bytes[] = (byte[]) ob;
      try {
        m_responder.respond(ConvertUtils.deserializePacket(bytes));
      } catch (InterruptedException e) {
        e.printStackTrace();
      } catch (IOException e) {
        e.printStackTrace();
      }
    }
  }

  @Override
  public void handleInfo(final Msg o) {
    Object ob = o.getInfo();
    String s = "Info";
    if (ob instanceof String)
      s = (String) (ob);
    if (showToasts)
      Toast.makeText(this, s, Toast.LENGTH_LONG).show();
  }

  @Override
  public void handleReceived(Msg o) {
    Object ob = o.getInfo();
    String s = "Received";
    if (ob instanceof String)
      s = (String) (ob);
    if (s.equals("update"))
      if (m_weathergallery != null)
        m_weathergallery.refresh();
    if (showToasts)
      Toast.makeText(this, s, Toast.LENGTH_LONG).show();
  }

  @Override
  public IBinder onBind(final Intent intent) {
    return mBinder;
  }

  @Override
  public void onCreate() {
    super.onCreate();
    nm = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
    if (showToasts)
      Toast.makeText(this, "Service created at " + time.getTime(),
          Toast.LENGTH_LONG).show();
    showNotification();
    // createTimer();
    connect();
  }

  @Override
  public void onDestroy() {
    super.onDestroy();
    // Cancel the persistent notification.
    destroyTimer();
    disconnect();
    nm.cancel(R.string.service_started);
    if (showToasts)
      Toast.makeText(
          this,
          "Service destroyed at " + time.getTime()
              + "; m_counter is at: " + m_counter,
          Toast.LENGTH_LONG).show();
    m_counter = null;
  }

  @Override
  public int onStartCommand(Intent intent, int flags, int startId) {
    return START_STICKY;
  }

  public void setWeatherGallery(WeatherGallery a) {
    m_weathergallery = a;
  }

}
package org.nws.aweather;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.ObjectInput;
import java.io.ObjectInputStream;
import java.io.ObjectOutput;
import java.io.ObjectOutputStream;
import java.net.DatagramPacket;
import java.net.InetAddress;

public class ConvertUtils {

  static public DatagramPacket deserializePacket(byte[] packetBytes)
      throws IOException {
    DatagramPacket udp = new DatagramPacket(new byte[1024], 1024);

    ByteArrayInputStream bis = new ByteArrayInputStream(packetBytes);
    ObjectInput in = null;
    try {
      in = new ObjectInputStream(bis);
      udp.setAddress((InetAddress) in.readObject());
      udp.setPort((Integer) in.readObject());
      udp.setData((byte[]) in.readObject());
    } catch (IOException e) {
      e.printStackTrace();
      return null;
    } catch (ClassNotFoundException e) {
      e.printStackTrace();
      return null;
    } finally {
      in.close();
      bis.close();
    }

    return udp;
  }

  public int numPackets = 0;

  public void serializePacket(byte[] packetBytes) {
    // record the number of packets and the time between packets received
    // avgInterval
    numPackets += 1;
  }

  public void serializePacket(DatagramPacket udp) throws IOException {
    // byte[] packetBytes = new byte[4096];
    //
    // byte[] tmp = udp.getAddress().getAddress();
    // int offset = 0;
    // System.arraycopy( tmp, 0, packetBytes, offset, tmp.length);
    // offset += tmp.length;
    //
    // ByteBuffer bb = ByteBuffer.allocate(4);
    // bb.putInt(udp.getPort());
    // System.arraycopy( bb.array(), 0, packetBytes, offset, 4);
    // offset += 4;
    //
    // bb.putInt(udp.getLength());
    // System.arraycopy( bb.array(), 0, packetBytes, offset, 4);
    // offset += 4;
    //
    // System.arraycopy( udp.getData(), 0, packetBytes, offset,
    // udp.getLength());
    // serializePacket(packetBytes);

    ByteArrayOutputStream bos = new ByteArrayOutputStream();
    ObjectOutput out = null;
    try {
      out = new ObjectOutputStream(bos);
      out.writeObject(udp.getAddress());
      out.write(udp.getPort());
      out.writeObject(udp.getData());
      serializePacket(bos.toByteArray());
    } catch (IOException e) {
      e.printStackTrace();
    } finally {
      out.close();
      bos.close();
    }
  }
}package org.nws.aweather.forecast;

import java.io.BufferedReader;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.ObjectInput;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.Socket;
import java.net.SocketException;
import java.net.UnknownHostException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Random;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.TimeUnit;

import org.nws.aweather.WeatherGallery;
import org.xml.sax.helpers.DefaultHandler;

import com.nws.nwsserver.Info;

public class FetchForecast extends DefaultHandler implements Runnable {

  private static final String BEGIN = "&begin=";
  private static final DateFormat DATE_FORMAT = new SimpleDateFormat(
      "yyyy-MM-dd'T'hh':00:00'");
  private static final String END = "&end=";
  private static final String HOST = "graphical.weather.gov";
  private static final String LAT = "&lat=";
  private static final String LON = "&lon=";
  private static final int MAX_RETRIES = 3;
  private static final int PORT = 5559;
  private static FetchForecast theInstance = null;

  private static final long TIMEOUT_SECS = 10;

  private static final int UDP_WC_PORT = 10000;
  private static final String UDP_LOCAL = "www.weatherpushupdates.com";
  private static final int UDP_WS_PORT = 5559;
  //public static String udpWsHost = "192.168.19.207";
  public static String udpWsHost = UDP_LOCAL;
  private static final String URL_PREFIX = "/xml/sample_products/browser_interface/ndfdXMLclient.php?"
      + "&product=time-series&icons=icons";
  public static WeatherGallery parent = null;
  public static FetchForecast getInstance() {
    if (theInstance == null) {
      theInstance = new FetchForecast();
      new Thread(theInstance).start();
    }
    return theInstance;
  }

  private final boolean activeFlag = true;
  private Info currentRequestMsg;
  private final BlockingQueue<Info> recvQueue = new LinkedBlockingQueue<Info>(
      1024);
  private int requestId = (new Random()).nextInt();
  private InetAddress serverAddress;

  private DatagramSocket socket;
  private String urlString;

  public boolean useUDPProtocol = true;

  private void getDirectNWS(String lat, String lon) {

    final StringBuilder url = new StringBuilder(URL_PREFIX);
    url.append(LAT).append(lat).append(LON).append(lon);
    final Calendar start = Calendar.getInstance();
    final Calendar end = Calendar.getInstance();
    end.add(Calendar.DAY_OF_MONTH, 7);
    url.append(BEGIN).append(DATE_FORMAT.format(start.getTime()));
    url.append(END).append(DATE_FORMAT.format(end.getTime()));
    urlString = url.toString();

    try {
      System.out.println("Ripping: " + urlString);
      final Socket s = new Socket(HOST, PORT);
      final OutputStream os = s.getOutputStream();
      final PrintWriter out = new PrintWriter(new OutputStreamWriter(os));
      final StringBuilder request = new StringBuilder("GET ").append(
          urlString).append("HTTP/1.1");
      request.append("\n");
      request.append("Host: ").append(HOST).append("\n");
      request.append("\n");
      out.print(request);
      out.flush();
      final InputStream in = s.getInputStream();
      // String t = getIn(in);
      //
      // int n = t.length( );
      final NWSWeatherXMLParser parser = new NWSWeatherXMLParser(in);
      parser.parse(parent);
      System.out.println("Finished parsing");
    } catch (final Exception e) {
      throw new RuntimeException(e);
    }
  }

  @SuppressWarnings("unused")
  private String getIn(final InputStream in) {
    final InputStreamReader is = new InputStreamReader(in);
    final StringBuilder sb = new StringBuilder();
    final BufferedReader br = new BufferedReader(is);
    String read = null;
    try {
      read = br.readLine();
    } catch (final IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }
    while (read != null) {
      // System.out.println(read);
      sb.append(read);
      try {
        read = br.readLine();
      } catch (final IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
      }
    }
    return sb.toString();
  }

  private void getUDPWeatherServce(String lat, String lon) {
    currentRequestMsg = new Info();
    currentRequestMsg.data = "REQW_GETFORCAST";
    currentRequestMsg.lat = lat;
    currentRequestMsg.lon = lon;

    int nTries = MAX_RETRIES;
    currentRequestMsg.id = requestId++;
    while (nTries > 0) {
      nTries--;
      // currentRequestMsg.id = requestId++;

      byte[] sendBuf = null;
      try {
        ByteArrayOutputStream bos = new ByteArrayOutputStream();
        ObjectOutputStream out = new ObjectOutputStream(bos);
        out.writeObject(currentRequestMsg);
        out.close();
        sendBuf = bos.toByteArray();
        bos.close();
      } catch (IOException e) {
        parent.logError(e.getMessage());
        return;
      }

      DatagramPacket sendPkt = new DatagramPacket(sendBuf,
          sendBuf.length, serverAddress, UDP_WS_PORT);

      try {
        if ((socket != null) && (serverAddress != null)) {
          socket.send(sendPkt);
        }
      } catch (IOException e1) {
        parent.logError(e1.getMessage());
        return;
      }

      Info recvMsg;
      do {
        try {
          recvMsg = recvQueue.poll(TIMEOUT_SECS, TimeUnit.SECONDS);
          if (recvMsg != null)
            if (isValid(recvMsg, currentRequestMsg)) {
              parent.gotWeather(recvMsg.data);
              return;
            }

        } catch (InterruptedException e) {
          // TODO Auto-generated catch block
          e.printStackTrace();
        } catch (Exception e) {
          // TODO Auto-generated catch block
          e.printStackTrace();
        }
      } while (!recvQueue.isEmpty());
    }
  }

  private boolean isValid(Info info_in, Info info_out) {
    return info_in.id == info_out.id;
  }

  public String getDefaultUrl() {
    return UDP_LOCAL;
  }

  public String getUrl() {
    return udpWsHost;
  }

  public void initializeUDP() {
    try {
      serverAddress = InetAddress.getByName(udpWsHost);
    } catch (UnknownHostException e) {
      parent.logError(e.getMessage());
      return;
    }
    try {
      socket = new DatagramSocket(UDP_WC_PORT);
    } catch (SocketException e) {
      parent.logError(e.getMessage());
    }
  }

  public void requestForecast(String lat, String lon) {
    if (!useUDPProtocol)
      getDirectNWS(lat, lon);
    else
      getUDPWeatherServce(lat, lon);
  }

  @Override
  public void run() {
    initializeUDP();
    while (activeFlag) {
      byte recvBuf[] = new byte[4096];
      DatagramPacket recvPkt = new DatagramPacket(recvBuf, recvBuf.length);

      try {
        socket.receive(recvPkt);
      } catch (IOException e) {
        parent.logError(e.getMessage());
        continue;
      }
      Info recvInfo = null;
      try {
        ByteArrayInputStream bis = new ByteArrayInputStream(
            recvPkt.getData());
        ObjectInput in = new ObjectInputStream(bis);
        recvInfo = (Info) in.readObject();
      } catch (Exception e) {
        parent.logError(recvPkt.getData());
        continue;
      }

      if (isValid(recvInfo, currentRequestMsg)) {
        recvQueue.offer(recvInfo);
      } else {
        parent.logError("Invalid id.");
      }
    }
  }

  public void setUrl(String url) {
    udpWsHost = url;
  }
}
package org.nws.aweather.forecast;

import java.util.ArrayList;
import java.util.Date;
import java.util.LinkedHashMap;
import java.util.List;

public class NWSWeatherObject {
  private final List<String> conditionIconLinks;
  private String conditionIconTimeLayout;
  private Date createdOn;
  private String moreInfoLink;
  private String mostRecentTimeLayout;
  private final LinkedHashMap<String, List<TimeLayoutBand>> timeLayouts;

  public void addConditionIconLink(final String link) {
    conditionIconLinks.add(link);
  }

  public void addNewTimeBand(final Date date) {
    final List<TimeLayoutBand> l = timeLayouts.get(mostRecentTimeLayout);
    final TimeLayoutBand tlb = new TimeLayoutBand(date);
    l.add(tlb);
  }

  public void addTimeLayout(final String name) {
    mostRecentTimeLayout = name;
    timeLayouts.put(name, new ArrayList<TimeLayoutBand>());
  }

  public void closeCurrentTimeBand(final Date end) {
    final List<TimeLayoutBand> l = timeLayouts.get(mostRecentTimeLayout);
    final TimeLayoutBand last = l.get(l.size() - 1);
    last.setEndDate(end);
  }

  public List<String> getConditionIconLinks() {
    return conditionIconLinks;
  }

  public String getConditionIconTimeLayout() {
    return conditionIconTimeLayout;
  }

  public Date getCreatedOn() {
    return createdOn;
  }

  public String getMoreInfoLink() {
    return moreInfoLink;
  }

  public void setConditionIconTimeLayout(final String name) {
    conditionIconTimeLayout = name;
  }

  public void setCreatedOn(final Date date) {
    createdOn = date;
  }

  public void setMoreInfoLink(final String moreInfoLink) {
    this.moreInfoLink = moreInfoLink;
  }

  public NWSWeatherObject() {
    timeLayouts = new LinkedHashMap<String, List<TimeLayoutBand>>();
    conditionIconLinks = new ArrayList<String>();
  }
}
package org.nws.aweather.forecast;

import java.io.InputStream;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import org.nws.aweather.WeatherGallery;
import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

/**
 * Responsible for parsing through the XML recovered from an NWS hit. There are
 * two core sections to look for:
 * <ul>
 * <li>time-layout: which contains sections/spans of time</li>
 * <li>parameters: which contains specific blocks of weather data</li>
 * </ul>
 *
 * @author keerat
 */
public class NWSWeatherXMLParser extends DefaultHandler {
  // condition icons
  private static final String CONDITIONS = "conditions-icon";
  private static final SimpleDateFormat CREATED_DATE_FORMATTER = new SimpleDateFormat(
      "yyyy-MM-dd'T'HH:mm:ss'Z'");
  // <creation-date
  // refresh-frequency="PT1H">2009-03-14T20:58:51Z</creation-date>
  private static final String CREATION_DATE = "creation-date";
  private static final String END_VALID_TIME = "end-valid-time";
  private static final String ICON_LINK = "icon-link";
  private static final String LAYOUT_KEY = "layout-key";
  // <moreWeatherInformation applicable-location="point1">
  // http://forecast.weather.gov/MapClick.php?textField1=38.99&textField2=-77.01
  // </moreWeatherInformation>
  private static final String MORE_INFO = "moreWeatherInformation";
  private static final String START_VALID_TIME = "start-valid-time";
  private static final String TIME_LAYOUT = "time-layout";
  // <start-valid-time>2009-03-14T08:00:00-04:00</start-valid-time>
  private static final SimpleDateFormat TIME_LAYOUT_DATE_FORMATTER = new SimpleDateFormat(
      "yyyy-MM-dd'T'HH:mm:ssz");
  private String currentTag;
  private final InputStream in;
  private final NWSWeatherObject nwo;
  private WeatherGallery parent;

  private void handleCreationDate(final String dateString) {
    try {
      final Date date = CREATED_DATE_FORMATTER.parse(dateString);
      nwo.setCreatedOn(date);
    } catch (final ParseException e) {
    }
  }

  private void handleEndValidTime(final String endTimeString) {
    try {
      final Date date = TIME_LAYOUT_DATE_FORMATTER.parse(endTimeString);
      nwo.closeCurrentTimeBand(date);
    } catch (final ParseException e) {
    }
  }

  private void handleStartValidTime(final String startTimeString) {
    try {
      final Date date = TIME_LAYOUT_DATE_FORMATTER.parse(startTimeString);
      nwo.addNewTimeBand(date);
    } catch (final ParseException e) {
    }
  }

  @Override
  public void characters(final char[] ch, final int start, final int length)
      throws SAXException {
    if (currentTag.equals(CREATION_DATE))
      handleCreationDate(new String(ch, start, length));
    if (currentTag.equals(MORE_INFO))
      nwo.setMoreInfoLink(new String(ch, start, length));
    if (currentTag.equals(LAYOUT_KEY))
      nwo.addTimeLayout(new String(ch, start, length));
    if (currentTag.equals(START_VALID_TIME))
      handleStartValidTime(new String(ch, start, length));
    if (currentTag.equals(END_VALID_TIME))
      handleEndValidTime(new String(ch, start, length));
    if (currentTag.equals(ICON_LINK))
      nwo.addConditionIconLink(new String(ch, start, length));
  }

  @Override
  public void endDocument() throws SAXException {
    parent.gotWeather(nwo);
  }

  @Override
  public void endElement(final String uri, final String localName,
      final String name) throws SAXException {
    currentTag = "";
  }

  public void parse(final WeatherGallery parent) throws Exception {
    final SAXParserFactory saxParserFactory = SAXParserFactory
        .newInstance();
    final SAXParser parser = saxParserFactory.newSAXParser();
    this.parent = parent;
    parser.parse(in, this);
  }

  @Override
  public void startElement(final String uri, final String localName,
      final String name, final Attributes attributes) throws SAXException {
    currentTag = localName;
    if (currentTag.equals(CONDITIONS)) {
      // rip the time layout
      final String timeLayoutName = attributes.getValue(TIME_LAYOUT);
      nwo.setConditionIconTimeLayout(timeLayoutName);
    }
  }

  public NWSWeatherXMLParser(final InputStream in) {
    this.in = in;
    currentTag = "";
    nwo = new NWSWeatherObject();
  }
}
package org.nws.aweather.forecast;

import java.util.Date;

public class TimeLayoutBand {
  private Date end;
  private final Date start;

  public Date getEnd() {
    return end;
  }

  public Date getStart() {
    return start;
  }

  public boolean isPointInTime() {
    if (start.equals(end))
      return true;
    else
      return false;
  }

  public void setEndDate(final Date end) {
    this.end = end;
  }

  public TimeLayoutBand(final Date start) {
    this.start = start;
    end = start;
  }
}
package org.nws.aweather;

import java.io.IOException;
import java.util.List;
import java.util.Locale;

import android.content.Context;
import android.location.Address;
import android.location.Geocoder;

public class GeocodePlace implements Runnable {
  private final Context context;
  private final AddLocation owner;
  private final String place;

  @Override
  public void run() {
    final Geocoder geocoder = new Geocoder(context, Locale.getDefault());
    List<Address> addresses = null;
    boolean found = false;
    try {
      addresses = geocoder.getFromLocationName(place, 1);
      if (addresses != null && addresses.size() > 0) {
        found = true;
        owner.geocodeStatus = AddLocation.GEOCODE_STATUS_SUCCESS;
      } else
        owner.geocodeStatus = AddLocation.GEOCODE_STATUS_LOOKUP_FAIL;
    } catch (final IOException e) {
      owner.geocodeStatus = AddLocation.GEOCODE_STATUS_NETWORK_ERROR;
      found = false;
    }
    if (found) {
      final Address a = addresses.get(0);
      owner.geocoded = a;
    }
    owner.handler.post(owner.grh);
  }

  GeocodePlace(final String place, final AddLocation owner,
      final Context context) {
    this.place = place;
    this.owner = owner;
    this.context = context;
  }
}
package org.nws.aweather;

import java.util.Map;
import java.util.Set;
import java.util.TreeMap;

import org.nws.aweather.forecast.FetchForecast;

import android.content.Context;
import android.content.SharedPreferences;
import android.location.Address;

public class LocationManager {
  private static final String LOCAL_MAP_LOCATION = "llm.";
  private static final String LOCATION_LATITUDE = "location.lat.";
  private static final String LOCATION_LONGITUDE = "location.lon.";
  private static final String LOCATION_NAME = "location.name.";
  private static final String SELECTED = "selected";
  private static final String SERVER = "server";
  private int locationCount;
  private final TreeMap<String, String[]> locations;
  private int selected = 0;
  private String server = FetchForecast.udpWsHost;
  private final SharedPreferences sharedPreferences;

  public String[] getLocation(final int location) {
    final String[] loc = locations.get(LOCAL_MAP_LOCATION + location);
    return loc;
  }

  public int getLocationCount() {
    return locationCount;
  }

  public TreeMap<String, String[]> getLocations() {
    return locations;
  }

  public int getSelected() {
    return selected;
  }

  public String getServer() {
    return server;
  }

  public synchronized void storeLocation(final Address address) {
    if (address == null)
      System.out.println("Could not geocode location!");
    final SharedPreferences.Editor editor = sharedPreferences.edit();
    final String adminArea = address.getAdminArea();
    final String locality = address.getLocality();
    final StringBuilder sb = new StringBuilder(locality).append(" (")
        .append(adminArea).append(")");
    final String latitude = Double.toString(address.getLatitude());
    final String longitude = Double.toString(address.getLongitude());
    final String nameKey = LOCATION_NAME + locationCount;
    final String latKey = LOCATION_LATITUDE + locationCount;
    final String lonKey = LOCATION_LONGITUDE + locationCount;
    editor.putString(nameKey, sb.toString());
    editor.putString(latKey, latitude);
    editor.putString(lonKey, longitude);
    editor.commit();
    locations.put(LOCAL_MAP_LOCATION + locationCount,
        new String[] { sb.toString(), latitude, longitude });
    locationCount++;
  }

  public synchronized void storeSelected(final int i) {
    final SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putInt(SELECTED, i);
    editor.commit();
    selected = i;
  }

  public synchronized void storeServer(final String s) {
    final SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString(SERVER, s);
    editor.commit();
    server = s;
  }

  public LocationManager(final AWeather owner) {
    sharedPreferences = owner.getPreferences(Context.MODE_WORLD_READABLE);
    final Map<String, ?> allParams = sharedPreferences.getAll();
    final Set<String> keys = allParams.keySet();
    locations = new TreeMap<String, String[]>();
    locationCount = 0;
    for (final String key : keys)
      if (key.startsWith(LOCATION_NAME)) {
        final String name = sharedPreferences.getString(LOCATION_NAME
            + locationCount, "Corrupted");
        final String latitude = sharedPreferences.getString(
            LOCATION_LATITUDE + locationCount, "0");
        final String longitude = sharedPreferences.getString(
            LOCATION_LONGITUDE + locationCount, "0");
        locations.put(LOCAL_MAP_LOCATION + locationCount, new String[] {
            name, latitude, longitude });
        locationCount++;
      } else if (key.startsWith(SELECTED)) {
        selected = sharedPreferences.getInt(SELECTED, 0);
      } else if (key.startsWith(SERVER)) {
        server = sharedPreferences.getString(SERVER,  FetchForecast.udpWsHost);
      }

  }
}
package org.nws.aweather;

public class Msg {
  private final Object info;

  public String getDescription() {
    if (info instanceof String)
      return String.format("InfoMsg: info = %s", info);
    return info.toString();
  }

  public Object getInfo() {
    return info;
  }

  public Msg(final Object info) {
    this.info = info;
  }
}package org.nws.aweather;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Date;

import org.nws.aweather.forecast.FetchForecast;

import android.os.Handler;
import android.os.Message;

public class NetworkListener extends Thread {
  private boolean domore = true;

  private final Handler m_handler = new Handler() {
    @Override
    public void handleMessage(final Message msg) {
      final Msg m = (Msg) msg.obj;
      ;
      switch (msg.what) {
      case AWeatherService.MESSAGE_DISCONNECTED:
        m_service.handleDisconnected(m);
        break;

      case AWeatherService.MESSAGE_CONNECTED:
        m_service.handleConnected(m);
        break;

      case AWeatherService.MESSAGE_ERROR:
        m_service.handleError(m);
        break;

      case AWeatherService.MESSAGE_INFO:
        m_service.handleInfo(m);
        break;

      case AWeatherService.MESSAGE_RECEIVED:
        m_service.handleReceived(m);
        break;
      }
    }
  };

  private int m_port;
  private AWeatherService m_service;
  private DatagramSocket socket = null;

  private void forwardMsg(final int type, final String s) {
    final Message m = Message.obtain(m_handler, type);
    m.obj = new Msg(s);
    m_handler.sendMessage(m);

  }

  public void close() {
    socket.close();
    System.out.println("Receiver closed.");
  }

  public void push() {
    byte[] greet = "Hello".getBytes();
    DatagramPacket p;
    try {
      p = new DatagramPacket(greet, greet.length,
          InetAddress.getByName(FetchForecast.udpWsHost), m_port);
      socket.send(p);
      System.out.println(new Date().toString() + " :  Forward: "
          + p.getAddress());
    } catch (Exception e) {
      System.out.print(String.format("Exception: %s\n", e.getMessage()));
    }
  }

  @Override
  public void run() {
    forwardMsg(AWeatherService.MESSAGE_CONNECTED,
        String.format("Receiver started on port %d.", m_port));
    push();
    String s;
    while (domore) {
      try {
        byte[] buf = new byte[256];

        // receive request
        DatagramPacket packet = new DatagramPacket(buf, buf.length);
        socket.receive(packet);
        s = new String(packet.getData()).trim();
        forwardMsg(AWeatherService.MESSAGE_RECEIVED,
            new Date().toString() + " : " + packet.getAddress()
                + ": " + s);
        if (s.equals("update"))
          forwardMsg(AWeatherService.MESSAGE_RECEIVED, s);

      } catch (IOException e) {
        // gets here when ReceiverThread::close() is called.
        domore = false;
      }
    }
    socket.close();
  }

  public NetworkListener(final AWeatherService a, int port)
      throws SocketException {
    super("NetworkListener");
    m_service = a;
    socket = new DatagramSocket(port);
    m_port = port;
  }
}
package org.nws.aweather;

public interface NetworkTaskEvents {
  public void handleConnected(Msg o);

  public void handleDisconnected(Msg o);

  public void handleError(Msg o);

  public void handleInfo(Msg o);

  public void handleReceived(Msg o);
}
package org.nws.aweather;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Date;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;

import android.os.Handler;
import android.os.Message;

import com.nws.nwsserver.Info;

public class Responder {
  private final Handler m_handler = new Handler() {
    @Override
    public void handleMessage(final Message msg) {
      final Msg m = (Msg) msg.obj;
      ;
      switch (msg.what) {
      case AWeatherService.MESSAGE_DISCONNECTED:
        m_service.handleDisconnected(m);
        break;

      case AWeatherService.MESSAGE_CONNECTED:
        m_service.handleConnected(m);
        break;

      case AWeatherService.MESSAGE_ERROR:
        m_service.handleError(m);
        break;

      case AWeatherService.MESSAGE_INFO:
        m_service.handleInfo(m);
        break;

      case AWeatherService.MESSAGE_RECEIVED:
        m_service.handleReceived(m);
        break;
      }
    }
  };
  private AWeatherService m_service;
  private final BlockingQueue<Object> queue = new LinkedBlockingQueue<Object>(
      1024);
  protected DatagramSocket socket = null;

  void begin() {
    try {
      socket = new DatagramSocket();
    } catch (SocketException e1) {
      // TODO Auto-generated catch block
      e1.printStackTrace();
    }
    new Thread(new Runnable() {
      @Override
      public void run() {
        while (true) {
          try {
            Object o = queue.take();
            process(o);
            Thread.sleep(750);
          } catch (InterruptedException e) {
            e.printStackTrace();
          }
        }
      }
    }, "Consumer").start();
  }

  private void forwardMsg(final int type, final String s) {
    final Message m = Message.obtain(m_handler, type);
    m.obj = new Msg(s);
    m_handler.sendMessage(m);
  }

  private void process(Object o) {
    if (o instanceof DatagramPacket)
      processDGP(o);
    else if (o instanceof String)
      processStr(o);
    else if (o instanceof Info)
      processInfo(o);
  }

  private void processDGP(Object o) {
    DatagramPacket packet = (DatagramPacket) o;
    InetAddress address = packet.getAddress();
    int port = packet.getPort();
    byte[] data = packet.getData();
    try {
      packet = new DatagramPacket(data, data.length, address, port);
    } catch (Exception e) {
      e.printStackTrace();
      return;
    }
    try {
      socket.send(packet);
    } catch (IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }
  }

  private void processInfo(Object o) {
    forwardMsg(AWeatherService.MESSAGE_INFO, new Date().toString() + " : "
        + ((Info) o).toString());
  }

  private void processStr(Object o) {
    String s = ((String) (o)).trim();
    if (s.equals("update"))
      forwardMsg(AWeatherService.MESSAGE_RECEIVED, s);
    else forwardMsg(AWeatherService.MESSAGE_INFO, new Date().toString() + " : "
        + ((String) (o)).trim());

  }

  public void close() {
    socket.close();
    System.out.println("Responder closed.");
  }

  public void respond(Object o) throws InterruptedException {
    if (o != null)
      queue.put(o);
  }

  public Responder(final AWeatherService a) {
    m_service = a;
  }
}
package org.nws.aweather;

import android.app.Activity;
import android.app.ActivityManager;
import android.app.ActivityManager.RunningServiceInfo;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class ServiceLauncher extends Activity {
  private Button start;
  private final OnClickListener startListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      startService(new Intent(ServiceLauncher.this, AWeatherService.class));
      update();
    }
  };

  private Button stop;

  private final OnClickListener stopListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      stopService(new Intent(ServiceLauncher.this, AWeatherService.class));
      update();
    }
  };

  private boolean serviceIsRunning() {
    ActivityManager manager = (ActivityManager) getSystemService(Context.ACTIVITY_SERVICE);
    for (RunningServiceInfo service : manager
        .getRunningServices(Integer.MAX_VALUE)) {
      if (AWeatherService.class.getName().equals(
          service.service.getClassName())) {
        return true;
      }
    }
    return false;
  }

  private void update() {
    boolean f = serviceIsRunning();
    start.setEnabled(!f);
    stop.setEnabled(f);
  }

  /** Called when the activity is first created. */
  @Override
  public void onCreate(final Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.servicelauncher);
    start = (Button) findViewById(R.id.startButton);
    stop = (Button) findViewById(R.id.stopButton);
    start.setOnClickListener(startListener);
    stop.setOnClickListener(stopListener);
    update();
  }

}
package org.nws.aweather;

import java.net.InetAddress;
import java.net.UnknownHostException;

import org.nws.aweather.forecast.FetchForecast;

import android.app.Activity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class SetServer extends Activity {
  private Button cancelButton;
  private final OnClickListener cancelListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      finish();
    }
  };
  private Button defaultButton;
  private final OnClickListener defaultListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      String s = FetchForecast.getInstance().getDefaultUrl();
      url.setText(s);
    }
  };
  private Button setUrlButton;

  private final OnClickListener setUrlListener = new OnClickListener() {
    @Override
    public void onClick(final View v) {
      if (isvalidurl()){
        String s = url.getText().toString();
        FetchForecast.getInstance().setUrl(s);
            setResult( RESULT_OK, getIntent( ) );
        finish();
      }
      else
        Toast.makeText(SetServer.this, "Invalid Url", Toast.LENGTH_LONG)
            .show();
    }
  };

  private EditText url;
  private boolean isvalidurl() {
    String s = url.getText().toString();
    try {
      @SuppressWarnings("unused")
      InetAddress serverAddress = InetAddress.getByName(s);
    } catch (UnknownHostException e) {
      return false;
    }
    return true;
  }

  private void update()
  {
    setUrlButton.setEnabled(isvalidurl());
  }

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.setserver);
      setResult( RESULT_CANCELED, getIntent( ) );
    url = (EditText) findViewById(R.id.enterurl);
    String s = FetchForecast.getInstance().getUrl();
    url.setText(s);
    setUrlButton = (Button) findViewById(R.id.setUrlButton);
    defaultButton = (Button) findViewById(R.id.defaultButton);
    cancelButton = (Button) findViewById(R.id.CancelButton);
    setUrlButton.setOnClickListener(setUrlListener);
    defaultButton.setOnClickListener(defaultListener);
    cancelButton.setOnClickListener(cancelListener);
    url.addTextChangedListener(new TextWatcher() {

      @Override
      public void afterTextChanged(Editable arg0) {

      }

      @Override
      public void beforeTextChanged(CharSequence arg0, int arg1,
          int arg2, int arg3) {

      }

      @Override
      public void onTextChanged(CharSequence arg0, int arg1, int arg2,
          int arg3) {
        update();

      }


    });
  }

}
package org.nws.aweather;

import java.io.BufferedInputStream;
import java.io.IOException;
import java.net.URL;
import java.net.URLConnection;
import java.util.List;

import org.nws.aweather.forecast.FetchForecast;
import org.nws.aweather.forecast.NWSWeatherObject;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Handler;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;

public class WeatherGallery {
  private final Handler handler;
  private final ArrayAdapter<String> locationAdapter;
  private final Spinner locationChooser;
  private final LocationManager locationManager;
  private final AWeather owner;
  private final Button refreshButton;

  private Bitmap getImage(final String httpLocation) throws IOException {
    Bitmap result = null;
    final URL url = new URL(httpLocation);
    final URLConnection con = url.openConnection();
    final BufferedInputStream in = new BufferedInputStream(
        con.getInputStream());
    result = BitmapFactory.decodeStream(in);
    in.close();
    return result;
  }

  private void updateWeather(final NWSWeatherObject nwo) {
    final ImageView i = (ImageView) owner.findViewById(R.id.image1);
    if (i == null)
      // view has bailed on us- just fail fast
      return;
    try {
      final List<String> conditions = nwo.getConditionIconLinks();
      final String first = conditions.get(0);
      final Bitmap bm = getImage(first);
      i.setImageBitmap(bm);
    } catch (final IOException e) {
      throw new RuntimeException(e);
    }
    final TextView t = (TextView) owner.findViewById(R.id.day);
    t.setText("Tomorrow");
  }

  private void updateWeather(final String s) {
    final ImageView i = (ImageView) owner.findViewById(R.id.image1);
    if (i == null)
      // view has bailed on us- just fail fast
      return;
    try {
      final Bitmap bm = getImage(s);
      i.setImageBitmap(bm);
    } catch (final IOException e) {
      throw new RuntimeException(e);
    }
    final TextView t = (TextView) owner.findViewById(R.id.day);
    t.setText("Tomorrow");
  }

  public void gotWeather(final NWSWeatherObject nwo) {
    final Thread X = new Thread() {
      @Override
      public void run() {
        updateWeather(nwo);
      }
    };
    handler.post(X);
  }

  public void gotWeather(final String s) {
    final Thread X = new Thread() {
      @Override
      public void run() {
        updateWeather(s);
      }
    };
    handler.post(X);
  }

  public void logError(Object p) {
    owner.forwardMsg(AWeatherService.MESSAGE_ERROR, p);
  }

  public void refresh() {
    final String[] location = locationManager.getLocation(0);
    FetchForecast.getInstance().requestForecast(location[1], location[2]);
  }

  public WeatherGallery(final AWeather owner) {
    this.owner = owner;
    FetchForecast.parent = this;
    locationManager = owner.getLocationManager();

    locationChooser = (Spinner) owner.findViewById(R.id.locationChooser);
    refreshButton = (Button) owner.findViewById(R.id.refreshButton);
    refreshButton.setOnClickListener(owner);
    handler = new Handler();
    locationAdapter = new ArrayAdapter<String>(
        owner.getApplicationContext(),
        android.R.layout.simple_spinner_item);
    final String[] location = locationManager.getLocation(locationManager
        .getSelected());
    locationAdapter.add(location[0]);
    String s = locationManager.getServer();
    FetchForecast.getInstance().setUrl(s);
    FetchForecast.getInstance().requestForecast(location[1], location[2]);
    locationChooser.setAdapter(locationAdapter);
  }
}
```

90%

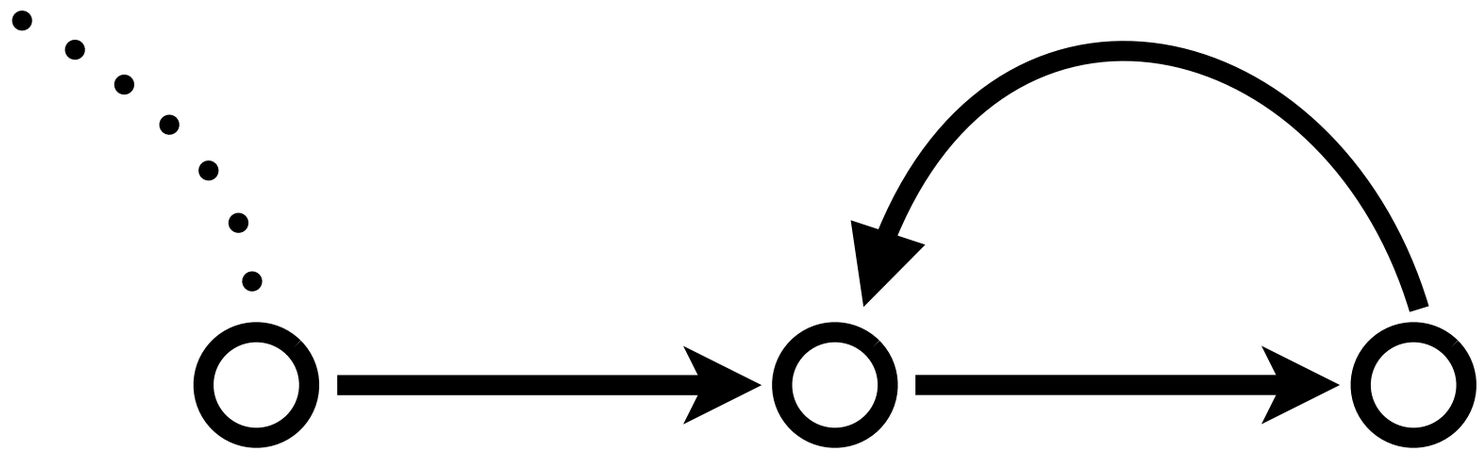
(Challenge 3A)

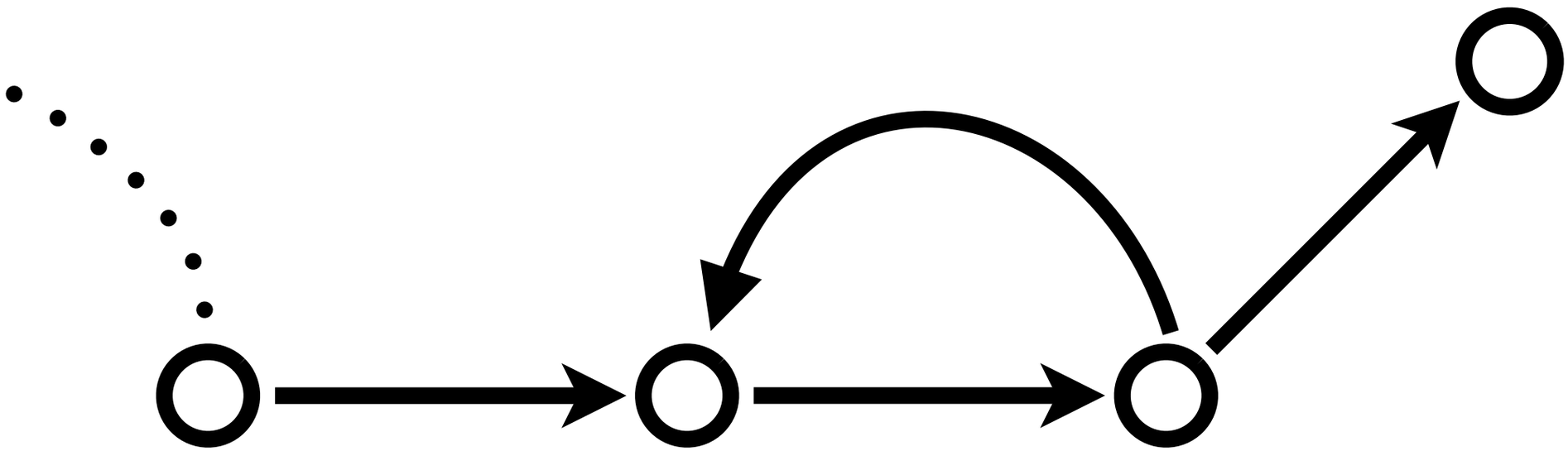
Terminal — zsh — 82×20 — ⌘1

```
</0>


<282!matt@maptop:~/Copy/APAC-Demo/Anadroid-AWeather>
% 
```

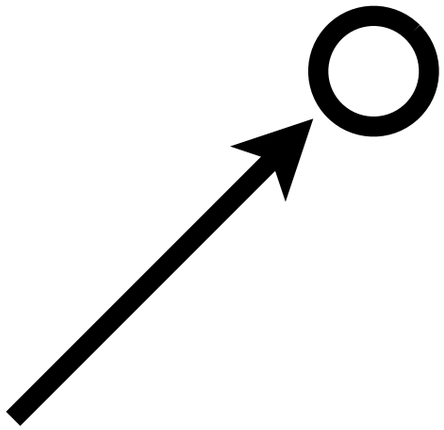
analyst

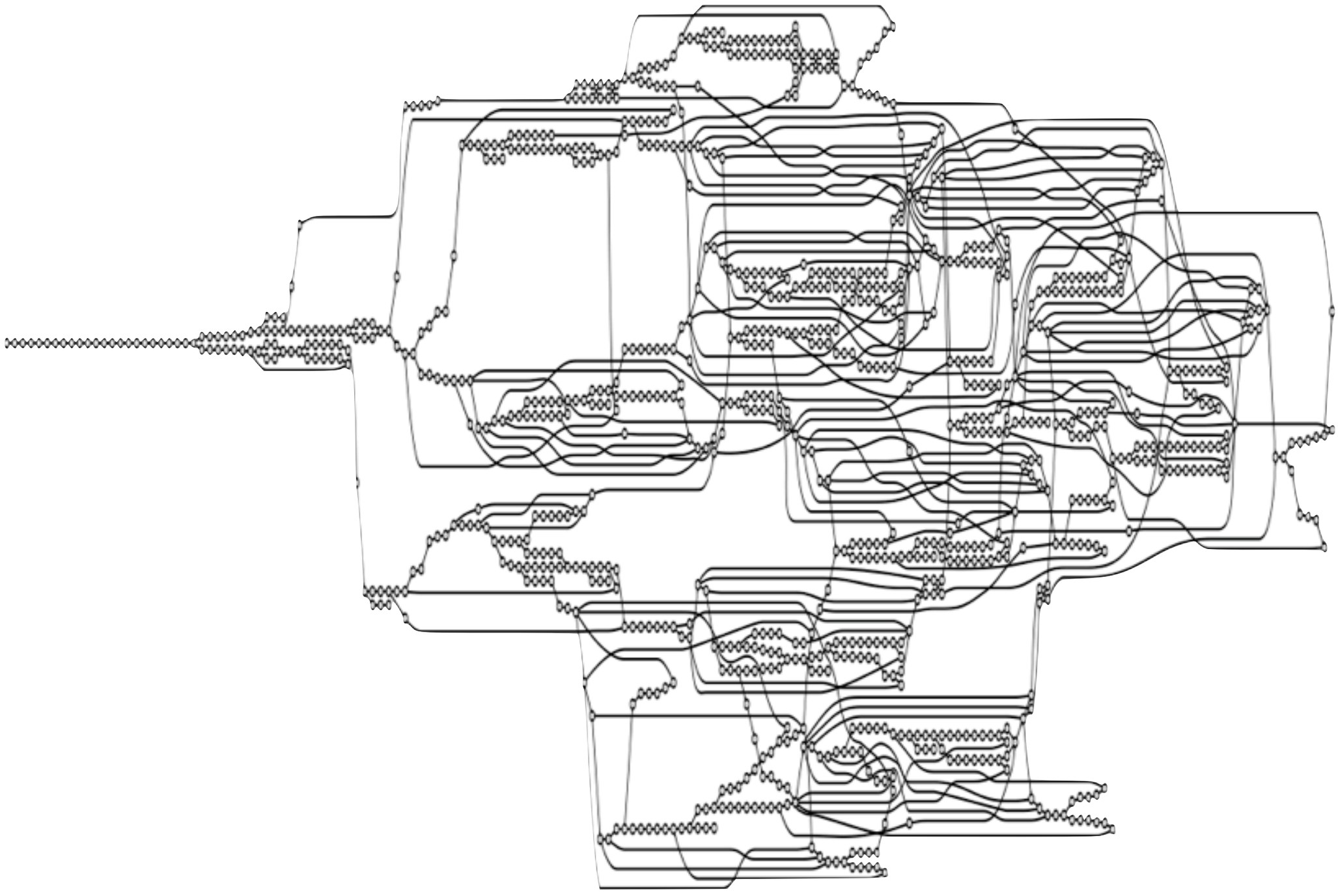
Trick 1

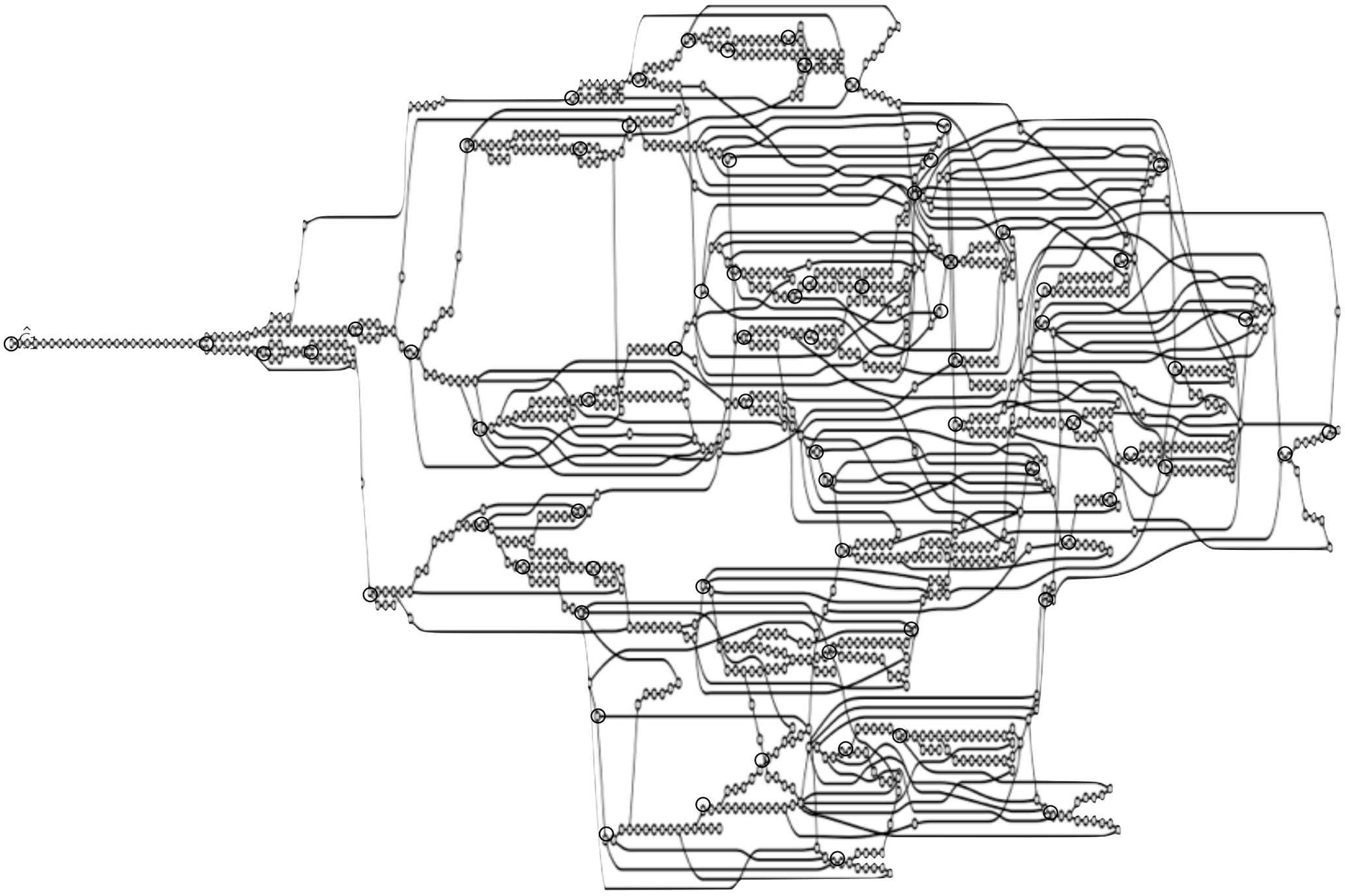


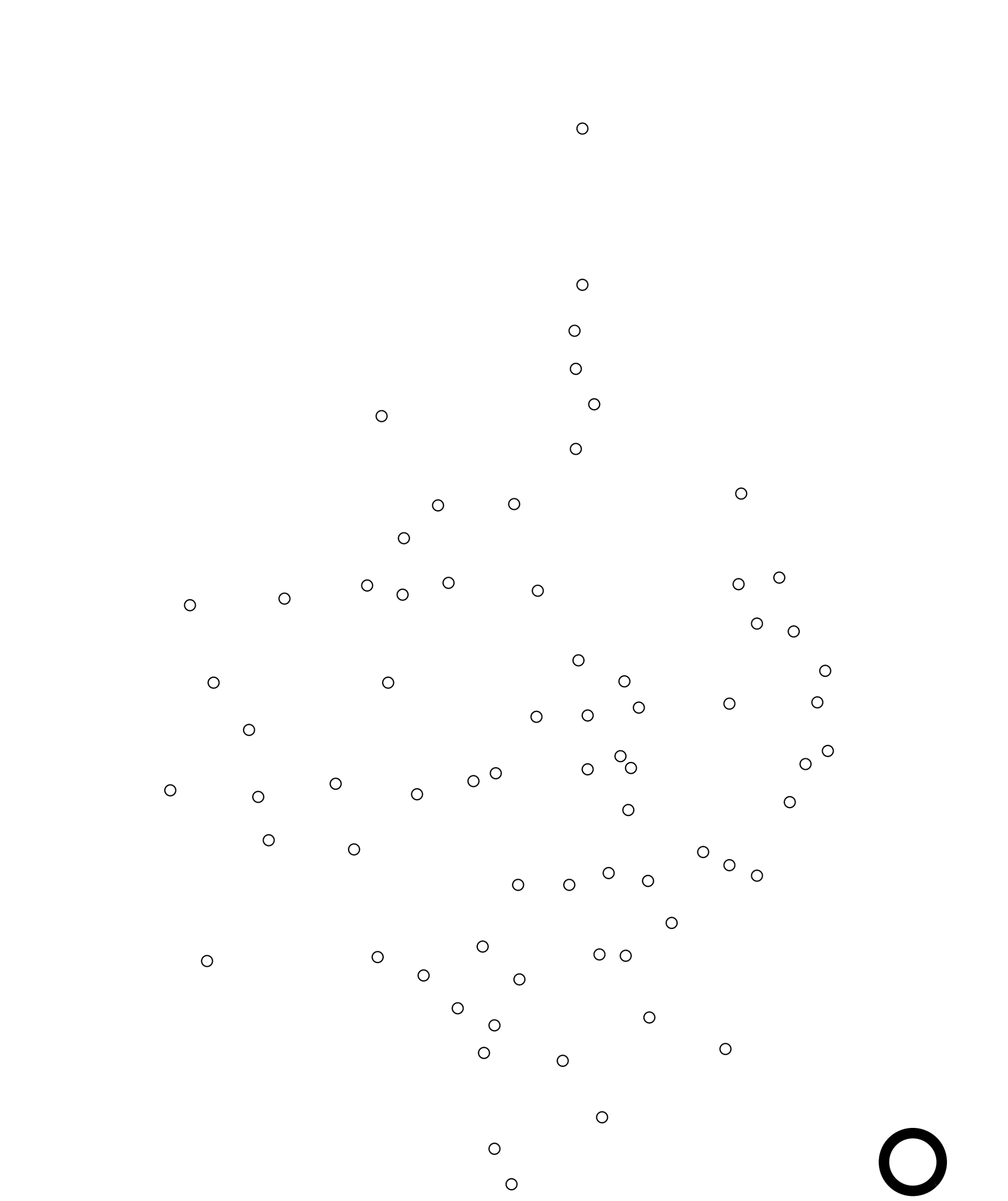


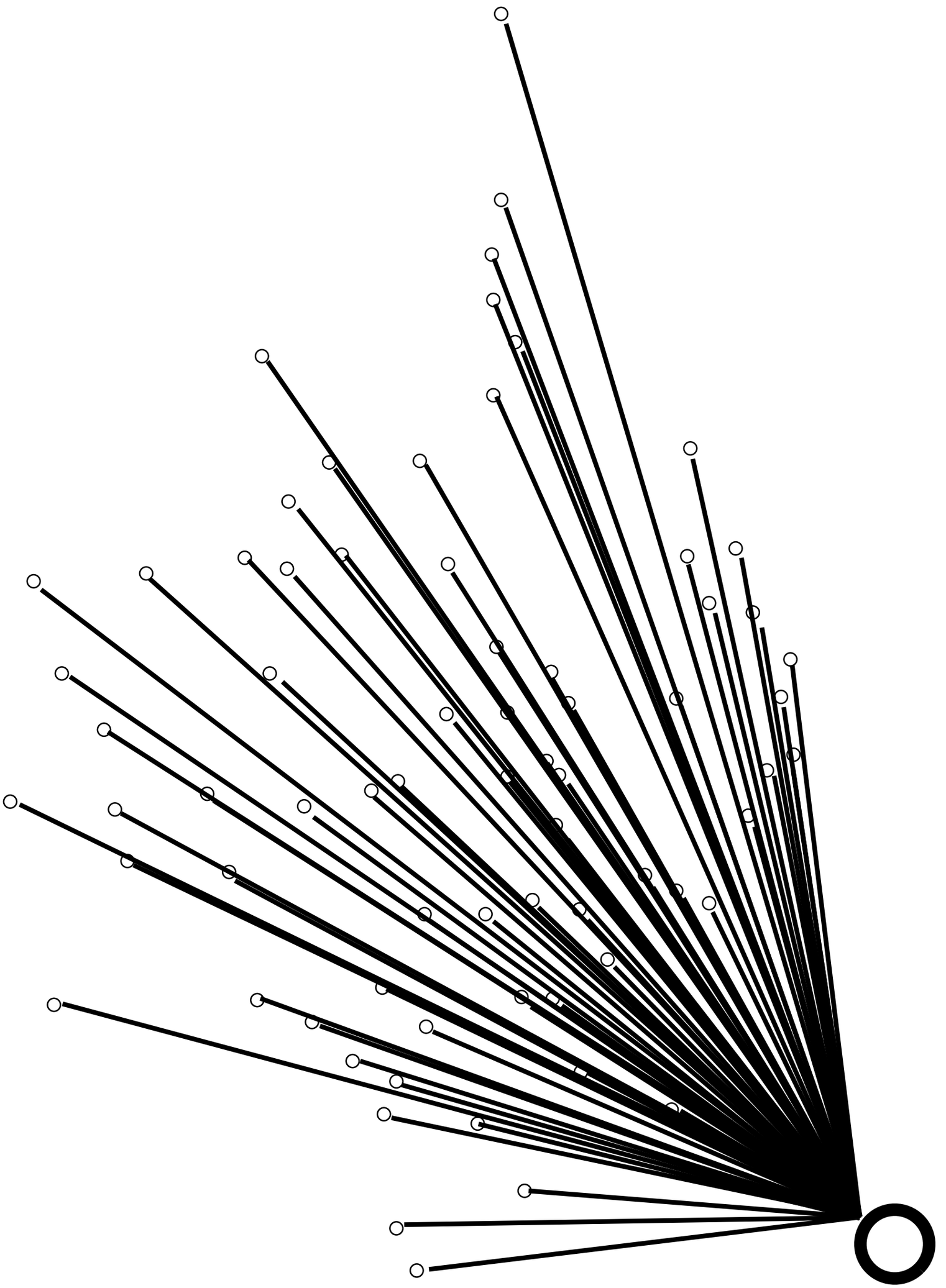
Problem

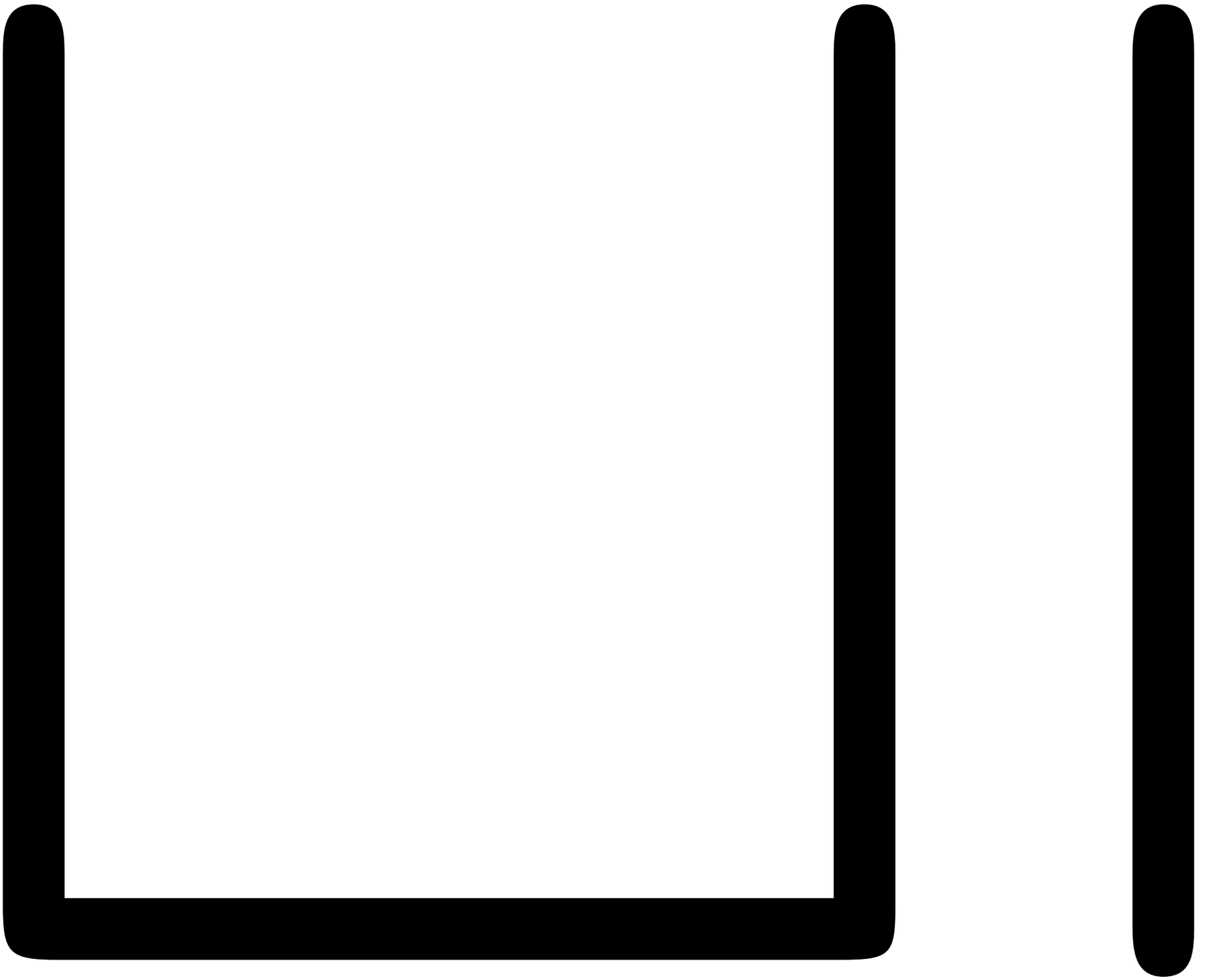


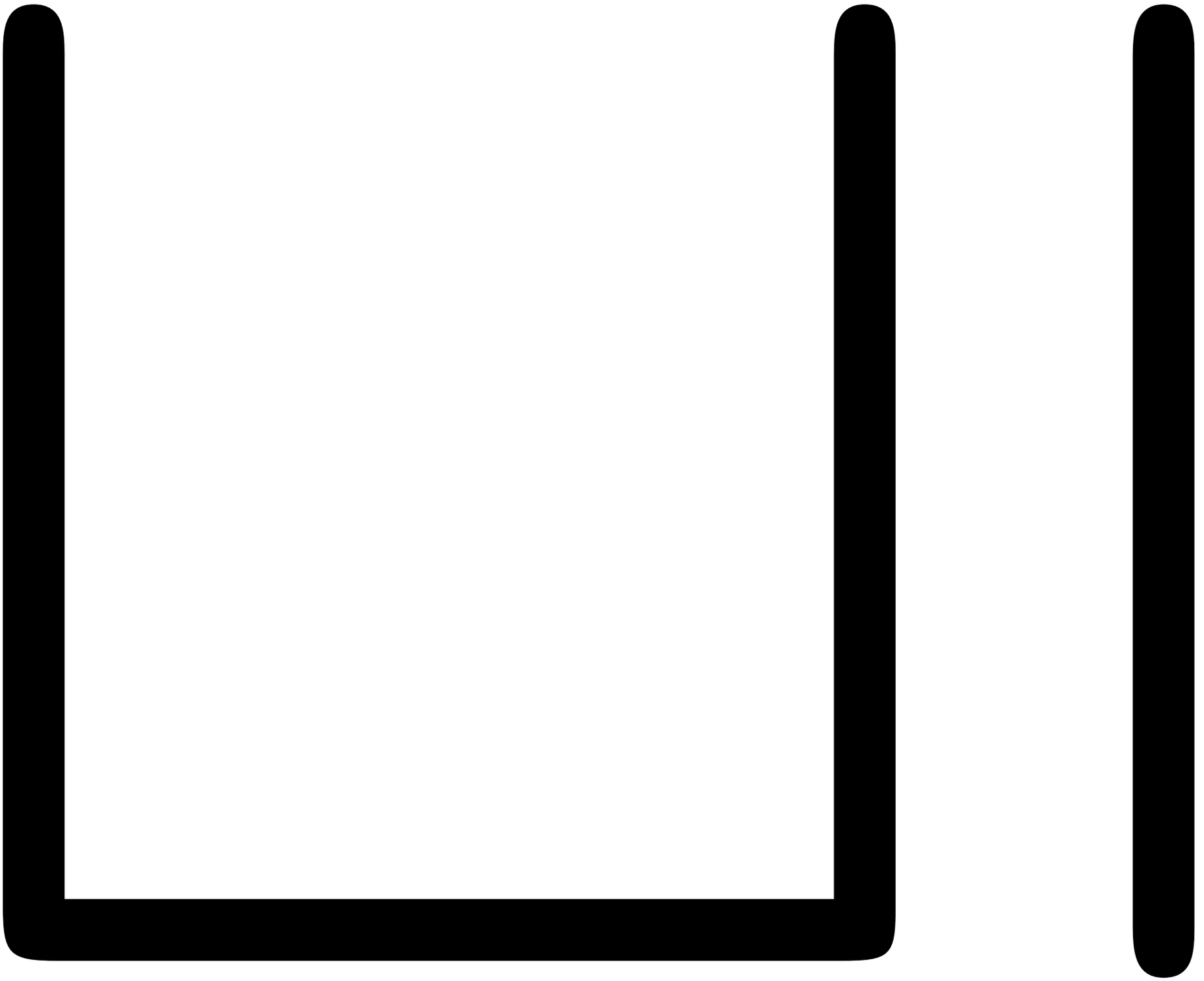



$\hat{C}_1$









$$\hat{\varsigma}_1 \sqsubseteq \hat{\varsigma}_2$$

$$\hat{\varsigma}_1 = \left(e, \hat{\rho}, \hat{\sigma}, \hat{\kappa}\right)$$

instruction

heap

$$\hat{\varsigma}_1 = (e, \hat{\rho}, \hat{\sigma}, \hat{\kappa})$$

frame pointer

stack

instruction

heap

$$\hat{\varsigma}_1 = (e, \hat{\rho}, \hat{\sigma}, \hat{\kappa})$$

frame pointer

stack

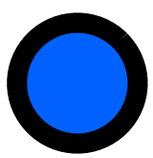
$$\hat{\varsigma}_1 \sqsubseteq \hat{\varsigma}_2$$

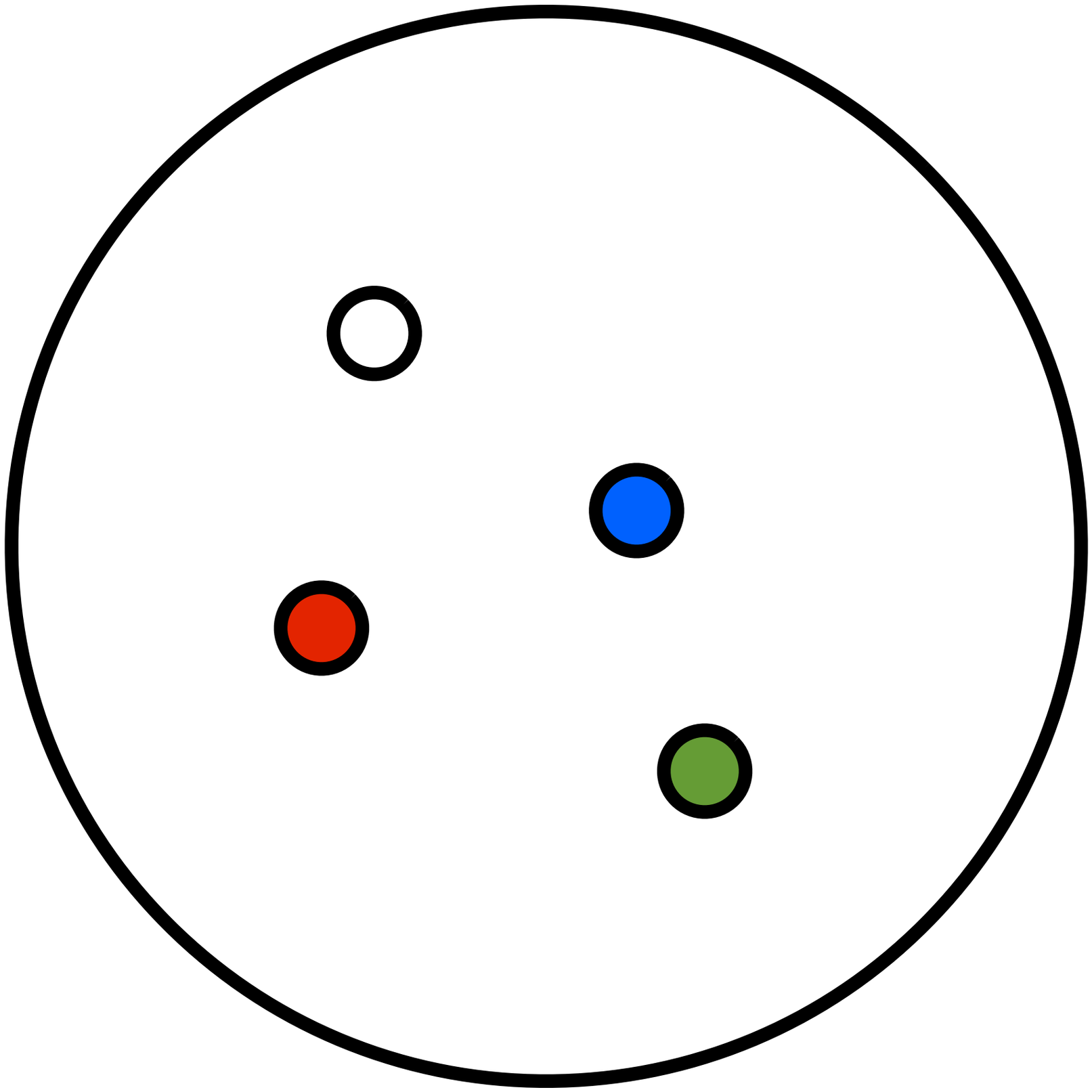
BEST PAPER

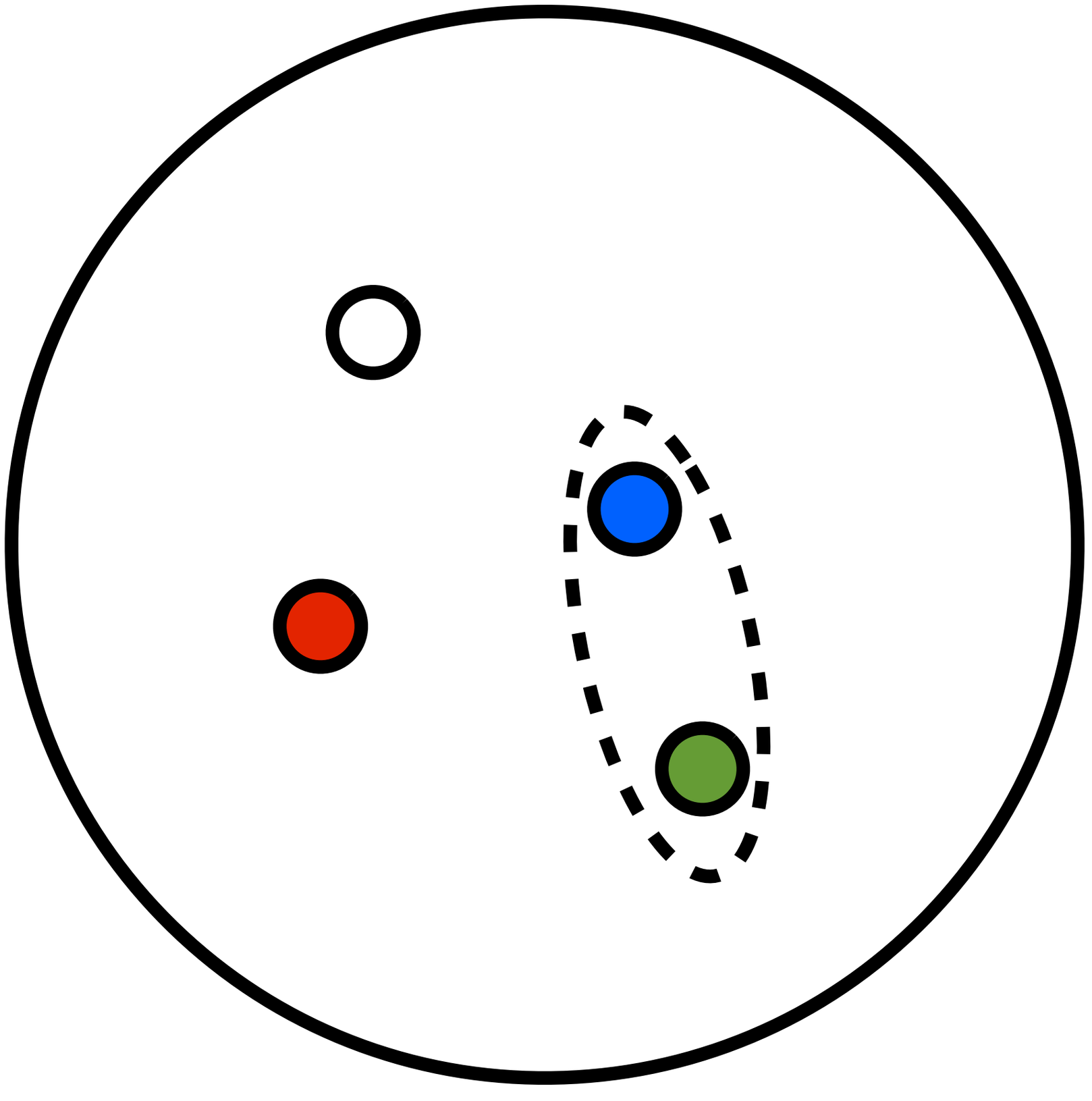
$$\hat{\S}_1 \sqsubseteq \hat{\S}_2$$

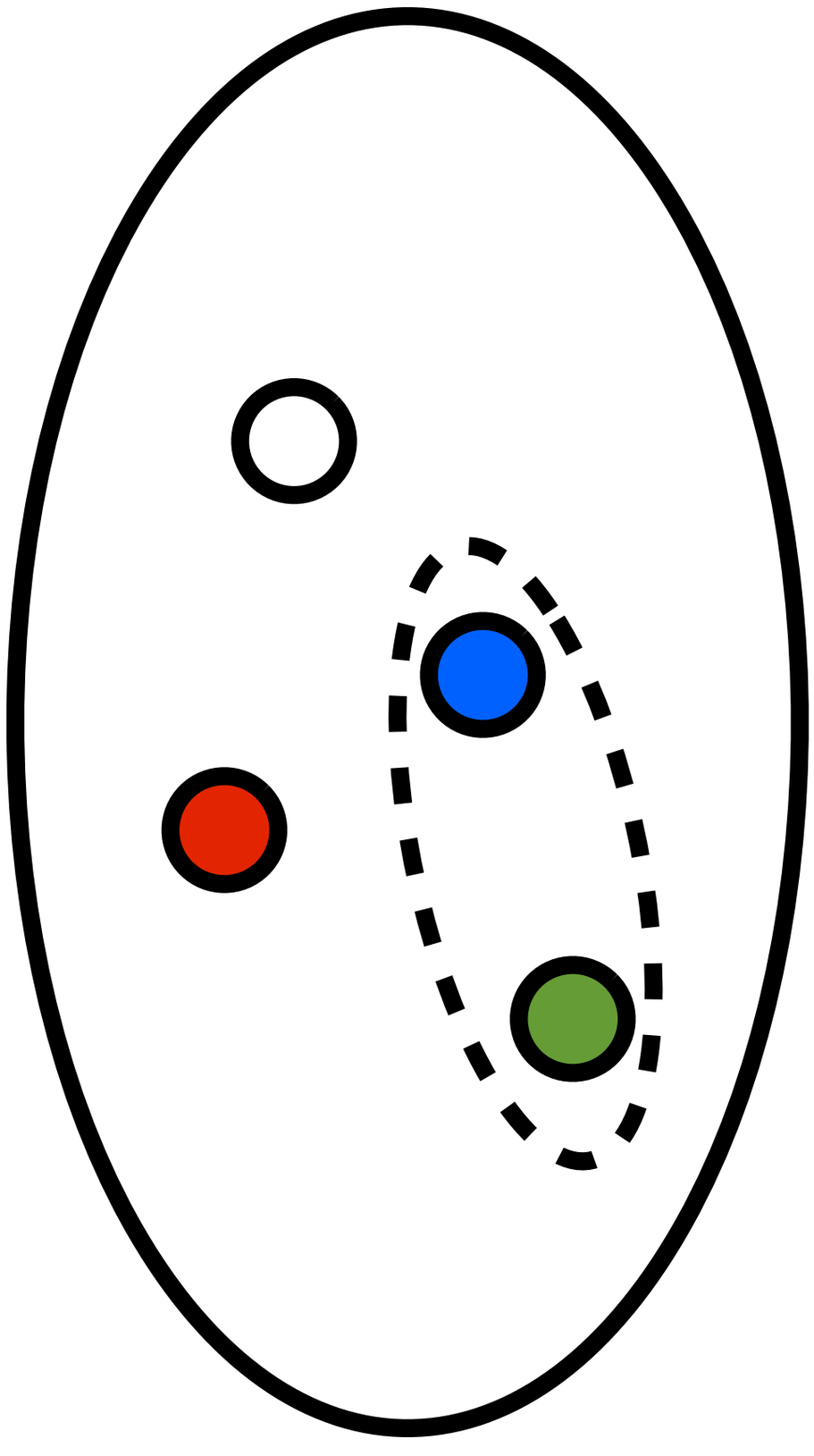
First: Hash sets

Prime decomposition

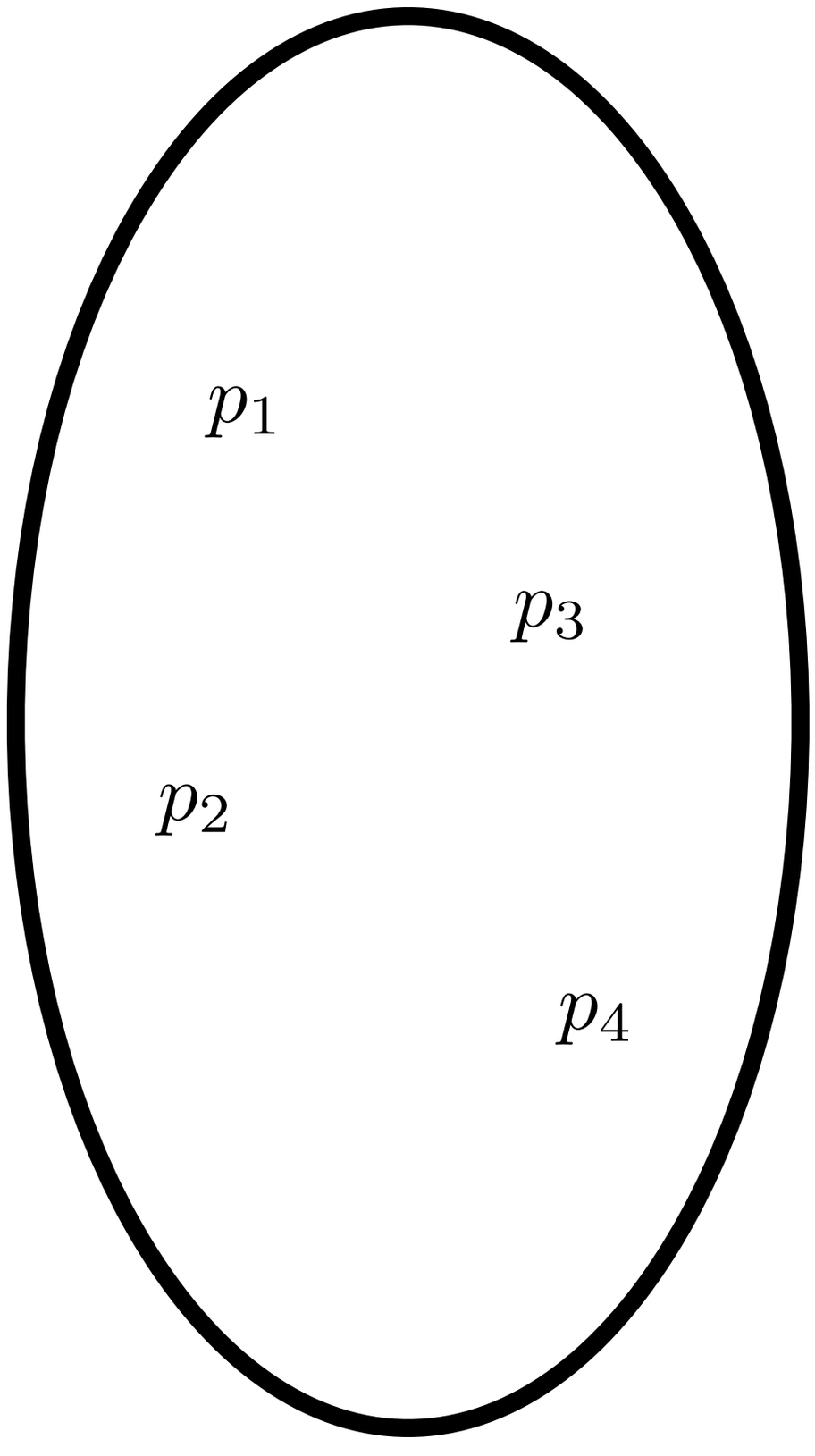


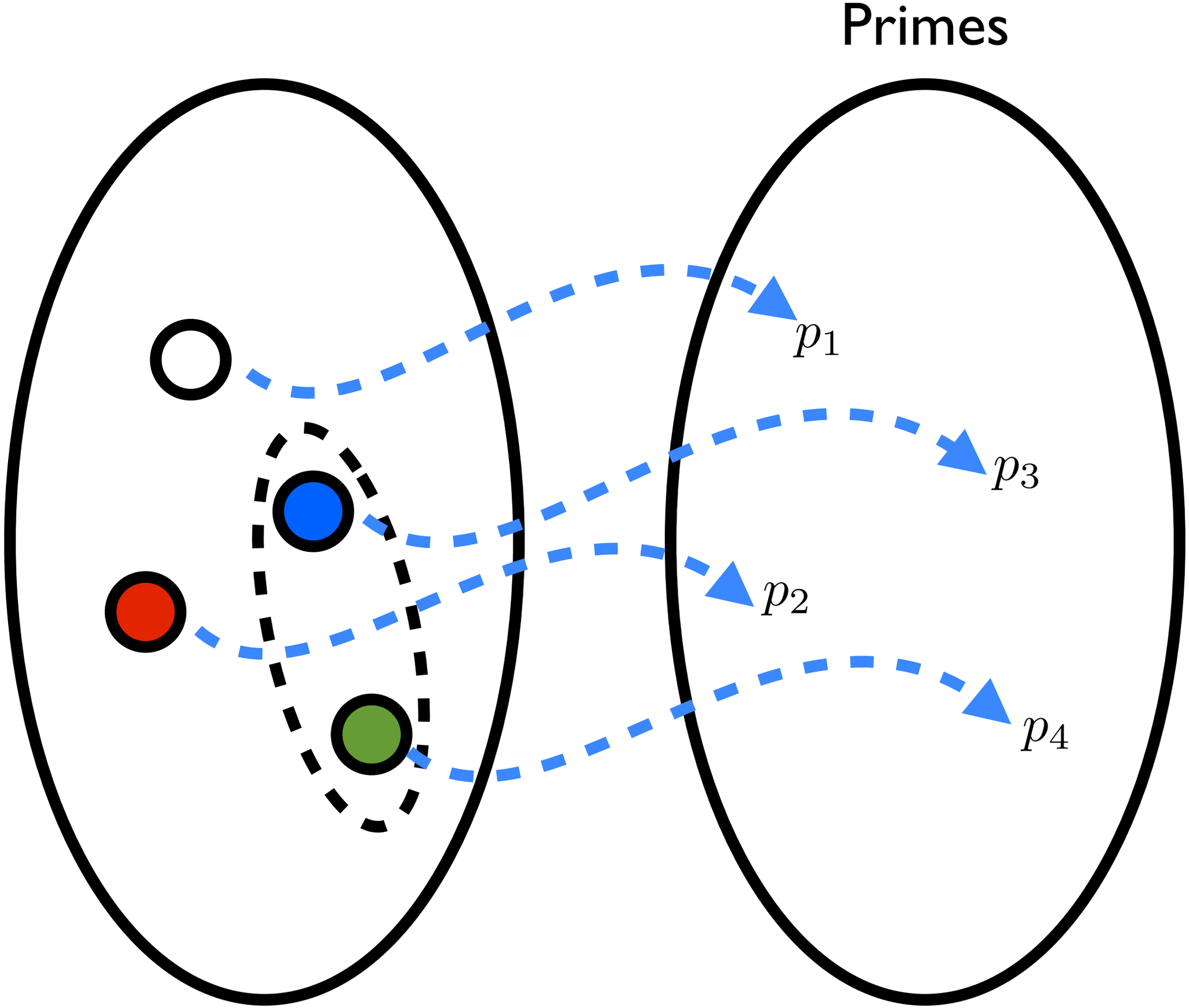


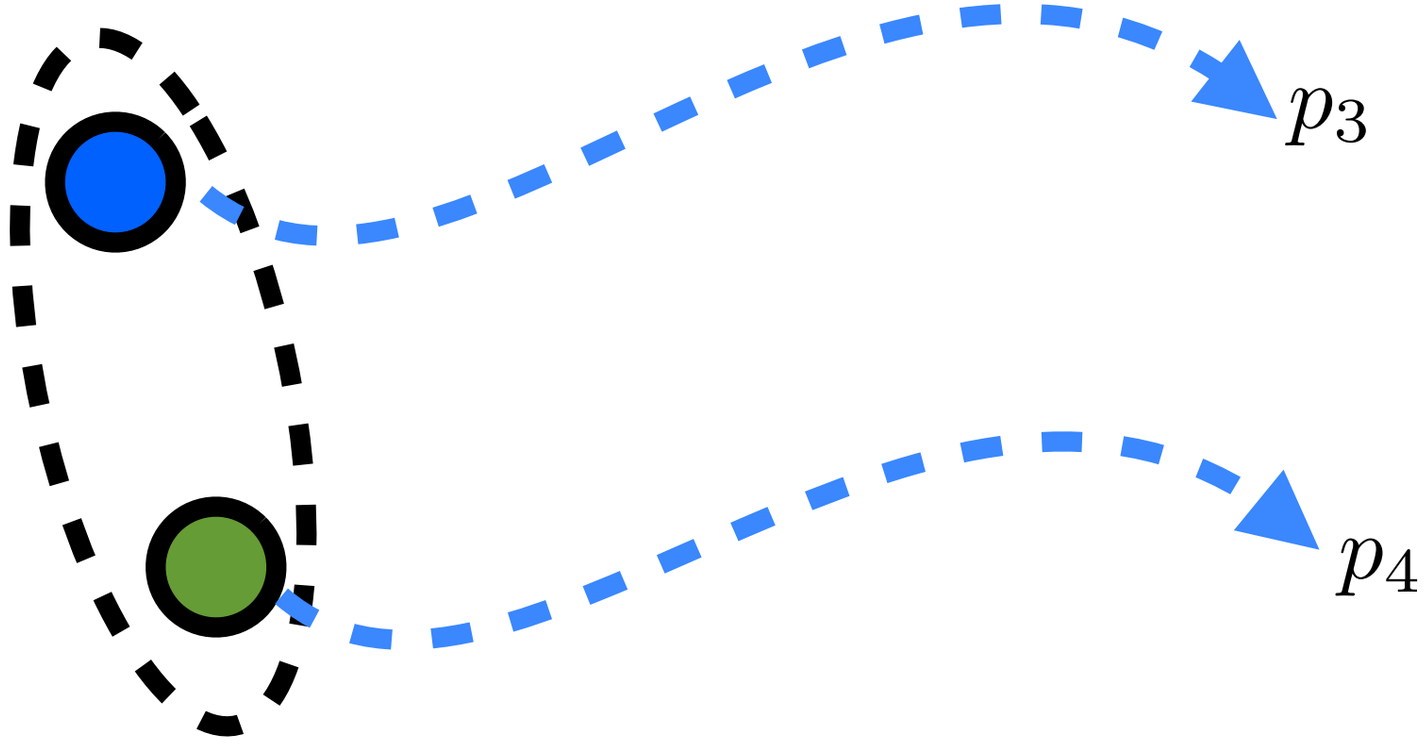




Primes




Primes
p_1
p_3
p_2
p_4


p_3
p_4

p_3 p_4

$$p_3 \times p_4$$

$$A \subseteq B$$

$$[\![B]\!] \bmod [\![A]\!] = 0$$

$$A \cap B$$

$$\gcd(\llbracket A \rrbracket, \llbracket B \rrbracket)$$

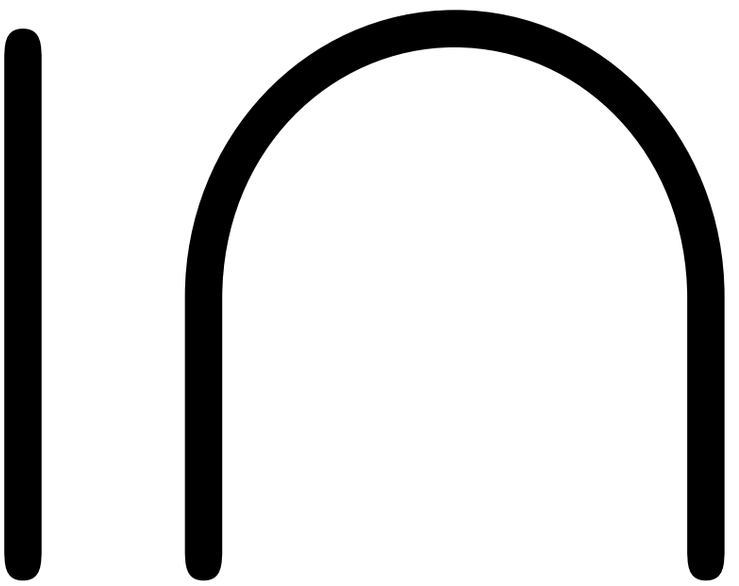
$$\operatorname{lcm}(\llbracket A \rrbracket, \llbracket B \rrbracket)$$

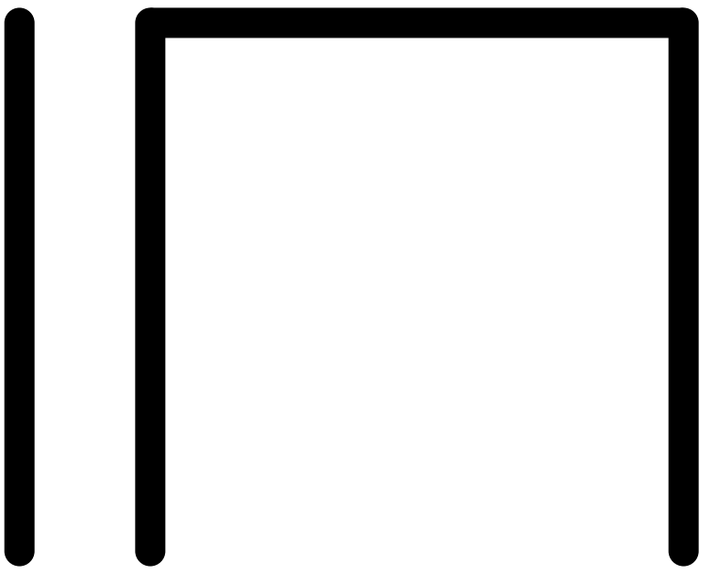
$$A \cup B$$

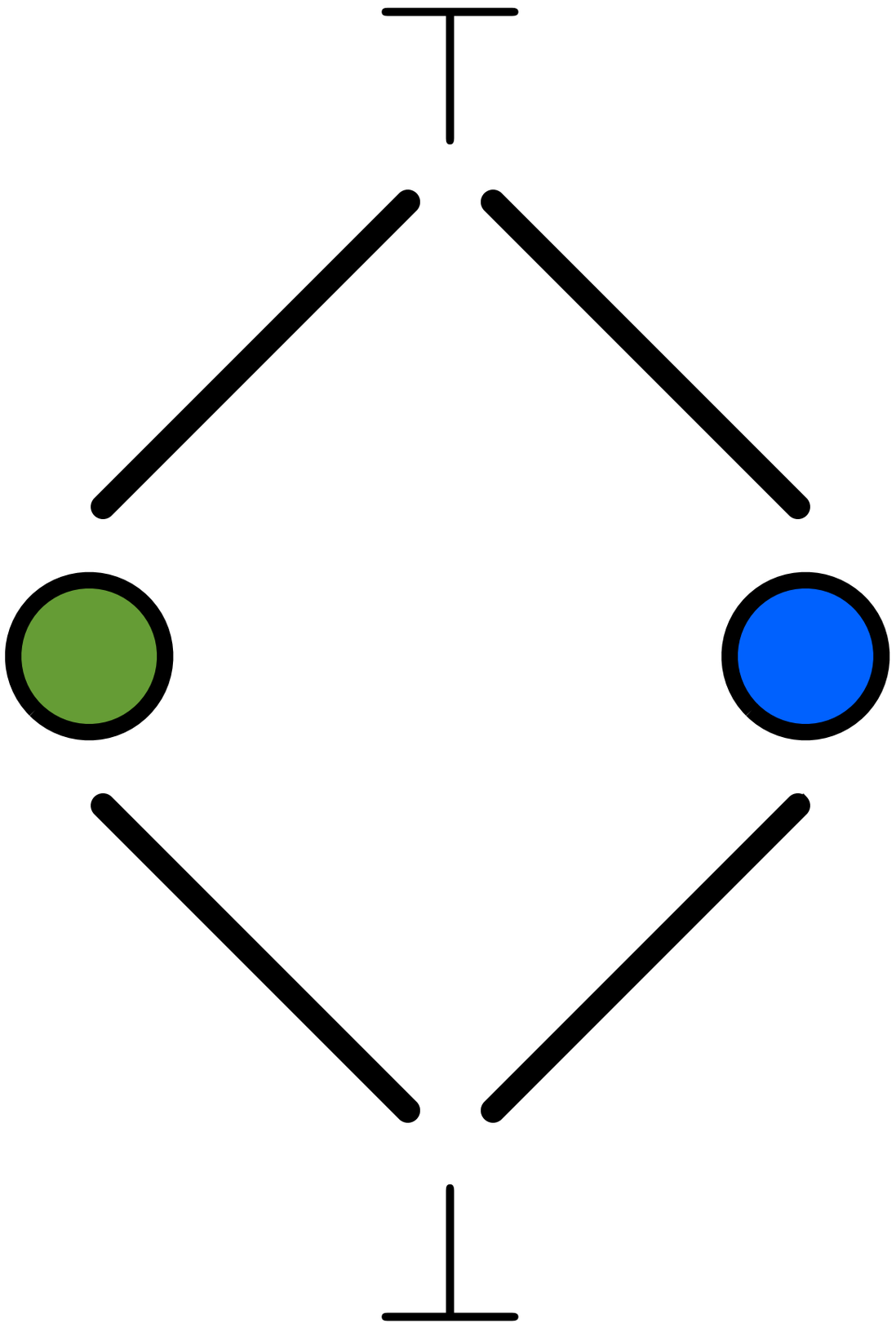
$$A \cup B$$

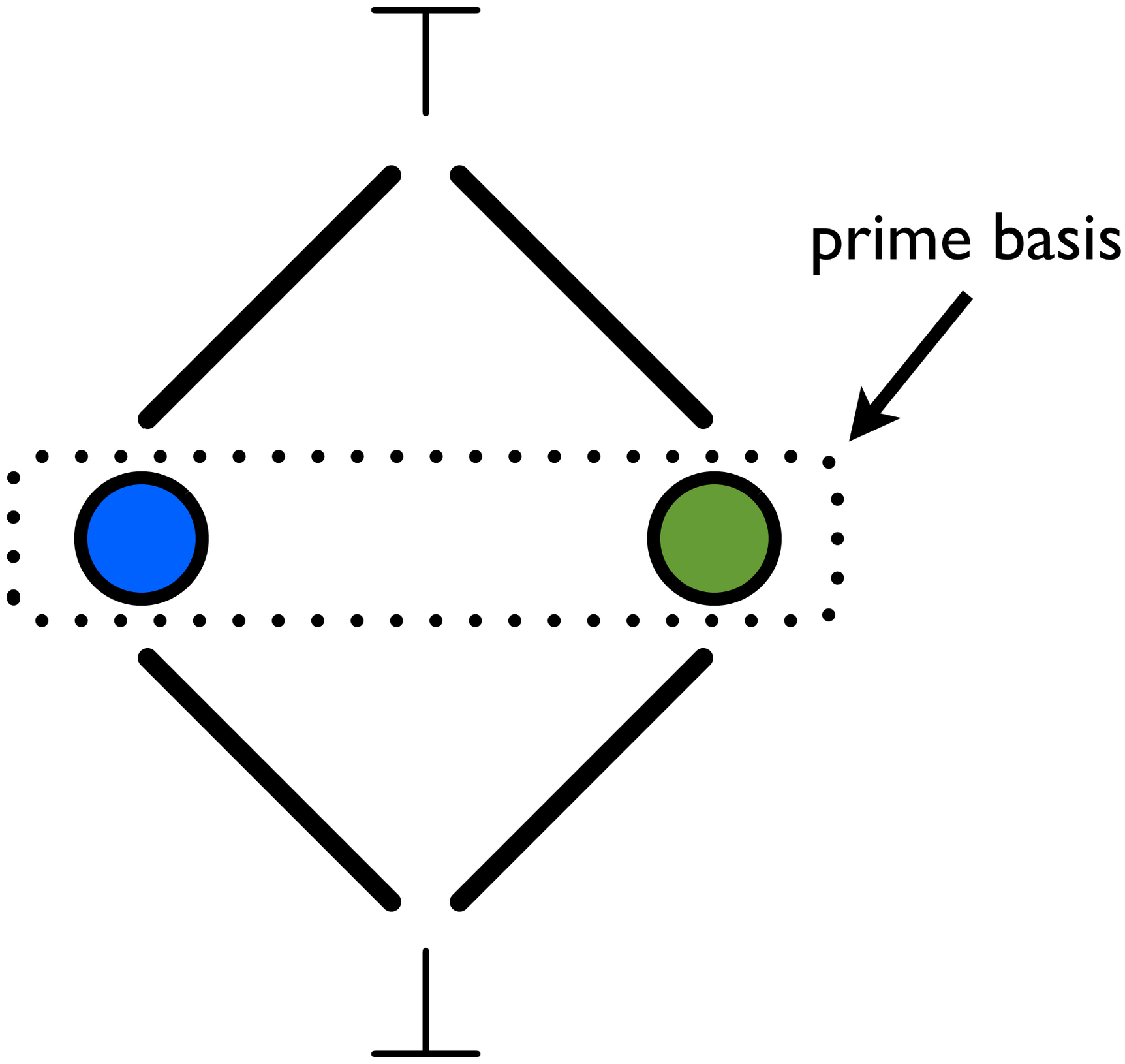
$$A - B$$

$$[\![A]\!]/\gcd([\![A]\!],[\![B]\!])$$

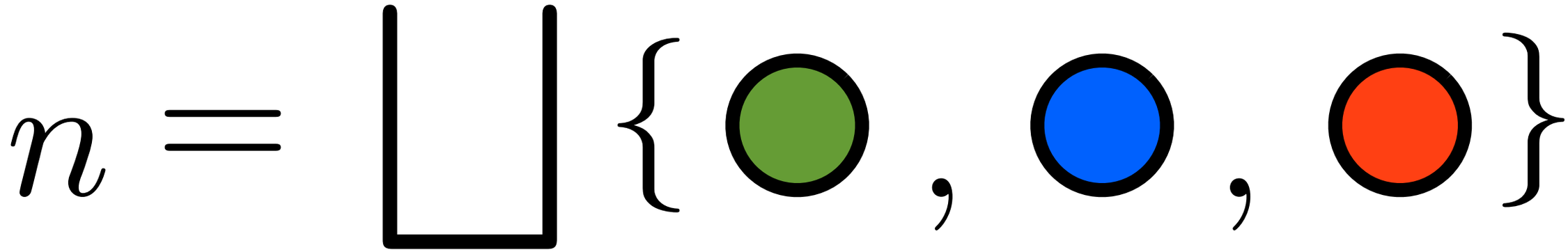


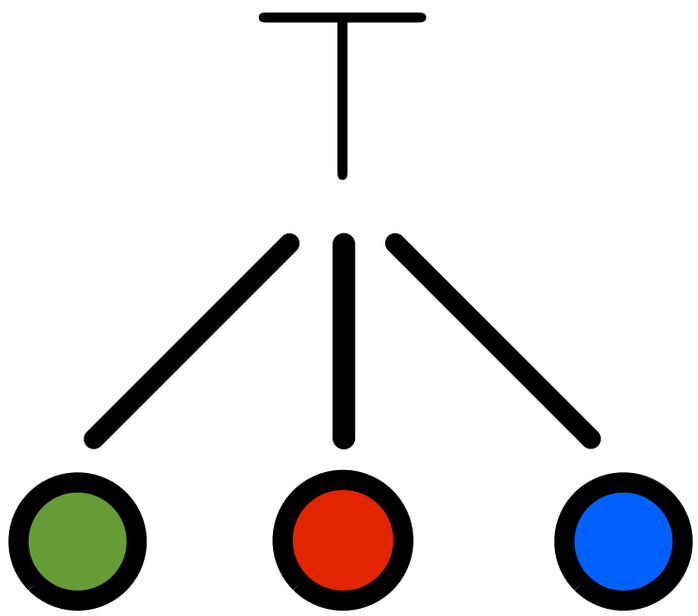


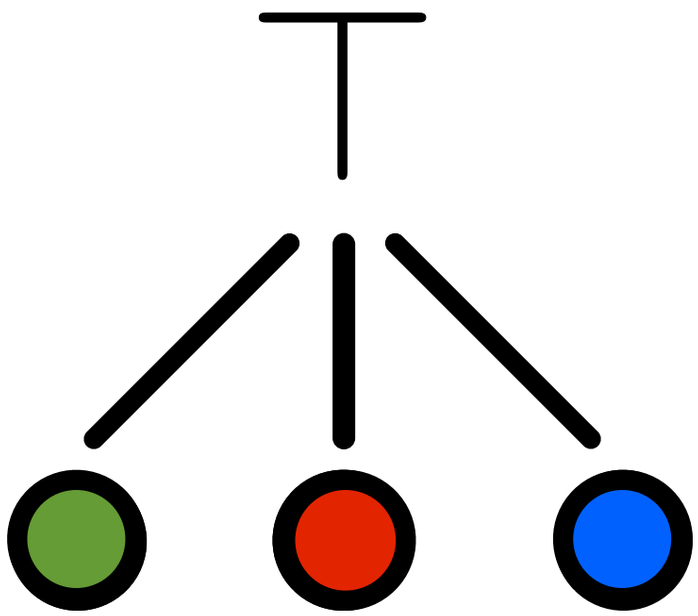


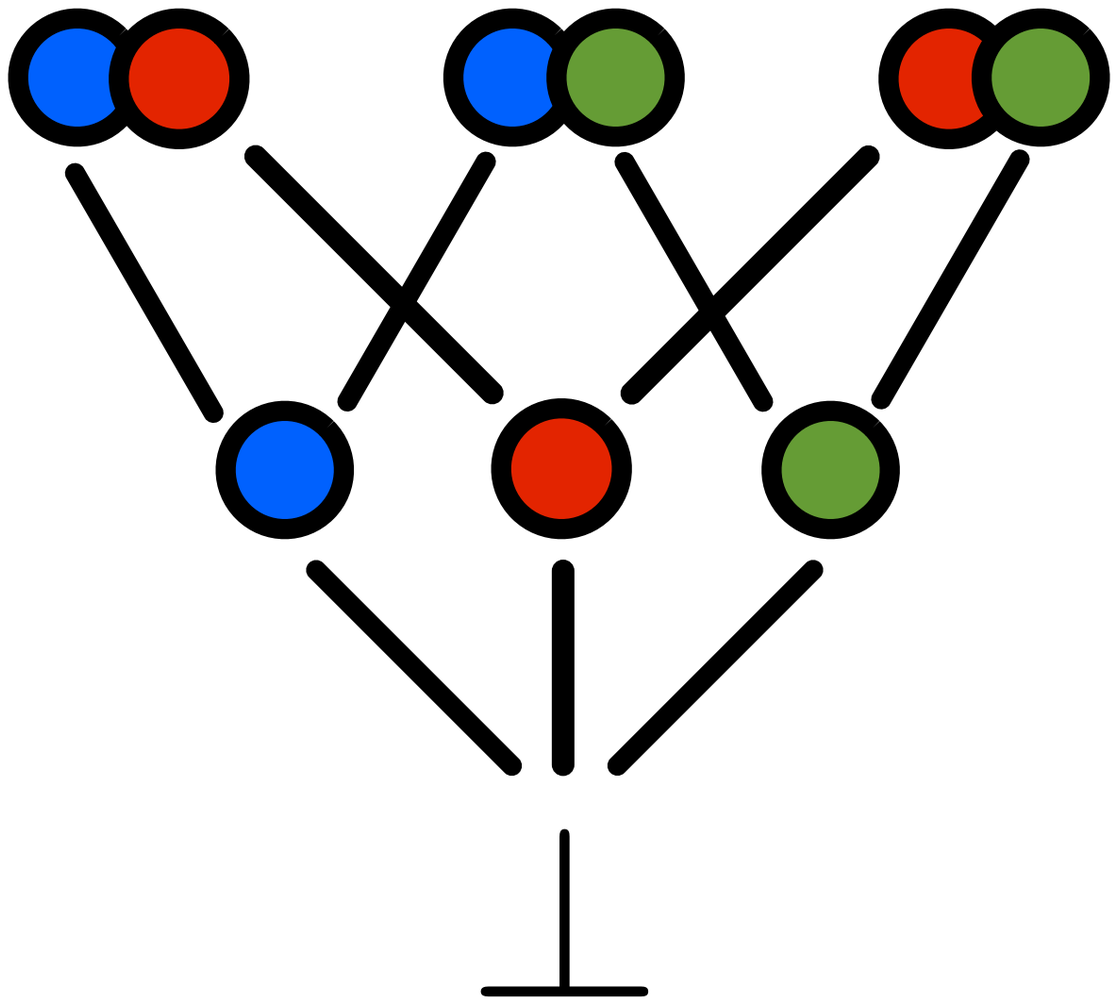

prime basis

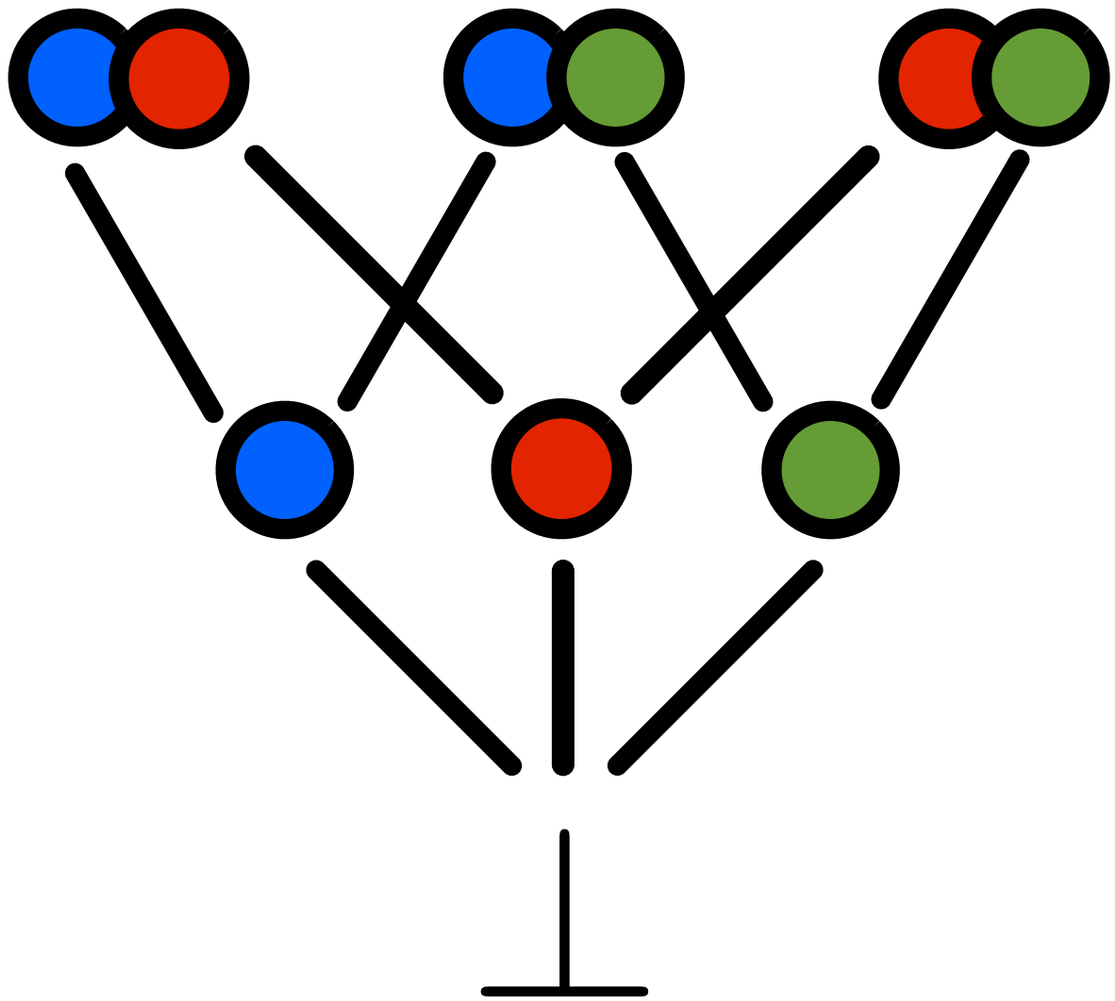
$$n = p_1^{m_1} p_2^{m_2} p_3^{m_3} \cdots$$

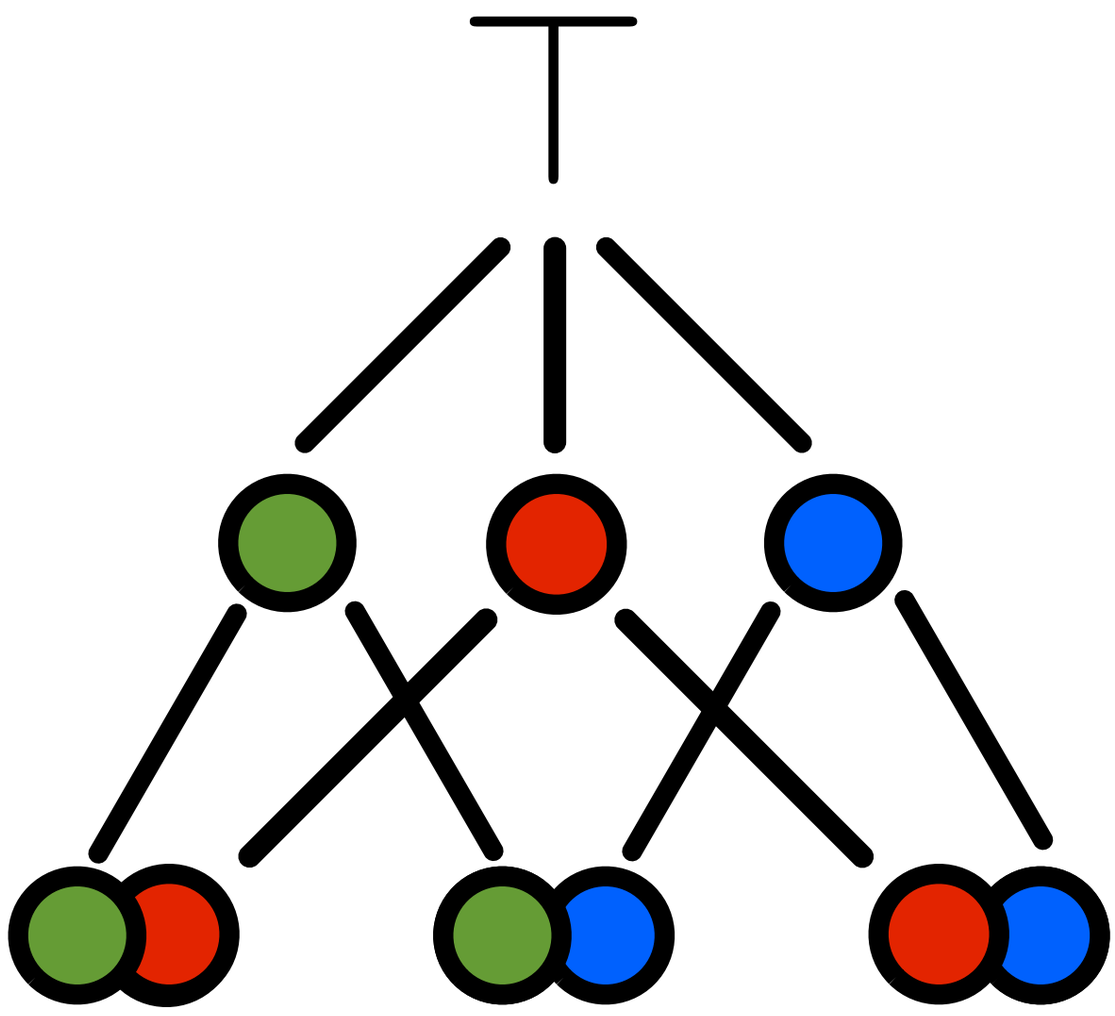


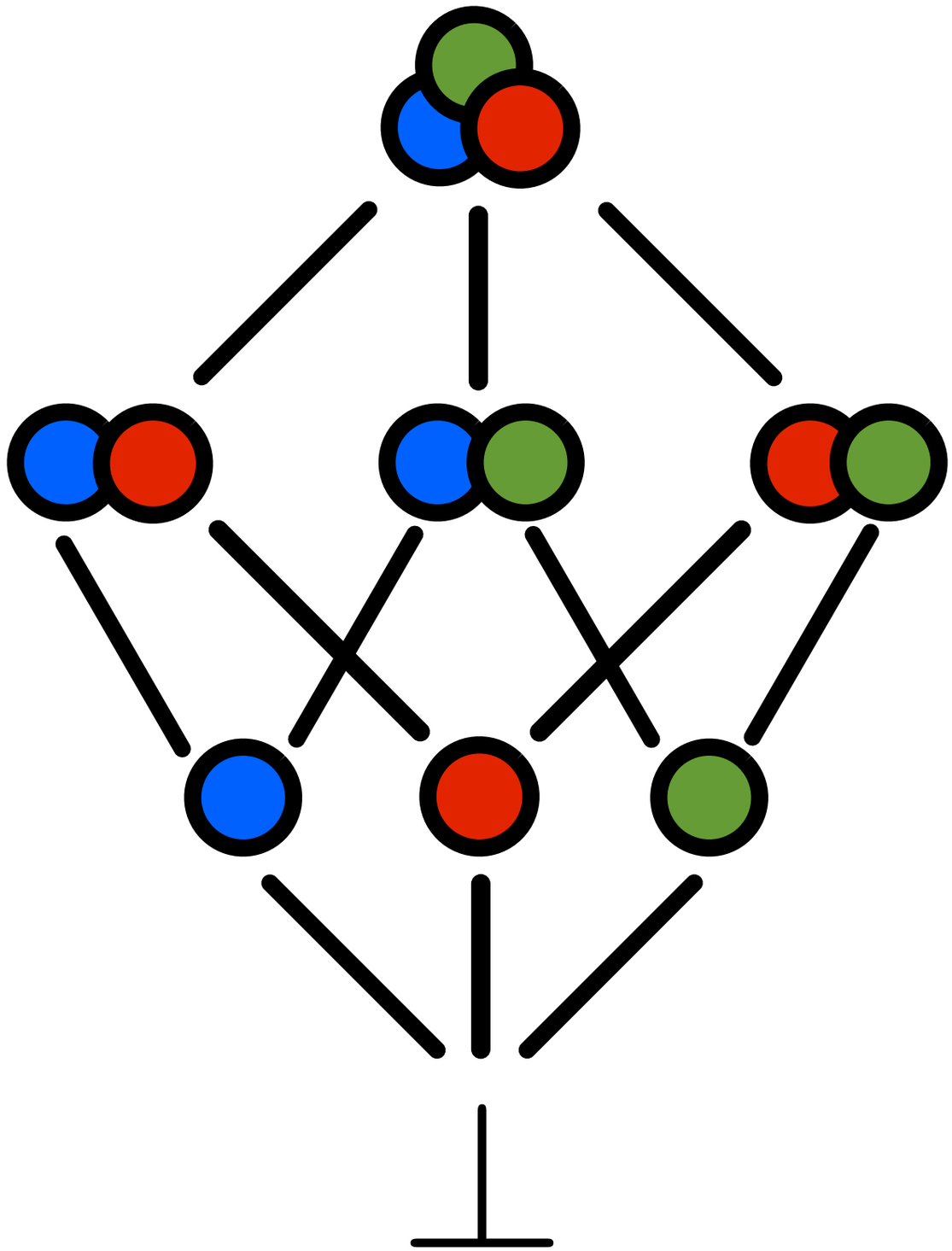


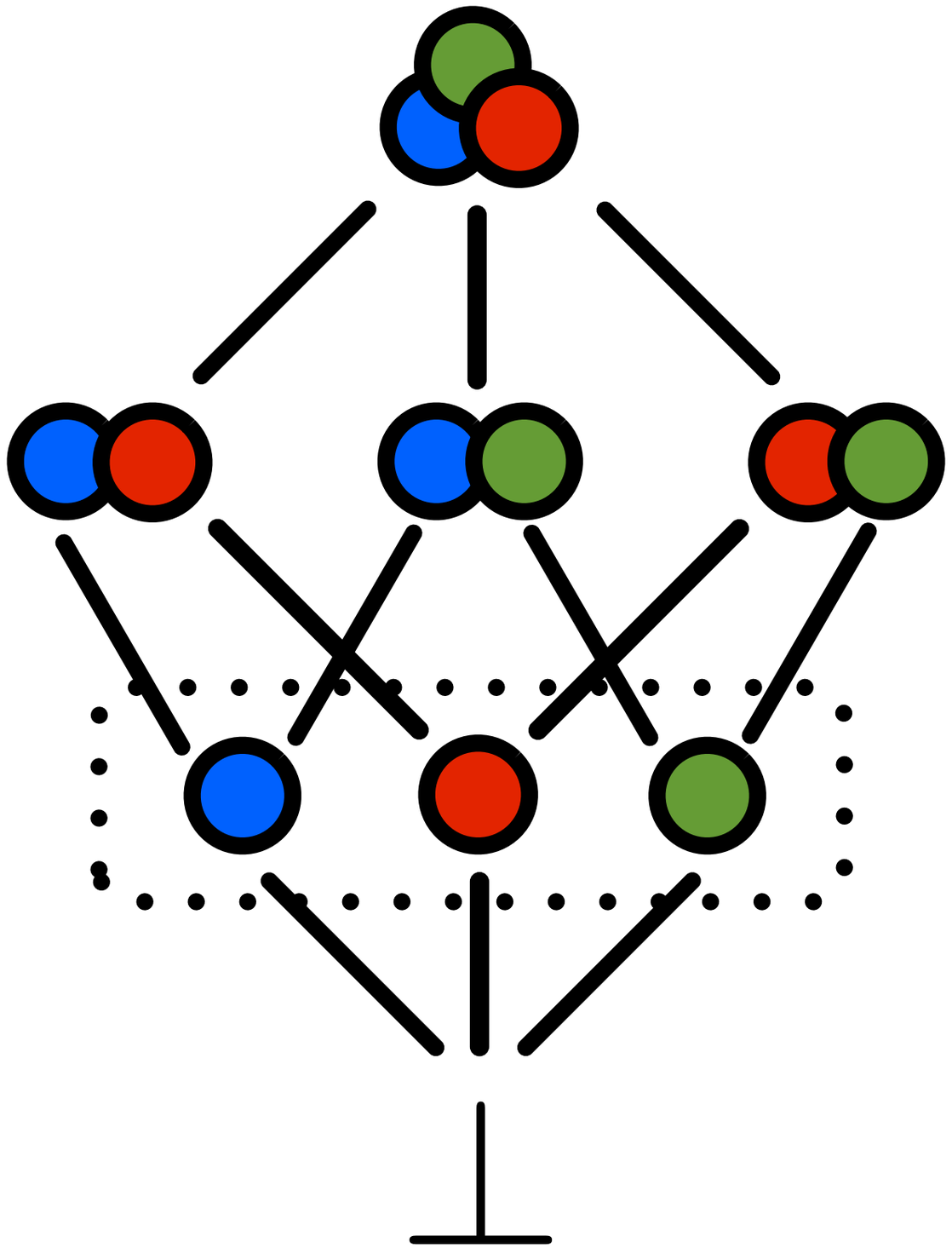


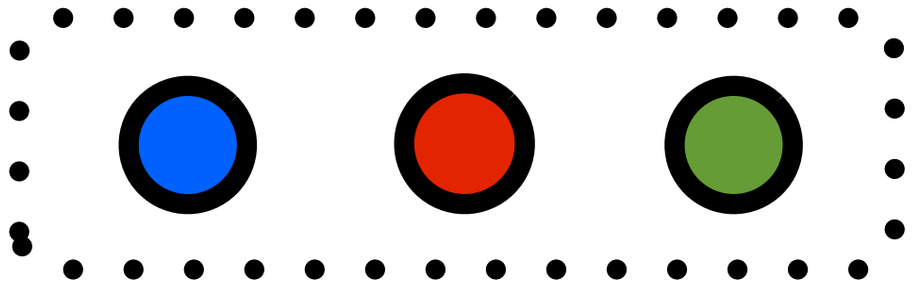


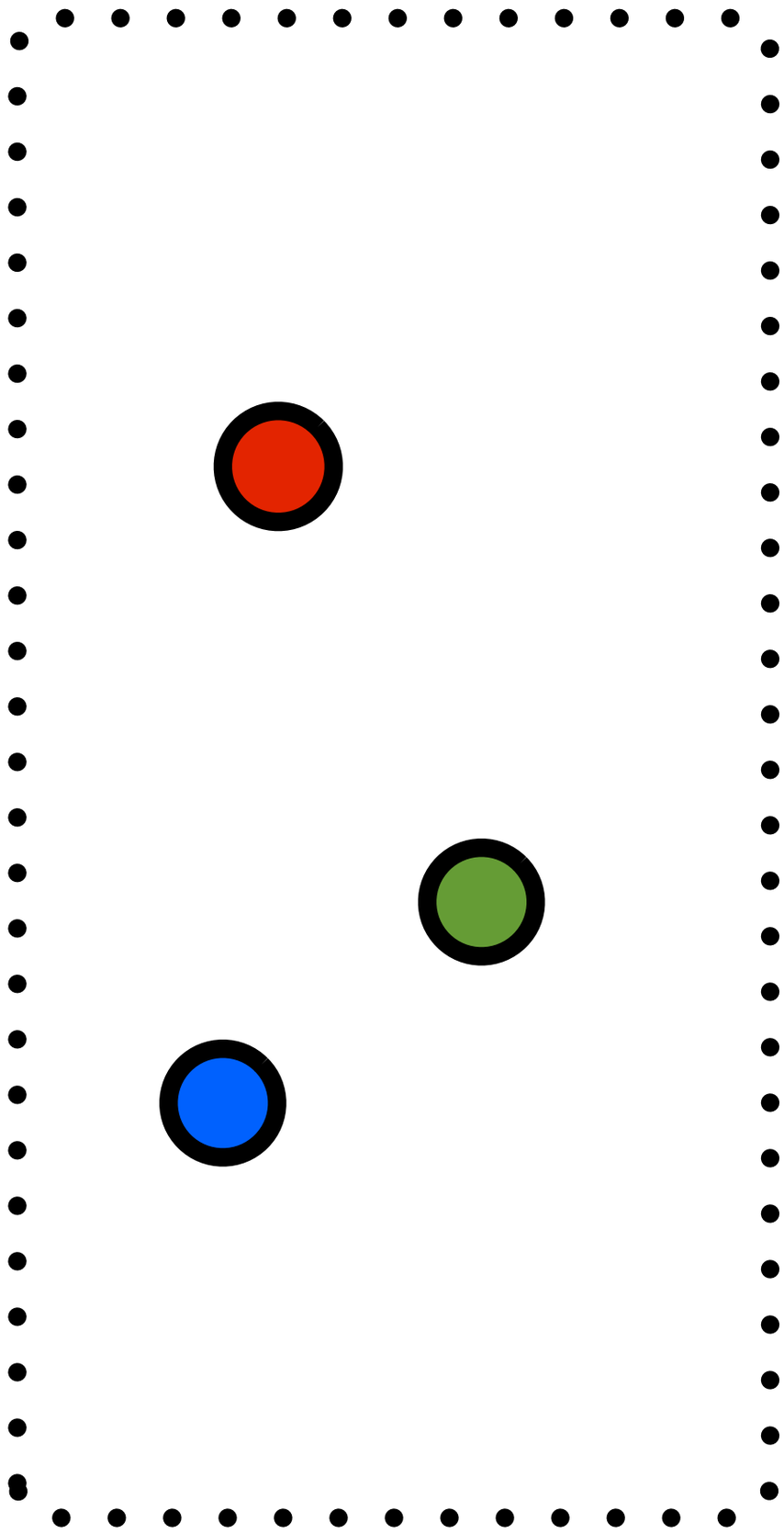


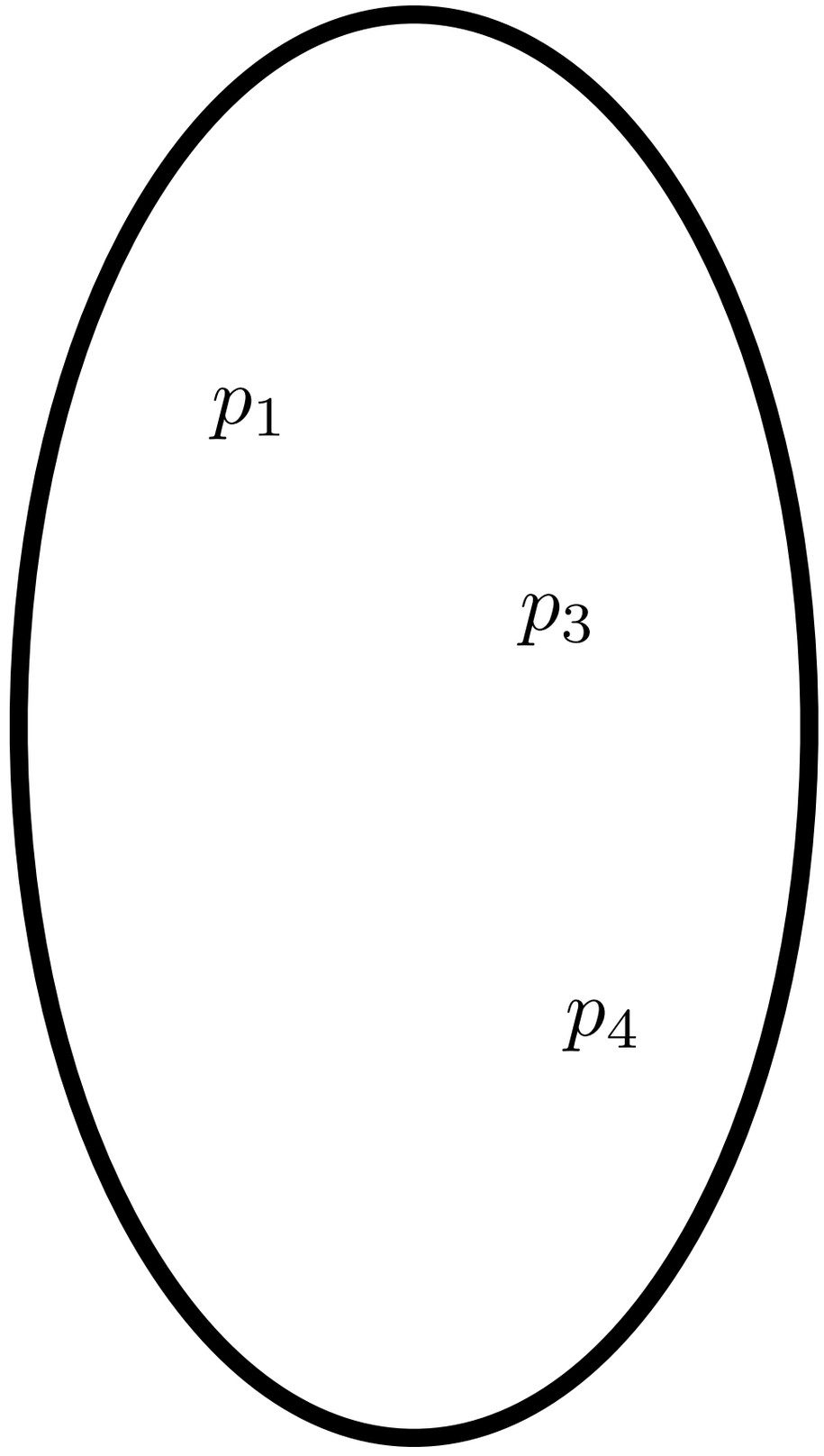


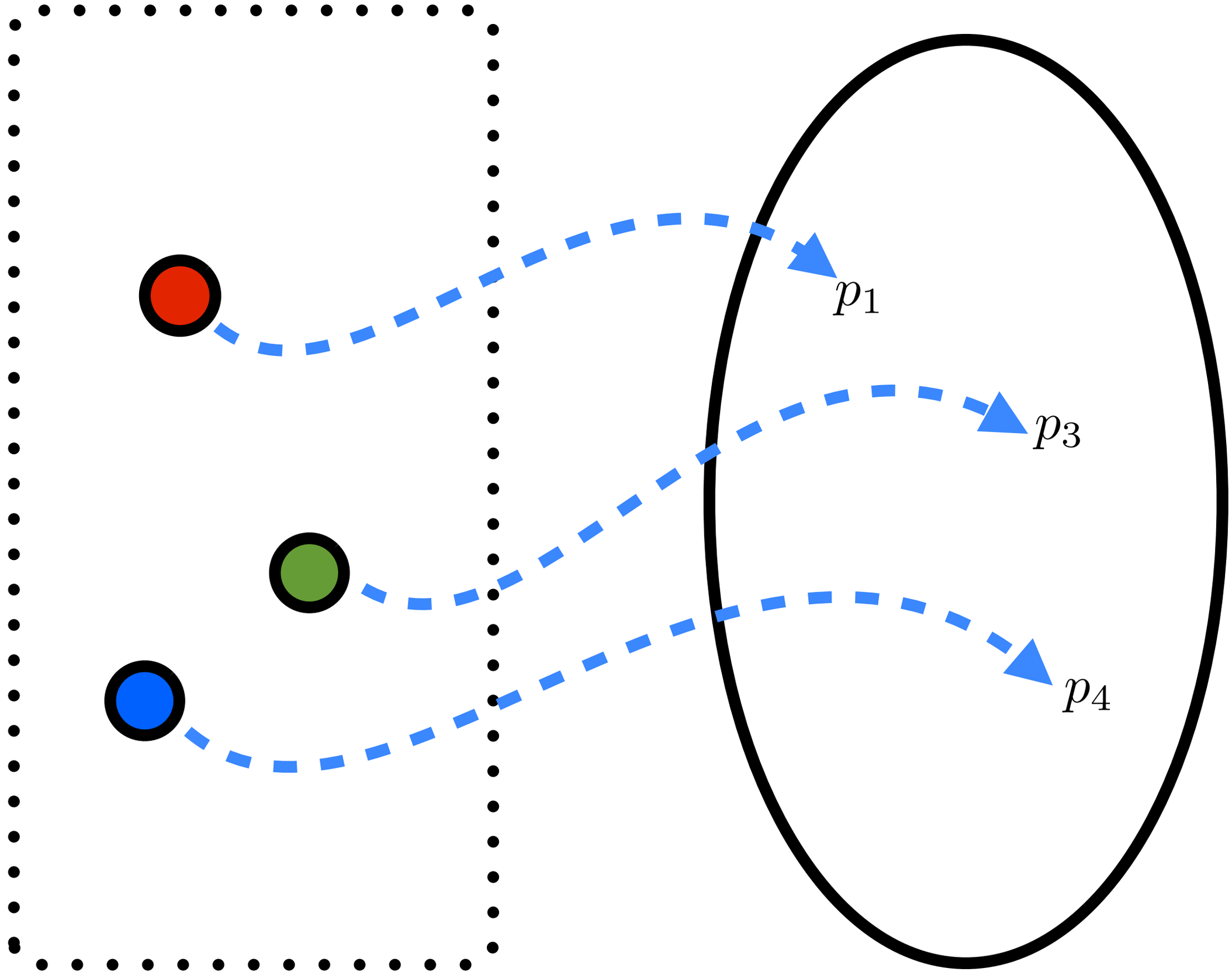









p_1
p_3
p_4


p_1
p_3
p_4

Trick 2

Control-flow still forks.

Return-flow still forks.

a.foo()

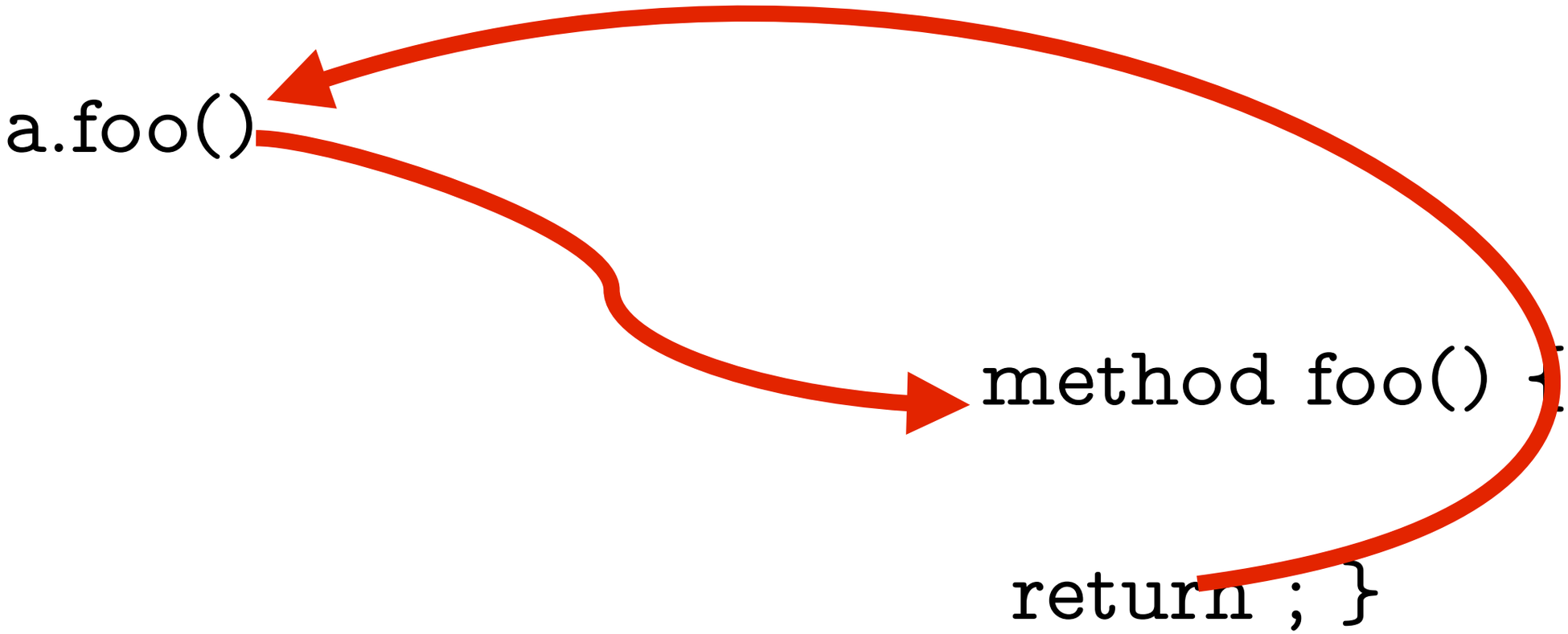
```
method foo() {
    return ; }
```

b.foo()

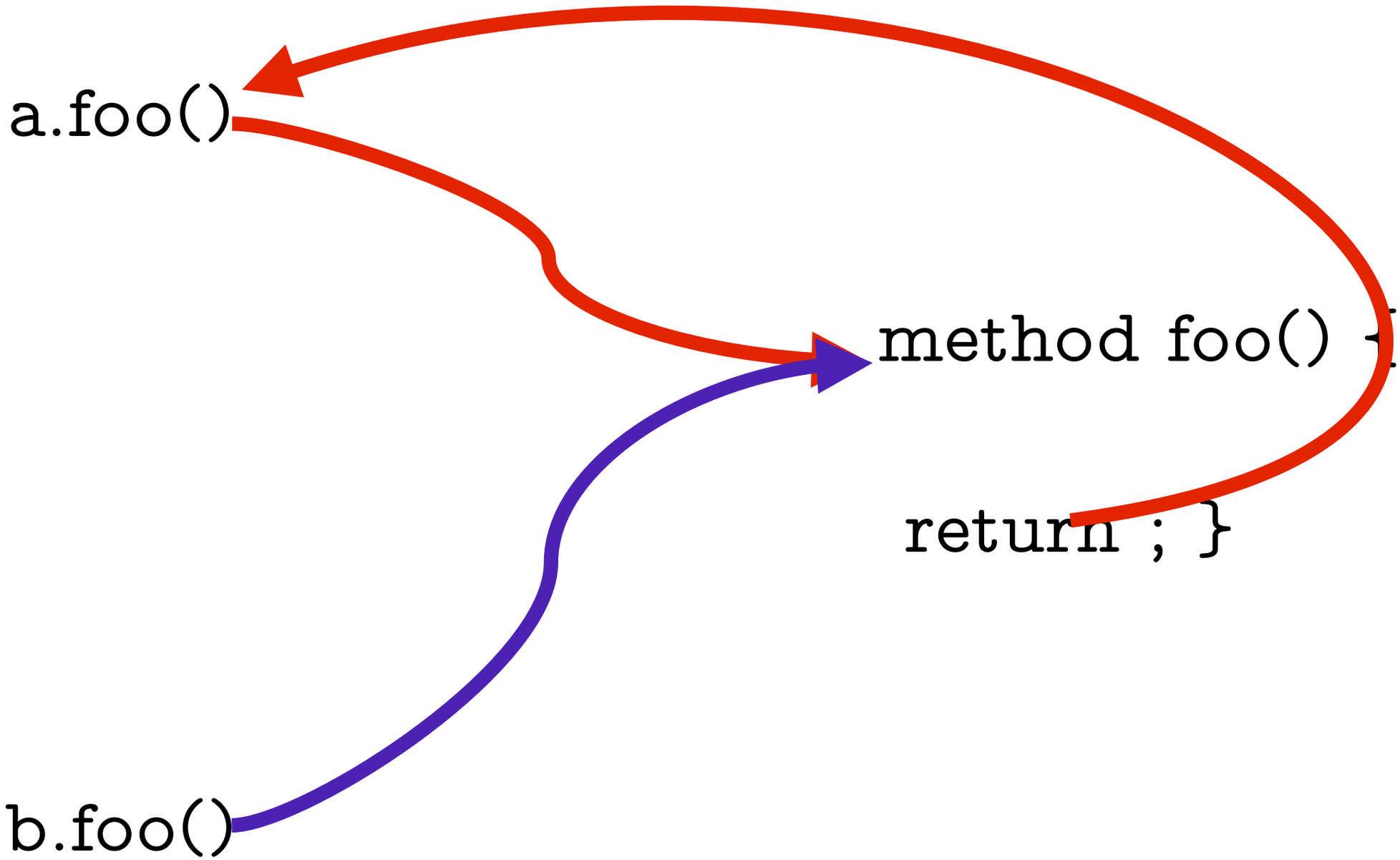
a.foo()

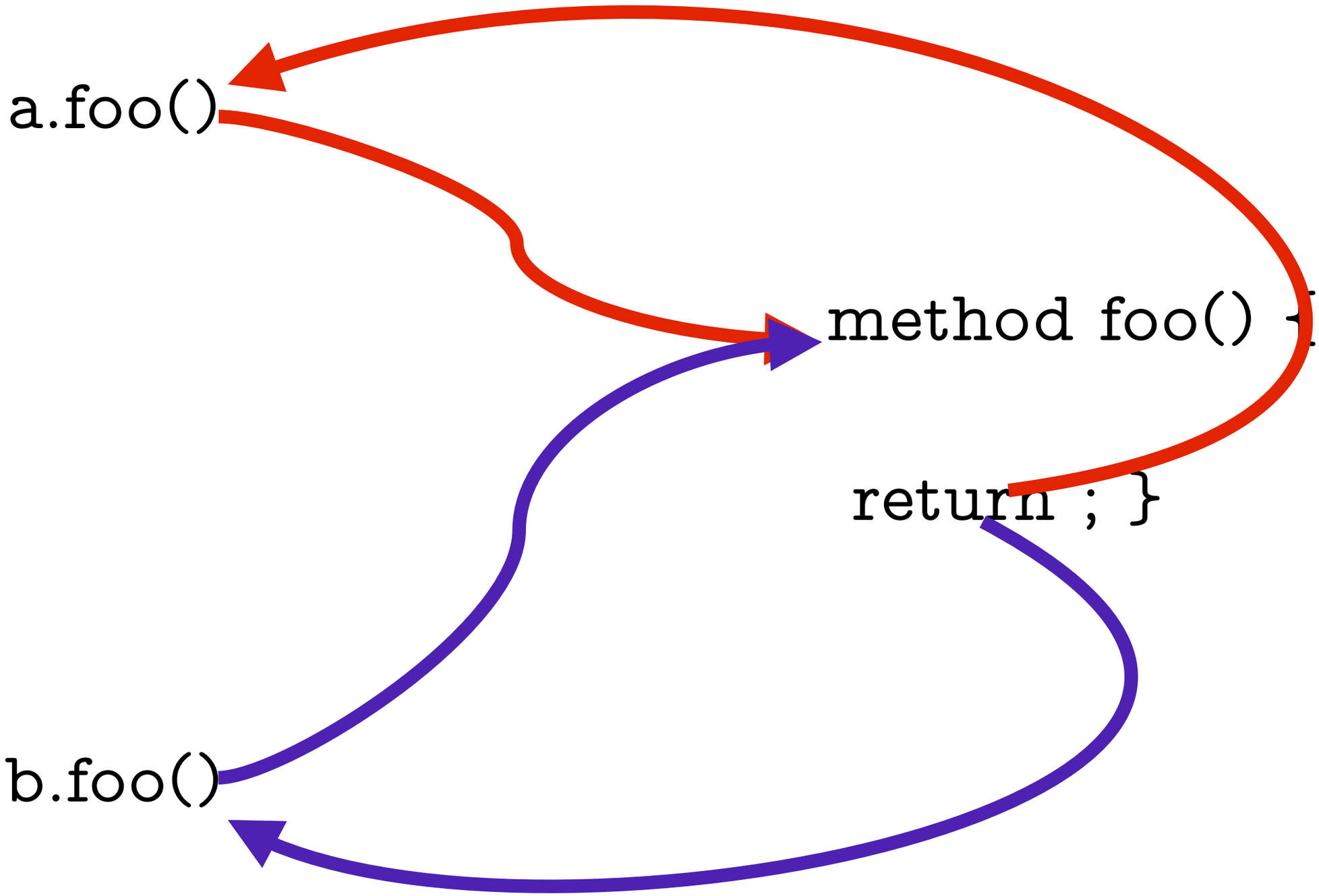
```
method foo() {
return ; }
```

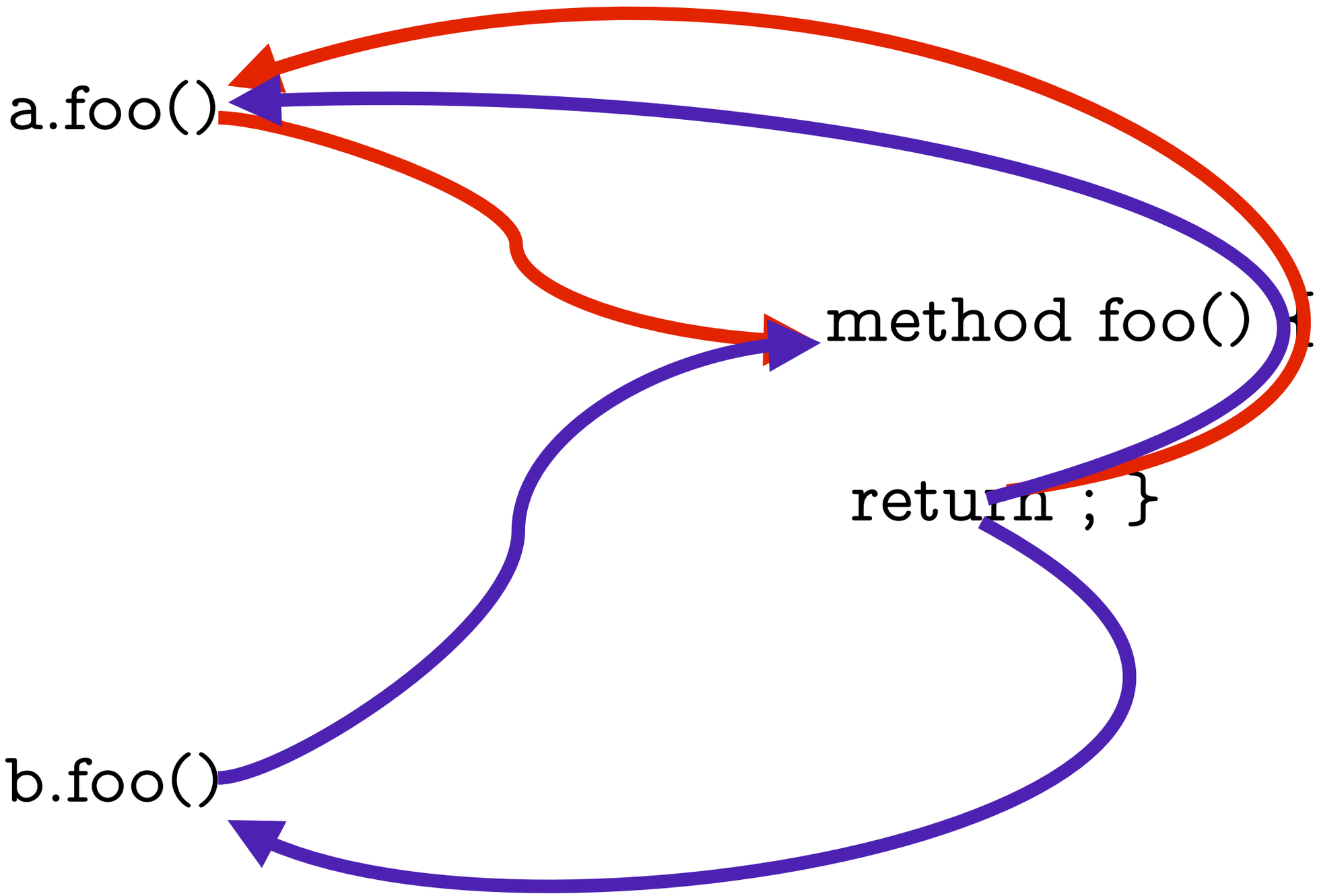
b.foo()



b.foo()


a.foo()
method foo() {
return ; }
b.foo()


a.foo()
method foo() {
return ; }
b.foo()


a.foo()
method foo()
return ; }
b.foo()

ČESK

CES K

$\widehat{\mathrm{CES}}$ $\widehat{\mathrm{K}}$

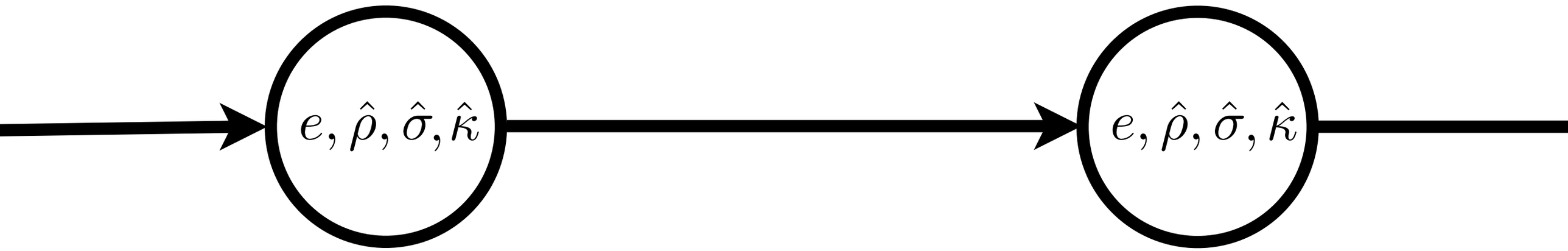
CESK

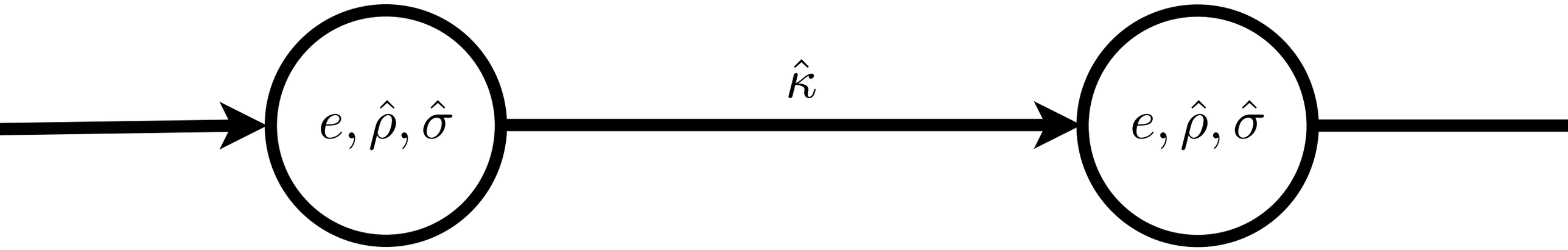
control state

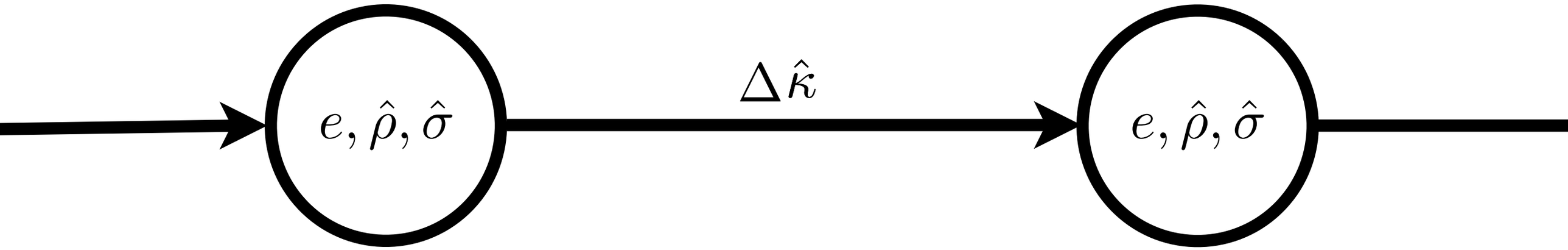
CESK

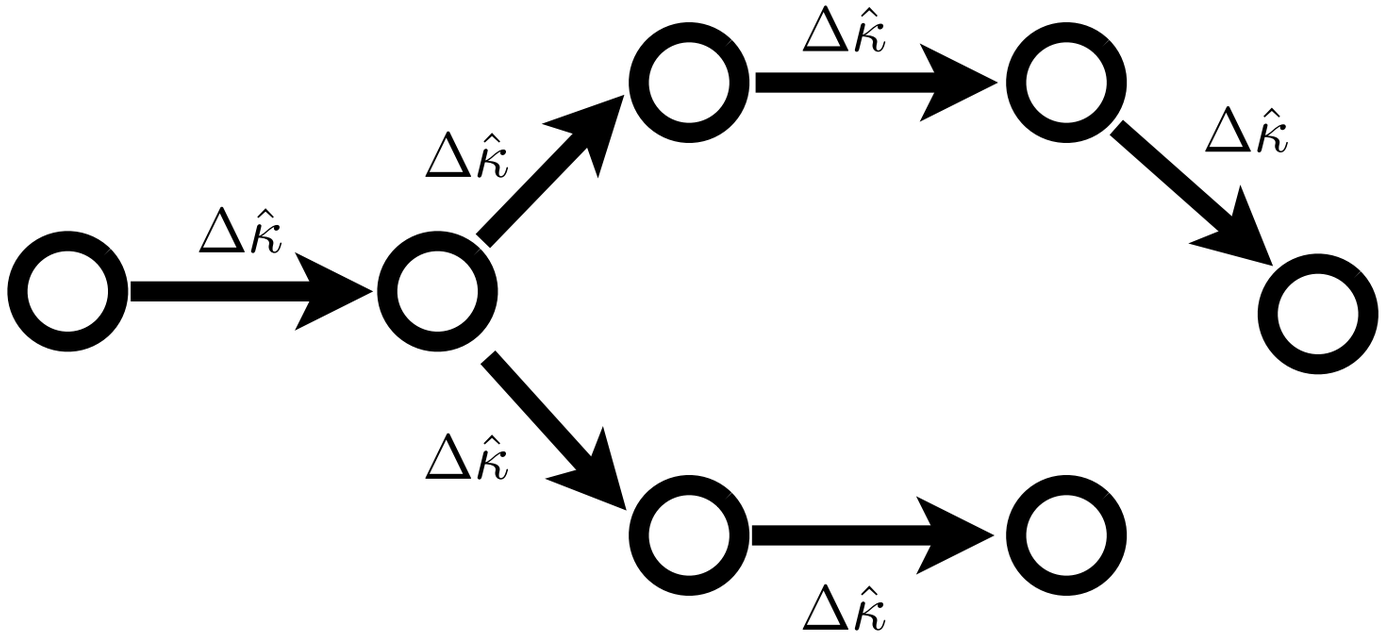
control state stack

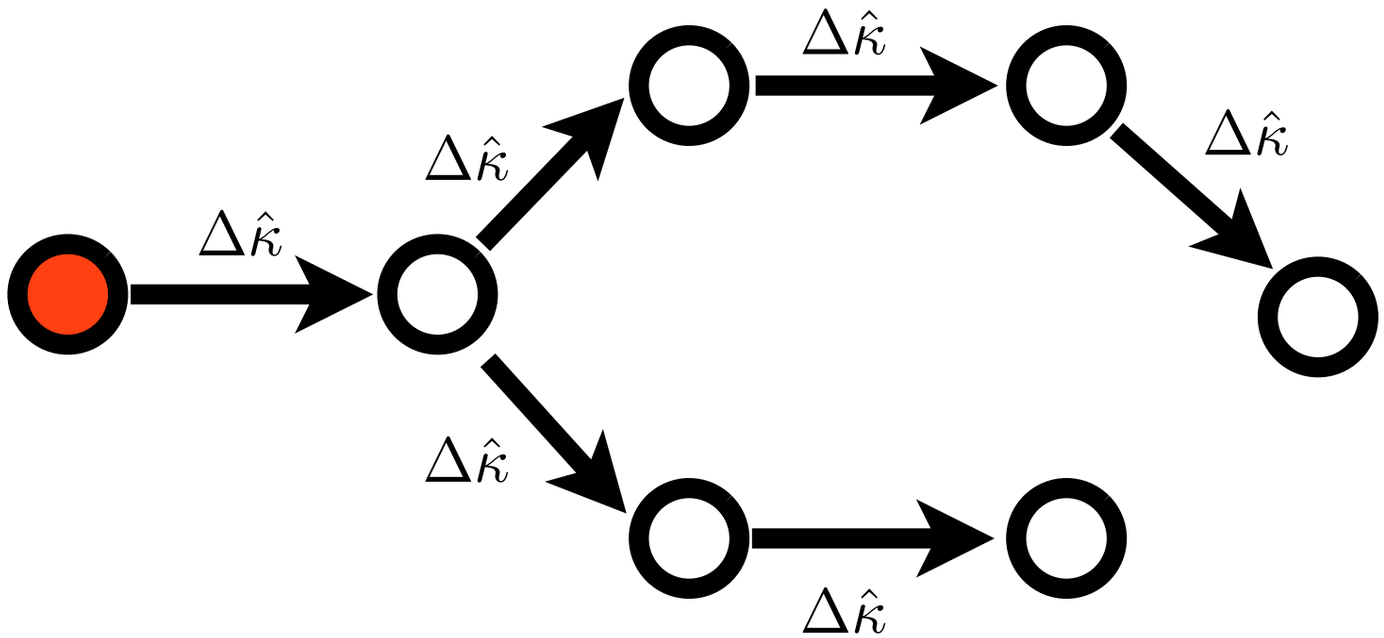
control state + stack = pushdown

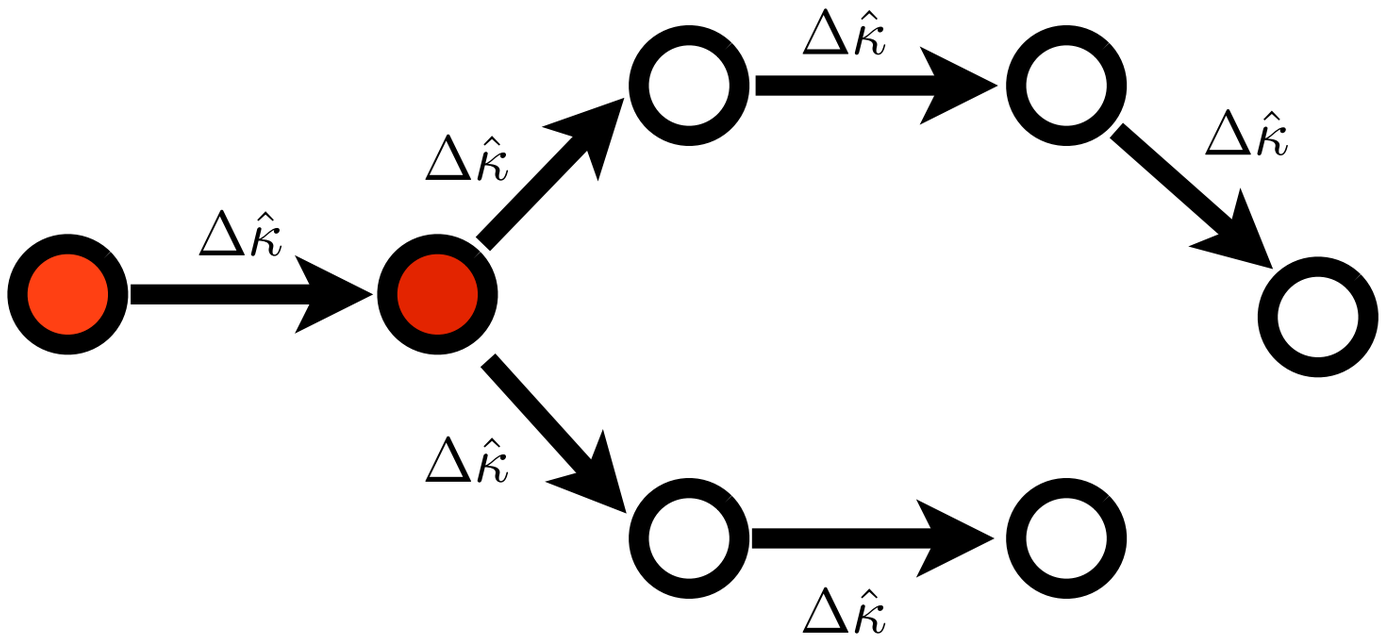

$e, \hat{\rho}, \hat{\sigma}, \hat{\kappa}$
$e, \hat{\rho}, \hat{\sigma}, \hat{\kappa}$

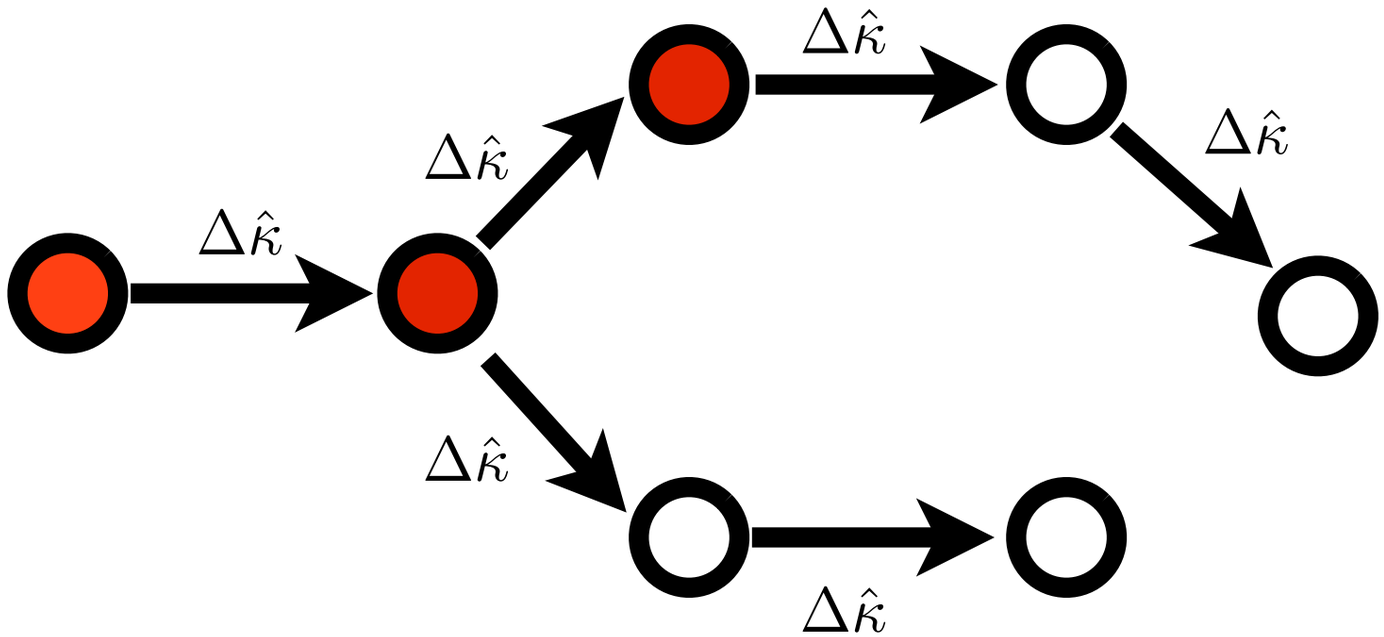

$e, \hat{\rho}, \hat{\sigma}$
$\hat{\kappa}$
$e, \hat{\rho}, \hat{\sigma}$

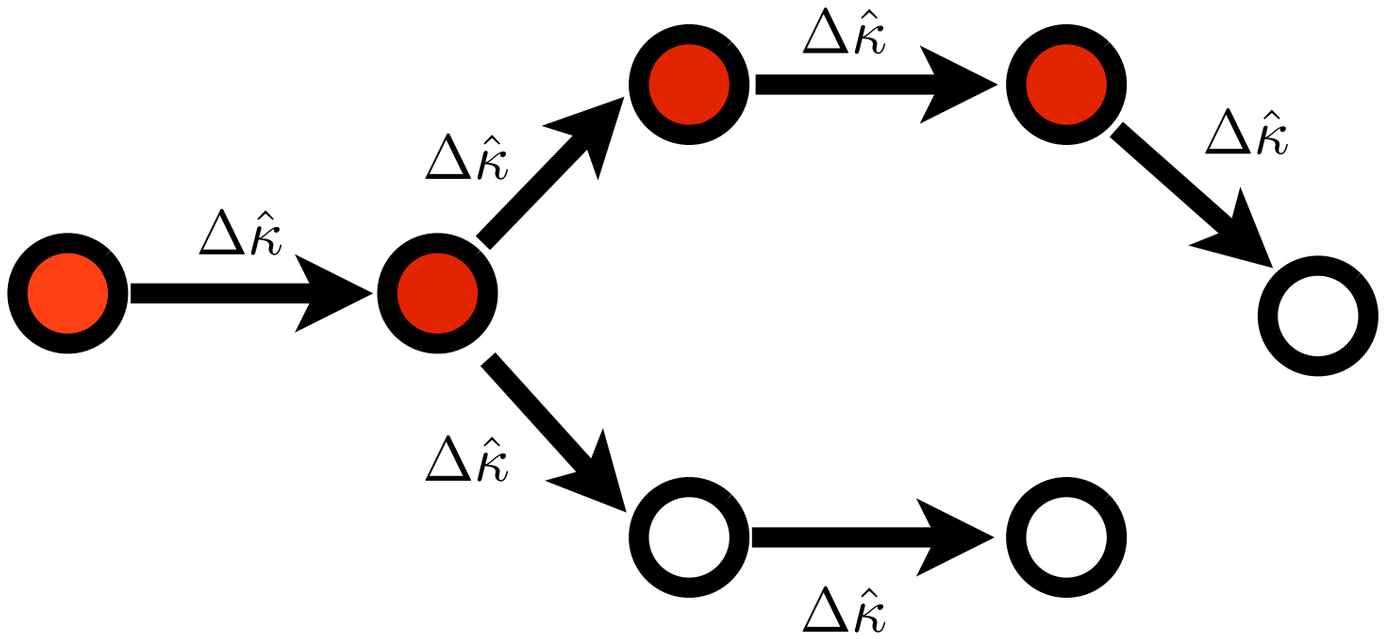

$e, \hat{\rho}, \hat{\sigma}$
$\Delta\hat{\kappa}$
$e, \hat{\rho}, \hat{\sigma}$

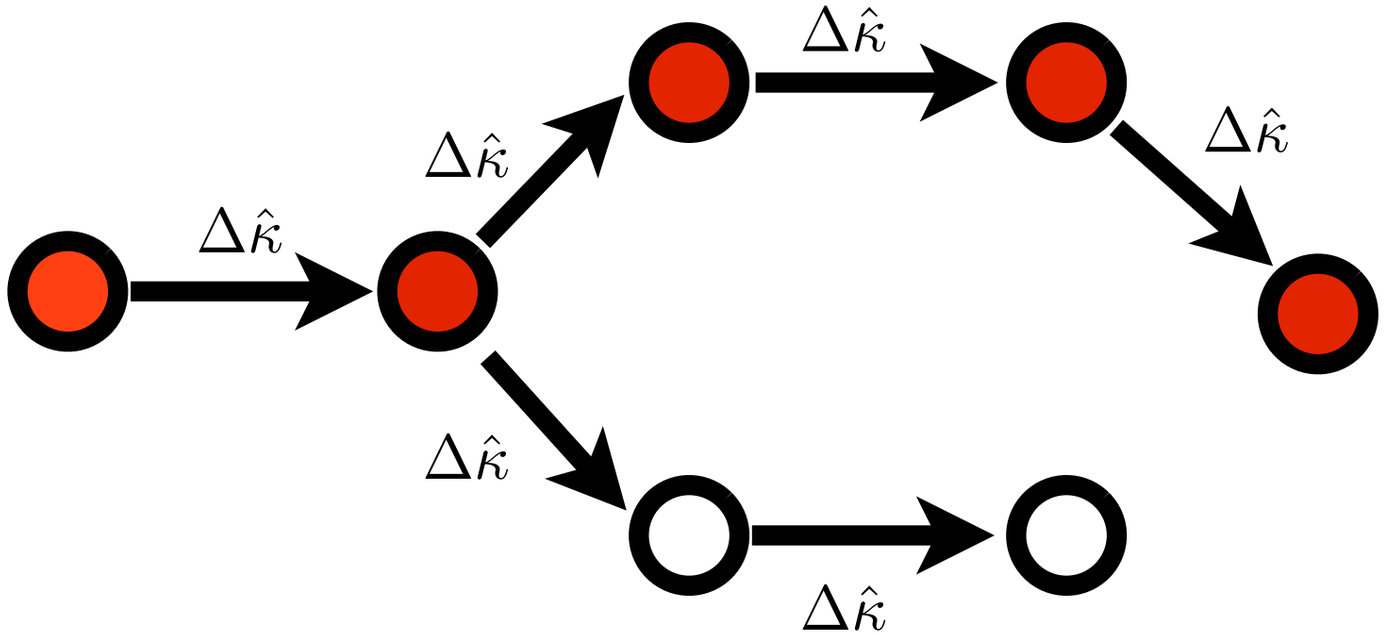

$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$

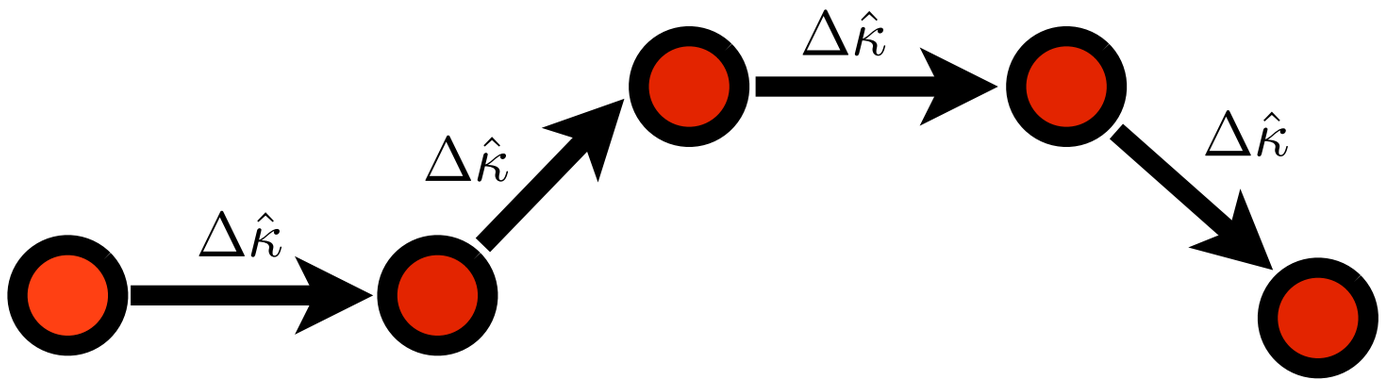

$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$

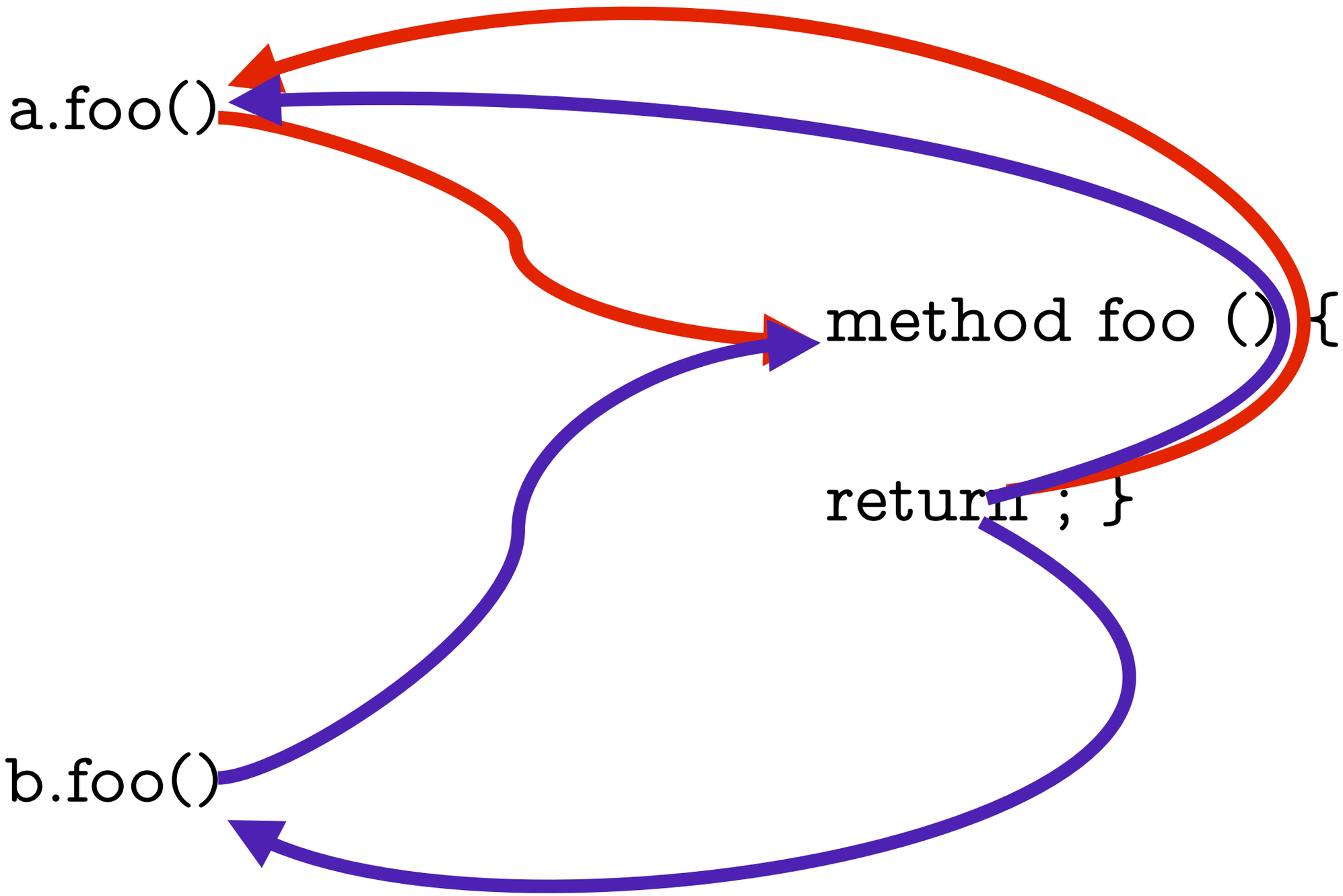

$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$

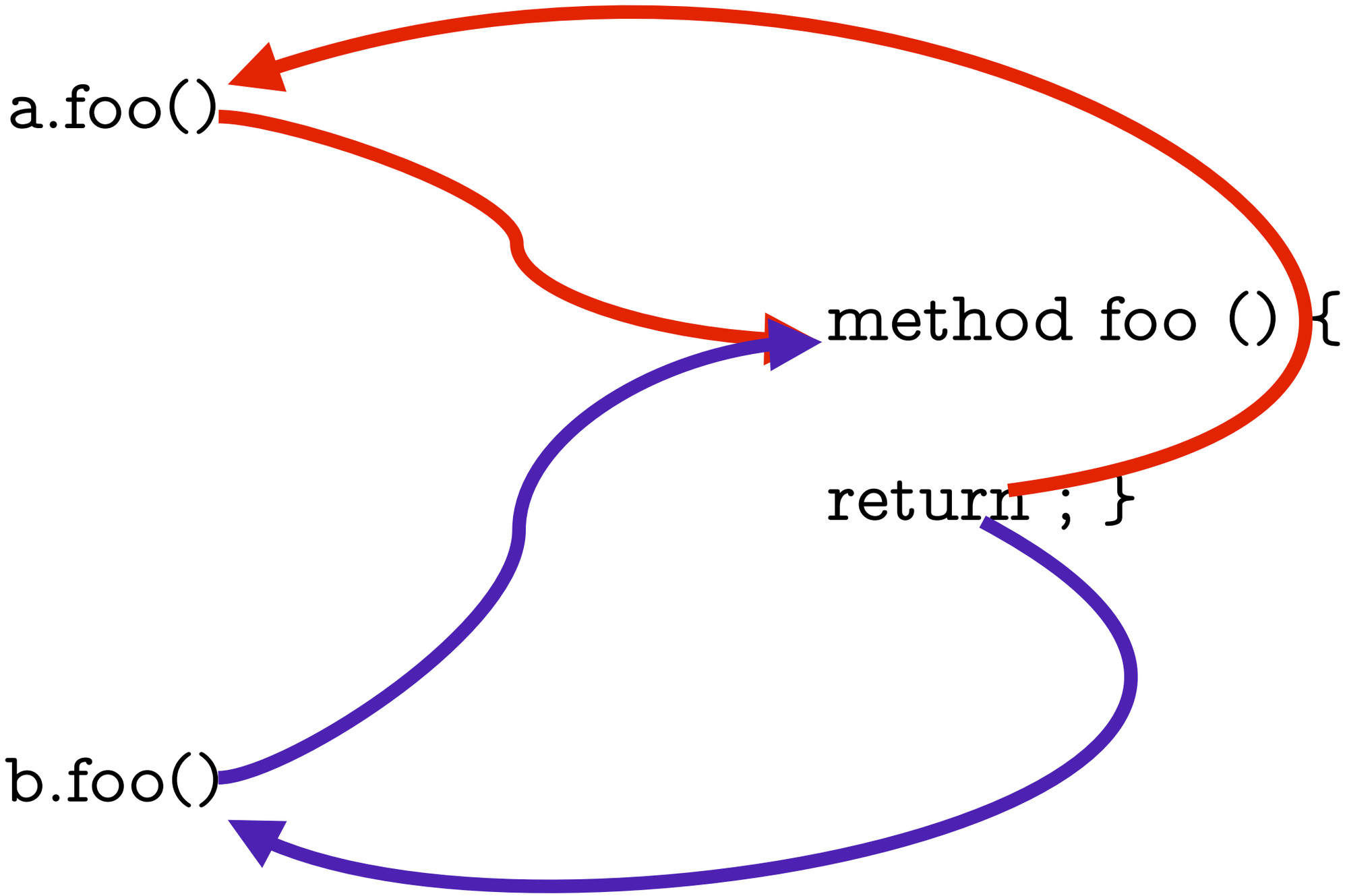

$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$


$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$


$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$


$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$
$\Delta\hat{\kappa}$


a.foo()
method foo () {
return ; }
b.foo()


a.foo()
method foo () {
return ; }
b.foo()

Terminal — zsh — 82×20 — ⌘1

```
</0>


<282!matt@maptop:~/Copy/APAC-Demo/Anadroid-AWeather>
%
```

shred

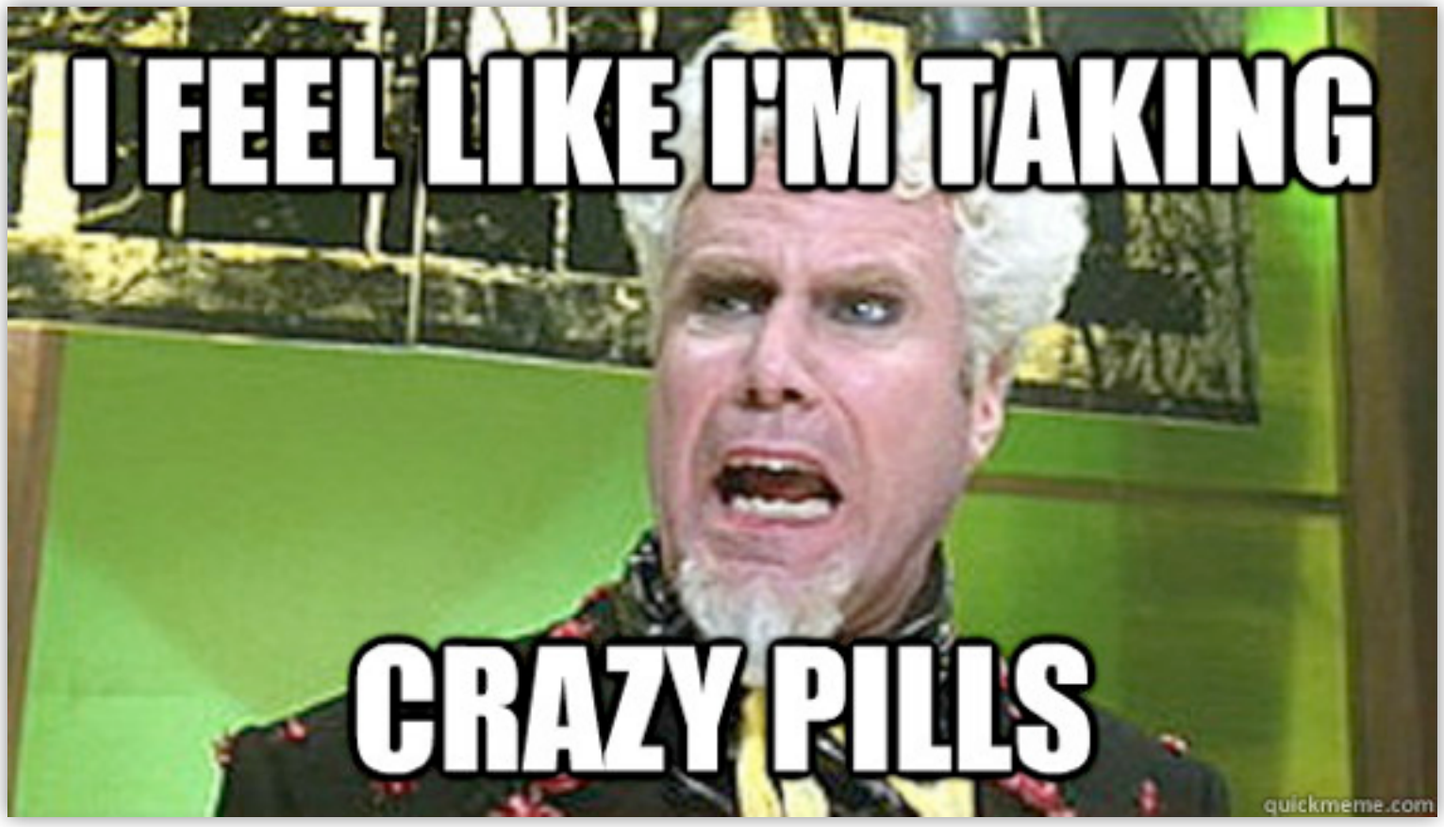
93%

(Challenge 4A)

Categorization


MEMOTIS!!!!!!!
memegenerator.net


FIND
ALL THE MALWARE
memegenerator.net


I FEEL LIKE I'M TAKING
CRAZY PILLS
quickmeme.com

Ongoing

Information

Information flow

```
sock.send32f(dat);
```

```
sock.send32f(dat);
```

dat *is-a* number

```
sock.send32f(dat);
```

```
-90.0 <= dat <= 90.0
```

```
sock.send32f(dat);
```

```
37.0 <= dat <= 43.0
```

```
sock.send32f(dat);
```

```
37.0 <= dat <= 43.0
```

```
sock.send32f(dat);
```

```
37.0 <= dat <= 43.0
```

```
sock.send32f(dat);
```

```
latitude influences dat
```

Tracking *influence*

Terminal — zsh — 82×20 — ⌘1

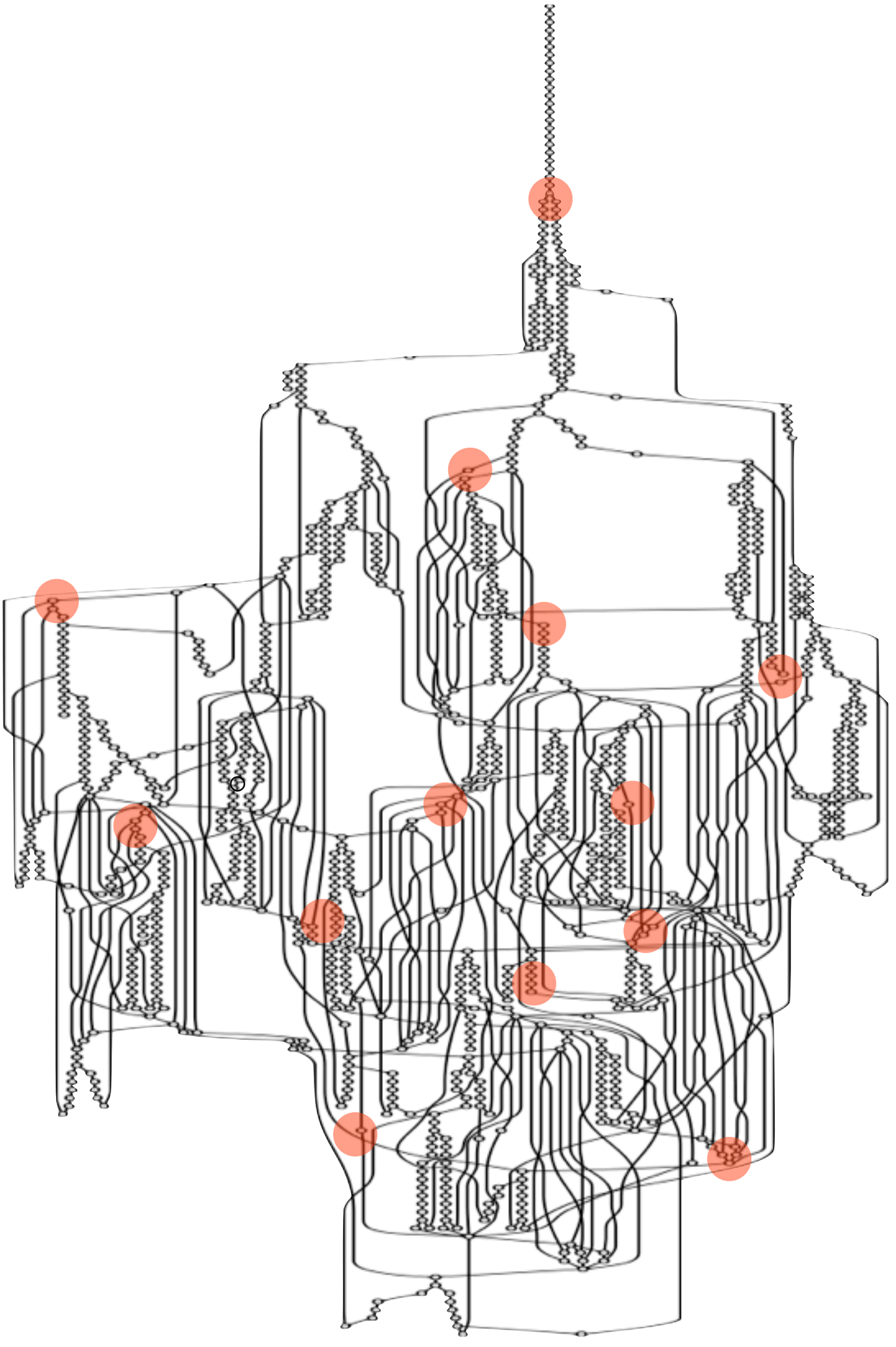
```
</0>


<282!matt@maptop:~/Copy/APAC-Demo/Anadroid-AWeather>
% 
```

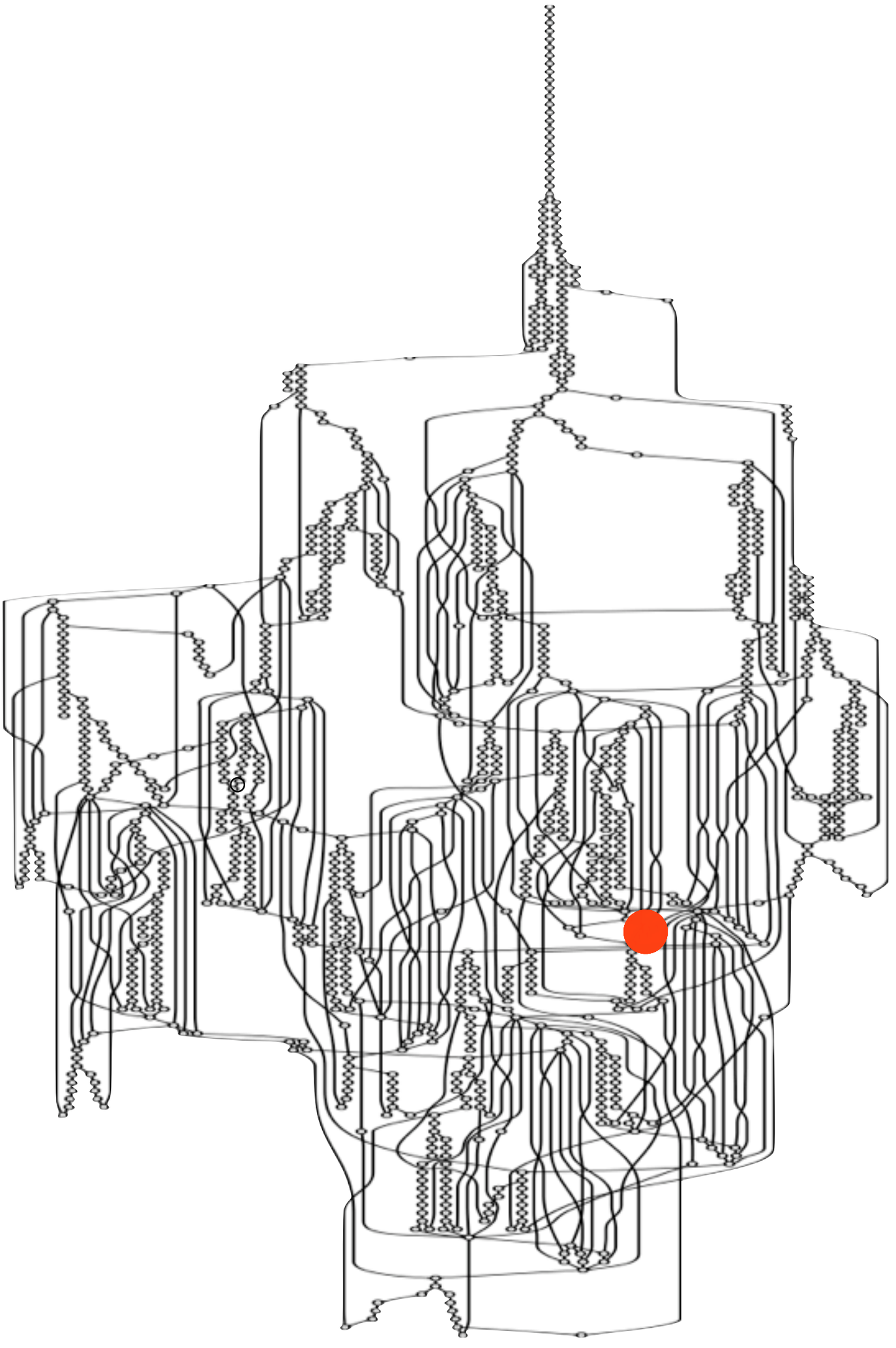
flows

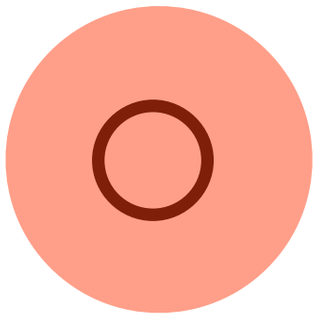
In progress

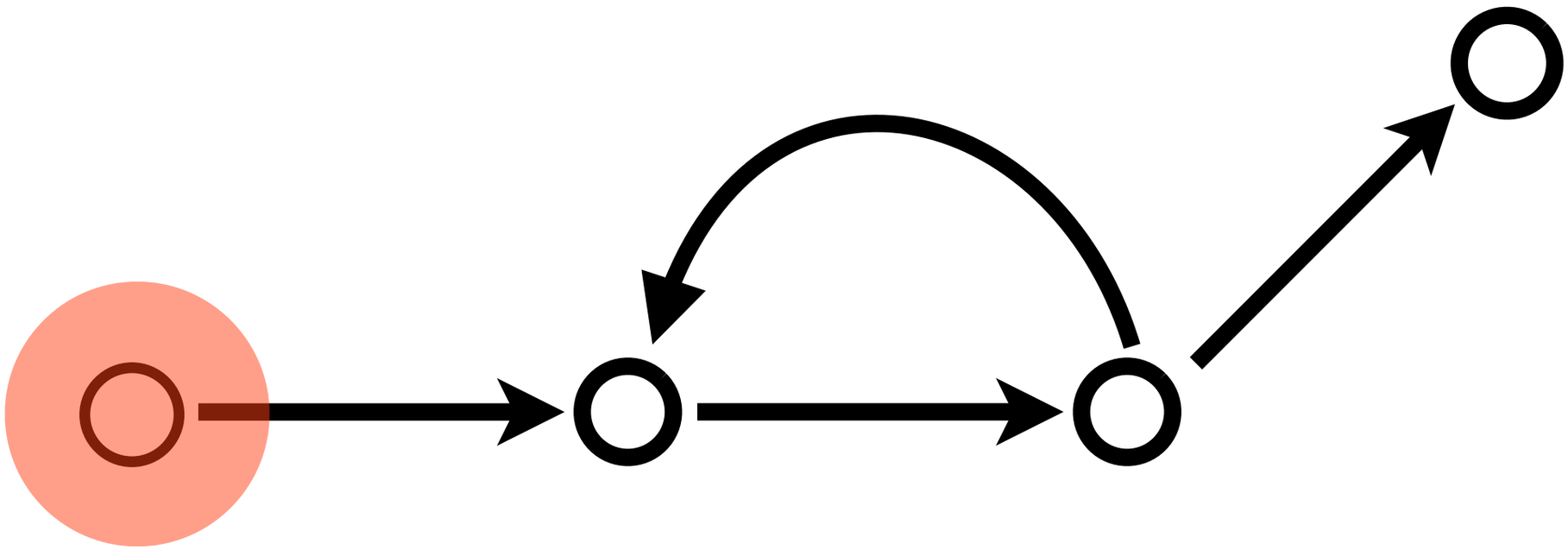

Terminal — zsh — 82×20 — ⌘1
</0>
<282!matt@maptop:~/Copy/APAC-Demo/Anadroid-AWeather>
%
catsup

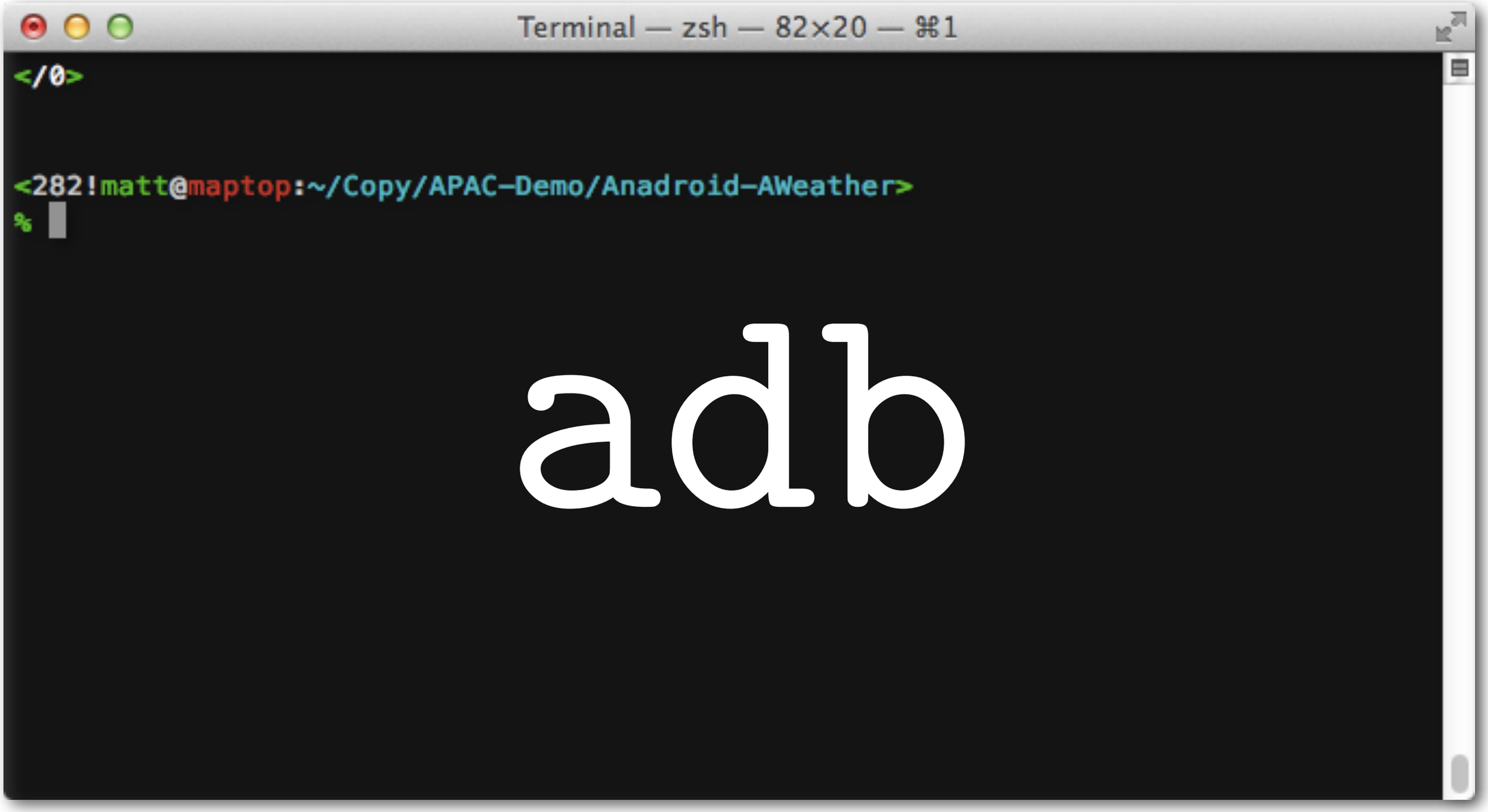
Abstract debugging










Terminal — zsh — 82×20 — ⌘1
</0>
<282!matt@maptop:~/Copy/APAC-Demo/Anadroid-AWeather>
%
adb

Implicit flows

```
void onClick () {

   if (sensitiveCheckBox.isChecked()) {
      send(true);
   } else {
      send(false);
   }

}
```

Terminal — zsh — 82×20 — ⌘1

```
</0>


<282!matt@maptop:~/Copy/APAC-Demo/Anadroid-AWeather>
% 
```

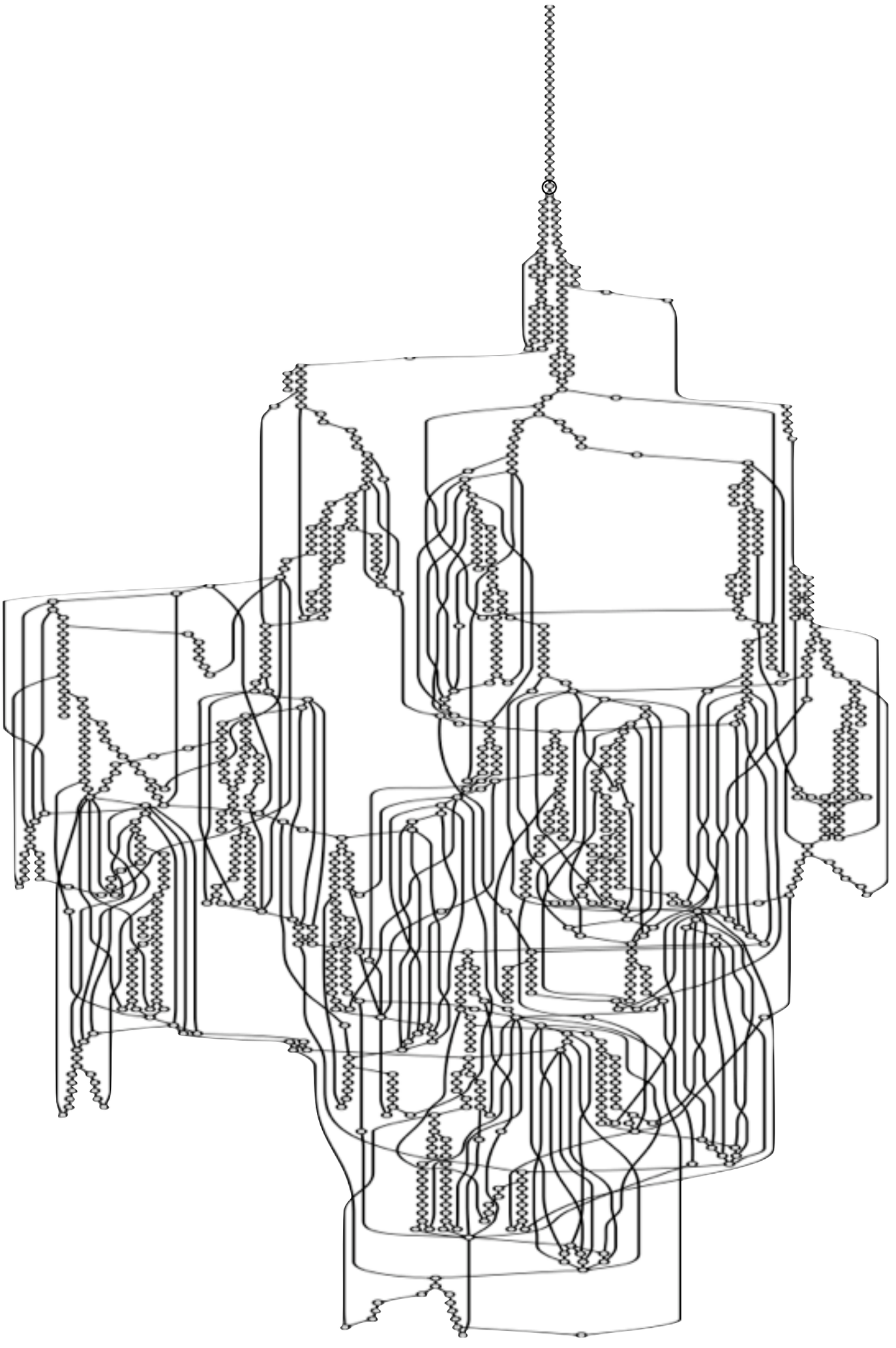
imp

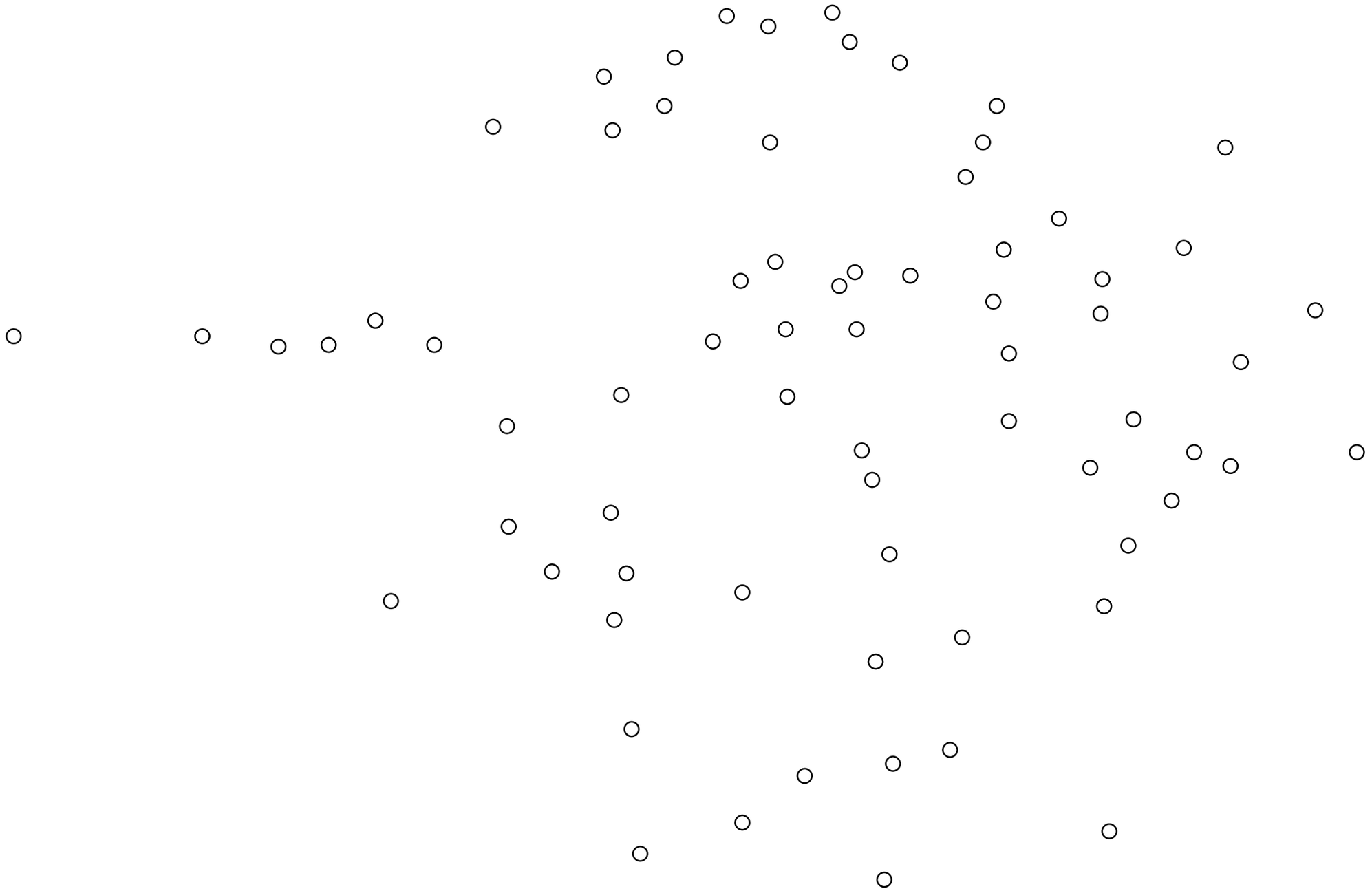
```
void onClick () {

   if (sensitiveCheckBox.isChecked()) {
      send(true);
   } else {
      send(false);
   }

}
```

```
void onClick () {

   if (sensitiveCheckBox.isChecked()) {
      send(true);
   } else {
      send(false);
   }

}
```

```
void onClick () {

    if (sensitiveCheckBox.isChecked()) {
      send(true);
    } else {
      send(false);
    }

}
```

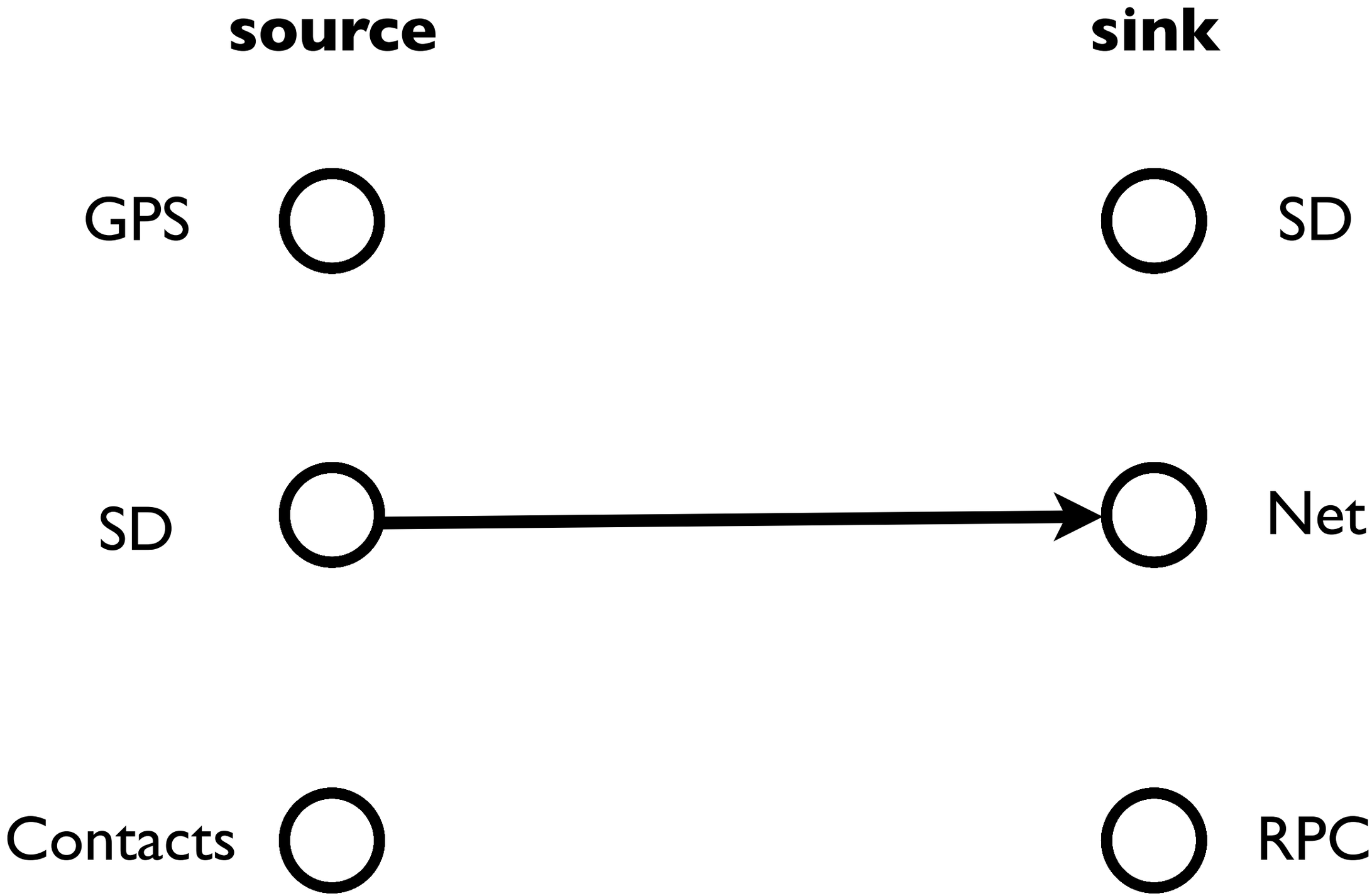

POTENTIAL





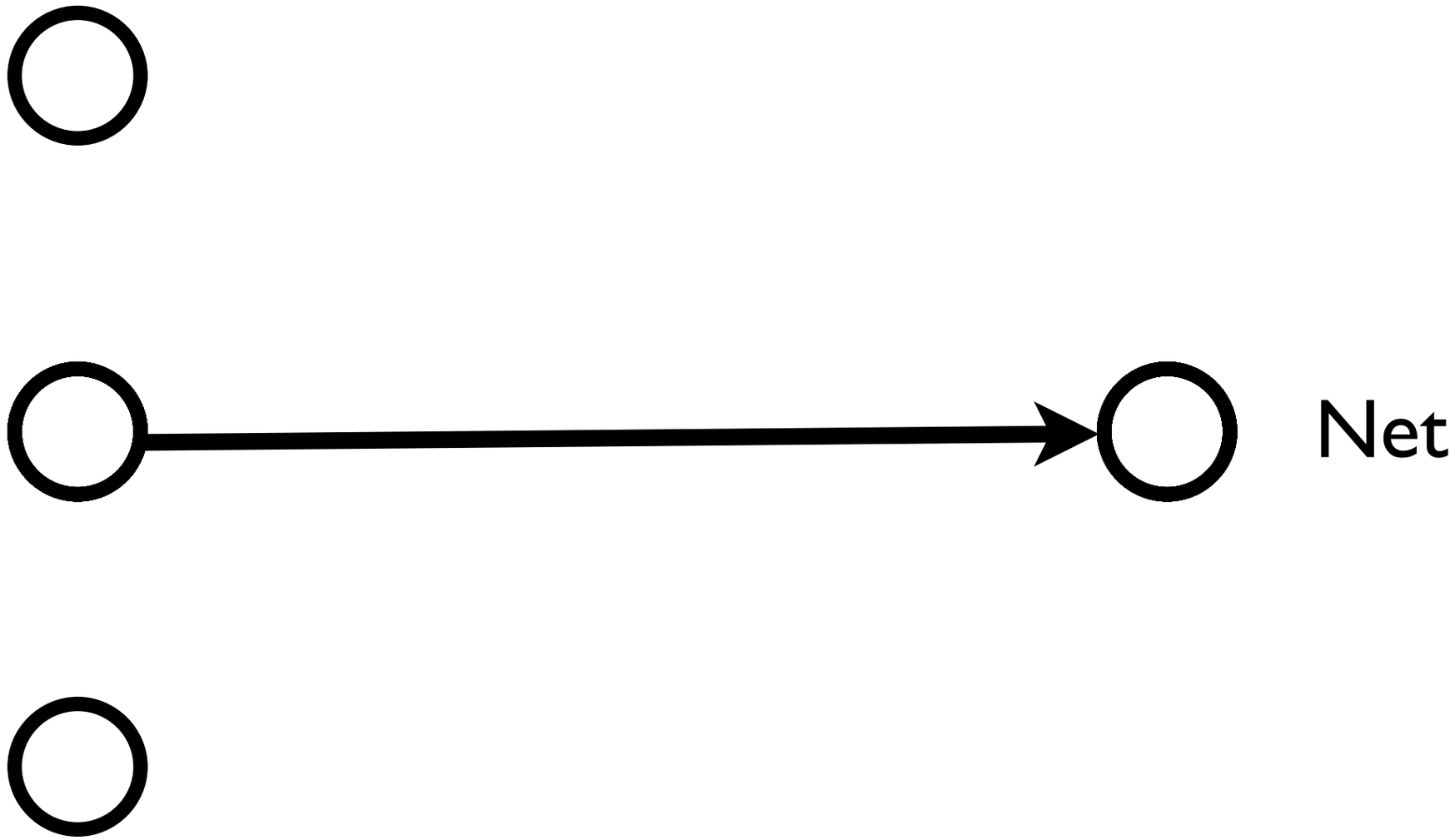
source

sink


source
sink
GPS
SD
SD
Net
Contacts
RPC

source **sink**

source **sink**



Net

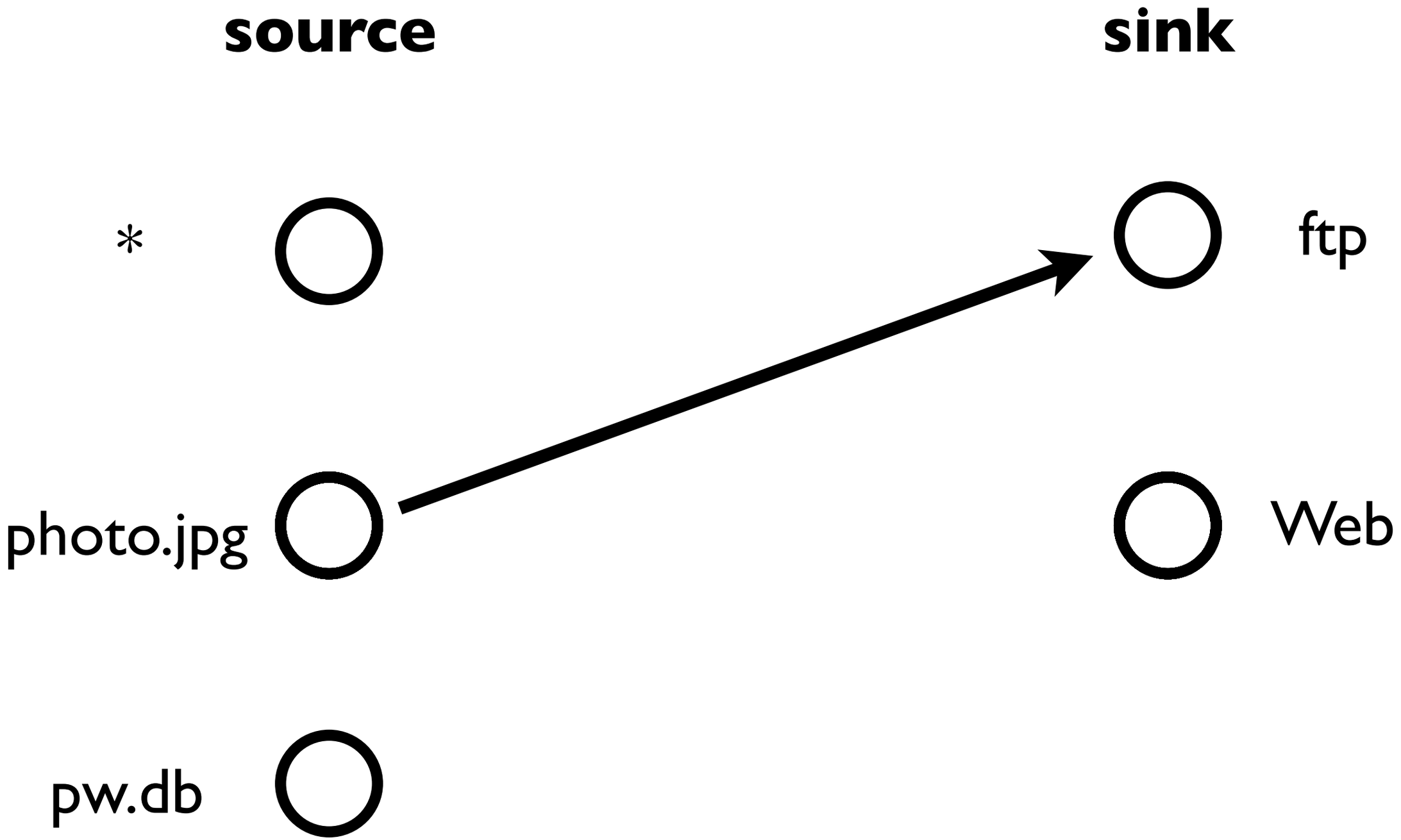
source **sink**

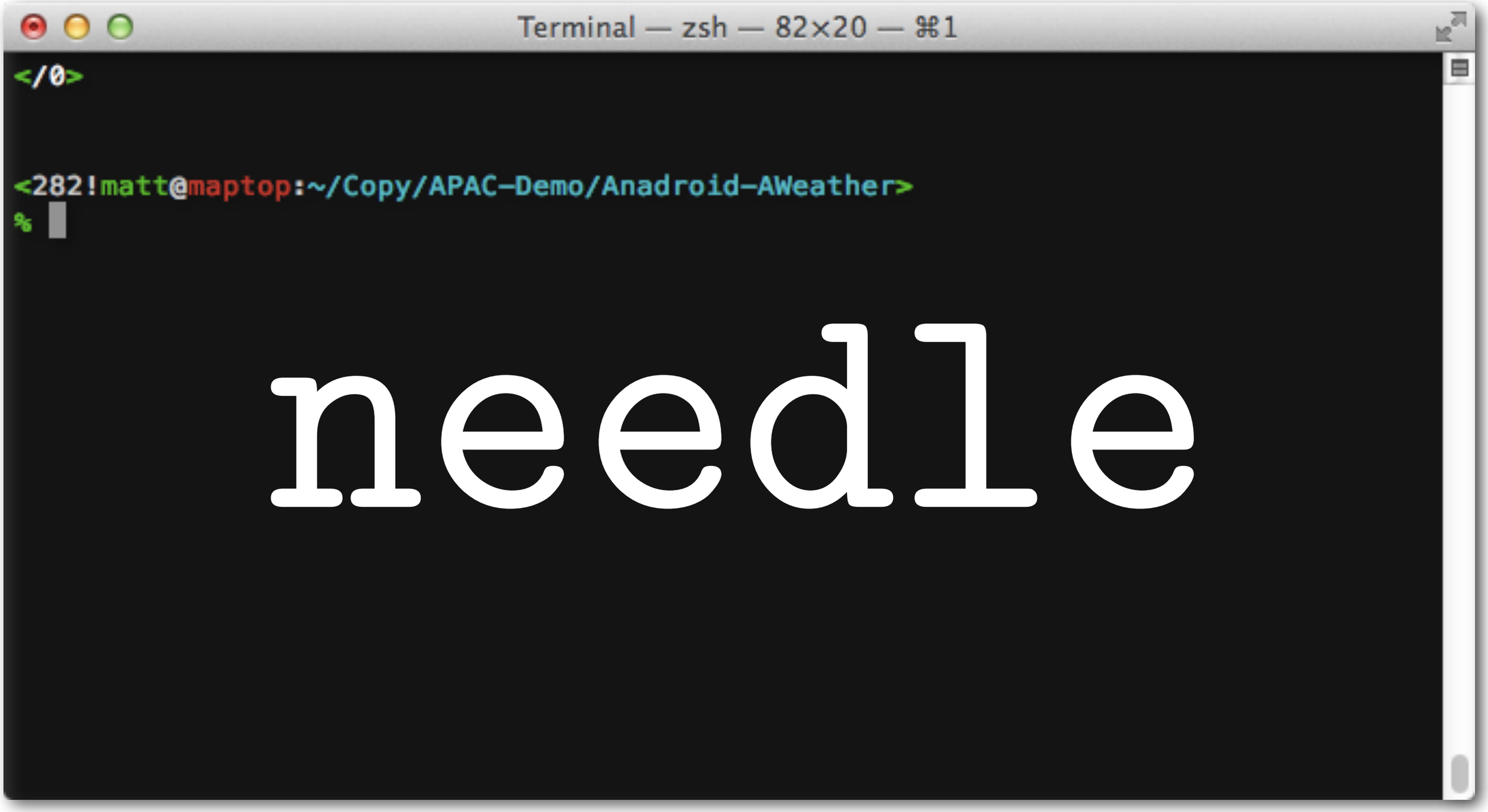
*

photo.jpg

Net

pw.db


source
sink
*
ftp
photo.jpg
Web
pw.db


Terminal — zsh — 82×20 — ⌘1
</0>
<282!matt@maptop:~/Copy/APAC-Demo/Anadroid-AWeather>
%
needle